Abstract

We consider the visual sentiment task of mapping an image to an adjective noun pair (ANP) such as “cute baby”. To capture the two-factor structure of our ANP semantics as well as to overcome annotation noise and ambiguity, we propose a novel factorized CNN model which learns separate representations for adjectives and nouns but optimizes the classification performance over their product. Our experiments on the publicly available SentiBank dataset show that our model significantly outperforms not only independent ANP classifiers on unseen ANPs and on retrieving images of novel ANPs, but also image captioning models which capture word semantics from co-occurrence of natural text; the latter turn out to be surprisingly poor at capturing the sentiment evoked by pure visual experience. That is, our factorized ANP CNN not only trains better from noisy labels, generalizes better to new images, but can also expands the ANP vocabulary on its own.

1. Introduction

Automatic assessment of sentiment from visual content has gained considerable attention [3, 4, 5, 26, 29]. One key element towards achieving this is the use of Adjective Noun Pair (ANP) concepts as a mid-level representation of visual content. We consider the task of labeling user-generated images by ANPs that visually convey a plausible sentiment, e.g. adorable girls in Fig. 1. This task can be more subjective and holistic, e.g. beautiful landscape [1], as compared to object detection [16, 24], scene categorization [9], or pure visual attribute analysis [7, 15, 2, 23]. It also has a simpler focus than image captioning which aims to describe an image as completely and objectively as possible [21, 13].

ANP labeling is related to broader and more abstract image analysis for aesthetics [6, 20], interestingness [11], affect or emotions [19, 28, 27]. Borth et al. [3] uses a bank of linear SVMs (SentiBank), and [4] uses deep CNNs. Both approaches aim to only detect known ANP from the dataset. Deep CNNs have also been used for sentiment prediction [26, 29], but they are unable to model sentiment prediction by a mid-level representation such as ANPs.

Our goal is to map an image onto embedding derived from the visual sentiment ontology [3] that is built completely from visual data and respects visual correlations along adjective (A) and noun (N) semantics, but also fills in the semantic blanks where there has been 0 training data.

![Figure 1](image_url)

Figure 1: Our task is to classify an image into Adjective Noun Pair (ANP) concepts. The numbers indicate the size of the ANP category in our dataset. Our goal is to develop an ANP classifier out of extremely noisy training data from the web that not only respects visual correlations along adjectives (A) and nouns (N) semantics, but also fills in the semantic blanks where there has been 0 training data.
pretty baby attractive face

Figure 2: User generated ANP tags of images are inherently noisy: The same noun (baby) could mean different entities, and a positive adjective (attractive) could modify the pairing noun with a negative sentiment when used sarcastically.

of nouns along adjectives, ANP forms a unique two-factor embedding for visual learning.

ANP images in Fig. 1 exhibit structured correlations. Along each N column is the same type of objects; across the N columns are related objects and parts. Along each A row is the same type of positive sentiment manifested in different objects; across the A rows are sometimes interchangeable sentiments but most times distinctive ones in their own ways. For example, not every ANP is popular on platforms like Flickr: adorable eyes and attractive baby are not frequent enough to have associated images in the visual sentiment dataset [3], suggesting that adorable is reserved more for overall impressions, whereas attractive is more for sexual appeal. When an ANP classifier captures the row-column structure, it can fill in the semantic blanks where there is no training data available and extend the concept consistent with other known ANPs.

Learning a data-driven factorized adjective-noun embedding is necessary not only for finding semantic structures, i.e., some ANPs are more similar than others (pretty girls and attractive girls vs. ugly girls), but also for filtering out annotation noise and removing inherent ambiguity. Fig. 2 illustrates issues common to ANP images. The same noun could mean different entities: baby often refers to human baby, but it could also refer to one’s pet or favorite thing such as cupcakes, whereas an adjective could be used in a sarcastic manner to indicate an opposite sentiment, and such usage is dependent on the particular noun that it is paired with: images tagged as attractive girls are mostly positive, but images tagged as attractive face are often negative, with people making funny faces.

We present a nonlinear factorization model for ANP classification based on the composition of two deep neural networks (Fig. 3). Unlike the classical bilinear factorization model [8] which decomposes an image into style and content variations in a generative process, our model is discriminative and nonlinear. Compared to the bilinear SVM classifiers [22] which represents the classifier as a product of two low-rank matrices, our model learns both the feature and the classifier in a deep neural network architecture.

We emphasize that our factorized ANP CNN is only seemingly similar to the recent bilinear CNN model [18]: we differ completely on the problem, the architecture, and the technical approach. 1) The bilinear CNN model [18] is a feature extractor; it takes the particular form of CNN products and the two CNNs have no particular meaning. Our bilinear model reflects structured outputs and is in fact independent of how we extract the features, i.e., we could additionally use their bilinear model for feature extraction. As a result, we can deal with unseen class labels, while
2. Sentiment ANP CNN Classifiers

We develop three CNN models that output a sentiment ANP label for an input image (Fig. 4). The first is a simple classification model that treats each ANP as an independent class, whereas the other two models have separate A and N processing streams and can thus predict ANPs never seen in the training data. The second model is based on a shared-CNN architecture with two forked output layers, and the third model further incorporates an explicit factorization layer for A and N which is subsequently multiplied together for representing the ANP class.

**ANP-Net: Basic ANP CNN Classifier.** Fig. 4a shows the baseline model that treats the ANP prediction as a straightforward classification problem. We use VGG 16-layer [25] as a base model and replace the final fully connected layer “fc8” from predicting 1,000 ImageNet categories to predicting 1,523 sentiment ANP classes. The model is fine tuned from the ImageNet pretrained version.

We explicitly map the output of A and N nets onto an individual representation that is to be combined bilinearly for final classification. Such a factorization provides our model not only the much needed regularization across different ANPs, but also the capability to classify and retrieve ANPs never seen during the training. Experimental results on the publicly available dataset [3] demonstrate that our model significantly outperforms independent ANP classification on unseen ANPs, and on retrieving images of new ANP vocabulary. That is, our model based on a factorized representation of ANP not only generalizes better, but can also expands the ANP vocabulary on its own.

**Fork-Net: Forked Adjective-Noun CNN Classifier.** Fig. 4b shows an alternative model which predicts A and N separately from the input image. The two streams share earlier layers of computation. That is, the network tries to learn first a common representation useful for both A and N, and then an independent classifier for A and N separately. As for ANP-Net, we use softmax cross-entropy loss for the A or N output, i.e., the network tries to learn universal A and N classifiers regardless of which N or A they are paired with, ignoring the correlation between A and N. At the test time, we calculate the ANP response score from the product of A output $y_A$ and N output $y_N$:

$$p(y = (i, j) | I) = p(y_A = i | I) \times p(y_N = j | I)$$

**Fact-Net: Factorized Adverb-Noun CNN Classifier.** Fig. 4c shows an alternative model which predicts A and N separately from the input image. The two streams share earlier layers of computation. That is, the network tries to learn first a common representation useful for both A and N, and then an independent classifier for A and N separately. As for ANP-Net, we use softmax cross-entropy loss for the A or N output, i.e., the network tries to learn universal A and N classifiers regardless of which N or A they are paired with, ignoring the correlation between A and N. At the test time, we calculate the ANP response score from the product of A output $y_A$ and N output $y_N$:

$$p(y = (i, j) | I) = p(y_A = i | I) \times p(y_N = j | I)$$

$$L(t, I, \theta) = -\log p(y = t | I)$$

$$p(y = k | I) = \text{softmax}(y)_k = \frac{\exp(y_k)}{\sum_{m=1}^{K} \exp(y_m)}$$

2. Sentiment ANP CNN Classifiers
Fact-Net: Bilinearly Factorized ANP CNN Classifier. Fig. 4c shows a model with early layers of Fork-Net followed by a new product layer that combines the A and N outputs bilinearly for the final ANP output. That is, with adjective i and noun j represented in the same M-dimensional latent space, \( a_i \in \mathbb{R}^M \) and \( n_j \in \mathbb{R}^M \) respectively, where \( a_{im} \) and \( n_{jm} \) denote m-th hidden variable for adjective i and noun j, the Fact-Net output \( y_{ij} \) is \( y_{ij} = \sum_{m \in M} a_{im} n_{jm} \). Let \( A, N \) denote the numbers of adjectives and nouns. We have in matrix notations:

\[
Y_{A \times N} = A_{A \times M} \cdot N'_{N \times M}, \quad (4)
\]

\[
A = \begin{bmatrix}
  a_1 \\
  a_2 \\
  \vdots \\
  a_A
\end{bmatrix}, \quad N = \begin{bmatrix}
  n_1 \\
  n_2 \\
  \vdots \\
  n_N
\end{bmatrix}. \quad (5)
\]

The Fact-Net learns to map an image to a factorized A-N matrix representation \( Y \) by minimizing a cross entropy loss \( L \), with gradients over latent A and N net outputs:

\[
\frac{\partial L}{\partial A} = \frac{\partial L}{\partial N'}, \quad \frac{\partial L}{\partial N} = \left( \frac{\partial L}{\partial N'} \right)' A. \quad (6)
\]

The entire network can be learned end-to-end with back propagation. We find the network to learn better with the softmax function normalizing only over ANPs seen in the training set, in order to ignore the effect of ANP activations \( Y_{ij} \) which are unseen during training.

N-LSTM-A and A-LSTM-N are two baseline recurrent algorithms, where networks predict ANPs sequentially. For example, Fig. 4d first predicts the best noun given an image (i.e., \( p(y_N = j|I) \)), and then conditioned on the noun, an adjective is predicted \( p(y_A = i|y_N = j, I) \). Likewise, Fig. 4e predicts first the best adjective, and then the best noun conditioned on that. These two networks are inspired by image captioning models, most of which are in response to the creation of the MSCOCO Dataset [17].

3. Experiments and Results

We describe our ANP ontology and its associated publicly available dataset, present our experimental details, and show our detection and retrieval performance.

ANPs from Visual Sentiment Ontology (VSO). VSO was created by mining online platforms such as Flickr and Youtube by the 24 emotions from Plutchik’s Wheel of Emotions [3]. Derived from an analysis of tags associated with retrieved images and videos from this mining process, an ontology of roughly 3,000 ANPs was established, e.g. beautiful flowers or sad eyes. See Table 1 for statistics.

ANP Dataset. We use the publicly available dataset of Flickr images introduced in [3] with SentiBank 1.1. Please note that we experiment on the larger “non-creative common (Non-CC)” also referred to as the “Full VSO” dataset and not the smaller “creative common (CC)” only dataset.

In the Non-CC dataset, for each ANP, at most 1,000 images tagged with the ANP have been downloaded, resulting in about one million images for the 3,316 ANPs of the VSO.

We first filter out the ANPs with fewer than 200 images, as such small categories are either non-representative or with poorly generalizable evaluation. We also remove ANPs which have unintended semantics against their general usage, e.g. dark funeral refers to images of a heavy-metal band. We then remove any ANP that have fewer than two supports on both the adjectives and the noun, i.e. two ANPs support each other if they share A or N. Such pruning results in 1,523 ANPs with 737, 264 images, 172 adjectives and 240 nouns. For each ANP, 20% of images are randomly selected for testing, while others are used for training. We do ensure that images of one ANP from an uploader (Flickr user) go to either training or testing but not both, i.e., there is no user sharing between training and testing images.

Our ANP labels come from Flickr user tags for images. These labels may be incomplete and noisy, i.e., not all true labels are annotated and there could be falsely assigned labels. We do not manually refine them; we use the labels as is and thus will refer to them pseudo ground truth (PGT).

Model Details. We fine tune the models in Fig. 4 from VGG-net pretrained on ImageNet dataset. For ANP-net, the fully connected layer for final classification is randomly initialized. For Fork-net and Fact-net, we initialize \( fc7-a \) and \( fc7-n \) and all the following layers randomly. The \( fc7-a \) and \( fc7-n \) layers are followed by a parametric ReLU (PReLU) layer for better convergence [10]. We use 0.01 for the learning rate throughout training except the learning rates of pretrained weights are reduced by a factor of 10. Our models are implemented using our modified branch of CAFFE [12]. We use the polynomial decay learning rate scheduler.

We train our models though stochastic gradient descent with momentum 0.9, weight decay 0.0005 and mini-batch size 256 for five epochs, taking 2-3 days for training convergence on a single GPU. For the two recurrent models, we expand and modify Andrej Karpathy’s “neuraltalk” Github branch [13]. After incorporating various hidden layer sizes, we settle on a hidden layer of 128, word+image encoding size of 128, and a single recurrent layer. However, we did not pretrain on any word/semantic data on other corporat

| Table 1: Visual Sentiment Ontology ANP statistics [3]. |
|-----------------|-----------------|-----------------|-----------------|
|                | Flickr          | YouTube         | VSO Statistics  |
| emotions        | 24              | 24              | 320,000         |
| images/videos   | 150,034         | 166,342         | 47,000          |
| tags            | 3,138,795       | 3,079,526       | 3,000 AnP tags  |
| distinct tags   | 17,298          | 38,935          | top ANP happy birthday |
| tags per image  | 20.92           | 18.51           | top positive A beautiful, amazing |
| tag re-usage    | 181.45          | 79.09           | top negative A sad, angry, dark |
| distinct top 100 tags | 1,146          | 1,047           | top N face, eyes, sky |
Table 2: Top-\(k\) accuracies (%) over all ANPs.

|                        | top-1 | top-5 | top-10 |
|------------------------|-------|-------|--------|
| **a) Seen ANP**        |       |       |        |
| DeepSentiBank [4]      | 9.77  | 22.11 | 29.59  |
| ANP-Net                | 13.56 | 30.61 | 40.16  |
| Fork-Net               | 10.82 | 25.76 | 34.53  |
| Fact-Net \(M = 2\)     | 12.51 | 28.95 | 38.32  |
| Fact-Net \(M = 5\)     | 11.68 | 27.80 | 37.28  |
| Fact-Net \(M = 10\)    | 13.22 | 30.36 | 39.92  |
| N-LSTM-A               | 3.32  | -     | -      |
| A-LSTM-N               | 3.32  | -     | -      |
| chance                 | 0.066 | 0.33  | 0.66   |

| **b) Unseen ANP**      |       |       |        |
| DeepSentiBank [4]      | -     | -     | -      |
| ANP-Net                | -     | -     | -      |
| Fork-Net               | 0.27  | 3.35  | 7.09   |
| Fact-Net \(M = 2\)     | **1.54** | **7.17** | **11.85** |
| Fact-Net \(M = 5\)     | 0.77  | 4.52  | 8.77   |
| Fact-Net \(M = 10\)    | 0.36  | 3.35  | 7.22   |
| N-LSTM-A               | 0.012 | -     | -      |
| A-LSTM-N               | 0.009 | -     | -      |

(e.g., MSCOCO), but rather only sequentially trained adjectives and nouns from Sentibank.

**Top-\(k\) Accuracy on Seen and Unseen ANPs.** ANP classes are either seen or unseen depending on whether the ANP concept was given during training. While images of an explicitly unseen ANP class, e.g., beautiful men, might be new to a model, images sharing the same A or N, e.g. beautiful girls or handsome men, have been seen by the model. Unseen ANPs come from those valid ANPs which are excluded from training due to their fewer than 200 examples. For the unseen dataset for our evaluation, we filter out the unseen ANPs with less than 100 examples. We have 293 unseen ANPs with 43,133 examples in total.

We use top-\(k\) accuracy, \(k = 1, 5, 10\), to evaluate a model. We examine whether the PGT label of an image is among the top \(k\) ANP labels suggested by a model output. The average hit rate for test images of an ANP indicates how accurately a model is at differentiating the ANP from others. The top-\(k\) accuracy on seen ANPs shows how good the model is fitting the training data, whereas that on unseen ANPs shows how well the model can generalize to new ANPs.

We take the DeepSentiBank model [4] as a baseline, which already outperforms the initial SentiBank 1.1 model [3]. It uses the AlexNet architecture [14] but fine-tuned to the ANP classes. We apply the same CNN architecture and setup to our set of 1,523 ANPs from the NON-CC dataset.

Table 2a shows top-\(k\) accuracy on seen ANPs. ANP-Net produces the best accuracies, since it is trained for directly optimizing the classification accuracies on these ANPs. Fact-Net outperforms Fork-Net for a number of choices of \(M\), suggesting that our factorized representation better captures discriminative information between ANPs. Also, our VGG-net based models all outperform the Alexnet based DeepSentiBank model, confirming that deeper CNN architectures build stronger classification models.

Table 2b shows top-\(k\) accuracy on unseen ANPs. Consistent with the results for seen ANPs, Fact-Net always outperforms Fork-Net. More importantly, the top-\(k\) accuracies on unseen ANPs decrease with increasing \(M\), with Fact-Net at \(M = 2\) significantly outperforming Fork-Net. That is, the larger the internal representation for A and N, the poorer the generalization to new ANPs. Since models like DeepSentiBank or the individual ANP-net are only capable of classifying what they have seen during training, we leave the corresponding entries in the Table blank.

The top-\(k\) accuracies on our two baseline image captioning models, while significantly above the chance level due to the large number of ANP classes, still seem surprisingly low. These poor results demonstrate the challenge of our ANP task, and in turn corroborate the effectiveness of factorized ANP CNN model.

Our ANP task differs from the common language+vision problems in two significant ways: 1) It aims to capture not so much the semantics of word adjective-noun pairs such as (bull shark, blue shark, tiger shark), but rather pairs of adjectives and nouns with the sentiment evoked by pure visual experience such as (cute dog, scary dog, dirty dog). In this sense, our adjectives just happen to be words, the subjective aspect of our labels for conditioning our nouns, in order to partition the visual space instead of the semantic space. Word semantics reflected in the co-occurrence of natural text has little to do with our visual sentiment analysis. Our task is thus entirely different from the slew of language model and image captioning works. 2) We are generalizing not along a conceptual hierarchy with obvious visual similarity basis, e.g. from boxer and dog to canine and animal, but across two different conceptual trees with subtle visual experience such as (beautiful + sky / landscape / person) and (dead / dry / ... / old + tree) to (beautiful tree). Our task is thus much more challenging.

**Best and worst ANPs by Fact-Net.** We look into the classification accuracy on individual ANPs and compare our Fact-Net with \(M = 2\) against the best alternative ANP-Net for the seen ANPs and Fork-Net for the unseen ANPs. The former could help us understand how the training data are organized and the latter how the model could fill in the blanks of the ANP space and generalize to new classes.

Fig. 5a lists the top and bottom 10 seen ANPs when they are sorted by the difference in top-10 accuracy between Fact-Net and ANP-Net, and Fig. 5b lists the top and bottom 10 unseen ANPs when they are sorted by the difference in
top-10 accuracy gap between Fact-Net and Fork-Net. The range of accuracy gap is \((-0.6, 0.4)\) for the unseen, much wider than \((-0.3, 0.3)\) for the seen ANP case. We analyze the accuracies with respect to the number of images as well as the number of different ANPs seen during the training, and obtain correlation coefficients at the order of 0.05, suggesting that the gain of individual ANPs cannot be explained by the amount of exposure to training instances, but it has more to do with the correlations between ANPs. Fig. 5c-d show sample images from the top and bottom 3 ANPs for the seen and unseen ANPs. Our Fact-Net always seems to have a larger gain over ANPs with fewer varieties.

Image Retrieval by Fact-Net and Fork-Net. We also compare models on retrieving images of a particular ANP. We rank the model output for all the images corresponding to the ANP, and return the images with top responses. The ANP could be seen or unseen in our dataset, or completely novel, e.g. dangerous summer. For Fork-net, we use the product of A-net and N-net output components corresponding to the ANP parts; for Fact-net, we use the output component directly corresponding to the ANP. Fig. 6 shows side-by-side comparisons of top retrievals for 1 seen ANP (beautiful sky) and 3 unseen/novel ANPs by Fact-Net and Fork-Net. 1) Images returned by Fact-Net are in general more accurate on the noun: e.g. ugly baby and ugly sky images contain mostly baby and sky scenes,
1) seen ANP query: **beautiful sky**

- awesome trip
- beautiful sky
- bright sky
- clear lake
- gentle flowers
- sunny sky
- beautiful clouds
- magnificent sky
- incredible beauty
- beautiful sunset
- heavy clouds
- amazing sky
- magical sunsets
- clear night
- colorful sunset
- magical sunset
- harsh sea
- cloudy sunrise
- incredible sunset
- clear night
- colorful sky
- beautiful sky
- serene winter
- smooth clouds

2) unseen ANP query: **beautiful men**

- hot body
- hot model
- sexy fashion
- hot girls
- tough guy
- talented student
- relaxing bath
- sexy dance
- traditional dress
- handsome guy
- strong hair
- sexy body
- heavy book
- heavy weight
- hot butt
- incredible adventure
- sexy body
- cold beer
- stupid graffiti
- smiling guy
- strong men
- sexy fashion
- fragile body
- handsome face

3) unseen ANP query: **ugly baby**

- precious gift
- chubby face
- chubby baby
- fresh baby
- ugly fish
- favorite animal
- dead pig
- dangerous spider
- fresh baby
- warm hat
- laughing baby
- crying baby
- favorite animal
- ugly guy
- clean baby
- favorite animal
- chubby baby
- funny face
- laughing baby
- fresh baby
- graceful animals
- chubby face
- cute animals
- creepy doll

4) unseen ANP query: **ugly sky**

- weird clouds
- magical moon
- beautiful sky
- beautiful sky
- clear sky
- lovely city
- ugly building
- strange clouds
- amazing clouds
- fluffy bed
- amazing clouds
- nice building
- marvelous architecture
- shiny city
- exterior architecture
- weird clouds
- pleasant surprise
- beautiful sky
- weird clouds
- shiny gold
- amazing architecture
- nice building
- lovely evening

---

Figure 6: Images with their PGT tags for 1 seen and 3 unseen ANPs retrieved by Fact-Net and Fork-Net.

Whereas Fork-Net results contain mostly fish and buildings instead. 2) Fact-Net retrievals have more varieties on the adjective: e.g. **beautiful sky** images have both warm and cool color tones, whereas Fork-Net results have mostly cool colors. 3) Fact-Net results correct more annotation mistakes: e.g. man with a tie tagged **hot girls** is rightly retrieved for **beautiful men**, whereas those mistakes such as females tagged **sexy fashion** and **fragile body** are retained.
in Fork-Net results for *beautiful men*. 4) Our Fact-Net can also be used for consensus re-tagging: while beauty is in the eyes of the beholder, we see that the images tagged *beautiful sky* become top retrievals for *ugly sky*, which do share characteristics with other gloomy scenes.

**Conclusions.** From our extensive experimentation, we gain two insights into the unique and challenging sentiment ANP detection task, unlike other well-defined image classification or captioning task. 1) For seen ANPs, the ANP-net is the winner, but it cannot handle unseen ANPs, a killing caveat. We set our initial goal to exceed the ANP-net baseline, after numerous trials, we realize that there will always be a baseline version that no factorized model could beat, since the former directly optimizes the performance over each ANP. However, such CNNs neither see the connections between As and Ns nor generalize as ours. 2) Our Fact-Net on unseen ANPs is substantially better than all baselines. In addition, due to noisy labels (Fig. 2 and Fig. 6), the results on unseen ANPs is substantially better than all baselines. In 2) between As and Ns nor generalize as ours.

## References

[1] Mohammad Al-Naser, Seyyed Saleh Mozafari Chanijani, Syed Saqib Bukhari, Damian Borth, and Andreas Dengel. What makes a beautiful landscape: Beautiful adjective noun pairs attention by eye-tracking and gaze analysis. In *Proceedings of the 1st International Workshop on Affect & Sentiment in Multimedia*, pages 51–56. ACM, 2015.

[2] T. Berg, A. Berg, and J. Shi. Automatic Attribute Discovery and Characterization from Noisy Web Data. In *ECCV*, pages 663–676. Springer, September 2010.

[3] D. Borth, R. Ji, T. Chen, T. Breuel, and S.-F. Chang. Large-scale Visual Sentiment Ontology and Detectors Using Adjective Noun Pairs. In *AMM*, pages 223–232, October 2013.

[4] T. Chen, D. Borth, T. Darrell, and S.-F. Chang. DeepSentiBank: Visual Sentiment Conceptualization with Deep Convolutional Neural Networks. *arXiv:1410.8586*, October 2014.

[5] T. Chen, F. Yu, J. Chen, Y. Cui, Y.-Y. Chen, and S.-F. Chang. Object-Based Visual Sentiment Concept Analysis and Application. In *AMM*, pages 367–376, November 2014.

[6] R. Datta, D. Joshi, J. Li, and J. Wang. Studying Aesthetics in Photographic Images using a Computational Approach. In *ECCV*, pages 288–301. Springer, May 2006.

[7] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing Objects by their Attributes. In *CVPR*, pages 1778–1785, June 2009.

[8] William T. Freeman and J. B. Tenenbaum. Learning bilinear models for two-factor problems in vision. In *CVPR*, 1997.

[9] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, pages 392–407. Springer, 2014.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.