Improving Object Recognition with Entity Co-Occurrence

Willie Boag
wboag@cs.uml.edu

Pete Maniatis
maniatisp@gmail.com

Abstract

State-of-the-art object detection is powerful, but limited. Most systems use a simple sliding window scheme to detect objects. These schemes ignore all relevant contextual information to disambiguate ”tricky” images. We propose a simple second-pass classification using entity co-occurrence statistics to enhance the prediction results for object detection and recognition. This procedure uses context to share topical information across all object predictions within the same image. In addition, we analyze what this procedure models about the ”common sense” interactions between related objects. We find that this enhancement yields modest, but consistent improvements in F1 score.

1. Introduction

Object Recognition and Object Detection are two of the most fundamental tasks in computer vision. Before systems can perform downstream tasks of: motion tracking, image captioning, robot guidance, and more, the objects within each image must be identified. Because early-pipeline prediction errors can cascade into larger errors for high-level tasks, it is important that the detection and recognition algorithms are as accurate as possible.

Historically, object identification tasks have been completed by non-learning algorithms reliant on feature based methods. The idea behind this is to take a known object, identify unique and invariant regions of interest, or keypoints, and extract a description that can then be used as a template and compared to objects in other images. If these regions and descriptions show a strong enough resemblance, then a positive match may be identified. Various methods such as the Canny Edge Detector, Harris Corner Detector, and the Laplacian of Gaussian Blob Detection can be used to identify areas which are robust enough to ensure uniqueness. The descriptions for these local regions may be built using histograms that identify things like edge orientation and magnitude. These methods are reliant on having good keypoints, descriptors, and comparison algorithms that can accommodate image variance such as rotation, scaling, and lighting.

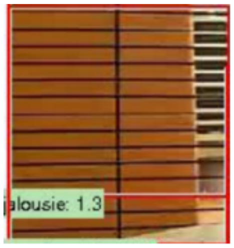

Figure 1. A ”tricky”, ambiguous image whose meaning is difficult to discern without context.

More recently, neural networks have taken the field by storm, significantly outperforming older approaches which use hand-crafted features [5]. Convolutional neural networks have become so successful because they are able to learn their own hierarchical feature representations automatically from the data, which gives them a lot of expressive power to identify patterns within the images they process [6]. The success of Deep Learning for Computer Vision has enabled the creation of numerous libraries for deep learning, including Caffe [4] and Theano [1]. In fact, Caffe’s model zoo [1] is a collection of pre-trained deep neural networks, which makes achieving state-of-the-art results easy.

However, even with Deep Neural Networks which learn advanced feature representations, there is still the issue of context. Most object detection algorithms work by running a sliding window across a given image in order to identify objects. These sliding windows only use information contained within the image to identify the object’s category. Consider the image in Figure 1. A neural network predicted the category of ”Venetian blind”, which seems pretty reasonable for the picture. However, Figure 2 shows the same image, but in the context of its full scene. Using this context, we can see that we were actually looking at a cage, not a Venetian blind. Certainly context plays a large role in disambiguating ”tricky” images.

There has been much effort to use context for disambiguating object recognition errors. In his 2012 Thesis, Divvala incorporates *scene gist* for object presence prediction by using logistic regression to predict a category label

[1] http://caffe.berkeleyvision.org/model_zoo.html

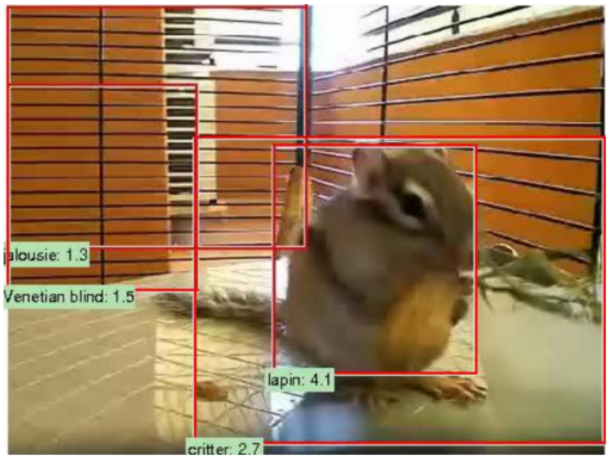

Figure 2. The full image to disambiguate what originally looked like venetian blinds.

when given tags from nearest neighbor images in the a labeled dataset as input [2]. Torralba et. al work jointly on the tasks of object recognition and scene recognition. Scene information is very helpful, and often essential, for recognizing objects, even for humans (e.g. easy to identify a coffee maker when you know it is in a kitchen) [11]. Yao and Fei-Fei perform still image action recognition using a mutual context model that captures the relationships and correlations between human poses and recognized objects. For instance, one leg lifted up and both arms above the head along with a baseball glove indicates that the person is pitching a baseball [12].

We propose a simple procedure for post-processing the prediction confidence scores of a multi-object image. By combining information from each object into a single ”full image” representation, we can form a better representation of the image’s scene than any single object prediction could. This topical information is combined with the original neural network’s confidence prediction to produce an updated prediction of the object’s category.

2. Data

We wanted a dataset that reflected a diverse set of scenes, where object co-occurrence would be able to provide important information. For this reason, we chose the validation set for the detection task of the ImageNet Large Scale Visualization Representation Challenge (ILSVRC 2012) [10]. For this challenge, participants must detect objects from 200 categories in images. In the interest of measuring just the effects of context, we treated the gold standard bounding box as given when we fed these images in to our neural network object recognition step, essentially reducing this to an object recognition problem.

Because our co-occurence-based approach requires many objects in a single image, we filtered out all images that had only a single object, leaving 5,586 multi-object images. The list of images that we used for this experiment can be found at `http://www.cs.uml.edu/~wboag/classes/vision/multiobject_context.txt`.

3. Approach

In order to determine the benefits that contextual information can provide to object recognition, we wanted to improve upon a state-of-the-art model. We use the pre-trained Caffe model described by Girshick et al. [3] [2] to predict the top-10 most likely category labels and their confidences for each bounding box in our data. Using these predicted confidences as input features, we train a SVM to predict the true label. Next, we augment the feature set by also including a ”context vector” associated with each object, and again train a SVM to predict the true label.

The non-context features for our SVM, *BBox Features*, are computed with the following two-step process:

1. build a 200-dimensional vector V of values (one for each category in the ILSVRC data) and store the top-10 confidence scores in their respective dimensions.

2. compute the softmax vector S of V, where

$$softmax(V)_i = \frac{exp(V_i)}{\sum_j exp(V_j)}$$

   Vector S now represents the predicted probability distribution over labels.

Vector S is used as the 200-dimensional feature representation for the BBox-SVM run. This process is illustrated in Figure 3.

Our representation of context uses a simple extension of the BBox features. As demonstrated in Figure 4, the feature representation for an object O for Context-SVM is a 400-dimensional concatenation of

1. the BBox vector of image O (200 dimensions)

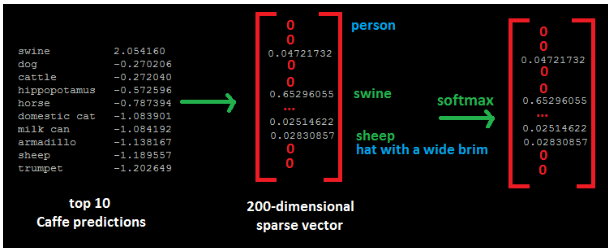
```
swine           2.054160
dog            -0.270206
cattle         -0.272040
hippopotamus   -0.572596
horse          -0.787394
domestic cat   -1.083901
milk can       -1.084192
armadillo      -1.138167
sheep          -1.189557
trumpet        -1.202649
```

Figure 3. Procedure for constructing a 200-dimensional sparse ”BBox” feature representation from the top 10 Caffe predictions for an object instance within an image.

[2] `https://github.com/rbgirshick/rcnn`

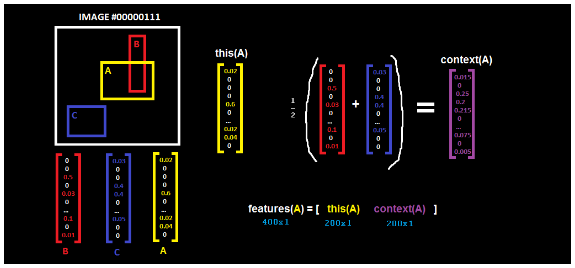

Figure 4. Procedure for constructing the full 400-dimensional feature representation of an object instance. The feature vector is the concatenation of its own BBox vector and the average of all BBox vectors from its image.

2. the average of the the BBox vectors for every object in the image, except for image O, itself (200 dimensions)

We then train our SVM on this 400-dimensional feature representation as our Context-SVM run.

4. Experimental Setup

All results are reported as the average of 5-fold cross validation. In other words, we partition our 5,586 images into 5 folds of equal size. We then run the following procedure five times and report the average scores as our results:

1. Train SVM_i on every fold except for fold i
2. Use SVM_i to predict category labels for the images in fold i
3. Compute and average the precision, recall, and F1-score for each of the 200 categories of the predicted images in fold i, where

   $precision = \frac{TP}{TP+FP}$

   $recall = \frac{TP}{TP+FN}$

   $F1 = \frac{2*precision*recall}{precision+recall}$

For our SVM implementation, we used scikit-learn's LinearSVC class with default hyper-parameters, namely C=1.0 [9].

5. Results

Table 1 shows the results of the Caffe baseline predictions, BBox-SVM, and Context-SVM. For computing the P, R, and F1 of the Caffe baseline, we treat the highest confidence category as the prediction.

	precision	recall	F1
Caffe	67.8	**72.1**	67.4
BBox-SVM	74.5	64.6	66.4
Context-SVM	**75.4**	67.5	**68.9**

Table 1. Macro-averaged P, R, and F1 for our our two runs, compared against the Caffe baseline. All scores are the average of 5-fold Cross Validation.

	TP	FP	FN		TP	FP	FN
lion	0	0	1	lion	1	0	0
ruler	1	0	0	ruler	0	0	1
tiger	0	0	1	tiger	1	0	0
binder	0	0	2	binder	1	2	1
chair	66	32	20	chair	57	16	29
table	94	40	20	table	80	11	34
dog	167	22	4	dog	166	24	5
person	765	117	48	person	726	36	87

Table 2. Prediction statistics of the BBox-SVM (left) and Caffe baseline (right) runs for some very infrequent (top) and very frequent (bottom) labels. These results are from fold 1 of the data. Context-SVM had similar results to BBox-SVM and are omitted for redundancy purposes.

One major difference between the Caffe baseline and both SVM approaches is that the SVMs tend to predict more, conservatively - that is, they tend to favor picking popular labels much more strongly than Caffe does. We can see this more closely by examining Table 2. This table displays the True Positives (TP), False Positives (FP), and False Negatives (FN) of the SVM (left) vs the Caffe baseline (right). The top-left quadrant of the table shows that the SVM has low TP and FP rates on infrequent labels; for those four labels, the SVM only predicts one positive instance whereas Caffe predicts 5 positive instances (though only 3 are correct). Similarly, the bottom half of the graph shows that the SVM predicts frequent labels more often than Caffe predicts them.

On closer inspection, it is not surprising to realize why the SVM would more strongly prefer popular labels - it sees more examples of those during training. Though fold 1 of the SVM had 39 instances of ”lion” to train on, it saw 12,010 ”person” objects. With this in mind, we can make sense of why Table 1 shows BBox-SVM and Context-SVM with much higher precisions and lower recalls. In fact, the recall score for the SVMs is artificially low, because it is computed as the macro-average of the 200 recalls for each category label. Although the SVMs tend to ignore infrequent labels, those labels have just as much weight in the final average as the dominating labels do.

Even still, we can see that Table 1 shows that Context-SVM outperforms both BBox-SVM and Caffe on the F1-score. From this, we can see that not only do context features show a clear improvement of 2.5 points over the same system without those features (BBox-SVM), but they also improve the results of the original neural network predictions (Caffe) by 1.5 points. This confirms the hypothesis that neural network features, though powerful, can still benefit from additional, unrepresented information.

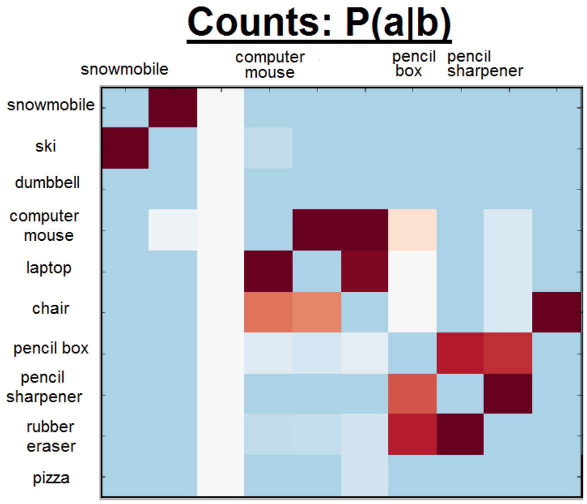

Figure 5. A small slice of the (actually 200x200) empirically-computed conditional probability of the training data for fold 1. Each column of the 200x200 matrix is a probability distribution over the 200 labels. Dark red entries indicate very high probabilities.

6. Analysis

The success of our context-based enhancement of predictions suggests that the data must exhibit topic-based co-occurrence. That is, there are a nontrivial number of images in the data set where knowing the presence of one object can better inform the prediction of another object. More formally, we can consider the conditional probability $P(I|C)$ of the image's category label given the initial guesses for its context labels. Since there are 200 different labels that could be in the context, there are really 200 different conditional probability distributions; a small subset of these distributions can be seen in Figure 5. In this figure, each column is a probability distibution over 200 labels. For instance, the the third column from the right says that given we see ”pencil sharpener” in our context, we are much more likely to be see ”pencil box” or ”rubber eraser” than anything else.

Figure 6 shows an example of when Context-SVM is able to correct one of Caffe's predictions. In this image, we can see that Caffe (red) incorrectly predicted ”pizza” for the pencil's eraser, but Context-SVM correctly classified the object as ”rubber eraser”. Not only does this relationship intuitively satisfy our ”common sense” knowledge that pencil sharpeners are more strongly related to rubber erasers than to pizza, but this relationship also respects the conditional probability distribution in Figure 5.

We can see another instance where Context-SVM corrects Caffe's mistake in Figure 7. In this case, Caffe predicts ”chair”, which does have *some* topical similarity with computer mouses because chairs are typically placed in front of computers. However, we can also see that Context-SVM identified ”laptop” to be even more strongly related to computer mice. Once again, this relationship is modeled in the empirical conditional probability distribution. These examples lead us to wonder to what extent Context-SVM is learning ”common sense” co-occurrence relationships.

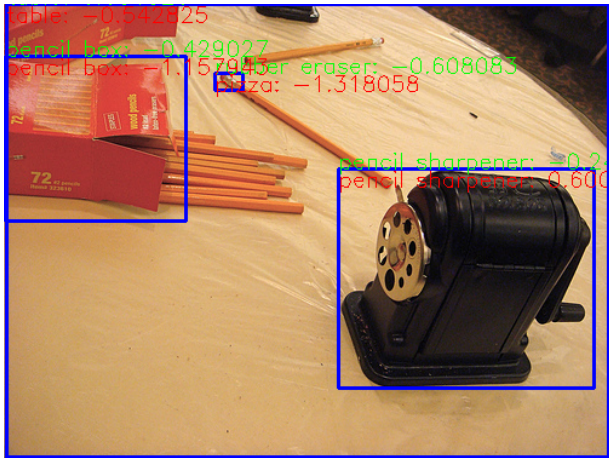

Figure 6. An image that Caffe misclassified, but Context-SVM corrected. The blue boxes are the given bounding boxes. Caffe's prediction (red) incorrectly guessed that the eraser as a ”pizza”, though Context-SVM's prediction (green) correctly identified it.

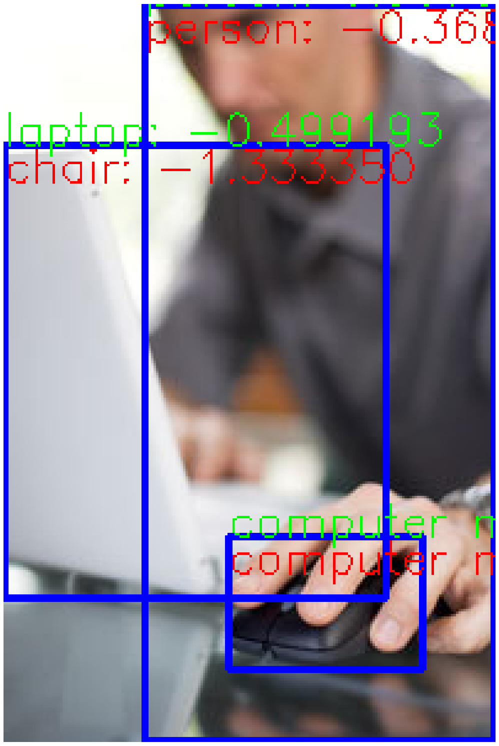

Figure 7. Another example where Caffe misclassified an object. Although Context-SVM (green) was able to correctly predict the laptop, Caffe (red) predicted that it was a chair.

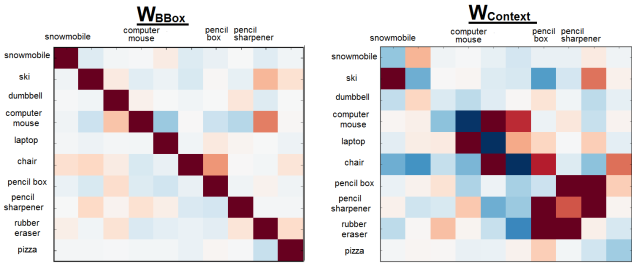

Figure 8. This image shows two small 10x10 submatrices from the trained SVM's parameter matrices. The left submatrix is representative of the first 200 BBox features of an image's 400-dimensional vector representation. The submatrix of the left is representative of the second 200-dimensional context features of an image.

Let's consider the 200x400 parameter matrix W and 200x1 bias vector b of Context-SVM. When we predict the label for a new data point, represented by a 400x1 dimensional feature x, we chose a label y according to:

$$h = Wx + b$$

$$y = argmax_i \quad [h_i]$$

However, since the feature vector x is the concatenation of two 200x1 vectors (one BBox and one BoW average), we can likewise treat parameter matrix W as the concatenation of two 200x200 matrices:

$$W = \begin{bmatrix} W_{BBox} & W_{context} \end{bmatrix}$$

so that now

$$h = \begin{bmatrix} W_{BBox} & W_{context} \end{bmatrix} \begin{bmatrix} x_{BBox} \\ x_{context} \end{bmatrix} + b$$

Figure 8 examines the same 10x10 submatrix slices as from Figure 5 but for Context-SVM parameter matrices W_{BBox} and $W_{context}$. The shapes of each of these submatrices tell interesting stories about how an input feature vector is able to predict an output label.

We can see that W_{BBox} is approximately a diagonal matrix, which indicates that the SVM performs a "pass through" operation - whichever label was predicted by Caffe is generally carried out as the prediction for the SVM. This result is certainly expected given that the Caffe model is the state-of-the-art for object recognition, and its predictions are already strongly correlated with the correct labels.

More surprisingly, there seems to be a similarity between the shapes of $W_{context}$ and $P(I|O)$. These two submatrices have been copied and placed next to one another for a side-by-side comparison in Figure 9. It's important to note that both of these submatrices were fitted on the same training data, so any idiosyncrasies in one would also show up in the other. We can see this occur in the rightmost column, where the context "pizza" yields a distribution over labels that strongly prefers "chair". This is because the training data only has 5 training examples of "pizza" and one of those examples is a photo of a pizza on a table next to a chair. Because of these kinds of idiosyncrasies, it is important to ensure that $P(I|O)$ and $W_{context}$ are trained on the same set of training data, as we do here.

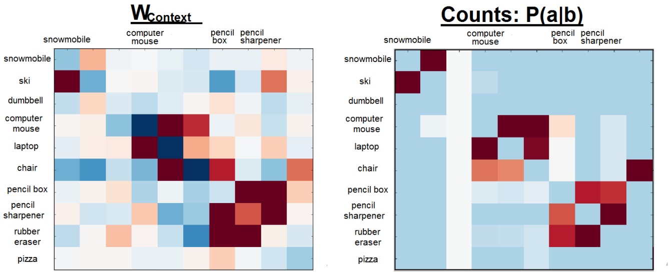

Figure 9. This image compares a 10x10 submatrix of the trained SVM's parameter matrix with the empirically-derived conidtional probability distribution from the training data. Each column of this matrix of the right corresponds to a conditional probability distribution, for instance, the "pencil sharpener" column shows that there is a high chance of seeing a "rubber eraser" or "pencil box" given that the context contains a pencil sharpener.

Almost every entry which $P(I|O)$ suggests should be strongly correlated is, in fact, given a large weight in $W_{context}$. Intuitively, this is exactly what we would hope our context-based model to learn - a set of predictive weights that capture the same correlations we see in the conditional probability distribution (e.g. when we see "snowmobile" the context representation, the SVM assigns a higher score for "ski").

However, it is unclear just how far these similarities go. When training Context-SVM, we experimented with varying settings for the regularization parameter C. While we did find that a strongly regularized model (low C) does result in a $W_{context}$ resembling the uniform distribution (analogous to strong uniform prior distribution), we were not able to see the opposite behavior for an un-regularized model. When the SVM's C hyperparameter was set to a very large value, we were hoping to see an even stronger semblance to $(P(I|O)$, but instead the weights were seemingly sporadically set to grossly overfit predicting the data. This suggests that perhaps our C=1.0 version of $W_{context}$ only resembles $(P(I|O)$ on an intuitive level, rather than carrying a deeper mathematical relationship. Further investigation of the SVM's training objective would need to be done in order to definitely say one way or the other.

7. Conclusion

This paper investigated a simple strategy for leveraging cooccurrence-based context information to improve the results of sliding-window object recognition systems. Using a well-known dataset and state-of-the-art object recognition model, we trained an SVM to classify the category label of objects in an image using features derived from both within the bounding box and from the co-occurence of other objects within the image. We found that even though the SVMs tended to prefer predicting the most frequent labels, there was still a modest improvement in the macro-average F1 score of the Context-SVM system. A closer look at what this trained SVM was modeling showed an implicit similarity with the empirically-derived conditional probability distribution. It is unclear to us whether this is simply an intuitive similarity or if there exists a deeper mathematical explanation.

8. Acknowledgments

This work was done for our Computer Vision semester project. Pete handled installing Caffe, downloading the ImageNet data, running the Caffe baseline predictions, and conducting an error analysis of the ”tricky” images that Context-SVM corrected Caffe's predictions on. Willie built the image representation, trained and ran both SVM runs, analyzed the precision/recall trade-off for the SVM method, and analyzed the SVM parameter matrices.

9. Future Work

Topic/Scene representation. How effective are our BoW context representations? If we cluster object instances based on their context vectors, will we get coherent topic/scene clusters?

Beyond blind BoW The context vector is very similar to the word2vec CBOW architecture [8]. Perhaps using a role-based context similar to Levy and Goldberg's dependecy-based word embeddings would allow us to capture some structured relationships between the roles of interaction between objects [7].

References

[1] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010.

[2] S. K. Divvala. *Context and Subcategories for Sliding Window Object Recognition*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 2012.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

[4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[6] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations.

[7] O. Levy and Y. Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *In Proceedings of NIPS*, 2013.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[11] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. *Computer Vision, IEEE International Conference on*, 1:273, 2003.

[12] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. *Conference on Computer Vision and Pattern Recognition*, 2010.