UNLEASHING THE POTENTIAL OF CNNS FOR INTERPRETABLE FEW-SHOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Convolutional neural networks (CNNs) have been generally acknowledged as one of the driving forces for the advancement of computer vision. Despite their promising performances on many tasks, CNNs still face major obstacles on the road to achieving ideal machine intelligence. One is the difficulty of interpreting them and understanding their inner workings, which is important for diagnosing their failures and correcting them. Another is that standard CNNs require large amounts of annotated data, which is sometimes very hard to obtain. Hence, it is desirable to enable them to learn from few examples. In this work, we address these two blind sides of CNNs by developing novel and interpretable models for few-shot learning. Our models are based on the idea of encoding objects in terms of visual concepts, which are interpretable visual cues represented within CNNs. We first use qualitative visualizations and quantitative statistics, to uncover several key properties of feature encoding using visual concepts. Motivated by these properties, we present two intuitive models for the problem of few-shot learning. Experiments show that our models achieve competitive performances, while being much more flexible and interpretable than previous state-of-the-art few-shot learning methods. We conclude that visual concepts expose the natural capability of CNNs for few-shot learning.

1 INTRODUCTION

After their debut (LeCun et al., 1998) in 1998, Convolutional Neural Networks (CNNs) have played an ever increasing role in computer vision, particularly after their triumph (Krizhevsky et al., 2012) on the ImageNet challenge (Deng et al., 2009). Some researchers have even claimed that CNNs have surpassed human-level performance (He et al., 2015), though other work suggests otherwise (Zhu et al., 2017). Recent studies also show that they are vulnerable to adversarial attacks (Goodfellow et al., 2015). Nevertheless, the successes of CNNs have inspired the computer vision community to develop more sophisticated models (He et al., 2016; Szegedy et al., 2017). Despite the gains in performance due to these increasingly complex models, We only have limited insights into why CNNs are effective and it is hard to determine the visual cues used in these networks. This is unsatisfactory from a scientific perspective and arguably slows the progress of developing more powerful CNNs.

In particular, the ever-increasing depth and complicated structures of CNNs mean that the internal representations produced by CNNs are highly abstract and hard to understand. This means that the problem of understanding deep neural networks is non-trivial. Notably, for conventional deep neural network structures, there are three major challenges for understanding their mechanisms. Firstly, the use of non-linear layers makes it almost impossible to obtain theoretical results for understanding neural networks. Secondly, for some commonly used networks such as ResNet (He et al., 2016) and Inception (Szegedy et al., 2017), the deep network features processed by many convolutional layers are squeezed into a vector before the prediction layer, which does not correspond to the $2D$ spatial structure of images. Thirdly, the intermediate-level feature maps within deep neural networks, despite preserving spatial information, are hard to relate to the final recognition decisions of the networks.

To address the difficulty of interpreting deep neural networks, we attempt to gain understanding by building on recent work on **Visual Concepts** (Wang et al., 2015). These Visual Concepts (VCs) are

(a) Sofa Cushion (b) Side Windows (c) Bicycle Wheel

Figure 1: Visualizations of VCs. Each group consists of patches from original images closest to a visual concept. In general, these patches roughly correspond to semantic parts of objects, e.g., cushion of sofa (a), side windows of trains (b) and wheel of bicycles (c).

extracted representations of object parts from CNNs (see visualizations in Figure 1). They relate to the findings that CNNs have internal representations corresponding to objects in scenes (Zhou et al., 2015). In Section 3, we will review VC in detail. Briefly speaking, VCs are extracted by clustering intermediate-level raw features of CNNs, e.g., features produced by the Pool-4 layer of VGG-16 (Simonyan & Zisserman, 2015). Serving as the cluster centers in feature space, VCs divide intermediate-level deep network features into a discrete dictionary. Visualizations (as in Figure 1) qualitatively suggest that these dictionary elements correspond to semantic parts of objects. VCs have also been quantified as detectors for semantic parts (Wang et al., 2017).

More specifically, our starting point is recent work in preparation on **VC-Encoding** which represents objects using binary codes of VCs. Through VC-Encoding we obtain the following three preliminary findings. Firstly, some VCs are closely related to specific object categories, *i.e.*, they mainly occur in images from a certain category. Secondly, VC-Encodings result in spatial patterns in images represented by the heat maps in Figure 3b. Combined with the first property, this indicates that VC-encoding capture spatial information. Thirdly, VCs are training cost effective in the sense that several images are sufficient for extracting VCs for VC-Encoding.

These preliminary findings encouraged us to investigate whether VC-Encoding can be applied to **Few-Shot Learning**. Traditionally, the fact that machine learning algorithms require large annotated datasets is problematic for some real world applications. Moreover, the ability to learn from a few examples is a characteristic of human intelligence and would be strongly desirable for an ideal machine learning system.

Previous work on few-shot learning mainly lies in two directions – *learning a metric* and *learning to learn*. These methods achieve high accuracies on few-shot benchmarks but they lack flexibilities (*i.e.* have to be trained separately for each task) and are not always easy to interpret. By exploiting VC-Encoding we develop a novel approach to few-shot learning that is simple, interpretable, and flexible. This is motivated by the three properties described above: (1) the third property suggests that VCs can be extracted from few images, and (2) the first two properties suggest that we can then perform image recognition using those encodings. Based on these considerations we propose two intuitive models, nearest neighbor and factorizable likelihood model based on the VC-Encoding. We emphasize that these models, unlike other few-shot learning methods, are not deliberately trained to address specific few-shot learning scenarios, e.g., 5-way 5-shot. Instead, they are flexible and can be applied to any-way any-shot without additional training. Despite their simplicity, these models achieve comparable results to the state-of-the-art few-shot learning methods (using only the simplest versions of our approach). From a deeper perspective, these results show that CNNs naturally have the potential for few-shot learning on novel classes but to achieve this potential requires studying the internal structures of CNNs to re-express them in simpler and more interpretable terms.

Overall, our major contributions are two-fold:

(1) We shed additional light on CNNs by studying VC-Encoding and finding three novel properties of these encodings.

(2) Based on these three properties, we present two simple, interpretable, and flexible models for few-shot learning. These models unleash the potential of CNNs for flexible and interpretable few-shot learning. They yield competitive results compared to the state-of-the-art methods on specific few-shot learning tasks but can also be applied directly, without additional training, to other few-shot scenarios.

2 Related Work

Our work lies at the intersection of attempts to understand neural networks and work on few-shot learning. Therefore, we review here previous literature on these two topics.

2.1 Neural Network Understanding

Recently, there have been numerous studies aimed at understanding the behavior of neural networks. The dominant theme of these explorations has been to probe the internal representations of CNNs. Some literature tries to visualize inner representations by sampling (Zeiler & Fergus, 2014), generating (Simonyan et al., 2013; Nguyen et al., 2016) or by backpropagating (Mahendran & Vedaldi, 2015) images maximizing the activations of hidden units. In addition, Zhou et al. (2015) shows that object detectors emerge in CNNs and tries to interpret scene classifications as compositions of object detections. These attempts mainly focus on tracing back the activities from hidden units to inputs while ignoring the causal relations between hidden units and recognition results. Conversely, other works investigate the discriminative power of the hidden features of CNNs by assessing them on specific problems (Sharif Razavian et al., 2014; Bau et al., 2017; Agrawal et al., 2014; Yosinski et al., 2014). The overall findings suggest that deep networks have internal representations of object parts. The most relevant work is the study of visual concepts which discovered mid-level visual cues in the internal features of CNNs and showed relationships between these visual cues and semantic parts (Wang et al., 2015; 2017).

2.2 Few-Shot Learning

There have been growing attempts to perform few-shot learning motivated by attempts to mimic human abilities and to avoid some of the limitations of conventional data-demanding learning. An early attempt was made building on probabilistic program induction (Lake et al., 2015) and another attempt exploited object parts (Wong et al., 2017). More recent efforts at few-shot learning can be broadly categorized into two classes. The first is to design methods to embed the inputs into a feature space friendly to few-shot settings (Koch et al., 2015; Vinyals et al., 2016). The second is meta-learning which efficiently trains an ordinary model with the budget of few examples (Ravi & Larochelle, 2017; Finn et al., 2017). In addition, Qiao et al. (2017) performs few-shot learning by estimating parameters of the prediction layer using regression from previously learned objects. We emphasize that our approach differs from these works, most of which are tailored for a few specific few-shot learning scenarios, while our methods are simple and flexible, so that they work both in normal and almost all few-shot settings.

3 Background: Visual Concepts

In Wang et al. (2015), VCs are discovered as internal representations within deep networks which roughly correspond to mid-level semantic visual cues. VCs play a core role in our work on understanding properties of CNNs and developing our interpretable few-shot learning approach, In this section, we formalize VCs following Wang et al. (2015).

We first summarize the formalization of VCs, which are also illustrated in Figure 2. CNNs contain a hierarchy of lattices $\mathcal{L}_l$, where $l \in 0, 1, \ldots$ stands for the layer of the lattice. In particular, the input images are defined over the lattice $\mathcal{L}_0$ and those lattices on which we derive VCs are specified as $\mathcal{L}_k$. We denote the spatial mappings from $\mathcal{L}_0$ to $\mathcal{L}_k$ by $\pi_{0 \mapsto k}$ and from $\mathcal{L}_k$ to $\mathcal{L}_0$ by $\pi_{k \mapsto 0}$. Thus we can define $\mathcal{F}_k = \{\mathbf{f}_p : p \in \mathcal{L}_k\}$ as the feature vector set at $\mathcal{L}_k$ of all images with p referring to a pixel in

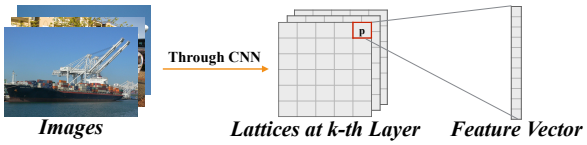


Figure 2: Essential terms in the VC formalization. At the left are input images of the CNN, noted as $\mathcal{L}_0$. At the middle are lattices at the kth layer of the CNN with all images as inputs, noted as $\mathcal{L}_k$. At the right is a feature vector at position p in $\mathcal{L}_k$, noted as f_p.

lattices $\mathcal{L}_k$. These feature vectors are computed by $\mathbf{f}_p = \mathbf{f}(\mathbf{I}_p)$, where the function $\mathbf{f}$ is specified by the neural network and $\mathbf{I}_p$ is a subregion of the input image $\mathbf{I}$, centered on a point $\pi_{k \mapsto 0}(p)$ on $\mathcal{L}_0$. Note that by collecting feature vectors into $\mathcal{F}_k$ we remove all spatial information.

Now we describe how VCs are extracted. The approach assumes that the visual concepts are represented by a population code of the CNN feature vectors. They can be extracted using an unsupervised grouping algorithm. In our implementation, we assume that the feature vectors are generated by a mixture of von Mises-Fisher distributions (Hasnat et al., 2017) and learn this mixture by the EM algorithm (Banerjee et al., 2005). This yields a set of VCs, which are mean directions of the learnt von Mises-Fisher distributions. By contrast, K-means clustering was used in Wang et al. (2015). Finally, we denote the set of VCs by $\mathcal{V} = \{f_v : v \in \mathcal{V}\}$.

To observe the basic semantic nature of VCs, we compute the distances from the original feature vectors to the VCs as follows:

$$d_{p,v} = 1 - \frac{f_p \cdot f_v}{\|f_p\|_2 \, \|f_v\|_2} \tag{1}$$

where $d_{p,v}$ denotes the distance between f_p and the visual concept v. We select those feature vectors with the smallest distances to each VC v and trace them back to the original input image using $\pi_{k \mapsto 0}$. This yields visualizations patches of VCs, shown in Figure 1. We observe that these patches roughly correspond to the semantic parts of objects, which justifies our assertion that the VCs are semantic visual cues.

Visual concepts have been applied to detection and classification tasks. In the original paper (Wang et al., 2015) VCs were evaluated as semantic part detectors. Later work (Wang et al., 2017) showed that VCs could be combined by voting to detect semantic parts and performed well even if the parts were partially occluded.

In more recent work (in preparation) VCs were used to encode semantic parts and objects using *VC-Encoding* and used for detection tasks in the presence of occlusion. This VC-Encoding is described in the next section. We emphasize that none of this prior work on visual concepts addressed few-shot learning and restricted itself to situations where there were many training examples.

4 Few-Shot Learning from VCs

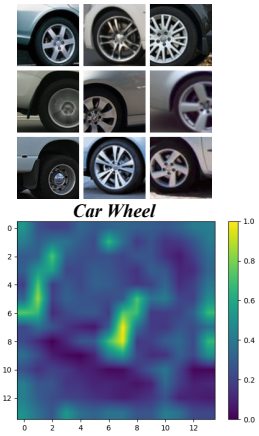
This section describes the technical ideas of our paper. In Section 4.1, we introduce VC-Encoding and then discuss three important properties which motivate its use for few-shot learning. We note that VC-Encoding was developed for another research project (in preparation) but the three properties are contributions of this paper. Then in Section 4.2 and Section 4.3, we propose two simple, interpretable models for few-shot learning based on VC-encoding.

4.1 VC-Encoding and Its Properties

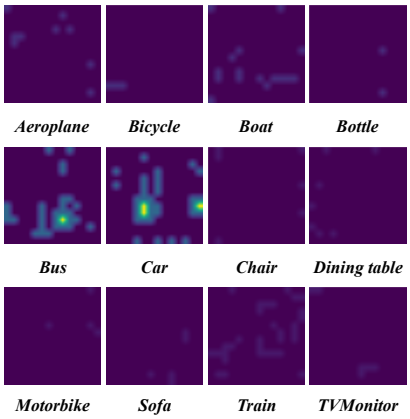
We assume that objects can be parsed into semantic parts. From the perspective of VCs, this means that most f_p should be assigned to *a single VC*. This requires specifying an explicit relationships between the f_p and the VCs. A natural choice is to compute the distances $d_{p,v}$ between the f_p and the VCs and threshold it to provide a binary value $b_{p,v}$ (*i.e.* $b_{p,v} = 1$ if $d_{p,v} < T$). We refer to $\{b_{p,v}\}$ as the **VC-Encoding**. We use two criteria to specify a good encoding, $coverage$ and $firerate$,

Airplane	**77.1%**	9.2%	11.5%
Bicycle	7.5%	2.4%	9.4%
Bus	0.5%	24.5%	12%
Car	3%	**40.5%**	11.8%
Motorbike	3%	13.2%	**45.2%**
Train	6.9%	10.3%	10.2%

(a) Class Sensitivity



(b) Spatially Clustering



(c) Training Cost Effectiveness

Figure 3: Properties of visual concepts. In (a), we illustrate 3 VCs by their closest patches and their distributions over 6 object classes out of the 12 in PASCAL3D+. In (b), we visualize the closest patches to a VC and show a negative distance heatmap derived from the pool-4 feature maps of a car image based on this VC. More precisely, the negative distance heatmap is given by $-d_p$ scaled to $(0, 1)$. In (c) we show the frequency distributions for the 12 object classes in PASCAL3D+ calculated from 5 images per class. These distributions correspond to the same VCs in b.

defined as following:

$$coverage = \frac{\sum_p \max_v b_{p,v}}{|\mathcal{F}_k|} \tag{2}$$

$$firerate = \frac{\sum_p \sum_v b_{p,v}}{|\mathcal{F}_k|} \tag{3}$$

The choice of the encoding threshold T is a trade-off between requiring sufficient coverage and a firing rate that it is close to one (the threshold will be found by a grid-search which outputs the smallest threshold ensuring that $coverage >= 0.8$ with step size 0.001). This yields $b_{p,v} = \mathbb{1}(d_{p,v} < T)$.

Next we analyze VC-encoding and find the following three properties.

Class Sensitivity The first insight is that many VCs tend to occur for a specific object class. In Figure 3a, we calculate the frequency distributions of several VCs for 6 object classes out the 12 in PASCAL3D+ (Xiang et al., 2014). In each column, a single object class dominates the frequency of VC occurrences. This suggests that VCs are useful for classification decisions. Moreover, the corresponding visualized patches on the top support our understanding that VCs have this class sensitivity because they capture the semantic parts of objects.

Spatial Clustering The second discovered property of VC-Encoding is that VCs, despite being learnt by ignoring spatial position, naturally cluster together to form regular spatial patterns (as shown in Figure 3b). This is consistent with the conjecture that the spatial patterns of semantic parts play a vital role in object recognition. More specifically, the VC-encoding capture the spatial patterns of the semantic parts of objects.

Training Cost Effectiveness After showing that VC-encoding is potentially useful for recognition (the first two properties) we also study how many examples are needed to learn good VCs. We find that we only need a few images in order to extract VCs which, when used for VC-encoding, obey the first two properties and hence are suitable for object classification. To illustrate this third property, we calculate the frequencies of VCs occurring on the 12 object classes (after thresholding) using only 5 randomly picked images per class in Figure 3c. These frequencies are very similar to those when many images are used to train the visual concepts. We observe, for example, that only the

car and bus respond to the wheel visual concept. As we will show, these frequencies can be used to build models learnt from few-shots.

4.2 NEAREST NEIGHBOR ON VC-ENCODINGS

First, we propose a simple template matching method which is similar to traditional nearest neighbor algorithms. The only novelty is that we use a similarity metric between VC-encodings which is "fuzzy" so that it can tolerate small spatial shifts of the semantic parts in images. Formally, the similarity metric takes the following form:

$$K(b, b') = \frac{1}{2}\left(\frac{\sum_{p,v} b_{p,v} \max_{q,q\in n(p)} b'_{q,v}}{\sum_{p,v} b_{p,v}} + \frac{\sum_{p,v} b'_{p,v} \max_{q,q\in n(p)} b_{q,v}}{\sum_{p,v} b'_{p,v}}\right) \tag{4}$$

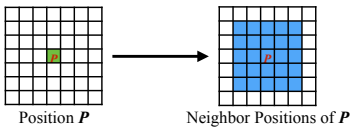


Figure 4: The green grid on the left is p. The blue grids on the right are $n(p)$.

where $K(b, b')$ is the similarity between the binary VC-encodings b and b'. $n(p)$ defines the set of neighboring positions of p (as shown in Figure 4). During testing, we classify an image to the category of the training example with the largest similarity.

One motivation for this method is that sequential convolutional operations carried in a neural network can be considered as embedding input images into a hierarchy of feature spaces. Each convolutional layer can be treated as a different level of decomposition of the inputs since the convolution along with non-linear activation, which take the form $\mathbf{Y} = \sigma(\mathbf{W} \cdot \mathbf{X} + \mathbf{B})$, is composed of matching templates $\mathbf{W}$ and using non-linear function σ to filter out patterns based on the threshold $\mathbf{B}$. In light of this interpretation, and the three properties described above, it is reasonable that VC-Encoding will yield an explicit semantic decomposition.

4.3 FACTORIZABLE LIKELIHOOD MODEL

Apart from the intuitive nearest neighbor method, we present a second method which models the likelihood of the VC-Encoding. We observe that we can specify a distribution over the VC-Encoding $b_{p,v}$ using a bernoulli distribution with probability $\theta_{p,v}$. Following Naïve Bayes, we assume all the elements of the VC-Encoding b are independent (making it possible to learn the distribution for a very small number of examples). Hence we can express the likelihood of b as following:

$$\mathcal{L}(b|\theta) = \prod_{p,v} b_{p,v} \cdot \theta_{p,v} + (1 - b_{p,v}) \cdot (1 - \theta_{p,v}) \tag{5}$$

For each object class y, we derive a probabilistic distribution θ_y from the training examples. Thus the prediction of object class given the VC-encoding b is given by:

$$y_b = \max_y \mathcal{L}(b|\theta_y) \tag{6}$$

Note that by doing this, we are in fact implementing a discriminative model obtained from a generative distribution. We smooth each distribution θ_y using a Gaussian filter to guard against unlikely events.

5 EVALUATIONS UNDER FEW-SHOT SETTINGS

Few-Shot learning, unlike conventional machine learning, resembles the learning process of human beings. It requires the efficiency to learn generalizable knowledge from strictly limited examples, such as a few training images. Hence few-shot learning is a challenging task. Nevertheless, the third property of VC-Encoding in Section 4.1 suggests that a few images may be enough for learning object models represented by VC-Encodings. Indeed our experiments show that both our two visual-concepts-based few-shot learning methods are competitive in performance with methods designed specifically for few-shot learning such as Ravi & Larochelle (2017). In addition, while previous few-shot methods always work in specific few-shot scenarios, such as 5-way classifications, our methods can be applied to a large range of few-shot scenarios using the same CNN trained only once. The experimental results convey the message that trained CNNs have the potential to recognize novel objects from few examples by exploiting VC-Encoding.

Method	5-class 1-shot	5-class 5-shot	10-class 5-shot
Baseline-finetune	$28.86 \pm 0.54\%$	$49.79 \pm 0.79\%$	—
Baseline-nearest-neighbor	$41.08 \pm 0.70\%$	$51.04 \pm 0.65\%$	$39.89 \pm 0.48\%$
Pool3-neareast-neighbor	$43.38 \pm 0.81\%$	$55.33 \pm 0.75\%$	$\mathbf{40.36 \pm 0.59\%}$
Matching Network	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$	—
Meta-Learner LSTM	$43.44 \pm 0.77\%$	$60.60 \pm 0.71\%$	—
MAML	$\mathbf{48.70 \pm 1.84\%}$	$\mathbf{63.11 \pm 0.92\%}$	—
VC-nearest-neighbor (Ours)	$\mathbf{46.39 \pm 1.09\%}$	$58.84 \pm 1.12\%$	$42.42 \pm 0.62\%$
VC-likelihood (Ours)	$45.61 \pm 1.14\%$	$\mathbf{63.07 \pm 1.02\%}$	$\mathbf{45.11 \pm 0.66\%}$

Table 1: Average classification accuracies on Mini-ImageNet with 95% confidence intervals. Evaluations of Baseline-finetune and Baseline-nearest-neighbor are from Ravi & Larochelle (2017). Pool3-neareast-neighbor stands for a nearest neighbor method based on raw Pool-3 features from the same VGG-13 as our methods. At the bottom are our factorizable likelihood method and nearest neighbor method based on VCs. Marked in bold at the top are the best published results for each scenario. Marked in bold at the bottom are our best results for the corresponding set-up. At the right is an extended setting for variance in the number of classes. Note in the last column we use the same models as in the middle and we omit those that are not directly applicable to this setting.

5.1 MINI-IMAGENET

To assess the capability of our few-shot methods, we first evaluate them on a common few-shot learning benchmark, namely Mini-ImageNet. The Mini-ImageNet dataset was first proposed by Vinyals et al. (2016) as a benchmark for evaluating few-shot learning methods. It selects 100 classes out of 1000 classes in ImageNet with 600 examples per class. We use the split proposed by Ravi & Larochelle (2017) consisting of 64 training classes, 16 validation classes and 20 testing classes. In accordance with the convention for Mini-ImageNet, we perform numerous trials of few-shot learning during testing. In each trial, we randomly sample 5 unseen classes from a preserved testing set. Each class is composed of 5 training images for the 5-shot setting and 1 training image for the 1-shot setting. During evaluation, we randomly select 15 images for each class following Ravi & Larochelle (2017).

As Table 1 illustrates, we compare our methods against two baselines in line with the ones in Ravi & Larochelle (2017). In addition, we present the performances of state-of-the-art few-shot learning methods including matching network (Vinyals et al., 2016), Meta-Learner (Ravi & Larochelle, 2017) and MAML (Finn et al., 2017). Regarding our methods, we train a VGG-13 on the training and validation set and then extract 200 visual concepts from the Pool-3 layer. The reason for choosing Pool-3 features is that a grid in Pool-3 lattices $\mathcal{L}_3$ correspond to a 36×36 patch in the original 84×84 image, which is a plausible size for a semantic part. For the gaussian filter used to smooth the factorizable likelihood model, we use a σ of 1.2. To directly examine the impact of the VCs, we also include the result of nearest neighbor matching using raw features from the pool-3 layer (referred as Pool3-Neareast-Neighbor in Table 1). Moreover, we attempt to evaluate the few-shot

Method	6-class 3-shot	8-class 4-shot	12-class 6-shot
Baseline-nearest-neighbor	$\mathbf{46.70 \pm 0.84\%}$	$42.48 \pm 0.74\%$	$38.49 \pm 0.49\%$
Pool3-neareast-neighbor	$44.25 \pm 0.73\%$	$\mathbf{43.30 \pm 0.73\%}$	$\mathbf{38.77 \pm 0.53\%}$
VC-nearest-neighbor (Ours)	$50.42 \pm 0.97\%$	$46.39 \pm 0.74\%$	$40.78 \pm 0.54\%$
VC-likelihood (Ours)	$\mathbf{52.41 \pm 0.93\%}$	$\mathbf{47.37 \pm 0.74\%}$	$\mathbf{43.42 \pm 0.54\%}$

Table 2: Average classification accuracies on Mini-ImageNet with 95% confidence intervals under randomly selected few-shot settings. All models used here and in Table 1 are the same set of models trained only once on the training set. Like the last column of Table 1, we omit those models not directly applicable to various few-shot settings.

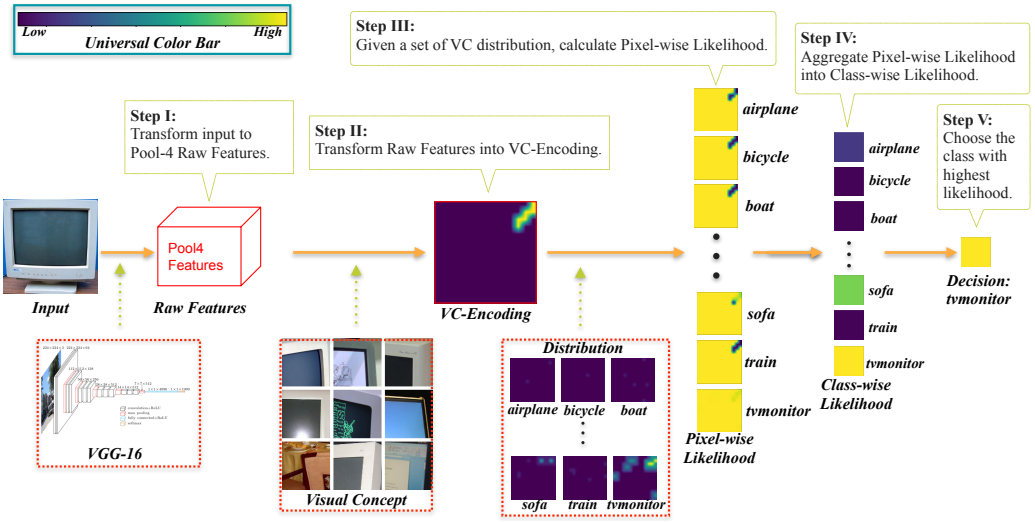


Figure 5: Visualizing the inference procedure of the factorizable likelihood model (using 1 VC for example). The original image is processed by VGG-16, and represented by its VC-Encoding. After calculating the likelihood for each pixel using distributions derived from a few examples, we obtain the pixel-wise likelihood. Then we use the likelihood to make the final decision. For better visualization, we rescale the variance of aggregated likelihood to 1. For all the visualizations, we use the same Universal Color Bar. This figure is best viewed in color.

learning ability with the variance in the number of classes (as shown in the last column of Table 1) and further extend the few-shot learning evaluation with other randomly selected settings (as shown in Table 2), where we need to use exactly the same model as the middle columns in Table 1 trained once on the training and validation set.

The results indicate our VCs-based methods compare well with prior methods which are specifically designed for few-shot learning. Considering meta-learning-based methods, we achieve higher accuracy than Meta-Learner both in 1-shot and 5-shot set-ups, while just slightly behind MAML. Compared with metric-based methods, which are more similar to ours, we marginally outperform the matching network, which is the state-of-the-art method of this category. These results confirm our assumption that low level visual cues within trained CNNs can naturally perform few-shot learning. We observe that on the 5-shot scenario, our likelihood model is significantly better than our nearest neighbor model. A possible explanation is that the likelihood model combines several training examples into a distribution while nearest neighbor can only use examples individually. For instance, if a front wheel of cars appears in one training example and a rare wheel occurs in another example, the likelihood can combine these two wheels into a distribution while nearest neighbor can only match testing examples with either the front wheel or the rare wheel. Finally, using the same model in the middle columns of Table 1, some previous methods like Meta-Learner LSTM are not applicable to various extended settings in Table 2 and the last column in Table 1. In fact, these methods can directly deal only with changes in the number of shots but cannot deal with changes in the number of classes. But unlike these methods, our few-shot learning methods based on VC-Encoding can be easily extended (with minimal re-training) to any number of shots and any number of classes.

5.2 PASCAL3D+

To delve deeper into our methods, we apply them to PASCAL3D+, a dataset with larger high quality images than Mini-ImageNet. PASCAL3D+ (Xiang et al., 2014) is a dataset augmenting 12 rigid categories of the PASCAL VOC 2012 (Everingham et al.). It is originally tailored for 3D object detection and pose estimation. We choose PASCAL3D+ as our testbed since it provides high quality images in comparable image size to ImageNet. We interpret our few-shot recognition mainly by visualizing every step of the inference. With input images of sufficient sizes, we can obtain large VC-Encoding distribution maps whose visualizations are friendly to human observation.

Method	Number of VCs	12-class	
		1-shot	5-shot
Pool4-neareast-neighbor	–	**36.12%**	52.30%
Pool4-SVM	–	32.66%	**52.46%**
VC-likelihood (Ours)	120	39.25%	64.37%
VC-likelihood (Ours)	200	**40.02%**	66.00%
VC-likelihood (Ours)	300	39.23%	**66.47%**
VC-nearest-neighbor (Ours)	120	40.74%	58.52%
VC-nearest-neighbor (Ours)	200	**42.36%**	59.47%
VC-nearest-neighbor (Ours)	300	41.18%	**61.07%**

Table 3: Average classification accuracies on PASCAL3D+. At the top is the group of baseline methods including nearest neighbor and Exemplar-SVM based on Pool-4 features from the same VGG-16 used in our methods. At the middle are our factorizable likelihood models using different number of VCs. At the bottom are our VCs-based nearest neighbor models. Marked in bold are best results within each group for each scenario.

The simplicity of our methods makes the inference process of few-shot recognition very transparent. On PASCAL3D+, we first qualitatively analyze this procedure based on VCs. In Figure 5, we thoroughly visualize every step of our method based on an example VC. Among the closest patches, the corner of TV Monitor occurs the most times. So we may assume this VC relate to the corner of TV Monitor. The distributions of this VC suggest it mainly responds to the upper right of TV Monitors since only the upper right corner of TV Monitor's distribution map shows high frequency (see the distributions in Figure 5). Using this VC, we convert the original deep network features into VC-Encoding. The VC-Encoding implies this VC fires on the upper right corner of the input (see the VC-Encoding map in Figure 5). After calculating the pixel-wise likelihood using the distributions from a few images, it is obvious that except for the TV Monitor, each class has a low likelihood area on that corner (dark parts of pixel-wise likelihood maps in Figure 5). Finally, we aggregate the likelihood and make the correct classification decision.

Meanwhile, we quantitatively evaluate our methods on PASCAL3D+. More specifically, we employ PASCAL3D+ as our testing dataset. For training, we use the ImageNet (Deng et al., 2009) classification dataset without object classes related to 12 rigid categories (956 classes left). We train an ordinary VGG-16 as our starting point which achieves 71.27% top-1 accuracy. For testing, we crop the objects out using annotated bounding boxes provided by Xiang et al. (2014) and resize them into 224×224. Then we use Pool-4 features produced by the VGG-16 to implement our few-shot methods instead of our configurations in 5.1. The main reason for this change is the increased input image size of 224 in this task. As a comparison, we propose 2 baseline models. One (referred to as Baseline-nearest-neighbor in Table 3) is a nearest neighbor method based on raw Pool-4 features using the cosine distance metric. The other (referred to as Baseline-SVM in Table 3) is an Exemplar-SVM trained using hinge loss. Both of these baselines use the same pre-trained VGG-16 as our methods. During evaluation, we set 20 trials of both 5-shot and 1-shot learning over 12 classes on PASCAL3D+. We also assess our methods using different numbers of VCs to see impacts of the number of VCs. The results are shown in Table 3.

In light of our testing results, we conclude that VC-Encoding is a powerful semantic decomposition of images. In general, our methods based on VCs outperform 2 baselines. In particular, the difference between our nearest neighbor methods and baseline baseline nearest neighbor methods is a series of operations based on VCs. In view of the accuracy gap between this pair, we claim that decomposing fuzzy features (*i.e.* deep network features) into explicit semantic cues helps both in interpretability and performance. We also notice that our methods are not sensitive to the number of VCs because changes of the number of VCs only cause slight differences among accuracies.

6 Conclusion

In this paper we address the challenge of understanding the internal visual cues of CNNs and making use of them for few-shot learning. We start by using visual concepts which enable us to represent

objects in terms of VC-Encodings. Then we demonstrate three properties of VC-Encoding, both qualitatively and quantitatively, which suggest that VCs can be used for few-shot learning. Following these ideas we propose two novel, but closely related, models to few-shot learning which are simple, interpretable, and flexible. Our methods show comparable performances to the state-of-the-art methods which are specialized for specific few-shot learning scenarios. We demonstrate the flexibility of our two models by showing that they can apply to a range of different few-shot scenarios with minimal re-training. In summary, we show that VC-Encodings enable ordinary CNNs to perform few-shot learning. We emphasize that in this paper we have concentrated on developing the core ideas of our two few-shot learning models and that we have not explored variants of our ideas which could lead to better performance by exploiting standard performance enhancing tricks. Future improvements would include improving the quality of extracted VCs and extending this approach to few-shot detection.

REFERENCES

Pulkit Agrawal, Ross Girshick, and Jitendra Malik. Analyzing the performance of multilayer neural networks for object recognition. In *European conference on computer vision*, pp. 329–344. Springer, 2014.

Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(Sep): 1345–1382, 2005.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.

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*, pp. 248–255. IEEE, 2009.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

Md Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, Liming Chen, et al. von mises-fisher mixture model-based deep learning: Application to face verification. *arXiv preprint arXiv:1706.04264*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

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

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5188–5196, 2015.

Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pp. 3387–3395, 2016.

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan Yuille. Few-shot image recognition by predicting parameters from activations. *arXiv preprint arXiv:1706.03466*, 2017.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *International Conference on Learning Representations*, 2017.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813, 2014.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pp. 4278–4284, 2017.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pp. 3630–3638, 2016.

Jianyu Wang, Zhishuai Zhang, Cihang Xie, Vittal Premachandran, and Alan Yuille. Unsupervised learning of object semantic parts from internal states of cnns by population encoding. *arXiv preprint arXiv:1511.06855*, 2015.

Jianyu Wang, Cihang Xie, Zhishuai Zhang, Jun Zhu, Lingxi Xie, and Alan Yuille. Detecting semantic parts on partially occluded objects. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

Alex Wong, Brain Taylor, and Alan Yuille. Exploring protrusion cues for fast and effective shape matching via ellipses. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pp. 75–82. IEEE, 2014.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pp. 3320–3328, 2014.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *International Conference on Learning Representations*, 2015.

Zhuotun Zhu, Lingxi Xie, and Alan L. Yuille. Object recognition with and without objects. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 3609–3615, 2017.