

Semantically-driven Automatic Creation of Training Sets for Object Recognition

Dong-Seon Cheng^a, Francesco Setti^b, Nicola Zeni^b, Roberta Ferrario^b,
Marco Cristani^c

^a*Hankuk University of Foreign Studies, Yongin, Gyeonggi-do, Korea*

^b*ISTC-CNR, via alla Cascata 56/C, I-38123 Povo (Trento), Italy*

^c*Università degli Studi di Verona, Strada Le Grazie 15, I-37134 Verona, Italy*

Abstract

In the object recognition community, much effort has been spent on devising expressive object representations and powerful learning strategies for designing effective classifiers, capable of achieving high accuracy and generalization. In this scenario, the focus on the training sets has been historically weak; by and large, training sets have been generated with a substantial human intervention, requiring considerable time. In this paper, we present a strategy for automatic training set generation. The strategy uses semantic knowledge coming from WordNet, coupled with the statistical power provided by Google Ngram, to select a set of meaningful text strings related to the text class-label (*e.g.*, “cat”), that are subsequently fed into the Google Images search engine, producing sets of images with high training value. Focusing on the classes of different object recognition benchmarks (PASCAL VOC 2012, Caltech-256, ImageNet, GRAZ and OxfordPet), our approach collects novel training images, compared to the ones obtained by exploiting Google Images with the simple text class-label. In particular, we show that the gathered images are better able to capture the different visual facets of a concept, thus encoding in a more successful manner the intra-class variance. As a consequence, training standard classifiers with this data produces performances not too distant from those obtained from the classical hand-crafted training sets. In addition, our datasets generalize well and are stable, that is, they provide similar performances on diverse test datasets. This process does not require manual intervention and is completed in a few hours.

Keywords: Object recognition, training dataset, semantics, WordNet, Internet search

1. Introduction

Object recognition has been since its beginnings and still is one of the main and most studied topics in computer vision and its applications are many and varied, ranging from image indexing and retrieval, to video surveillance, robotics and medicine.

Even though at a first glance one may think that what is being recognized is an object that is given “out there in the world”, at a closest look one may see that what is detected and then assigned to a certain class of objects is something that is constructed out from an aggregation of features that a classifier has been trained to recognize as that particular kind of object [1]. As a consequence, the fact that a certain aggregation of features is recognized as a dog or as a building, strongly depends on the images that have been chosen to be part of the training set [2].

Traditionally, classifiers have been and often are still trained with datasets that were created *ad-hoc* by computer vision scientists, whose expertise drives the choice towards images with certain characteristics (being class-prototypical instances or making the recognition particularly challenging, see [3]); important examples are the Caltech-101/256 [4, 5], MSRC [6], the PASCAL VOC series [7], LabelMe [8] and Lotus Hill [9]. Of course such choice is not arbitrary, but the criteria of choice are left implicit and so are the criteria of identity of the target object which is detected (or, better, constructed). As long as we are only concerned with object recognition tasks, probably this is not such a big issue, but when such tasks are part of more complex processes that include visual inference, this could constitute a drawback. Another relevant drawback is that building object recognition datasets is costly and thus the number of images that are collected is limited.

To overcome the disadvantage of having few training images per class and, in general, few object classes, in the last years projects have emerged, which exploit the so called “wisdom of crowd” to populate object recognition datasets, through *web-based* data collection methods. The idea is to employ web-based annotation tools that provide a way of building large annotated datasets by relying on the collaborative effort of a large population of users [10, 11]. The outcome consists of millions of tagged images, but usually of these only few are accessible, and they are not organized into classes by a proper taxonomy.

Differently, one of the most important web-based projects which focuses on the concept of class is ImageNet [12]. ImageNet takes the tree-like structure in which words are arranged in WordNet [13] and assigns to each word (or, better, to each *synset* of WordNet) a set of images that are taken to be instantiations of the class corresponding to the synset. The candidate images to be assigned to a class are quality-controlled and human-annotated through the service of the Amazon Mechanical Turk (AMT), an online platform on which everyone can put up tasks for users, to be completed in order for them to get paid. Nowadays, ImageNet is the largest clean image dataset available to the vision research community, in terms of the total number of images, number of images per category, as well as the number of categories (80K synsets).

Apart from these advantages, an important fact that should be discussed is *where the images come from*. In ImageNet, the source of data is Internet, so that the ImageNet project *partially* falls in the category of those approaches which build training sets by performing *automatic retrieval* of the images [14, 15]. In very general terms, the idea consists in using a term denoting the class of a target object as keyword for an image search engine and forming with the images retrieved in this way the training set. Search engines index images on the basis of the texts that accompany them and of users' tags, when they are present.

The obvious advantage of these approaches is that they can use a great amount of images to form the training set; on the other hand, the training set obtained in this way depends on the ranking of the images, that is, the first images provided by a search engine (say Google Images) are those which rank high in its indexing system. This is not beneficial for our purpose, since we would like to obtain a set of images covering the visual heterogeneity of a visual concept, and not only prototypical instances. As an example, we can take a look to the first 20 images retrieved by Google Images when using the keyword "cat" (Fig. 1). As visible, in most of the cases the cat is frontal, on a synthetic background, focusing on the snout.

These considerations suggest that, starting from the simple image search of Google, many steps ahead could be taken towards the creation of an expressive dataset.

So, the challenging question we will try to answer is: how is it possible to exploit the big amount of images that are available on the web and to automatize the search, providing a training set of pictures which mostly represent the variety of a given concept?



Figure 1: First 20 images obtained by searching “cat” in Google Images. The order (row-major) follows the ranking given by the search engine.

Our proposal is to refine the web search by adding to the standard keyword denoting the class of objects to be detected some other related terms, in order to make the search more expressive. However, we would like these terms to be added not to be arbitrarily chosen, but rather selected with a criterion that has to be explicit and meaningful. More specifically, we would like such accompanying keywords to have three important features:

1. to be frequently associated with the word denoting the target object (otherwise, too few images would be retrieved by the association of the two keywords);
2. to be meaningful from a visual point of view (as usually people tag pictures on the basis of what is depicted in them)
3. to capture the maximum possible level of variability of the addressed class.

Our approach can be summarized as follows: in the first step, we consider a large textual dataset (Google Ngram¹), containing 930 Gigabytes of text material; from Google Ngram we extract bi-grams containing the word denoting the target object (for simplicity, let’s call it “target word”) plus other terms, associated with their frequency in the dataset. In the second step, this input is filtered in various ways, distilling information useful for capturing the visual variability of the object of interest. To this aim, WordNet will be exploited. More specifically, among the most frequent *nouns* that accompany the target word in the bi-grams, *hyponyms* will be kept, thus capturing entities which belong to subclasses of the object of interest. *Adjectives* denoting visual properties will be also kept, that is, adjectives which characterize visible aspects of objects (their color, their patterns). Finally, among *verbs*, present participles are kept, in order to capture actions that

¹<https://books.google.com/ngrams/>.

can be performed or are performed by the entity of interest.

In the third step, these aspects will be fused together following two different criteria: in the first “frequency based” one we choose, among all selected words, those that, coupled with the target word, have the highest score in terms of frequency (disregarding whether they are visual adjectives, verbs or hyponyms). The final result of such process will be a list of pairs of words, composed by the target word plus an accompanying word, chosen with explicit and semantic criteria that, fed into image search engines, will provide semantically rich shots for training the object classifiers.

In the second strategy, we build three separate image sets, including bigrams formed by target word + visual properties, by target word + hyponyms, and by target word + verbs, respectively. These are then fed into three separate classifiers, whose classification decisions on a given test sample are subsequently fused using standard fusion rules. In addition, a “grounding” operation is adopted to reduce polysemy issues: it is assumed that, at the moment of the definition of a target word, a more generic term is also given (an *hypernym*). This term is added to all the strings created so far. Experimentally, this ensures a semantically more coherent image collection.

The aim of the experiments is to validate the goodness of the training datasets automatically built by our method, under different respects. We take inspiration from the ImageNet paper [12], following some of its experimental protocols. In first instance, we analyze the object classification accuracy derived from our data, mainly focusing on the PASCAL VOC 2012 “comp2” competition. This is carried out evaluating different classifiers, from very straightforward (K-Nearest-Neighbor, KNN) to more advanced (Convolutional Neural Networks, CNN [16]); we also evaluate the number of outliers produced by our system. In addition, we explore how the performance varies when the number of images employed changes; finally, we focus on different datasets, evaluating how generalizable the results on different visual scenarios are. In all cases, the results are encouraging, obtaining classification performances not too distant from those obtained from the man-made training set.

The rest of the paper is organized as follows: in Sec. 2 we report the related literature, formed by a very few approaches; in Sec. 3 we present our framework, detailing all the steps and fusion strategies that characterize it. In Sec. 4 we discuss the experimental results obtained and, finally, in Sec. 5 conclusions and issues to be addressed for future developments of the approach are discussed.

2. Related literature

Building object recognition training sets in an automatic fashion is a very recent challenge, born in the robotic field within at least two robot competitions: the *Semantic Robot Vision Challenge* (SRVC)², and *RoboCup@Home*³. Both competitions consist in letting a robot explore autonomously a previously unknown environment, locating specific objects, based on training data collected online or from Internet image searches. One of the most well-known system is Curious George [14]: the starting point of the self-training process consists in crawling a pool of images of the selected target word from Google. After that, the sequence of images is processed by a set of noise removal and ranking operations, which essentially cluster similar images, pruning away groups with too few elements. Groups with more images are ranked first. This system is especially suited for dealing with the robotic scenario, where the robot can acquire multiple shots, which are then matched with the image clusters; having highly populated clusters ensures a robust matching. The approach in [17] extends Curious George, by implementing an attention scheme that allows it to identify interesting regions that correspond to potential objects in the world. In both cases, the recognition scenario is different from ours, since multiple images of the same object are used as input of the classifier system, while we expect a single test image. Anyway, in both cases the first processing step for learning the appearance of an object is retrieving a set of images with the Google Images search engine, fed with a single target word. In [18], the problem of populating an image dataset for learning visual concepts is faced by focusing on images with explicit tags; in particular, they propose a way to predict the relevance of the tag list associated with the images w.r.t. a target concept. In our work, we prefer to disregard the investigation of tags already associated to the images; instead, our aim is to produce textual tags which are semantically relevant for the key concepts that we are considering, and feeding an image search engine with those tags. A massive automatic retrieval of images for the training of object detectors is proposed in [15], where, similarly as in [14, 17], simple image search by Google is used to populate the classes, but, differently from the the latter methods, no postprocessing is implemented. For this reason, we consider this process of data acquisition as competitor to our approach.

²<http://www.semantic-robot-vision-challenge.org/>.

³<http://www.robocupathome.org/>.

3. Method

Our system aims at extracting from Internet a set of images representing the input target word \mathbf{x} ; in order to reduce ambiguity, such word is associated with its hypernym \mathbf{h} . Both the words are selected by a human user and expressed in English⁴. The approach is formed by three steps, the first two of them are in common, while the third one is different depending on which one of the two versions is considered, that is, the *frequency-based* combination version (outlined in Fig. 2) and the *classification-based* combination version (outlined in Fig. 3).

In the first step of our approach, the target word \mathbf{x} is used to extract and filter from Google Ngram all the bi-grams in the form

$$\{\mathbf{x}\mathbf{y}_n\} \cup \{\mathbf{y}_n\mathbf{x}\} \cup \{\mathbf{y}_a\mathbf{x}\} \cup \{\mathbf{y}_v\mathbf{x}\} \quad (1)$$

where \mathbf{y}_n is a noun, \mathbf{y}_a is an adjective and \mathbf{y}_v is a verb, and the order of the variables matters, meaning that the noun can both follow and precede the target word, while the adjective and the verb must precede it. In addition, occurrence frequencies of the bi-grams are also collected as metadata. The number of bi-grams filtered is K , and is not selected a priori, since it depends on the number of entries in the corpus.

The second step consists in performing a set of three operations of semantic filtering: in the case of nouns, the set $\{\mathbf{y}_n\}$ will be filtered and turned into $\{\mathbf{y}'_n\}$, thus obtaining a set of M_n hyponyms of \mathbf{x} ; in the case of adjectives, $\{\mathbf{y}_a\}$ will be filtered and turned into $\{\mathbf{y}'_a\}$, containing M_a *visual* adjectives only, that is, adjectives expressing visual properties of the object of interest that can be observed with a camera. Finally, $\{\mathbf{y}_v\}$ will be transformed in $\{\mathbf{y}'_v\}$, distilling M_v verbs and obtaining only present participles, i.e. the linguistic form in which actions and states are usually expressed. Even in this case, M_n , M_a and M_v are not predefined, but depend on the content of the corpus.

In the third step, two choices are available, corresponding to two different versions of our system: the *frequency-based* combination (Fig. 2) and the *classification-based* combination version (Fig. 3); please note that in all cases,

⁴Experiments with other languages have not been yet performed, since the sentence structure may vary a lot from language to language and this should also be taken into account. Anyway, analogous procedures can be easily found.

the bi-grams so far obtained are now enriched with the hypernym h attached at the top of them, to handle polysemy.

In the *frequency-based* combination, the bi-grams are collected together in the same ensemble, and used to download N images; the mechanism that brings from the total number of bi-grams $M = M_n + M_a + M_v$ to N images will be detailed in the following. After that, the resulting images are employed to train a single classifier, which is associated to the input word, and used subsequently to classify previously unseen images. It is worth noting that the system may need also a negative set of images (in the case of a binary classifier), for the training process, which is not given here. The idea is that, in a typical classification challenge where C concepts have to be recognized, the negative set of a class is given by the pool of positive images of the remaining $C - 1$ classes, as done in our experiments. Alternatively, one can choose to use a generative classifier, or a one-class discriminative classifier, in which case a negative set is not needed anymore. Following these considerations, the system is fully automated.

The *classification-based* combination consists in downloading N images from the hyponyms, visual adjectives and participles bi-grams, respectively, and use them as training data for three different binary classifiers (one for each kind of bi-gram: hyponyms, visual adjectives, participles). Once trained, they will be used to classify a given test image, averaging their confidence score and producing the final decision.

In the following, each phase of the approach will be fully detailed.

3.1. Corpus interrogation and filtering

The initial input is the keyword x and related hypernym h . The first step of the process consists in downloading from Google Ngram all the bi-grams in the form xy or yx , that is, having x as first or second term. As an example, let us focus on $x = \text{“cat”}$. For each bi-gram, Google Ngram provides a pool of metadata, among which there is the frequency of occurrence of that bi-gram in the corpus.

Subsequently, from all bi-grams, only those of the form xy_n , y_nx , y_ax , y_vx , where y_n is a noun, y_a is an adjective and y_v is a verb, are retained and the order of the variables matters, since in English usually a “specifying” noun can both follow and precede the target word, while a qualifying adjective and an adjectival verb precedes it⁵. This operation provides K bi-grams:

⁵The choice of selecting these precise orders is motivated by widely known and long-

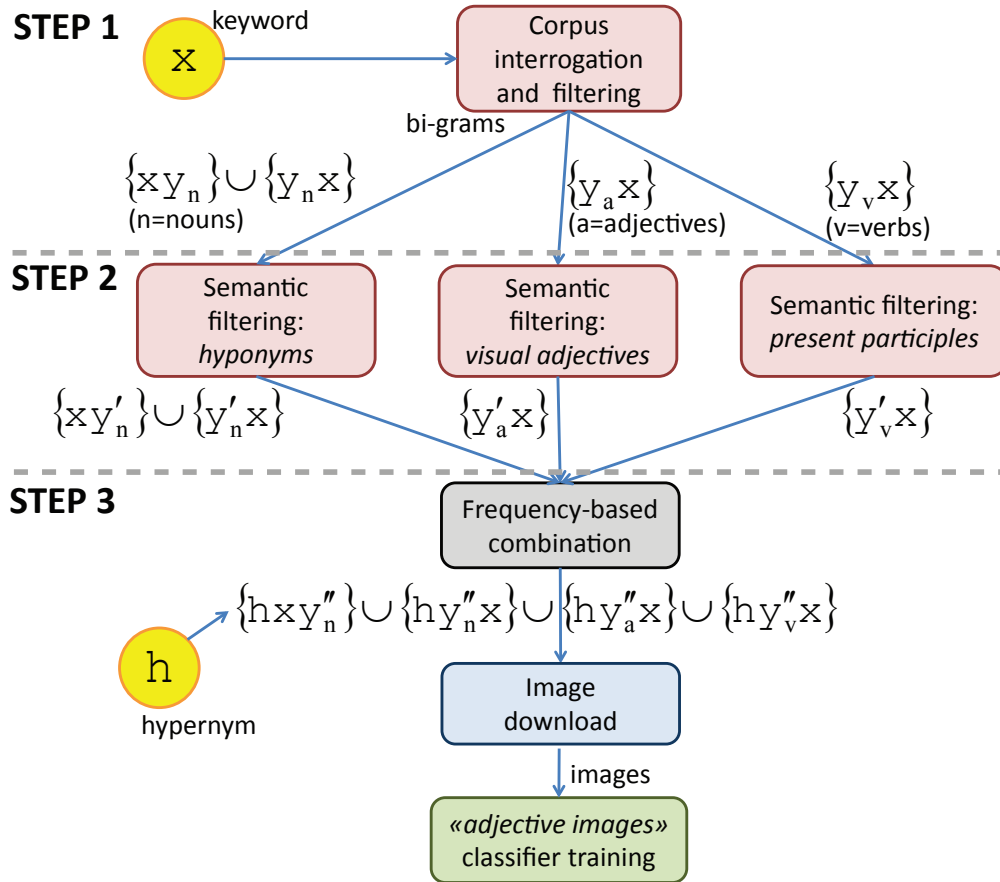


Figure 2: Adopted method, *frequency-based* combination version.

actually, in our experiments, this step prunes away around the 70% of bi-grams initially collected. Table 1 shows the first 10 bi-grams ordered by frequency obtained, using x ="cat"; in this case $K = 11970$, starting from an initial number of 510499 elements.

In this work we have chosen to use Google Ngram, as it is publicly available and already annotated (each word is labeled with its grammatical form, like adjective, noun, etc.), while other corpora, like *Linguistic Data Consor-*

lasting studies in linguistics, such as [19] and [20] (in particular Chapter 4), just to name a few.

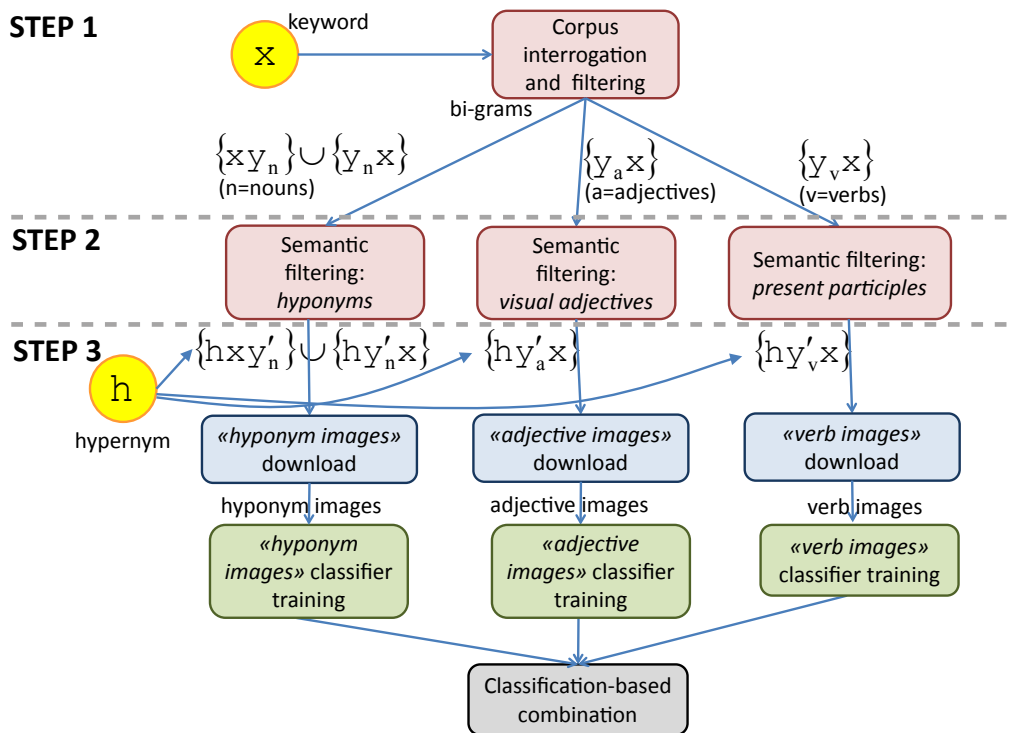


Figure 3: Adopted method, *classification-based* combination version.

*tium Gigawords*⁶, are proprietary. Finally, the Google Ngram corpus is based on *Google books*⁷, so on very heterogeneous sources.

x	Bi-grams after the corpus interrogation and filtering
cat	black cat, wild cat, white cat, old cat, fast cat, big cat, little cat, gray cat, domestic cat, dead cat.

Table 1: Extracted bi-grams: the first 10 bi-grams ordered by frequency obtained when using $x = \text{“cat”}$.

The second phase consists in a set of three semantic filtering operations, which restrict the pool of bi-grams to have the additional words xy_n , y_nx , y_ax , y_vx , belonging to the following sets: hyponyms, visual adjectives, and

⁶<http://catalog.ldc.upenn.edu/LDC2003T05>

⁷<http://books.google.com/>

present participles respectively.

3.2. Semantic filtering: hyponyms

In this case we focus on noun-noun bi-grams $\{\mathbf{x}y_n\}$, $\{y_n\mathbf{x}\}$. Among all the bi-grams of this kind, the interest is focused on those in which the noun y_n is a hyponym of \mathbf{x} . This is aimed at capturing many diverse specifications of the target word under analysis, and as a consequence highly heterogeneous images. To this sake, WordNet is deployed [13], checking whether y_n s are *hyponyms* of \mathbf{x} .

WordNet is a lexical resource structured as a tree, whose nodes are connected by lexical and semantic relations; each node in WordNet is a *synset*, and some of the relations connecting synsets are *hyponymy* (linking a more generic concept to more specific ones) and its opposite relation, *hypernymy* (linking a more specific concept to more general ones), *meronymy* (linking concepts denoting a certain entity with concepts denoting its parts), and so on.

We decided not to use hypernyms at this stage, because, given the fact that they are more general, the risk is that they would retrieve images of objects that do not belong to the class of interest, but to some “sibling” class. In addition, a correct hypernym is already given as input to the system, that is, \mathbf{h} , which will be used directly in the image collection step.

Moreover, we decided not to use meronymy, both because parts of the objects are very often not visible in pictures and, when they are, if the term denoting them is used in association with the target word, the search would probably render many images of the part itself rather than of the object. This is due to the fact that linguistically people tend to disambiguate the reference of the name of a part specifying the object it is part of, rather than vice versa. From this pruned dataset, we obtain M_n bi-grams, ranked in descending order of frequency, obtaining a subset of $\{\mathbf{x}y_n\}$, $\{y_n\mathbf{x}\}$, namely, $\{\mathbf{x}y'_n\}$, $\{y'_n\mathbf{x}\}$. Table 2 shows the first 10 hyponym bi-grams ordered by frequency obtained when using \mathbf{x} = “cat”, out of the $M_n = 611$ total bi-grams retrieved for this target word.

3.3. Semantic filtering: visual adjectives

For the subset of bi-grams adjective + target word $\{y_a\mathbf{x}\}$, ranked according to their frequency, we can filter those that are relevant from a visual point of view. We will do this by “climbing up their WordNet tree of hypernyms”, until the upper-most level is reached. In case we find among the

x	Hyponym bi-grams
cat	domestic cat, house cat, wild cat, siamese cat, persian cat, european cat, sand cat, egyptian cat, angora cat, maltese cat.

Table 2: Hyponym bi-grams: the first 10 hyponym bi-grams ordered by frequency obtained when using \mathbf{x} ="cat".

hypernyms "visual property" or "bodily property", we keep the bi-gram and use it for the search, otherwise we discard it. The choice to use adjectives as first components of bi-grams is motivated by the fact that we want to search for objects on the basis of the qualities that are most often used to describe them. The decision to filter out all those qualities that are not specifications of a visual or a bodily property is a consequence of the fact that we are going to search for images, and so what we are mainly interested in are the adjectives used to describe the visual appearance of the objects they depict. Finally, we have chosen to constrain the order of the words by making the adjective precede the target word, as in discourse the adjective referred to a noun most of the times precedes it, rather than following it. From this pruned dataset, we obtain a subset composed by M_a entries, ranked in descending order of frequency. Thus, we end with a selection of the visual adjective + target word set $\{\mathbf{y}'_a \mathbf{x}\}$. Table 3 shows the first 10 visual adjective bi-grams ordered by frequency obtained when using \mathbf{x} ="cat", out of the $M_a = 1949$ total bi-grams retrieved for this target word.

x	Visual adjective bi-grams
cat	black cat, white cat, gray cat, orange cat, grey cat, blue cat, red cat, green cat, brown cat, pink cat.

Table 3: Visual adjective bi-grams: the first 10 visual adjective bi-grams ordered by frequency obtained when using \mathbf{x} ="cat".

3.4. Semantic filtering: present participles

Bi-grams containing verbs $\{\mathbf{y}_v \mathbf{x}\}$ are also useful to improve the quality of the image search, in order to capture the target objects in their contexts. A huge amount of images in the Web have been uploaded by users and depict objects in certain situations, like being in a particular state (for instance sitting) or performing an action (e.g. running, being eaten, etc.). But even in this case, we are interested in words that specify the search, so in a certain

sense we would like to use verbs as if they were properties associated to the target object. In discourse this is accomplished by using the adjectival form of verbs, therefore using them in the present participle form. Like true adjectives, they usually precede the object they refer to, so we constrain their order of appearance in the bi-grams. From this pruned dataset, we obtain a set of M_v bi-grams, ranked in descending order of frequency. This produces a subset $\{y'_v \mathbf{x}\}$. Table 4 shows the first 10 present participle bi-grams ordered by frequency obtained when using \mathbf{x} ="cat", out of the $M_v = 587$ total bi-grams retrieved for this target word.

\mathbf{x}	Present participle bi-grams
cat	playing cat, sleeping cat, purring cat, looking cat, hunting cat, talking cat, using cat, missing cat, fishing cat, prowling cat.

Table 4: Present participle bi-grams: the first 10 present participle bi-grams ordered by frequency obtained when using \mathbf{x} ="cat".

3.5. Combining the bi-grams: two policies

After collecting the subsets $\{\mathbf{x}y'_n\}$, $\{y'_n \mathbf{x}\}$, $\{y'_a \mathbf{x}\}$, $\{y'_v \mathbf{x}\}$, we propose two ways to proceed: the former, *frequency-based* combination, where the pool of bi-grams are collected together and used to crawl images from the web; the latter, *classification-based*, where the bi-grams sets are kept separated, and used to download three separate image datasets. These two strategies (visible in Fig. 2 and Fig. 3, respectively), are detailed in the following.

Frequency-based combination strategy. In this strategy, all bi-grams are pooled together, keeping trace of the frequency scores associated to them. These scores allow to perform a ranking, from which we take the first ten bi-grams, independently from their semantic nature (nouns, verbs, adjectives). This gives a new set formed by $\{\mathbf{x}y''_n\}$, $\{y''_n \mathbf{x}\}$, $\{y''_a \mathbf{x}\}$, $\{y''_v \mathbf{x}\}$. Table 5 shows the 10 bi-grams ordered by frequency obtained when using \mathbf{x} ="cat", resulting from the *frequency-based* combination strategy.

At this point, for each bi-gram we take $N/10$ images (enriching each bi-gram with the hypernym \mathbf{h}). As an alternative, we try to fix the number of images proportionally to the frequency of the bi-grams, but experimentally this brought to slightly inferior results. Our composite pool of images is fed into a single binary classifier, which can be trained without a negative class (ex.: one-class Support Vector Machine, a generative classifier) or with an

x	Frequency-based combination bi-grams
cat	black cat, white cat, domestic cat, house cat, gray cat, playing cat, orange cat, grey cat, sleeping cat, blue cat.

Table 5: Frequency-based combination bi-grams: the 10 bi-grams ordered by frequency obtained when using $x=$ “cat”, resulting from the frequency-based combination strategy.

arbitrary negative class. In the case of a standard object classification task with C classes, the negative class may be composed by pooling together the remaining $C - 1$ classes. In Fig. 4, 20 images resulting from the image search are reported, and in particular, in row-major order two images corresponding to the related bi-grams listed in Table 5, for each bi-gram.



Figure 4: “Cat” images obtained by the frequency-based combination strategy. In row-major order are reported two images corresponding to the related bi-gram listed in Table 5, for each bi-gram.

As visible, comparing these images with that of Fig. 1, one can immediately notice the higher heterogeneity, in pose, appearance and scale.

Classification-based combination strategy. Here the idea is to design a specific classifier for each of the three subgroups of bi-grams so far obtained, using as positive set $\{xy'_n\}$, $\{y'_n x\}$, $\{y'_a x\}$, $\{y'_v x\}$, respectively, with N images, where each bi-gram is enriched with the hypernym \mathbf{h} ; as negative sets, the same considerations made for the previous strategy are applied. When a test image has to be evaluated, the three classifiers generate three values, expressing the probability of belonging to that class. A final classification is performed by applying the standard average vote (experimentally, we observed that the majority, min and max fusion rules perform worse).

4. Experiments

In this section, we intend to show the quality of the produced training sets under different perspectives. First, we use the simple K -Nearest-Neighbor

(KNN) classifier to get a better insight into our approach, analyzing the intermediate results of the process. Then, we employ a state-of-the-art classifier to compare and contrast the performances of our training sets.

4.1. Dataset creation

For our experiments, we rely on the Google Images search engine to automatically gather images from the Internet. We take the image classes contained in the PASCAL VOC 2012 dataset⁸ [7]: “aeroplane”, “bicycle”, “bird”, “boat”, “bottle”, “bus”, “car”, “cat”, “chair”, “cow”, “dog”, “horse”, “motorbike”, “dining table”, “person”, “potted plant”, “sheep”, “sofa”, “train”, “tv/monitor”. Three are the reasons of our choice of the PASCAL VOC 2012: the object of interest is not always in the center of the image, it is not restricted to have as the only instance of its class the object in the picture (a typical setting of the Caltech datasets and the older repositories [4, 5]), and it is a very popular benchmark in the literature.

As first analysis, we automatically generate 5 different training sets of $N = 100$ images each, for all the VOC classes; each dataset corresponds to one particular intermediate result of our strategy, in particular:

Basic filter (*basic*) we use as keywords the top 10 bi-grams obtained from the Ngram corpus, by applying the basic filter described in Sec. 3.1, (see Table 13);

Hyponyms (*hyp*) keywords are the top 10 bi-grams obtained by applying the hyponyms selection filter described in Sec. 3.2, (see Table 14).

Visual adjectives (*vadj*) we use as keywords the top 10 bi-grams obtained by applying the visual adjectives selection filter described in Sec. 3.3, (see Table 15).

Present participles (*prepar*) keywords are the top 10 bi-grams obtained by applying the present participles selection filter described in Sec. 3.4, (see Table 16).

Frequency combination (*fcomb*) keywords are the top 10 bi-grams obtained by applying the frequency-based combination strategy described in Sec. 3.5.

⁸<http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/>

Please note that the *classification-based* version, *ccomb*, has not a dataset on its own, as it consists of three classifiers trained on images of the *hyp*, *vadj*, *prepar*, respectively.

In the five strategies listed above, we add for each bi-gram the hypernym *h*; in particular, for the class “person” *h* is “being”, for the classes “bird, cat, cow, dog, horse, sheep” *h* is “animal”, for the classes “aeroplane, bicycle, boat, bus, car, motorbike, train” *h* is “vehicle”, and, finally, for the classes “bottle, chair, sofa, tv/monitor” *h* is “physical object”. Please note that we do not consider the classes “dining table” and “potted plant” since they are already in the form of bi-gram: adding another term to their specification would generate tri-grams and the comparison would not be meaningful anymore. For each bi-gram in the considered pool (basic filter, hyponym, visual adjective, present participle, frequency combination) we keep the first 10 images provided by Google; this allows to collect $N = 100$ images per class; in case the bi-grams are less than 10, say n , we select the top-ranked $\lfloor N/n \rfloor$ images per bi-gram.

In order to provide an example of how the final *frequency combination* dataset is obtained with our method, we show here an excerpt for each dataset, formed by 20 images each; here, the i -th pair of images (in row-major order) derives from the two top-rank images of the i -th bi-gram being analyzed. In cases in which the number of bi-grams is less than 10, say n , we show the top $\lfloor 20/n \rfloor$ images per bi-gram. The bi-grams and the related images are reported in Fig. 6 for the class “aeroplane”, and in Figg. 7-8 for the classes “cat” (partially discussed in the introduction), and “sofa”, respectively.

In addition, for each class we show the first 20 top-rank images for the Google basic approach (*Google*), that is, images obtained from the Google Images search with the target word. Please note that this approach is also used in [14, 15], so that it has to be considered a standard competitor of our strategy. Finally, we also plot 20 random images of the PASCAL VOC 2012 dataset (*VOC*), as further term of comparison.

Looking at the images of the *Google* dataset, one can immediately notice how the typology of the images is restricted, centered on aircrafts for public transportation, mostly flying, where the dominant color of the vessels is white; this represents an important limitation, since aeroplanes can also be taking on/off, on the floor in the hangar, on maintenance etc. Our methodology solves this problem: starting with the basic filtering on the bi-grams (which exhibits many outliers), the hyponym bi-grams introduce








Type	Top 10 bi-grams and related images
basic	<p>german aeroplane, two aeroplane, curtiss aeroplane, model aeroplane, enemy aeroplane, first aeroplane, aeroplane company, british aeroplane, one aeroplane, aeroplane engines.</p> 
hyp	<p>jet aeroplane, fighter aeroplane</p> 
vadj	<p>light aeroplane, red aeroplane, white aeroplane, silver aeroplane, blue aeroplane, black aeroplane, navy aeroplane, green aeroplane, gray aeroplane, gold aeroplane.</p> 
prepar	<p>flying aeroplane, bombing aeroplane, fighting aeroplane, making aeroplane, carrying aeroplane, scouting aeroplane, building aeroplane, manufacturing aeroplane, wing aeroplane, using aeroplane.</p> 
fcomb	<p>light aeroplane, flying aeroplane, bombing aeroplane, fighting aeroplane, making aeroplane, jet aeroplane, carrying aeroplane, red aeroplane, white aeroplane, scouting aeroplane.</p> 
Google	
VOC	

Table 6: Qualitative analysis of the different datasets related to the class “aeroplane” obtained by applying our strategy of frequency-based combination (*fcomb*), but also showing the intermediate *basic*, *hyp*, *vadj* and *prepar* datasets, together with the competitor *Google* and the original *VOC*. For each bi-gram, two images have been reported, following their row-major ranking in the list.

other kinds of aeroplanes (the military ones); the visual adjective bi-gram set adds some other typologies (light) and provides planes of different colors. Finally, the *prepar* set makes it possible to focus on aeroplanes in many different scenarios. The final *fcomb* dataset takes elements from these previous datasets, exhibiting images definitely more various than those of *Google*, and in this sense most similar to those of the *VOC* dataset.

Anyway, this comes with a price: in facts, in some cases outliers are produced, especially in the case of the *prepar* dataset, in which some verbs are clearly connected to the term “aeroplane” as direct object, and are not used for better specifying the term “aeroplane”. This is the case of “building, manufacturing aeroplane” and “using aeroplane”, that indicate the fact that someone else is building and using the aeroplane, respectively: this brings to images where parts of the aeroplane are portrayed, or images where a toy model of a plane is built, or that show people on a plane. Anyway, the overall effect in terms of classification accuracy (see later) and in terms of outliers suggests that this is not a crucial issue, and that having more various pictures of the object of interest is more important. This reasoning brings in the problem of outliers, what they are, how they are defined, when an image is dubbed as outlier, etc. Such issues will be discussed and analyzed later on in the paper.

The other case analyzed is that of the “cat” category (Fig. 7), whose bi-grams have been already shown in Sec. 3. Even in this case, one can notice that the final *frequency-based* combination dataset is richer in terms of visual heterogeneity with respect to the *Google* search results. It is interesting to note that in some cases “strange” images pop out, for example in correspondence of the green cat; looking at *Google*, many images report cats with green eyes, but the images portrayed here are the most ranked ones. A similar argument holds for “pink cat”, which in the text usually specifies the Sphynx cat, but here we have these painted-pink cats as the highest ranked image. Apparent outliers are also present here, like the “talking cat”, represented by synthetic images.

The last case analyzed is that of the “sofa” category (Fig. 8). The considerations that could be assessed in this case are similar to those reported for the other target words, that is, our pool of images appear to report more typologies of the target word taken into account (“sofa bed”, “convertible sofa”), with many images where the object denoted by the target word is embedded in a real scenario; sofas are often in a room and the illumination, scale, pose are diverse; in some cases we can see also people seated on them;








Type	Top 10 bi-grams and related images
basic	black cat, wild cat, white cat, old cat, fast cat, big cat, little cat, gray cat, domestic cat, dead cat. 
hyp	domestic cat, house cat, wild cat, siamese cat, persian cat, european cat, sand cat, egyptian cat, angora cat, maltese cat. 
vadj	black cat, white cat, gray cat, orange cat, grey cat, blue cat, red cat, green cat, brown cat, pink cat. 
prepar	playing cat, sleeping cat, purring cat, looking cat, hunting cat, talking cat, using cat, missing cat, fishing cat, prowling cat. 
fcomb	black cat, white cat, domestic cat, house cat, gray cat, playing cat, orange cat, grey cat, sleeping cat, blue cat. 
Google	
VOC	

Table 7: Qualitative analysis of the different datasets related to the class “cat” obtained by applying our strategy of frequency-based combination (*fcomb*), and showing the intermediate *basic*, *hyp*, *vadj* and *prepar* datasets, together with the competitor *Google* and the original *VOC*. For each bi-gram, two images have been reported, following their row-major ranking in the list. Dead cats images have been removed for ethical reasons.

Type	Top 10 bi-grams and related images
basic	leather sofa, room sofa, sofa bed, old sofa, sofa cushions, two sofa, small sofa, horse-hair sofa, comfortable sofa, sofa beside. 
hyp	sofa bed, convertible sofa, divan sofa. 
vadj	green sofa, white sofa, red sofa, blue sofa, brown sofa, black sofa, pink sofa, gray sofa, orange sofa, purple sofa. 
prepar	matching sofa, sagging sofa, looking sofa, facing sofa, inviting sofa, spring sofa, reclining sofa, including sofa, lounging sofa, imposing sofa. 
fcomb	sofa bed, green sofa, convertible sofa, white sofa, red sofa, blue sofa, matching sofa, sagging sofa, brown sofa, looking sofa. 
Google	
VOC	

Table 8: Qualitative analysis of the different datasets related to the class “sofa” obtained by applying our strategy of frequency-based combination (*fcomb*), and showing the intermediate *basic*, *hyp*, *vadj* and *prepar* datasets, together with the competitor *Google* and the original *VOC*. For each bi-gram, two images have been reported, following their row-major ranking in the list.

all this is absolutely absent in the images of *Google*.

Summing up these qualitative observations, we can state that the dataset produced by our method is actually a compromise between those benchmarks which focus mainly on the item of interest, discarding the rest (like the Caltech series, see later in the paper) and the ones which capture the objects in their context (like PASCAL VOC series). Each of these two paradigms of object visualization (1-discarding the background, 2-including the background) have pros and cons: in the former, the classifier can capture the precise essence of the object of interest, without being distracted by other entities in the scene. On the other hand, capturing the context is without any doubt a key element for inferring the nature of an object (given the fact that I recognize a road in the image, it is more probable to observe a motorbike than a shark on top of it). That is to say, the datasets produced by our approach seem to be more general than those hand-crafted by scientists so far. In the following, we will validate this assumption experimentally.

4.2. Evaluating the number of outliers

When evaluating a procedure which builds a dataset for object recognition, it is important to check how many outliers have been produced. The lower is the number of outliers in a dataset, the more precise is the classification model in avoiding false positives.

This introduces a much more intriguing question, that is, how to distinguish true positives from outliers. In some cases the decision is straightforward: images in which the target object is the main subject are positive, those in which no instance of the target object is present are negative. But what about more ambiguous cases, like photos of parts of the object, pictures that are caricatures or cartoons, images in which the object is not in the foreground and is surrounded by several other different objects? Deciding which images to include in a positive or in a negative training set is a general problem, which lacks *the* best solution. The goodness of the choice strongly depends on the purpose of the classification. Suppose the goal of the classification is to retrieve the largest number of representations of the target object; probably one would like to have a “permissive” classifier that includes as instances of the objects all the examples mentioned above. But if the classification task is part of a more complex endeavor, like for instance that of enabling a robot to recognize an object, grab it and use it for accomplishing a precise action, then we would want the classifier to work in a more “restrictive” way. Our long term vision is to use classification as a first

step of a reasoning process on the connections of the various objects in an environment and on the events in which they are involved. Ontology-based approaches provide the formal tools to distinguish an object from its parts, from the event it participates to and from the representations of it – just to name a few – and allow to infer new properties and relations of such object by leveraging on the axioms that explain the connections between all these elements.

This is the main reason why we have chosen to use a restrictive strategy in dubbing as outliers (see some examples in Table 9:

- images completely unrelated with the object;
- irrelevant parts of the object, that is, parts that alone are not sufficient to make the object identifiable;
- internal parts of the object (like the cockpit of an aeroplane);
- the object in the background;
- drawings and caricatures of the object.






Type	Example images
unrelated	
irrelevant part	
internal part	
background	
drawings	

Table 9: Example of outliers images for class “car”.

Following these annotation guidelines, we analyze all the images of the classes found by our *fcomb* approach and those found by the *Google* method,

reported in Table 10⁹. As a general note, we can see that we reduce the outliers rate only in half of the classes, while we allow more outliers in the second half. In particular, in two classes we increase the number of outliers of a significant amount (person and tv/monitor), but, as we will see, we do not decrease performances in the classification task. In our opinion this is because our method, though increasing the number of outliers for such cases, at the same time ensures a wide variety in terms of training images: different kinds of the target objects and different viewpoints. In this way we are able to avoid problems related to overfitting of a particular kind of target object – i.e. 90% of the images of person collected with the *Google* method are actually ‘faces’.

Class	<i>Google</i>		<i>fcomb</i>	
	outliers	good	outliers	good
Aeroplane	69	131	64	136
Bicycle	65	135	43	157
Bird	50	150	51	149
Boat	20	180	51	149
Bottle	17	183	52	148
Bus	30	170	64	136
Car	30	170	47	153
Cat	80	120	55	145
Chair	5	195	23	177
Cow	52	148	47	153
Dog	48	152	46	154
Horse	66	134	65	135
Motorbike	69	131	60	140
Person	11	189	102	98
Sheep	58	142	53	147
Sofa	7	193	13	177
Train	76	124	72	128
Tvmonitor	37	163	116	84

Table 10: Comparison between *Google* and *fcomb* with respect to outliers’ handling.

Concluding this section, we believe that being restrictive in labeling an

⁹In general, the outliers of *fcomb* and of *ccomb* are in similar proportions.

image as inlier is also a good practice, given that it is generally easier to lessen constraints rather than to strengthen them.

4.3. Object recognition by KNN classification

Inspired by [12], a KNN approach is used to test the dataset produced by our method, considering both the *frequency-based* combination strategy (*fcomb*), and adding the *classification-based* combination strategy *ccomb*; we consider also the datasets obtained by the intermediate steps of our method discussed in the previous qualitative experiment, that is, *basic*, *hyp*, *vadj*, *prepar*.

In the experiment evaluating the *fcomb* methodology, for each class, we build a binary classifier by using N positives, where N is the dimensionality of the PASCAL VOC 2012 training set for each class, and the same number of negative training samples; the positives taken from the *fcomb* dataset, the negatives randomly taken from the positive samples of the other classes, in a uniform way (that is, each class contributes with the same number of elements in creating the negative class). We resize all the images, both from the training and the testing sets, to 32×32 pixels; we compute then the feature descriptors simply by considering the RGB coordinates of each pixel. For selecting the neighbors, we use as metrics the sum of squared distances (SSD). Each positive training sample that has been individuated as neighbor of a test image votes ‘+1’ for that image, a negative neighbor gives ‘-1’; the summation of all the votes individuates the winning class (considering the sign) and a sort of “confidence” by considering the module.

For evaluating the *ccomb* methodology, a classifier for each of the positive datasets *hyp*, *vadj*, *prepar* is instantiated, the negative being the same dataset of the previous trial. This way, each classifier gives a signed score measuring the *confidence* of having a test set belonging to a particular class (or its negative). These three confidences are then mediated to get the final classification score.

To indicate the number of neighbors, we select $K = 49$. To evaluate performances, we employ PASCAL VOC’s interpolated *average precision* (AP) [21]: the precision/recall curve is interpolated by using the maximum precision observed across all cutoffs with higher recall, and the AP is the value of the area under this curve; in practice this metric penalises approaches which classify only a subset of images with high precision (see [7] for more details).

As competitive approaches, we include the *Google* approach [14, 15], that is, considering as positive the N top ranked images obtained by searching the target word with Google Images search; as reference, we consider also the results obtained with the PASCAL VOC 2012 training set. As testing set, the whole PASCAL VOC 2012 validation set has been considered. The results are shown in Fig. 5.

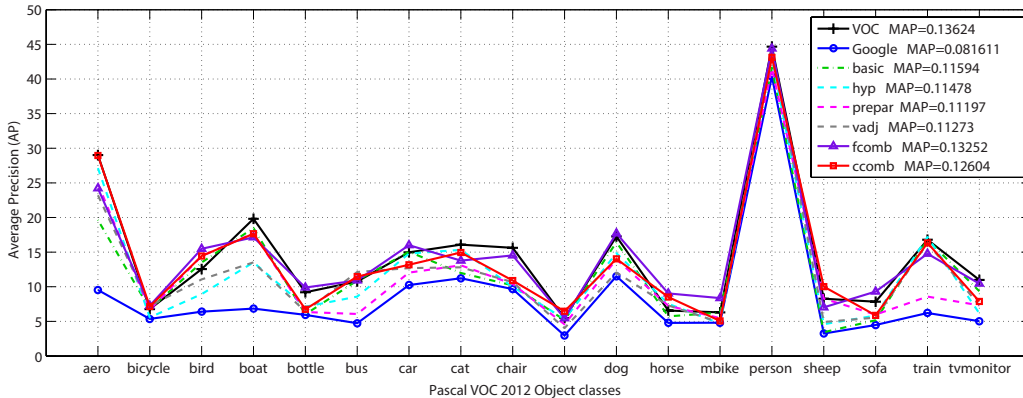


Figure 5: AP values on each Pascal VOC class obtained by the KNN classifier, comparing the two strategies of our approach (*fcomb* and *ccomb*), the intermediate strategies *basic*, *hyp*, *vadj*, *prepar*, the comparative approach *Google* [14, 15] and the reference *VOC*. MAP stands for mean AP, computed on all the per-class APs. Better viewed in colors.

Even if the scores are quite low (we are facing a hard problem with a straightforward classifier), the results lead to some evident conclusions: 1) using solely the Google Images search engine for creating an object recognition dataset is not very effective; in practice, the reasons are explained in the previous qualitative experiment - technically speaking, our datasets capture in a better way the intra-class visual variance; 2) enriching the target word with some additional terms coming from one of our intermediate strategy *basic*, *hyp*, *vadj*, *prepar* boosts the performance; 3) the *frequency-based* fusion *fcomb* version gives the highest performance (MAP=0.13252) among our strategies, followed by the *classification-based* combination version *ccomb* (MAP=0.12604). 4) Our two strategies are not so far from the performance obtained by the PASCAL VOC training set, especially the *fcomb* version. In particular, looking at the curves, we can observe that in some cases the AP obtained with our two strategies is slightly higher than that obtained by the VOC dataset (see the AP related to the classes “bird”, “bottle”, “car”,

“dog”, “horse”, “motorbike”, “sofa”). This fact can be explained once again by the high heterogeneity enforced by our semantically driven image collection system. As a confirmation, one can simply observe Table 8 concerning the “sofa” class: here the typology of our images (see the *fcomb* row) match better than the other methods the VOC’s typology.

4.4. Object recognition using Convolutional Neural Networks

In this experiment, we follow one of the leading approaches in large scale object recognition, namely Convolutional Neural Networks (CNNs). Popularized by the performance of [22] on the ImageNet 2012 classification benchmark, CNNs have been shown to be excellent features extractors when used on different datasets w.r.t. the one originally used for training [23]. In particular, we use a publicly available pre-trained CNN [24] to retrieve the weights in the 7th layer of the network when it is forward-fed with input images (see [16] for more details). We then use these 4096-dimensional sparse vectors to train a linear SVM [25] for each object class, optimized on a random half of the VOC validation set and tested on the remaining half.

In Fig. 6, we compare the AP values obtained with our classification-based combination strategy *ccomb* against the stock VOC training data and against the training sets obtained by the *Google* approach [14, 15], using a similar amount and distribution of images as in VOC, with less populated classes, like “cow” and “sheep”, and more populated ones, like “person”. For the sake of visual clarity, we do not report here the performance of the frequency-based combination version *fcomb*, obtaining systematically lower results than *ccomb*. As first evident fact, performance is much higher if compared to the KNN results, due to the sophisticated features extracted from the images by the CNN. At the same time, the difference among the *ccomb* and the VOC approach is higher, with *ccomb* trailing behind all the time. This effect can be understood by considering two facts: first, the CNN feature extractor (in its original version [24]) has been trained on ImageNet clean images, which do not include outliers like drawings, synthetic images etc. Second, and especially for few classes, our approach collects a consistent number of outlier images which actually are drawings, 3D models etc: see for example Table 10, the class “person”, of which some outliers are shown in Fig. 7a.

These two facts may have caused the high discrepancy between the “person” results shown in Fig. 5 and those reported here (Fig. 6). Another observation is that the *ccomb* approach in some cases do not outperform

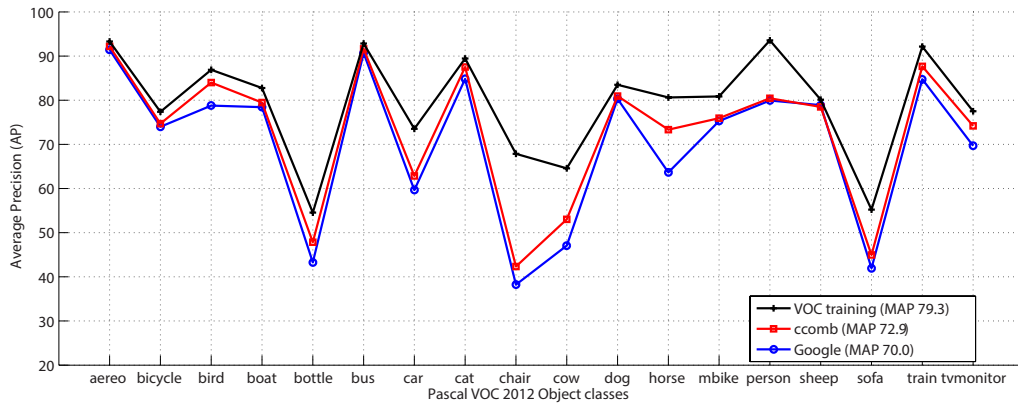


Figure 6: AP values on each Pascal VOC class by the CNN-based classifier trained on the stock training data and the training sets obtained by our proposed method using Google (mean AP is indicated in parentheses).

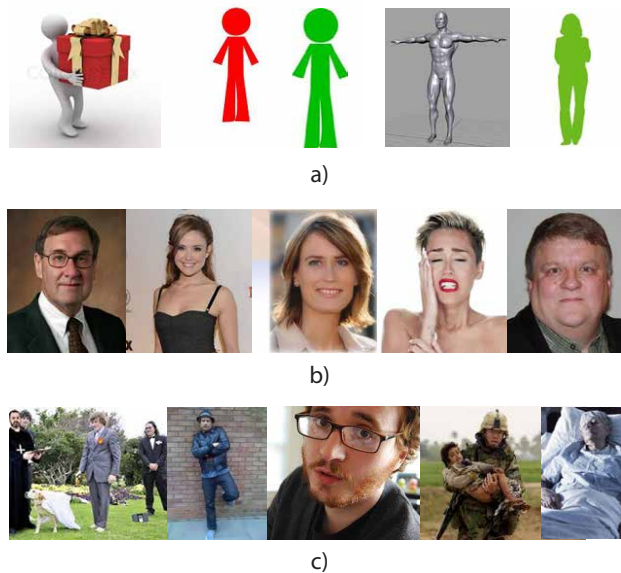


Figure 7: Some outliers of the class “person” regarding the *ccomb* approach (a) and some inliers for the *Google* approach (b) and *ccomb* approach (c).

drastically the *simple* approach. Even in this case, the reason may lie in the higher number of outliers in few cases. Still, it is worth noting that even with noisy samples, our system ensure higher variability, allowing to systematically overcome the *Google* method; as an example, we can focus again on

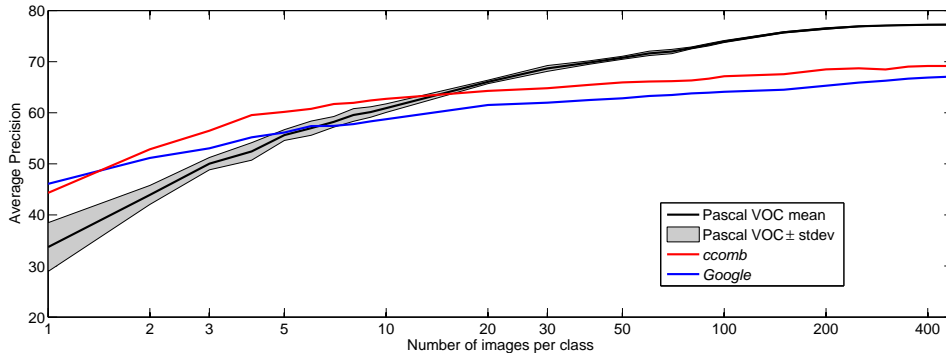


Figure 8: Classification performances while changing the cardinality of the dataset; the abscissa is scaled logarithmically to better show the behavior at low cardinalities.

the class “person”, where the *Google* approach exhibits very similar pictures (Fig. 7b), if compared to our set of inlier images (Fig. 7c).

4.5. Changing the dataset cardinality

A crucial aspect worth investigating in our framework is the classification behaviour when changing the cardinality of the training dataset. In particular, it is interesting to analyze what happens both when the number of available images is small (mimicking a fast, time-constrained system), and when it is arbitrarily large (asymptotic behavior).

Thus, we fix a set of predetermined cardinalities¹⁰ for the size of each class, and evaluate our datasets against *Google* and PASCAL VOC 2012. Since many images in the latter are shared – they contain multiple objects within – we choose to pick the class-exclusive images first and the shared ones after those are exhausted, to keep the negative sets balanced as long as possible. Since the PASCAL VOC images are not ranked like the searched images, every experiment at a given cardinality is repeated 10 times on random draws (giving priority to the class-exclusive first) and averaged.

For *Google* and *ccomb*, the approach is deterministic and amounts to taking one image without replacement from each bi-gram in turn (where the bi-grams are ordered by descending frequency), then starting again from the first bi-gram until all the images have been exhausted. When the desired

¹⁰We choose $\{1, 2, \dots, 10, 20, \dots, 100, 150, \dots, 450, 476\}$ as class sizes; larger sizes are inconvenient to handle and add little information.

cardinality is bigger than the available images for a given class, this class stops growing, and only acquires negative samples through the enlarging of the other classes.

In Fig. 8, we show the resulting APs averaged over all the classes: as expected, PASCAL VOC has considerable variance with low cardinalities and becomes stable as the size increases; *Google* and *ccomb* perform better than PASCAL VOC at the beginning – thanks to the images being more relevant – but are overtaken after size 7 and 15, respectively. Towards bigger sizes, they show a very slight uptrend, but it is unlikely they will reach the PASCAL VOC performance. *ccomb* shows to be an improvement w.r.t. *Google*, due to the bigger number and variety of starting images, which is promising in light of future expansions of our approach. Note that Fig. 6 has slightly different APs because of the mismatched class sizes.

4.6. Evaluating the generalization capabilities

Of significant interest for the practical usefulness of our approach is how well the training datasets generalize beyond the insights gathered on PASCAL VOC. One way to gain an approximate idea is by performing cross-dataset evaluations between different benchmark datasets, and comparing the relative performance of our training sets. Following [3], we set out to explore cross-dataset generalization on two test classes: “person” and “cat” (note that the results of [3] will not be directly comparable). For each class, we perform 10 randomized experiments with 200 positive and 400 negative samples split into 50% for training, 25% for cross-validation and 25% for testing. As source of the negative samples, we still use the “other” classes of PASCAL VOC, and while the negative sets are kept the same, we swap the positive sets from the benchmark datasets, or provide our training sets.

<i>Train on:</i>	<i>Test on:</i>				Mean others
	PASCAL VOC	Caltech-256	ImageNet	GRAZ	
PASCAL VOC	88.46	96.52	93.20	92.10	93.94
Caltech-256	85.96	99.95	98.15	92.78	92.30
ImageNet	85.74	99.75	98.58	93.29	92.93
GRAZ	81.50	97.65	92.91	97.51	90.69
<i>Google</i>	73.29	98.91	95.47	84.71	88.09
<i>ccomb</i>	75.25	99.57	96.56	84.84	89.06

Table 11: Cross-dataset generalization on the class “person”.

In Table 11, we report the APs averaged on 10 runs for the datasets PASCAL VOC, Caltech-256, ImageNet and GRAZ on class “person”. Despite underperforming on PASCAL VOC, the same training sets of *Google* and *ccomb* do very well on Caltech-256 and ImageNet, exceeding PASCAL VOC itself. On average, they display good generalization, on par with hand-crafted training sets, and *ccomb* edges ahead of *Google* by a small margin.

<i>Train on:</i>	<i>Test on:</i>			Mean others
	PASCAL VOC	ImageNet	OxfordPet	
PASCAL VOC	93.87	96.07	98.27	97.17
ImageNet	92.24	96.90	98.15	95.20
OxfordPet	92.36	96.07	99.34	94.22
<i>Google</i>	80.99	92.76	96.32	90.02
<i>ccomb</i>	86.48	96.10	97.93	93.50

Table 12: Cross-dataset generalization on the class “cat”.

In Table 12, we report the APs averaged on 10 runs for the datasets PASCAL VOC, ImageNet, and OxfordPet on the class “cat”. This time, *ccomb* has a larger advantage on *Google* and it is explained by a very good performance of the *prepar* and *vadj* keywords.

5. Conclusions

In this paper we face the new problem of building a training dataset for an object classifier, in a completely automatic fashion. Given a list of objects, for each one of them we want a binary classifier; following the well-known PASCAL VOC structure, the idea is to collect a positive image dataset, where the images portray different visual aspects of the object of interest. So far, this task has been pursued by crawling image search engines by using the target word denoting the object of interest. Here, we make a substantial step ahead, enriching the text strings with elements that model many different aspects of the object under analysis. The results, obtained by using both a standard KNN classifier and a most powerful convolutional net trained on our dataset, promote our idea: specifying objects with hyponyms, visual adjectives and present participles allows to increase the visual variability, getting different information to be encoded by a classifier. The whole process takes around 20 minutes for the PASCAL VOC 2012 dataset, getting performances that are not too far from the figure of merits obtained with the

Class	# bi-grams	Top 10 bi-grams
aeroplane	872	german aeroplane, two aeroplane, curtiss aeroplane, model aeroplane, enemy aeroplane, first aeroplane, aeroplane company, british aeroplane,one aeroplane, aeroplane engines
bicycle	2994	bicycle riding, bicycle shop, bicycle wheel, bicycle ergometer, new bicycle, bicycle race, stationary bicycle, bicycle ride, bicycle pump, riding bicycles
bird	16500	little bird, young bird, small bird, wild bird, game bird, bird sing, migratory bird, white bird, old bird, bird species
boat	3542	small boat, little boat, fishing boat, open boat, torpedo boat, motor boat, flying boat, patrol boat, ferry boat, canal boat
bottle	7964	two bottle, water bottle, glass bottle, empty bottle, one bottle, beer bottle, small bottle, wine bottle, plastic bottle, another bottle
bus	2701	school bus, data bus, address bus, local bus, greyhound bus, city bus, shuttle bus, montgomery bus, pci bus, tour bus
car	5320	new car, motor car, police car, street car, sports car, used car, rental car, patrol car, old car, passenger car
cat	3101	black cat, wild cat, white cat, old cat, fast cat, big cat, little cat, gray cat, domestic cat, dead cat
chair	16158	two chair, rocking chair, easy chair, leather chair, folding chair, empty chair, wooden chair, comfortable chair, one chair, backed chair
cow	31958	two cow, dairy cow, one cow, milk cow, per cow, milch cow, sacred cow, old cow, mad cow, cow dung
dog	45412	dog is, hot dog, little dog, mad dog, prairie dog, old dog, two dog, like dog, other dog, bulldog
horse	5111	white horse, black horse, crazy horse, old horse, good horse, trojan horse, wild horse, light horse, dark horse, dead horse
motorbike	29	small motorbike, new motorbike, old motorbike, red motorbike, honda motorbike, little motorbike, powerful motorbike, bmw motorbike, davidson motorbike, big motorbike
person	8777	young person, single person, second person, average person, particular person, older person, human person, sick person, different person, white person
sheep	8287	black sheep, lost sheep, mountain sheep, bighorn sheep, hundred sheep, thousand sheep, one sheep, two sheep, many sheep, merino sheep
sofa	5347	leather sofa, room sofa, sofa bed, old sofa, sofa cushions, two sofa, small sofa, horsehair sofa, comfortable sofa, sofa beside
train	4731	wagon train, freight train, long train, passenger train, special train, express train, night train, railroad train, railway train, pack train, o'clock train
tvmonitor	3308	cable tv, watching tv, watch tv, color tv, satellite tv, local tv, watched tv, national tv, screen tv, network tv

Table 13: Top 10 bi-grams (when available, otherwise all available bi-grams) obtained from the Ngram corpus, by applying the basic filter.

Class	Top 10 hyponym bi-grams
aeroplane	jet aeroplane, fighter aeroplane
bicycle	safety bicycle, tandem bicycle, ordinary bicycle
bird	flying bird, night bird, aquatic bird, flightless bird, passerine bird, gallinaceous bird, cock bird, hen bird, ratite bird, carinate bird
boat	small boat, ferry boat, canal boat, river boat, pilot boat, mail boat, packet boat, tug boat, police boat, guard boat
bottle	water bottle, beer bottle, wine bottle, whiskey bottle, soda bottle, ink bottle, pop bottle, pill bottle, ketchup bottle, bottle gourd
bus	school bus
car	police car, sports car, patrol car, squad car, race car, touring car, electric car, electric car, stock car, racing car
cat	domestic cat, house cat, wild cat, siamese cat, persian cat, european cat, sand cat, egyptian cat, angora cat, maltese cat
chair	rocking chair, folding chair, swivel chair, lawn chair, side chair, straight chair, barber chair, garden chair, fighting chair, feeding chair
cow	heifer cow
dog	hunting dog, puppy dog, mongrel dog, working dog, toy dog, poodle dog, pug dog, cur dog, dalmatian dog, coasch dog
horse	wild horse, saddle horse, race horse, bay horse, riding horse, sorrel horse, chestnut horse, roan horse, harness horse, female horse
person	good person, dead person, deceased person, deceased person, innocent person, religious person, best person, bad person, married person, male person
sheep	black sheep, domestic sheep, ewe sheep, wether sheep, sheep ram
sofa	sofa bed, convertible sofa, divan sofa
train	freight train, passenger train, subway train, mail train, car train, boat train, hospital train, streamliner train
tvmonitor	cable tv

Table 14: Top 10 bi-grams (when available, otherwise all available bi-grams) obtained by applying the hyponyms selection filter.

Class	Top 10 visual adjective bi-grams
aeroplane	light aeroplane, red aeroplane, white aeroplane, silver aeroplane, blue aeroplane, black aeroplane, navy aeroplane, green aeroplane, gray aeroplane, gold aeroplane
bicycle	red bicycle, blue bicycle, black bicycle, green bicycle, white bicycle, pink bicycle, purple bicycle, silver bicycle, light bicycle, orange bicycle
bird	white bird, black bird, blue bird, red bird, brown bird, canary bird, gray bird, green bird, grey bird, silver bird
boat	light boat, white boat, black boat, red boat, green boat, blue boat, gray boat, orange boat, brown boat, navy boat
bottle	green bottle, brown bottle, black bottle, blue bottle, white bottle, red bottle, light bottle, pink bottle, purple bottle, orange bottle
bus	address bus, blue bus, white bus, red bus, black bus, green bus, orange bus, gray bus, silver bus, brown bus
car	black car, red car, blue car, white car, green car, light car, gray car, brown car, grey car, maroon car
cat	black cat, white cat, gray cat, orange cat, grey cat, blue cat, red cat, green cat, brown cat, pink cat
chair	red chair, green chair, blue chair, white chair, black chair, gold chair, silver chair, brown chair, orange chair, light chair
cow	red cow, white cow, black cow, brown cow, purple cow, blue cow, green cow, gray cow, silver cow, grey cow
dog	red dog, black dog, white dog, brown dog, blue dog, green dog, gray dog
horse	white horse, black horse, light horse, gray horse, red horse, brown horse, grey horse, blue horse, green horse, pink horse
motorbike	red motorbike, black motorbike
person	white person, black person, light person, brown person, red person, green person, polish person, blue person, straw person, bearing person
sheep	black sheep, white sheep, red sheep, blue sheep, brown sheep, green sheep, gray sheep, bearing sheep, grey sheep, silver sheep
sofa	green sofa, white sofa, red sofa, blue sofa, brown sofa, black sofa, pink sofa, gray sofa, orange sofa, purple sofa
train	black train, blue train, white train, red train, light train, green train, gold train, purple train, sable train, silver train
tvmonitor	white tv, color tv, black tv, light tv, colour tv, polish tv, blue tv, red tv, green tv, gray tv

Table 15: Top 10 bi-grams (when available, otherwise all available bi-grams) obtained by applying the visual adjectives selection filter.

Class	Top 10 present participle bi-grams
aeroplane	flying aeroplane, bombing aeroplane, fighting aeroplane, making aeroplane, carrying aeroplane, scouting aeroplane, building aeroplane, manufacturing aeroplane, wing aeroplane, using aeroplane
bicycle	stealing bicycle, renting bicycle, buying bicycle, fixing bicycle, bring bicycle, wheeling bicycle, parking bicycle, dodging bicycle, owning bicycle, having bicycle
bird	singing bird, humming bird, breeding bird, flying bird, mocking bird, nesting bird, wading bird, eating bird, migrating bird, living bird
boat	fishing boat, flying boat, sailing boat, rowing boat, passing boat, rocking boat, moving boat, cruising boat, sinking boat, racing boat
bottle	nursing bottle, weighing bottle, feeding bottle, washing bottle, collecting bottle, smelling bottle, dropping bottle, throwing bottle, sampling bottle, remaining bottle
bus	morning bus, passing bus, sightseeing bus, moving bus, connecting bus, waiting bus, including bus, during bus, grounding bus, oncoming bus
car	sleeping car, dining car, touring car, racing car, moving car, passing car, waiting car, speeding car, oncoming car, approaching car
cat	playing cat, sleeping cat, purring cat, looking cat, hunting cat, talking cat, using cat, missing cat, fishing cat, prowling cat
chair	rocking chair, folding chair, reclining chair, dining chair, matching chair, reading chair, revolving chair, rolling chair, lounging chair, looking chair
cow	milking cow, lactating cow, producing cow, grazing cow, breeding cow, feeding cow, keeping cow, looking cow, herding cow, including cow
dog	hunting dog, barking dog, sleeping dog, running dog, working dog, looking dog, sporting dog, living dog, howling dog, fighting dog
horse	rocking horse, galloping horse, riding horse, running horse, trotting horse, stalking horse, flying horse, walking horse, kicking horse, bucking horse
motorbike	passing motorbike, speeding motorbike
person	living person, dying person, missing person, looking person, thinking person, interesting person, loving person, controlling person, caring person, charming person
sheep	herding sheep, grazing sheep, wandering sheep, raising sheep, tending sheep, counting sheep, bleating sheep, shearing sheep, keeping sheep, killing sheep
sofa	matching sofa, sagging sofa, looking sofa, facing sofa, inviting sofa, spring sofa, reclining sofa, including sofa, lounging sofa, imposing sofa
train	moving train, morning train, evening train, approaching train, passing train, speeding train, oncoming train, following train, waiting train, departing train
tvmonitor	watching tv, morning tv, using tv, including tv, making tv, viewing tv, running tv, doing tv, producing tv, existing tv

Table 16: Top 10 bi-grams (when available, otherwise all available bi-grams) obtained by applying the present participles selection filter.

PASCAL VOC training set (whose data collection required definitely more time), and are systematically better than the simple search by name performed on Google. As improvements, we plan to employ Flickr as image source repository, adopting multilingual strategies, and possibly building or reusing an ontology of the visual connected to a foundational ontology, in order to be able to leverage not only on the semantic relations holding between words, but also on the ontological relations between the objects such words are referred to.

References

- [1] C. Vondrick, A. Khosla, T. Malisiewicz, A. Torralba, Hoggles: Visualizing object detection features, in: *IEEE 14th International Conference on Computer Vision (ICCV 2013)*, 2013.
- [2] À. Lapedriza, H. Pirsiavash, Z. Bylinskii, A. Torralba, Are all training examples equally valuable?, *CoRR abs/1311.6510* (2013).
- [3] A. Torralba, A. A. Efros, Unbiased look at dataset bias, in: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 2011, pp. 1521–1528.
- [4] L. Fei-Fei, R. Fergus, P. Perona, One-shot learning of object categories, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28 (2006) 594–611.
- [5] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset (2007).
- [6] J. Shotton, J. Winn, C. Rother, A. Criminisi, Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation, in: *Computer Vision–ECCV 2006*, Springer, 2006, pp. 1–15.
- [7] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, *International journal of computer vision* 88 (2010) 303–338.
- [8] B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman, Labelme: a database and web-based tool for image annotation, *International journal of computer vision* 77 (2008) 157–173.

- [9] B. Yao, X. Yang, S.-C. Zhu, Introduction to a large-scale general purpose ground truth database: methodology, annotation tool and benchmarks, in: *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer, 2007, pp. 169–183.
- [10] L. Von Ahn, L. Dabbish, Labeling images with a computer game, in: *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 2004, pp. 319–326.
- [11] L. Von Ahn, R. Liu, M. Blum, Peekaboom: a game for locating objects in images, in: *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM, 2006, pp. 55–64.
- [12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: *Computer Vision and Pattern Recognition*, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 248–255.
- [13] C. Fellbaum (Ed.), *WordNet: an electronic lexical database*, MIT Press, 1998.
- [14] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, D. G. Lowe, Curious george: An attentive semantic robot, *Robotics and Autonomous Systems* 56 (2008) 503–511.
- [15] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, J. Yagnik, Fast, accurate detection of 100,000 object classes on a single machine, in: *Computer Vision and Pattern Recognition (CVPR)*, 2013 IEEE Conference on, IEEE, 2013, pp. 1814–1821.
- [16] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A deep convolutional activation feature for generic visual recognition, *CoRR abs/1310.1531* (2013).
- [17] P.-E. Forssén, D. Meger, K. Lai, S. Helmer, J. J. Little, D. G. Lowe, Informed visual search: Combining attention and object recognition, in: *Robotics and Automation*, 2008. ICRA 2008. IEEE International Conference on, IEEE, 2008, pp. 935–942.
- [18] S. Zhu, G. Wang, C.-W. Ngo, Y.-G. Jiang, On the sampling of web images for learning visual concept classifiers, in: *Proceedings of the*

ACM International Conference on Image and Video Retrieval, ACM, 2010, pp. 50–57.

- [19] J. H. Greenberg, Some universals of grammar with particular reference to the order of meaningful elements, in: J. H. Greenberg (Ed.), *Universals of Human Language*, MIT Press, Cambridge, Mass, 1963, pp. 73–113.
- [20] B. Comrie, *Language universals and linguistic typology: Syntax and morphology*, Basil Blackwell, Oxford, 1981.
- [21] G. Salton, M. J. McGill, *Introduction to modern information retrieval* (1983).
- [22] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, in: *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.
- [23] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional neural networks, arXiv preprint arXiv:1311.2901 (2013).
- [24] Y. Jia, Caffe: An open source convolutional architecture for fast feature embedding, 2013. [Http://caffe.berkeleyvision.org/](http://caffe.berkeleyvision.org/).
- [25] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, LIBLINEAR: A library for large linear classification, *Journal of Machine Learning Research* 9 (2008) 1871–1874.