# OBJECT RECOGNITION IN ART DRAWINGS: TRANSFER OF A NEURAL NETWORK

*Rujie Yin*[⋆] *Eric Monson*[‡] *Elizabeth Honig*[♯] *Ingrid Daubechies*[⋆‡] *Mauro Maggioni*[⋆†‡∗]

[⋆]Department of Mathematics, [†]Electrical and Computer Engineering, [‡]Computer Science, Duke University, U.S.A.
[♯]History of Art, University of California, Berkeley, U.S.A.

**ABSTRACT**

We consider the problem of recognizing objects in collections of art works, in view of automatically labeling, searching and organizing databases of art works. To avoid manually labelling objects, we introduce a framework for transferring a convolutional neural network (CNN), trained on available large collections of labelled natural images, to the context of drawings. We retrain both the top and the bottom layer of the network, responsible for the high-level classification output and the low-level features detection respectively, by transforming natural images into drawings. We apply this procedure to the drawings in the Jan Brueghel Wiki, and show the transferred CNN learns a discriminative metric on drawings and achieves good recognition accuracy. We also discuss why standard descriptor-based methods is problematic in the context of drawings.

***Index Terms*—** Object recognition, neural network, transfer learning, signal processing.

## 1 Introduction

In visual arts, objects created by artists are not exact representations of objects in the external world. However, it takes little effort for viewers to recognize objects and scenes in paintings and drawings up to a high level of abstraction and distortion, recognizing even objects that do not exist in the real world. In the case of natural images and photographs, object recognition in computer vision has recently achieved a sequence of breakthroughs, with an ensemble of ideas and techniques, including feature selection and dictionary learning [1], hierarchical representations [2], Convolutional Neural Networks (CNNs) [3] with large rich image repositories and challenges (e.g. ImageNet [4, 5] database, Pascal challenges [6], among many others). Taking advantage of its hierarchical structure inspired by the layers of processing in primate brains, deep CNNs have even surpassed human-level performance [7], automatically learning low-level features such as blobs and directional high-frequency filters used in classical image processing. For recognition of more abstract objects like drawings and sketches, it is unclear whether CNN can work equally well as for natural images, or how one can utilize object structures encoded in CNN trained on natural images to perform efficient recognition on drawings with minimal or no supervision.

To investigate these questions, we examine a data set of drawings from the Jan Brueghel Wiki[1]. We first discuss the difficulty in using standard descriptor based techniques. We then move on to applying CNNs: this is problematic too since the Berkeley data set is limited in size per class, hence it is impossible to train a CNN directly on it. We then propose to adapt a CNN trained with ImageNet, an artificial visual system trained to recognize a variety of objects in photos of real-world scenes, to our drawing dataset. In order to do this, we generate an artificial drawing dataset for training by performing signal processing operations on ImageNet; the transferred CNN achieves very good accuracy. It is important to remark that *no labels on the drawings are needed for this transfer process* (they are only used to measure predictive performance): we note however that the process of transfer we describe cannot be considered unsupervised, since the set of transformations and processing performed on ImageNet to create "drawing-like" images does use knowledge about the test set of drawings. Moreover, our analysis of the structure of the transferred CNN shows that it may be useful to construct effective discriminative metrics on images, which is encouraging towards the goal of constructing a search and recognition engine for objects in drawings and artworks.

The paper is organized as follows: in Section 2 we introduce the Berkeley drawing dataset and describe the collection procedure. In Section 3, we discuss the applicability of descriptors constructed for natural image processing to drawings. In particular, we show that SIFT descriptors are not robust for object recognition in drawings. In Section 4, we show two different ways of adapting the bottom layer of a CNN and analyze how categories of objects are structured through layers of processing by visualizing the inter-category and intra-category distance of deep features obtained from the network.

## 2 Berkeley Drawing Dataset

The Berkeley dataset contains over a thousand paintings, drawings and prints of baroque artist Jan Brueghel and his vast network of assistants, relatives, and collaborators, ranging from world-famous (Pieter Brueghel, Rubens) to utterly obscure. The images are available online at Jan Brueghel Wiki[1]. This collection contains drawings of highly varying contexts (from seascapes to flower still lifes) and quality (from compressed low-resolution, low-contrast prints to high-resolution photos; see Fig. 1).

**Fig. 1:** Sample Drawings in the Berkeley JB dataset

Featured objects appearing across different drawings are manually selected and cropped from the original drawings, in a size range of $38 \times 38$ to $1700 \times 1700$ pixels. Among all the available drawings, we collected sets of drawn objects consisting of 92 cows, 105 sail-

∗This research was partially supported by award NSF-DMS-1320655.

[1]http://www.janbrueghel.net/Main_Page

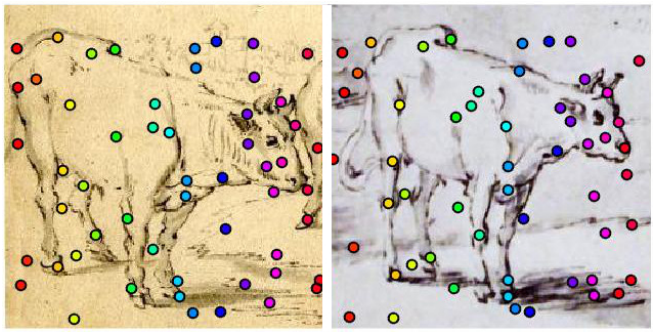

**Fig. 2:** SIFT descriptors matched in two drawings of cows, each pair of matched descriptors (algorithms and code from [12]) is shown in a unique color

boats and 34 windmills. We also collected sets of combined objects (scenes): 61 boats with people and 236 horse carts. Each cropped region contains at least one instance in the corresponding category in the center, occupying at least 70% of the whole region. Any instance identified by human perception is cropped and labelled regardless of the quality or size of the cropped image.

## 3 Descriptor-based Methods

A natural and popular approach to object recognition problems is through the use of descriptor-based classifiers. We describe here our attempts and the problems we encountered with this approach; these observations may apply more generally to image data sets that are significantly different from those consisting of natural images (to which these descriptors were tuned and successfully applied).

Descriptor-based image classifiers are very successful in computer vision tasks such as face recognition [8], Bag of Words model for object matching [9], and deformable part model for object detection [10]. The building blocks of these classifiers are task-oriented descriptors that efficiently encode feature information of objects in images. The ideal descriptor should be robust/invariant to variations of an object and be discriminative to different classes of objects. It has been noted that discriminative descriptors that are not highly frequent in the dataset suffer from high quantization error, resulting in degradation of their discrimination power [11]. We therefore focus on models based on generic descriptors.

Similar to object recognition in natural images, we are looking for representations of object features in drawings that are invariant to scale, contrast and small rotations. These requirements are satisfied by the SIFT descriptor [13] (we use the implementation of [14]), which favors local "corner-like" features and is widely used in computer vision and image processing. In our dataset, as in most others, salient SIFT descriptors are generated both at features of interest and other irrelevant locations due to variations of objects or background. When comparing the similarity of two images, we seek to consider the set of consistent SIFT descriptors in both images. Such a robust SIFT descriptor matching is possible when two images differ by an affine transform with bounded distortion [12]. Unfortunately, we find that this model of distortions is not sufficient to tackle the variability of shapes in our drawing dataset. Even for drawings looking highly similar to the human eye, the two sets of SIFT descriptors extracted are highly affected by the background and by local deformations of the objects of interest. Fig. 2 shows that even the consistency between robustly matched SIFT descriptors is weak. This inconsistency is caused by local and global feature deformations, background, different relative sizes of object parts. Although these issues occur in natural image object recognition where these techniques have been successfully applied, the flexibility/elasticity seems much larger in drawings, making regularization of deformations while matching SIFT descriptors particularly problematic.

## 4 Transfer of a Neural Network

Instead of manipulating and matching unsupervised descriptors, we use a deep convolutional neural network where discriminative descriptors are automatically learned. However we have too few labeled cropped drawings to train such a network. The idea is then to use an "off-the-shelf" CNN model [3] that has been trained on a large data set of natural images (ImageNet) and shown to have the ability to correctly recognize a large number of objects with huge variations in appearance in the training and test data. Here we view the pre-trained CNN as an "artificial vision system" that, while trained on natural images, could possibly be adapted to significantly different sets of images. Similarly to humans taking advantage of their knowledge from the real world in recognizing abstract objects in art [15], transferring knowledge from a pre-trained CNN might be able to assist recognition in art drawings that are significantly different from the natural images used to trained the CNN model. The different nature of the classes of objects requires re-training the top layers of the CNN model; because the fine structures and features of drawings are different as well, this motivates us to also re-train the bottom layer of the CNN model. In order to do this, we create a new training set bridging the two, by transforming ImageNet labelled images into drawing-looking images, and we use these (already labelled!) images – or rather, a subset containing only the classes of interest – to partially retrain the CNN. While no new labels are needed in this step, we do not consider this as completely unsupervised learning, since the transformation that generates "drawing-like" images from natural images is purposefully designed. We now describe these steps in detail, then present and discuss classification results and their dependence on how we re-train the CNN model, and finally present a quantitative investigation on how the drawings are structured by the transferred CNN.

### 4.1 Adapting and transferring the highest layer

We use BVLC Reference CaffeNet in the `caffe` package [16], an implementation of AlexNet trained on ImageNet ILSVRC 2012 in [3], as our pre-trained model, and we call it $CNN_{\text{ref}}$ in this paper. Since cow and windmill are not included in the 1000 categories of ILSVRC 2012, we substitute the last fully connected layer in $CNN_{\text{ref}}$ with layers for classification of categories in our drawing dataset. This technique was first used in [17] to transfer features in middle layers, while substituting the top layer for two sequential layers.

To train the new top layer, we need a training set of images that share the same low- and mid-layer features with the ILSVRC dataset. Therefore, for each category in the drawing dataset, we find the closest synset in ImageNet, and collect the corresponding sets of images as our ImageNet dataset, which we denote by $\mathcal{I}$. We selected the cow, cart, rowboat, sailboat and windmill synsets in ImageNet (ID n02403454, n02970849, n03199901, n04128499, n04587559 respectively). For the drawing category "people & boat", we pick the "rowboat" synset, which frequently includes people. With the same procedure of fine-tuning CaffeNet for style recognition on "Flickr Style" data[2],

[2] `caffe` tutorial: `http://caffe.berkeleyvision.org/gathered/examples/finetune_flickr_style.html`

we re-train $CNN_{\text{ref}}$ with a new top layer and call it $CNN^5_{\text{top}}$.

Fig. 3 (green bars, right panel) shows the prediction accuracy when testing on drawings. Despite the high accuracy in the horse cart category, $46\%$ of cow drawings are recognized as cart, and the network performance is severely biased. We conjecture that this is due to drawings lacking color, with objects often sketched with rough curves instead of the well-delineated closed geometric shapes filled with color occurring in natural images. To let the CNN better adapt to these features of drawings, that are significantly different from those of natural images, we push the CNN transfer mechanism further by re-training also the bottom layer in $CNN_{\text{ref}}$, which is responsible for the extraction of low-level features. This goes beyond the work of [17] where only the top layer (a fully connected layer whose main purpose is classification) is changed. The transferred CNN's obtained from $CNN_{\text{ref}}$ by re-training the bottom and top layers, are called $CNN_{\text{new}}$.

### 4.2 Adapting the bottom layer in $CNN_{\text{new}}$

We are unable to train a new bottom layer in $CNN_{\text{ref}}$ on drawings, due to data scarcity. We thus propose to generate "drawing-like" images $\mathcal{I}^{\text{draw}}$ from the subset of ImageNet images $\mathcal{I}$ so that there is sufficient data for training and validating the CNN.

We apply two transformations on the natural images from $\mathcal{I}$ (processing done with ImageMagick) to obtain $\mathcal{I}^{\text{draw}}$. First we convert each color image $I \in \mathcal{I}$ to a gray scale $256 \times 256$ image, by scaling with fixed width-height ratio and mirroring across boundaries to pad. The grayscale image is then mapped to a "drawing-like" image $I^{\text{draw}}$ by taking the difference of the gray images and their Gaussian blur, to enhance edges, followed by a contrast-normalization step.

We experiment training $CNN_{\text{new}}$s using the "drawing-like" images by themselves, or together with subImageNet, on a subset of categories (cow, sailboat, windmill), yielding $CNN^{\text{draw},3}_{\text{new}}$ and $CNN^{\text{all},3}_{\text{new}}$ respectively. In the training phase, the new layers are initialized using random weights, and are assigned a fast learning rate of 10, and the remaining layers are initialized using weights from $CNN_{\text{ref}}$, and assigned a slow learning rate of 1. The training and validation sets are kept fixed, and no pair $(I, I^{\text{draw}})$ is ever split across different sets. During the training phase the optimization parameters are tuned to achieve the best possible prediction performance on the validation set, typically higher than $92\%$ and we stop optimizing when a model starts to overfit. To compare different settings of the training data, we trained $CNN_{\text{new}}$ on three categories, cow, sailboat and windmill, that are quite distinct from each other. Fig. 3 shows the confusion matrices for the predictions of the transferred CNNs when tested on drawings.

The $CNN^3_{\text{top}}$ is the baseline network where only the top layer, but not the bottom layer, are transferred, and its prediction is biased to windmill category. The accuracy of $CNN^{\text{draw},3}_{\text{new}}$ (trained on "drawing-like" images $I^{\text{draw}}$ only) improves over that of $CNN^3_{\text{top}}$, with accuracy on cow drawings boosted from $70\%$ to $93\%$, at the price of a drop in the accuracy of windmill drawings, $82\%$. As there are three times as many drawings of cow as that of windmill, the overall prediction accuracy also has increased. $CNN^{\text{all},3}_{\text{new}}$, trained on both $\mathcal{I}$ and $\mathcal{I}^{\text{draw}}$, has even better accuracy in the sailboat category, and overall prediction accuracies over $90\%$ for all 3 categories. See Fig. 3.

We also trained a $CNN_{\text{new}}$ on all 5 categories, called $CNN^{\text{all},5}_{\text{new}}$, using both $\mathcal{I}$ and $\mathcal{I}^{\text{draw}}$, obtaining our best accuracy of $77\%$ over the drawings. As shown in Fig. 3, the bias of $CNN^5_{\text{top}}$ in drawings

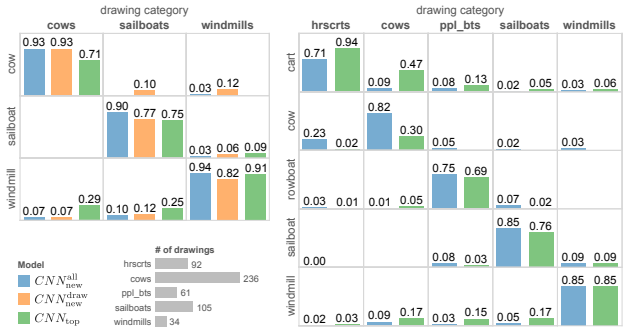


**Fig. 3:** Confusion matrix of drawings predicted by $CNN^{\text{draw},3}_{\text{new}}$, $CNN^{\text{all},3}_{\text{new}}$, $CNN^3_{\text{top}}$(left) and $CNN^{\text{all},5}_{\text{new}}$, $CNN^5_{\text{top}}$(right). The percentages are computed within each drawing category.

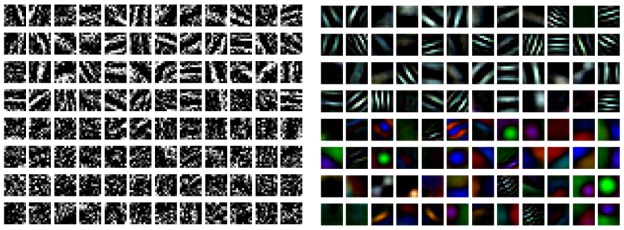

**Fig. 4:** Visualization of the bottom (convolutional) layer in networks trained on five categories. Left: $CNN^{\text{all},5}_{\text{new}}$ trained with "drawing-like" and color ImageNet images. Right: $CNN^5_{\text{top}}$ trained with color ImageNet images.

disappears when using this $CNN^{\text{all},5}_{\text{new}}$, with prediction accuracy increasing in all categories but horse cart. We were however unable to obtain accuracy for these 5 classes of drawings comparable to the one obtained for 3 classes. We conjecture that the main difficulty brought by the additional two categories, horse carts and rowboats, is that they are similar in content to cows and sailboats categories and the object composition is more complicated. The loss of color information and high-frequency details due to the transformation to "drawing-like" images may also limit the discriminating power. We also note that in this case the accuracy on the validation set is over $90\%$ (which does not include drawings), much higher than on the test set of drawings.

### 4.3 Feature space structure of $CNN_{\text{new}}$

We are interested in understanding some of the changes in the CNN upon transfer. Understanding the learned features, and their relationships, in a trained CNN is not an easy task, with no existing standard procedures. We start by comparing the first convolution layer of any of the $CNN_{\text{new}}$s to that of the $CNN_{\text{top}}$. Fig.4 shows the 96 filters of size $11 \times 11$ for $CNN^{\text{all},5}_{\text{new}}$ and $CNN^5_{\text{top}}$ in the same order. Almost half of the filters in $CNN^5_{\text{top}}$ are color blobs and the remaining are high frequency filters in various directions. In comparison, the filters of $CNN^{\text{all},5}_{\text{new}}$ show patterns similar to the high-frequency filters of $CNN^5_{\text{top}}$ in the corresponding location, but they are more noisy and present no high-frequency features. On the other hand, the filters look like textures in the channels where original color blobs are: when gray scale drawing-like images are used in training, color information is no longer helpful but the textures are.

To further investigate the feature space structure of $CNN_{\text{new}}$, we consider the output $f_k \in \mathbb{R}^{N_k \times C_k}$ of each layer $k$ for an image pro-

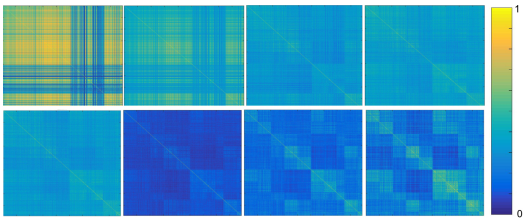


**Fig. 5:** Matrices of $cos(\theta_k)$ of the $CNN_{\text{new}}$. Network layers increase in depth from left to right, top to bottom. Images are grouped into blocks of "drawing-like" images, (cart, cow, rowboat, sailboat, windmill), drawings (in the same categorical order)

cessed through the network, where $C_k$ is the number of channels and $N_k$ is the dimension of output in each channel. For convenience, we reshape $f_k$ into a vector. Each convolutional layer of a CNN generates non-linearly a new feature space from the previous one, hence each $f_k$ is a feature descriptor of the image at a certain level of the hierarchical structure. We define the similarity at layer $k$ of two images $I_i, I_j$ as the cosine of the angle $\theta_k^{i,j}$ between $f_k^i$, $f_k^j$ at each layer $k$. Note that since all the entries of $f_k$ are nonnegative, $\cos(\theta_k^{i,j}) \in [0, 1]$. If images are embedded in an ideal Euclidean feature space, images from the same category should be close, with a small angle between them, whereas those from different categories should be farther away. We monitor the behavior of angles between $f_k$s of images from the same and across different categories, to decide if the feature space learned by a CNN is helpful in discriminating among images in different categories. We randomly selected 100 images from each of the five categories of "drawing-like" images and drawings (if the total number of drawings in a category is smaller than 100, we use all the drawings in that category) and in Fig. 5 we show the similarity matrices $[M_k]_{i,j} = \cos(\theta_k^{i,j})$ for the layers of $CNN_{\text{new}}$. The images are arranged in the order of categories. At the top left we see that the input images are not well-clustered in category: even drawings in the same group have low similarity (the dark blue pattern). Going through the panels from left to right, top to bottom, structures emerge in the similarity matrices; in the bottom right matrix five blocks on the upper left diagonal, corresponding to five categories in "drawing-like" images, are visible. The two big blocks on the lower right diagonal correspond to drawings being well-separated into two subgroups. Furthermore, there is strong correlation off-diagonal between "drawing-like" images and drawings in the corresponding categories. This demonstrates that the network learns features from "drawing-like" images and transfers them to drawings. To summarize, *a transferred discriminative metric emerges*.

To further quantify the linearity of the deepest feature space, we build a binary linear SVM classifier for each drawing category using only the output features of the same "drawing-like" images as in Fig. 5 from the second-to-last fully-connected layer of $CNN_{\text{new}}^{\text{all},5}$. The binary classifier is then tested on both "drawing-like" images and drawings. The result is shown in the top of Fig. 6, where the classifiers are very robust to "drawing-like" images, meaning that different categories are well separated by linear subspaces in the deepest feature space of $CNN_{\text{new}}^{\text{all},5}$. Furthermore, the classifiers achieve comparable performance on the drawings, indicating that "drawing-like" images and drawings in the same category are close in the feature space. In addition, we can train binary classifiers directly on the deepest feature of drawings to obtain better classification performance, see the bottom of Fig. 6.

Finally, we quantify the similarity in style between drawings

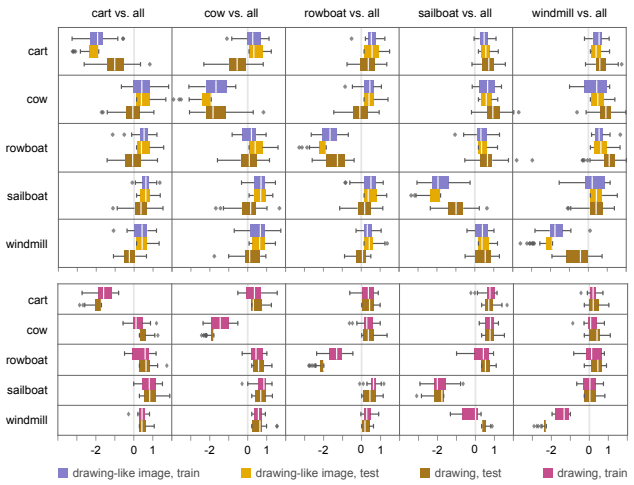


**Fig. 6:** Binary linear SVM's classification result of five categories trained on the deepest output feature of $CNN_{\text{new}}$. Top: trained with drawing-like images, bottom: trained with drawings.

and "drawing-like" images. It is observed in [18] that information about image content and style are encoded separately in a CNN. In particular, images of different content but the same style have small Euclidean distance between their covariances $f_k^T f_k$, where $f_k$ is as above the output from different channels in a convolution layer, but is now formatted to be a matrix $N_k \times C_k$ instead of a vector as above. As the layer goes deeper, the "style" information stored in the covariance evolves from local to global. Fig. 7 shows the distance matrices of covariance of the top and the bottom convolution layers from the same images in Fig. 5. For the low-level local style, the local features of drawings and "drawing-like" images look different as inter-source distance is bigger than intra-source distance. For the high-level global style, the inter-source distance is almost the same as the distance between "drawing-like" images, but it is bigger than the distance between drawings: the "drawing-like" images have higher diversity in composition than drawings.

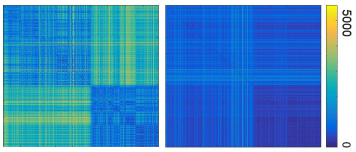


**Fig. 7:** Distance matrices of channel covariance from the top and the bottom convolution layers in $CNN_{\text{new}}^{\text{all},5}$

## 5 Conclusion

We have introduced a novel way of transferring a CNN trained on a large collection of pictures of real world images to perform recognition tasks on images of a very different nature, specifically art drawings from the Jan Brueghel Wiki. We do so without the need of any label on the drawings, by re-training only the top and bottom layers of the CNN on "drawing-like" versions of the real world images, jointly with the original image set. The transferred network has significantly increased performance in recognizing objects, compared to the original one. We also study the changes in the layers of the transferred CNN, showing it learns useful features that may be used in future work to introduce a discriminative metric for developing a search engine for objects in drawings.

## 6 References


[1] Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Ming Yang, Timothee Cour, Kai Yu, Liangliang Cao, and Thomas Huang, "Large-scale image classification: fast feature extraction and svm training," in *In IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[2] J. Zhang, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," *International Journal of Computer Vision*, vol. 73, pp. 2007, 2007.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li 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.

[5] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, pp. 1–42, April 2015.

[6] Mark Everingham, Luc Van Gool, C. K. I. Williams, J. Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," 2009.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *arXiv preprint arXiv:1502.01852*, 2015.

[8] Lior Wolf, Tal Hassner, and Yaniv Taigman, "Descriptor based methods in the wild," in *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008.

[9] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1470–1477.

[10] Pedro Felzenszwalb, David McAllester, and Deva Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[11] Oren Boiman, Eli Shechtman, and Michal Irani, "In defense of nearest-neighbor based image classification," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[12] Yaron Lipman, Stav Yagev, Roi Poranne, David W Jacobs, and Ronen Basri, "Feature matching with bounded distortion," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 3, pp. 26, 2014.

[13] David G Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Ieee, 1999, vol. 2, pp. 1150–1157.

[14] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," `http://www.vlfeat.org/`, 2008.

[15] Aaron Kozbelt, "Object recognition in picassos abstract art," *The Sloping Halls Review*, vol. 2, 1995.

[16] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[17] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1717–1724.

[18] L. A. Gatys, A. S. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *ArXiv e-prints*, Aug. 2015.