International Journal of Computer Vision (2021) 129:736–760
https://doi.org/10.1007/s11263-020-01401-3

Compositional Convolutional Neural Networks: A Robust and Interpretable Model for Object Recognition Under Occlusion

Adam Kortylewski[1] · Qing Liu[1] · Angtian Wang[1] · Yihong Sun[1] · Alan Yuille[1]

Received: 20 January 2020 / Accepted: 4 November 2020 / Published online: 24 November 2020

Abstract

Computer vision systems in real-world applications need to be robust to partial occlusion while also being explainable. In this work, we show that black-box deep convolutional neural networks (DCNNs) have only limited robustness to partial occlusion. We overcome these limitations by unifying DCNNs with part-based models into Compositional Convolutional Neural Networks (CompositionalNets)—an interpretable deep architecture with innate robustness to partial occlusion. Specifically, we propose to replace the fully connected classification head of DCNNs with a differentiable compositional model that can be trained end-to-end. The structure of the compositional model enables CompositionalNets to decompose images into objects and context, as well as to further decompose object representations in terms of individual parts and the objects' pose. The generative nature of our compositional model enables it to localize occluders and to recognize objects based on their non-occluded parts. We conduct extensive experiments in terms of image classification and object detection on images of artificially occluded objects from the PASCAL3D+ and ImageNet dataset, and real images of partially occluded vehicles from the MS-COCO dataset. Our experiments show that CompositionalNets made from several popular DCNN backbones (VGG-16, ResNet50, ResNext) improve by a large margin over their non-compositional counterparts at classifying and detecting partially occluded objects. Furthermore, they can localize occluders accurately despite being trained with class-level supervision only. Finally, we demonstrate that CompositionalNets provide human interpretable predictions as their individual components can be understood as detecting parts and estimating an objects' viewpoint.

Keywords Compositional models · Robustness to partial occlusion · Image classification · Object detection · Out-of-distribution generalization

1 Introduction

Advances in the architecture design of deep convolutional neural networks (DCNNs) (Krizhevsky et al. 2012; Simonyan and Zisserman 2014; He et al. 2016) increased the performance of computer vision systems at object recognition enormously. This led to the deployment of computer vision models in safety-critical real-world applications, such as self-driving cars and security systems. In these application areas, we expect models to reliably generalize to previously unseen visual stimuli. However, in practice we observe that deep models do not generalize as well as humans in scenarios that are different from what has been observed during training, e.g., unseen partial occlusion, rare object pose, changes in the environment, etc.. This lack of generalization may lead to fatal consequences in real-world applications, e.g. when driver-assistant systems fail to detect partially occluded pedestrians (Economist 2017).

In particular, a key problem for computer vision systems is how to deal with partial occlusion. In natural environments, objects are often surrounded and partially occluded by each other. The large variability of occluders in terms of their

Communicated by Mei Chen.

✉ Adam Kortylewski
akortyl1@jhu.edu

Qing Liu
qingliu@jhu.edu

Angtian Wang
angtianwang@jhu.edu

Yihong Sun
ysun86@jhu.edu

Alan Yuille
ayuille1@jhu.edu

[1] Johns Hopkins University, Baltimore, MD, USA

shape, appearance and position introduces an exponential complexity in the data distribution (Yuille and Liu 2018) that is unfeasible to be exhaustively represented in finite training data. Recent works (Zhu et al. 2019; Kortylewski et al. 2020b) have shown that deep vision systems are not as robust as humans at recognizing partially occluded objects. Moreover, our experiments show that this limitation persists even when deep networks have been exposed to large amounts of partial occlusion during training. Hence, this reveals a fundamental limitation of current approaches to computer vision that needs to be addressed.

While robustness to partial occlusion is crucial, safety-critical applications also require AI systems to provide human interpretable explanations of their prediction. Such explanations can help to understand failures and enable the further advancement of the performance of the models, while potentially also supporting the scientific understanding of the vision process. This insight motivated recent work to focus on developing interpretable vision models (Ross et al. 2017; Hu et al. 2016; Stone et al. 2017; Zhang et al. 2018a; Zhang and Zhu 2018). However, most often interpretable models do not perform as well as black-box DCNNs and can only be applied in a very specific domain.

In this work, we propose a general deep architecture that recognizes partially occluded objects robustly even when it has not been exposed to partial occlusion during training, while also being able to provide human interpretable explanations of its prediction. Here, we refer to interpretability in terms of the definition provided by Montavon et al. (2018) as *the mapping of an abstract concept (e.g. a predicted class) into a domain that the human can make sense of* (e.g. in the image space instead of an abstract neural feature space) and an explanation as *a collection of features from the interpretable domain that have contributed to the decision* (e.g. object part detections and occluder locations in the image space).

Our key contribution is that we *unify* compositional models and DCNNs into an architecture that we term *Compositional Convolutional Neural Network*. Our model is inspired by a number of works that demonstrated how the modularity of compositional representations enables efficient learning (Jin and Geman 2006), interpretability (George et al. 2017) and strong generalization at classifying partially occluded 2D patterns (George et al. 2017; Kortylewski 2017; Wang et al. 2017; Zhang et al. 2018c) and 3D objects (Kortylewski et al. 2020b). In particular, we propose to replace the fully-connected classification head of a DCNN with a differentiable compositional model that can be trained end-to-end. The compositional model represents objects as a set of parts that are composed spatially. This enables a robust classification based on the spatial configuration of a few visible parts. The compositional model is regularized to be fully generative in terms of the neural feature activations of the last convolutional layer. The generative nature of the model enables the network to localize occluders in an image and subsequently focus on the non-occluded parts of the object for recognition. In addition, the structure of our compositional model enforces the decomposition of the image representation as a mixture of the context and object representation. This context-aware image representation enables us to control the influence of the context on the models' prediction, which turns out to be important for the detection of partially occluded objects. Figure 1 illustrates the robustness of CompositionalNets at classifying and detecting partially occluded objects, while also being able to localize occluders in an image. In particular, it shows several images of partially occluded objects from the MS-COCO dataset (Lin et al. 2014). Next to these images, we show occlusion scores that illustrate the position of occluders as estimated by the CompositionalNets. Note how the occluders are accurately localized and provide a human interpretable explanation of the models' perception of the image.

Our work on CompositionalNets includes several important contributions:

1. We propose **a differentiable compositional model** that can be trained end-to-end and that is regularized to be generative in terms of the feature activations of a DCNN. This enables us to integrate compositional models with popular deep network architectures into **compositional convolutional neural networks**, a unified deep model with innate robustness to partial occlusion.
2. We evaluate the robustness to partial occlusion on images of artificially occluded objects from the PASCAL3D+ and ImageNet datasets, as well as real images of partially occluded vehicles from the MS-COCO dataset. Our extensive experiments with popular DCNN backbones (VGG-16 (Simonyan and Zisserman 2014), ResNet50 (He et al. 2016), ResNext (Xie et al. 2017)) demonstrate that **CompositionalNets consistently outperform their non-com- positional counterparts by a large margin** at the classification and detection of partially occluded objects.
3. We propose to decompose the image representation in CompositionalNets as a mixture model of context and object representations. We demonstrate that such **context-aware CompositionalNets** allow for fine-grained control of the influence of the object context on the model prediction, which increases the robustness when detecting strongly occluded objects.
4. We show that **CompositionalNets are human interpretable**, because their predictions can be understood in terms of object part detection, occluder localization and object viewpoint estimation. We perform qualitative and quantitative experiments that demonstrate the ability

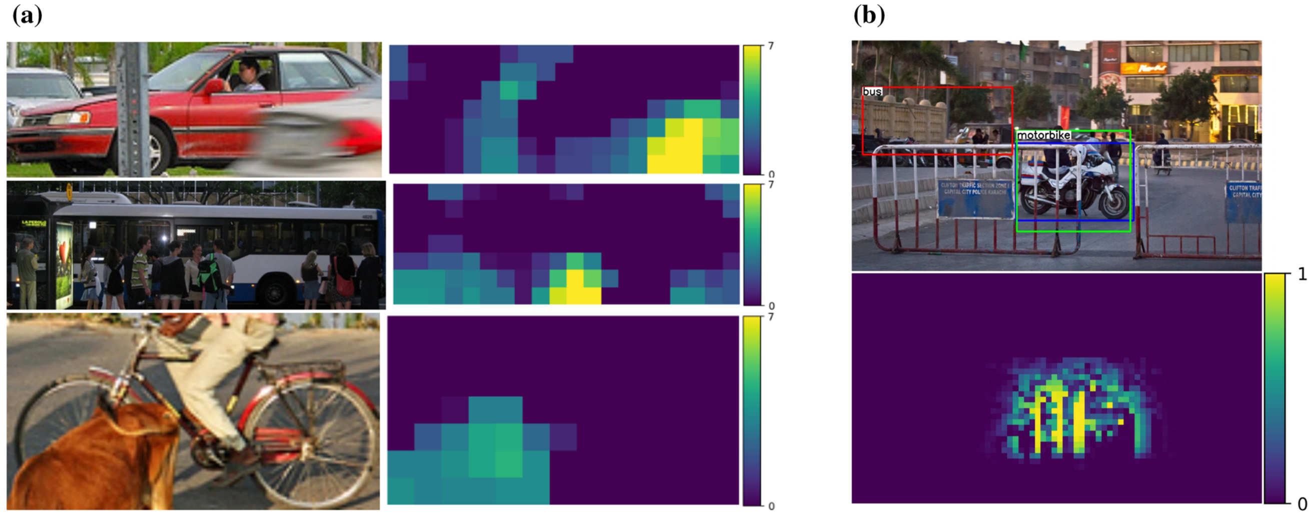


Fig. 1 Example of images for the classification (**a**) and detection (**b**) of partially occluded objects from the MS-COCO dataset (Lin et al. 2014). A standard DCNN misclassifies the images in (**a**) and does not detect the motorbike in (**b**), while also having a false-positive detection of a bus in the background. In contrast, CompositionalNets provide correct predictions in all cases. Intuitively, a CompositionalNet can localize the occluders (see visualization of occlusion scores) and subsequently focus on the non-occluded parts of the object to make a robust prediction

of CompositionalNets to localize occluders accurately, despite being trained with class labels only.

Finally, we note that this article extends the conference papers (Kortylewski et al. 2020a; Wang et al. 2020) in multiple ways: (1) We present the models of both papers coherently in a common theoretical framework and perform a number additional experiments, including ablation studies. (2) We show that a bad generalization to out-of-distribution examples in terms of partial occlusion is not just a limitation of the VGG-16 network. Instead, our experiments show that it is a general challenge for a variety of advanced deep architectures. (3) We find that CompositionalNets learned from residual backbones can use two fundamentally different approaches to achieve robustness to partial occlusion: Invariance to occlusion and localization of occluders. We observe that the combination of both approaches enables these models to achieve the highest robustness. (4) We study the interpretability of CompositionalNets quantitatively and show that the predictions of CompositionalNets are highly interpretable in terms of part detection, occluder localization and pose estimation. (5) We study large-scale classification of non-vehicle objects with CompositionalNets and achieve very promising results.

In summary, this article shows that the recognition of partially occluded objects poses a general and fundamental challenge to deep models. We give important insights into how this limitation can be overcome by unifying deep models with compositional models, and we show that the resulting CompositionalNets are not just more robust but also much more interpretable compared to their non-compositional counterparts.

2 Related Work

2.1 Object Recognition Under Occlusion

Many recent studies (Zhu et al. 2019; Kortylewski et al. 2020b) have shown that, current deep neural networks are much less robust to partial occlusion compared with humans at object classification. Fawzi and Frossard (2016) show that DCNNs are vulnerable to partial occlusions simulated by masking small patches of the input image. Several following works (DeVries and Taylor 2017; Yun et al. 2019) propose to augment the training data by masking out patches from the image. Our experimental results in Sect. 6.2 show that such data augmentation approaches only have limited beneficial effects. Moreover, data augmentation increases the amount of training data and thus the training time and cost. Therefore, it is desirable to develop novel neural network architectures that are inherently robust to partial occlusion. Xiao et al. (2019) propose TDAPNet, a deep network with an attention mechanism that masks out occluded features in lower layers to increase the robustness of the model against occlusion. Though it can work reliably on artificial occlusion, our results show that this model does not perform well on images with real occlusion.

Compared to image classification, object detection additionally involves the estimation of the object location and bounding box. While a search over the image can be implemented efficiently, e.g. using a scanning window (Lampert et al. 2008), the number of potential bounding boxes is combinatorial in the number of pixels. Currently, the most widely used approach for solving this problem is to use region proposal networks (RPNs) (Girshick et al. 2014) which enable the training of fast approaches for object detection (Girshick 2015; Ren et al. 2015; Cai and Vasconcelos 2018). However, our experiments in Sect. 6.3 demonstrate that RPNs do not estimate the location and bounding box of an object correctly under occlusion, which consequentially deteriorates the performance of these approaches.

To resolve this problem, a boosted cascade-based method for detecting partially visible objects has been proposed by Yan and Liu (2015). However, this approach is based on hand-crafted features and can only be applied to images which are artificially occluded by cutting out patches. A number of deep learning based approaches have also been proposed for detecting occluded objects (Zhang et al. 2018b; Narasimhan 2019), but they require detailed part level annotation to reconstruct the occluded objects. The work of Xiang and Savarese (2013) proposes to use 3D models and treat occlusion as a multi-label classification problem. However, in real-world scenarios, the classes of the occluders can be difficult to model and are often not known a-priori. Other approaches focus on videos or stereo images (Li and Yuan 2018; Jian Sun and Kang 2018). In this work, we consider the problem of partial occlusion in still images.

In contrast to deep learning approaches, generative compositional models (Jin and Geman 2006; Zhu et al. 2008b; Fidler et al. 2014; Dai et al. 2014; Kortylewski et al. 2019) have been shown to be inherently robust to partial occlusion. Such models have been successfully applied for detecting object parts (Wang et al. 2017; Zhang et al. 2018c) and recognizing 2D patterns (George et al. 2017; Kortylewski and Vetter 2016) under partial occlusion. Such part-based voting approaches (Wang et al. 2017; Zhang et al. 2018c) perform reliably for semantic part detection under occlusion, but they assume a fixed size bounding box and are viewpoint specific, which limits their applicability in the context of object detection.

2.2 Relation of CompositionalNets to And-Or Graphs

And-Or graphs (AOG) (Nilsson et al. 1980; Jin and Geman 2006; Dechter and Mateescu 2007) have been investigated e.g. to build hierarchical part-based models for human parsing and for object detection. Intuitively, the or-nodes allow the model to learn different object/part configurations, while the and-node decomposes them into smaller components. Early works in this direction (Chen et al. 2008; Zhu et al. 2008a; Li et al. 2013) relied on pre-defined parts and pre-defined graph structures. To learn the AOG model with less supervision, Zhu et al. (2008b) use recursive compositional clustering. However, this method may lead to unexplainable parts and structures. Song et al. (2013) used an over-complete set of shape primitives to quantize the image lattices and then organized them into an AOG by exploiting their compositional relations through iterative cutting. Xia et al. (2016) explicitly defined parts and part compositions, which correspond to the leaf node and non-leaf node in the AOG respectively. Then they used a score function with pre-defined adjacent part pairs to learn the structure of the AOG, which still required considerable amount of human input. Wu et al. (2015) made use of a large number of synthetic images generated by CAD simulations, on which 17 semantic parts were manually labeled. They enumerated all configurations observed from the synthetic data and then used a graph compression algorithm to get the refined AOG structure. All these models learned the AOG in two steps: after the graph structure was decided, variants of latent structural SVM was used to learn model parameters. In a different approach, Lin et al. (2014) learned AOG by joint optimization of both model structures and parameters. The resulting model worked on object shape detection with moderate performance and may not be easily applied to general object detection.

Most of the works on AOGs used low-level features (e.g., HOG features) to model the part appearance, which may limit their capacity and discriminative power. Furthermore, none of these works modeled occluders explicitly or tested their method on images with different level of occlusions, therefore it is unclear how those models can be made robust to occlusion. CompositionalNets can be considered to be complementary to and-or-graphs. In fact, our mixture model can be interpreted as simple two-layered and-or-graph, where each mixture components combines parts (vMF kernels) for a certain object pose, and the final class score is an "or"-combination over the different object poses (mixture components). While our model could be generalized to have multiple layers with and-or-nodes to introduce more flexibility in the representation, the focus of this work is robustness to partial occlusion. Furthermore, our experiments show that a two-layered and-or graph seems to be good enough to achieve high performance at image classification and detection on popular datasets such as PASCAL3D+, MS-COCO and ImageNet. Moreover, and-or-graphs often require considerable amounts of supervision for the graph structure (Chen et al. 2008; Zhu et al. 2008a; Li et al. 2013) or are not well interpretable (Zhu et al. 2008b; Song et al. 2013), whereas our graph is learned from class supervision only and still learns a meaningful human-interpretable representation.

2.3 Deep Compositional Models in Computer Vision

An early work from Liao et al. (2016) proposes to integrate compositionality into DCNNs by regularizing the feature representations of DCNNs to cluster during learning. Their qualitative results show that the resulting feature clusters resemble part-like detectors. Zhang et al. (2018a) also demonstrate that part detectors emerge in DCNNs by restricting the activations in feature maps to have a localized distribution. However, these approaches have not been shown to enhance the robustness of deep models to partial occlusion. Other related works propose to regularize the convolution kernels to be sparse (Tabernik et al. 2016), or to force feature activations to be disentangled for different objects (Stone et al. 2017). As the compositional model is not explicitly incorporated but rather implicitly encoded within the parameters of the DCNNs, the resulting models remain black-box and not expedcted to be robust to partial occlusion. A number of works (Li et al. 2019; Tang et al. 2018, 2017) use differentiable graphical models to integrate part-whole compositions into DCNNs. However, these models are purely discriminative and thus are also deep networks with no internal mechanism to account for partial occlusion. Girshick et al. (2015) discussed that compositional deformable part models can be formulated as neural networks. However, they do not evaluate their models' robustness to partial occlusion nor its explainability. Kortylewski et al. (2020b) propose to learn a generative dictionary-based compositional model using the features of a DCNN. Instead of merging the compositional model into the DCNN, they use it as a "backup" for an independently trained DCNN. Only when the DCNN classification score falls below a certain threshold, the prediction will be substituted by the output of the compositional model.

2.4 Explainable Computer Vision Models

Many post-hoc methods have been proposed to explain what has been encoded in the intermediate layers of DCNNs. Several works (Le 2013; Zhou et al. 2015) visualize a real or generated input that activates a filter most to study the roles of individual units inside neural networks. Similarly, Nguyen et al. (2016) synthesize prototypical images for individual units by learning a feature inversion mapping, while Bau et al. (2017) visualize segmentation masks extracted from filter activations and assign concept labels automatically. On the other hand, the works of (Mahendran and Vedaldi 2015; Simonyan et al. 2013; Zeiler and Fergus 2014) use variants of back-propagation to identify or generate salient image features. Moving beyond studying individual hidden units, Wang et al. (2015) use clusters of activations from all units in a layer and shows that the cluster centers yield better part detectors. Alain and Bengio (2016) probe mid-layer filters by training linear classifiers on the intermediate activations. They also analyze the information dynamics among layers and its effect on the final prediction. The work of Fong and Vedaldi (2018) shows that filter embeddings better characterize the meaning of a representation and its relationship to other concepts. Most of these works evaluate their results using human judgments.

Unlike the post-hoc methods that focus on visualizing/analyzing pre-trained DCNNs, other approaches aim to learn more meaningful representations during the network training stage. The work of Ross et al. (2017) explains and regularizes differentiable models by examining and selectively penalizing their input gradients, but this me- thod requires extra annotation from human experts. Hu et al. (2016) regularize the learning process by introducing an iterative distillation method that transfers the structured information of logic rules into the weights of neural networks. Stone et al. (2017) encourage networks to form representations that disentangle objects from their surroundings and from each other, but they do not obtain part-level semantics explicitly. Sabour et al. (2017) propose a capsule model, which used a dynamic routing mechanism to parse the entire object into a parsing tree of capsules, and each capsule may encode a specific meaning. The work of Zhang et al. (2018a) invents a generic loss to regularize the representation of a filter to improve its interpretability.

In this work, we unify generative compositional models and deep convolutional neural networks into a joint architecture with innate robustness to partial occlusion. The generative nature of the model enables it to localize occluders and to recognize objects based on the spatial configuration of visible object parts. CompositionalNets are naturally interpretable as their predictions can be understood in terms of part detection, occluder localization and viewpoint estimation.

3 CompositionalNets for Image Classification

In this section, we introduce CompositionalNets, a neural architecture design that replaces the fully-connected classification head of deep networks with a differentiable generative compositional model. We extend CompositionalNets to object detection in Sect. 4 and discuss how CompositionalNets can be trained in an end-to-end manner in Sect. 5.

3.1 A Generative Compositional Model of Neural Feature Activations

We denote a feature map $F^l \in \mathbb{R}^{H\times W\times D}$ as the output of a layer l in a DCNN, with D being the number of channels. $f_i^l \in \mathbb{R}^D$ is a feature vector in F^l at position i on the 2D lattice $\mathcal{P}$ of the feature map. In the remainder of this section we omit the superscript l for notational simplicity because this is fixed a-priori in our experiments.

Our goal is to learn a generative model $p(F|y)$ of the real-valued feature activations F for an object class y. In the following, we assume the viewpoint of the object to be known. Later, we generalize the model to 3D objects with varying viewpoints. We define the probabilistic model $p(F|y)$ to be a mixture of von-Mises-Fisher (vMF) distributions:

$$p(F|\theta_y) = \prod_i p(f_i|\mathcal{A}_{i,y}, \Lambda) \tag{1}$$

$$p(f_i|\mathcal{A}_{i,y}, \Lambda) = \sum_k \alpha_{i,k,y} p(f_i|\lambda_k), \tag{2}$$

where $\theta_y = \{\mathcal{A}_y, \Lambda\}$ are the model parameters and $\mathcal{A}_y = \{\mathcal{A}_{i,y}\}$ are the parameters of the mixture models at every position $i \in \mathcal{P}$ on the 2D lattice of the feature map F. Note that the probabilistic model defined in Eq. 1 has a tree-like independence structure and therefore enables efficient inference (Kortylewski et al. 2019). Moreover,

$$\mathcal{A}_{i,y} = \left\{ \alpha_{i,0,y}, \ldots, \alpha_{i,K,y} | \sum_{k=0}^{K} \alpha_{i,k,y} = 1 \right\} \tag{3}$$

are the mixture coefficients, K is the number of mixture components and $\Lambda = \{\lambda_k = \{\sigma_k, \mu_k\} | k = 1, \ldots, K\}$ are the variance and mean of the vMF distribution:

$$p(f_i|\lambda_k) = \frac{e^{\sigma_k \mu_k^T f_i}}{Z(\sigma_k)}, \|f_i\| = 1, \|\mu_k\| = 1, \tag{4}$$

where $Z(\sigma_k)$ is the normalization constant. As $Z(\sigma_k)$ is difficult to compute for high-dimensional data (Banerjee et al. 2005), we assume σ_k to be fixed a-priori. Hence, the normalization constant is the same for each mixture component and does not need to be computed explicitly when optimizing the parameters. After learning the vMF cluster centers $\{\mu_k\}$ with maximum likelihood optimization, they resemble feature activation patterns that frequently occur in the training data. Interestingly, feature vectors that are similar to one of the vMF cluster centers, are often induced by image patches that are similar in appearance and often even share semantic meanings (see Fig. 2). This property was also observed in a number of related works that used clustering in the neural feature space (Wang et al. 2015; Liao et al. 2016; Wang et al. 2017). Subsequently, we learn the mixture coefficients $\alpha_{i,k,y}$ with maximum likelihood estimation from the training images. Intuitively, $\alpha_{i,k,y}$ describes the expected activation of a cluster center μ_k at a position i in a feature map F for a class y.

3.2 Viewpoint Modeling

An important property of convolutional networks is that the spatial information from the image is preserved in the feature maps. Hence, the set of mixture coefficients $\mathcal{A}_y$ intuitively can be thought of as being a 2D template that describes the expected spatial activation pattern of parts in an image for a given class y - e.g. where the tires of a car are expected to be located in an image. Therefore, our proposed vMF model intuitively accumulates the part detections spatially into a hypothesis about the objects' presence. Note that this implements a part-based voting mechanism.

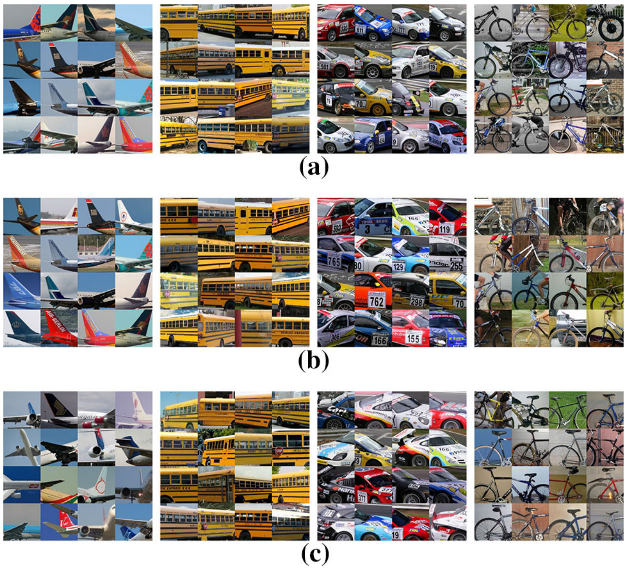


Fig. 2 Illustration of vMF kernels μ_k learned from: **a** VGG-16, **b** ResNet50 and **c** ResNext. We visualize image patterns from the training data that activate a vMF kernel the most. Note that feature vectors that are similar to one of the vMF kernels, are often induced by image patches that have similar geometry and appearance. Furthermore, we were able to find vMF kernels of each backbone that represent similar visual concepts (vertical stabilizer, school bus side, number sticker on car, top of bicycle wheel)

As the spatial pattern of parts varies significantly with the 3D pose of the object, we represent 3D objects as a mixture of compositional models:

$$p(F|\Theta_y) = \sum_m \nu^m p(F|\theta_y^m), \tag{5}$$

with $\mathcal{V} = \{\nu^m \in \{0, 1\}, \sum_m \nu^m = 1\}$ and $\Theta_y = \{\theta_y^m, m = 1, \ldots, M\}$. Here M is the number of compositional models in the mixture distribution and ν_m is a binary assignment variable that indicates which mixture component is active. Intuitively, each mixture component m will represent a different viewpoint of an object (see Experiments in Sect. 6.4.3).

The parameters of the mixture components $\{\mathcal{A}_y^m\}$ need to be learned in an EM-type manner by iterating between estimating the assignment variables $\mathcal{V}$ and maximum likelihood estimation of $\{\mathcal{A}_y^m\}$. We discuss how this process can be performed in a neural network in Sect. 5.2.

3.3 Occlusion Modeling

Following the approach presented in Kortylewski (2017), compositional models can be augmented with an occlusion model. The intuition behind an occlusion model is that at each position i in the image either the object model $p(f_i|\mathcal{A}_{i,y}^m, \Lambda)$

or an occluder model $p(f_i|\beta, \Lambda)$ is active (note that this is closely related to robust statistics (Huber 2011)):

$$p(F|\theta_y^m, \beta) = \prod_i p(f_i, z_i^m = 0)^{1-z_i^m} p(f_i, z_i^m = 1)^{z_i^m}, \quad (6)$$

$$p(f_i, z_i^m = 1) = p(f_i|\beta, \Lambda)\, p(z_i^m = 1), \quad (7)$$

$$p(f_i, z_i^m = 0) = p(f_i|\mathcal{A}_{i,y}^m, \Lambda)\,(1 - p(z_i^m = 1)). \quad (8)$$

The binary variables $\mathcal{Z}^m = \{z_i^m \in \{0, 1\} | i \in \mathcal{P}\}$ indicate if the object is occluded at position i for mixture component m. The occlusion prior $p(z_i^m = 1)$ is fixed a-priori. We use a mixture of several occluder models that are learned in an unsupervised manner:

$$p(f_i|\beta, \Lambda) = \prod_n p(f_i|\beta_n, \Lambda)^{\tau_n} \quad (9)$$

$$= \prod_n \Big(\sum_k \beta_{n,k}\, p(f_i|\sigma_k, \mu_k) \Big)^{\tau_n}, \quad (10)$$

where $\{\tau_n \in \{0, 1\}, \sum_n \tau_n = 1\}$ indicates which occluder model explains the data best. Note that the model parameters β are independent of the position i in the feature map and thus the model has no spatial structure.

The parameters of the occluder models β_n are learned from clustered features of random natural images that do not contain any object of interest (see Fig. 3a). Hence, the mixture coefficients $\beta_{n,k}$ intuitively describe the expected activation of μ_k in regions of natural images. While it is possible to use just a single occluder model (Kortylewski et al. 2020b), we found that having a mixture model slightly increases the classification performance and the occluder localization performance. The reason is that different occluder models can focus at explaining part distributions for different typical regions in images e.g. uniform colored image patches or textured regions. We illustrate this in Fig. 3b by visualizing patches from the training data in Fig. 3a that have the highest likelihood for five different occluder models. Note that the purpose of the occluder models is not to be purely specific to one particular texture type, they also need to be general enough to explain a variety of local image patterns which the object model (Eq. 1) is not able to explain well. Therefore, there is a trade-off between specificity and generality of the occluder models. We found that using five models balances this trade-off well.

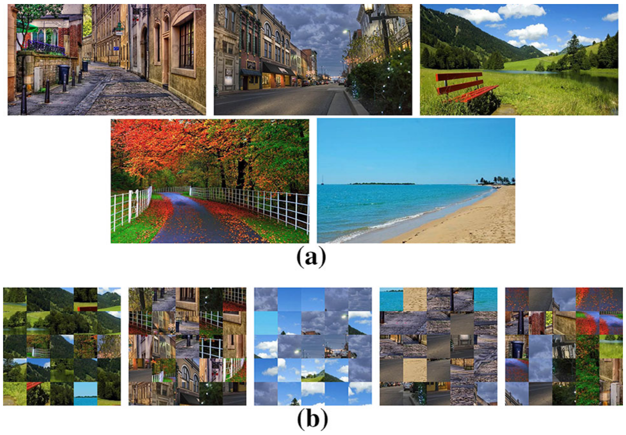


Fig. 3 The occluder models are learned from natural images (**a**). Note that no target object is present in any of these images. **b** Illustrates patches from the training data in (**a**) that have the highest likelihood for each of five occluder models. Note how some models focus more on uniform colored patches while others focus more on textured patches

4 Object Detection with CompositionalNets

Object detection involves the estimation of the object class, the object center, and the object scale (a bounding box around the object). We find that partial occlusion can have significant negative effects on all three tasks. In the following section, we generalize CompositionalNets to enable robust object detection under partial occlusion. We first discuss how CompositionalNets can be generalized to locate objects in images by introducing a detection layer (Sect. 4.1). Subsequently, we show how robust bounding box estimates can be obtained with an advanced compositional voting mechanism in Sect. 4.2. Finally, we discuss the importance of separating the representations of objects and their context in object detection, and show how this can be achieved with CompositionalNets in Sect. 4.3.

4.1 CompositionalNets for Object Localization

A natural way of generalizing CompositionalNets to object detection is to combine them with region proposal networks (RPNs). However, RPNs are typically trained end-to-end and therefore cannot predict the bounding box of strongly occluded objects reliably (see our experiments in Sect. 6.3). Figure 4 illustrates this limitation by depicting the detection results of Faster R-CNN trained with CutOut (DeVries and Taylor 2017) (red box) and a combination of RPN and CompositionalNet (yellow box). We address this limitation by generalizing CompositionalNets to object localization. In particular, we introduce a detection layer that accumulates votes for the object center for all mixtures m over the positions i in the feature map F. In order to achieve this, we compute the object likelihood in a scanning window manner. Thus, we shift the feature map, w.r.t. the object model along all points i from the 2D lattice of the feature map. This process will generate a spatial likelihood map:

$$R = \{p(F_i|\Theta_y) | \forall i \in \mathcal{P}\}, \quad (11)$$

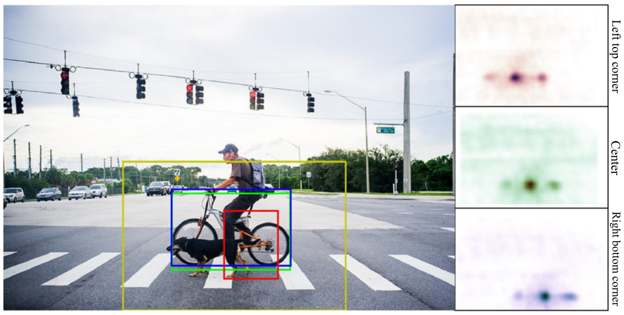


Fig. 4 Example of robust bounding box voting results. Blue box: Ground truth. Red box: Bounding box by Faster R-CNN. Yellow box: RPN + CompNet. Green box: CompNet + part-based bounding box voting. Our proposed part-based voting mechanism generates probability maps (right) for the object center (cyan point), the top left corner (purple point) and the bottom right corner (yellow point) of the bounding box. The final bounding box estimate is generated by combining the voting results

where F_i denotes the feature map centered at the position i. Using this simple generalization we can perform object localization by selecting all maxima in R above a threshold t after non-maximum suppression. Our proposed detection layer can be implemented efficiently on modern hardware using convolution-like operations (see Sect. 6.3 for more details).

4.2 Robust Compositional Bounding-Box Voting

While CompositionalNets can be generalized to localize partially occluded objects using our proposed detection layer, estimating the bounding box of an object under occlusion is more difficult because a significant amount of the object might not be visible (Fig. 4). We solve this problem by generalizing the part-based voting mechanism in CompositionalNets to vote for the bounding box corners in addition to voting for the object center. In this way, we can leverage the viewpoint-specific spatial distribution of part activations for bounding-box estimation. This makes the process highly robust to missing parts. In particular, we learn additional mixture components that model the expected feature activations around bounding box corners $p(F_i|\Theta_y^c)$, where $c = \{ct, br, tl\}$ are the object center ct and two opposite bounding box corners $\{br, tl\}$. Figure 4 illustrates the spatial likelihood maps R^c of all three models. We generate a bounding box using the two points that have maximal likelihood. Note how in Fig. 4 the bounding box can be localized accurately, despite the partial occlusion of the object. We discuss how the model can be learned in an end-to-end manner in Sect. 5.3.

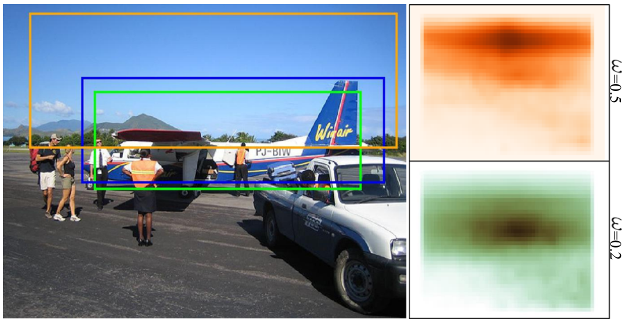


Fig. 5 Influence of context in airplane detection under occlusion. Blue box: Ground truth. Orange box: Bounding box of CompositionalNets ($\omega = 0.5$). Green box: Bounding box of Context-aware CompositionalNets ($\omega = 0.2$). Probability maps of the object center are on the right. Note how reducing the influence of the context improves the localization response

4.3 Context-Aware CompositionalNets

While in image classification, the object of interest often dominates a large part of the image, in object detection the object is embedded in a larger context that is often biased in the training data (e.g. airplanes often have blue background). This gives rise to a critical problem when aiming to detect partially occluded objects. Intuitively, the objects' context can be useful for recognizing objects due to biases in the data. However, relying too strongly on context can be misleading, because if objects are strongly occluded (Fig. 5) the detection thresholds must be lowered. This, in turn, increases the influence of the objects context and leads to false-positive detections in regions with no object. Hence, it is important to have control over the influence of contextual cues on the detection result.

We overcome this issue by explicitly separating the representation of the context from that of the object, to control the influence of the contextual information on the detection result. In particular, we represent the feature map F as a mixture of two models:

$$p(f_i|\mathcal{A}_{i,y}^m, \chi_{i,y}^m, \Lambda) = \omega\, p(f_i|\chi_{i,y}^m, \Lambda) + (1-\omega) p(f_i|\mathcal{A}_{i,y}^m, \Lambda). \quad (12)$$

Here $\{A_{i,y}^m, \chi_{i,y}^m\}$ are the parameters of the context-aware model that is defined to be a mixture of vMF likelihoods (Eq. 2). The parameter ω controls the trade-off between context and object, and is fixed a-priori at test time. Note that setting $\omega = 0.5$ retains the CompositionalNet as proposed in Sect. 3 as the context and object would be weighted equally. Figure 5 illustrates the benefits of reducing the influence of the context on the detection result under partial occlusion. The context parameters $\chi_{i,y}^m$ and object parameters $\mathcal{A}_{i,y}^m$ can be learned from the training data using maximum likelihood estimation, assuming a binary assignment π_i of the

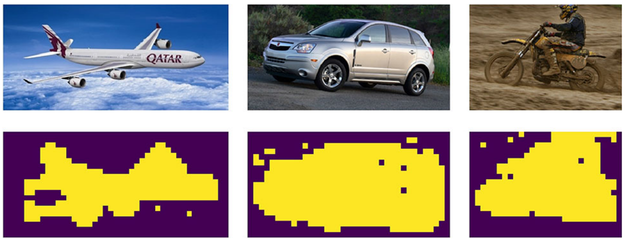


Fig. 6 Context segmentation results. A standard CompositionalNet learns a joint representation of the image including the context. Our context-aware CompositionalNet will separate the representation of the context from that of the object based on the illustrated segmentation masks

feature vectors f_i in the training data to either the context or the object. To obtain this binary assignment, we need to segment the training images into context and object based on the available bounding box annotation. We do so, by assuming that any feature vector with a receptive field outside of the scope of the bounding boxes can be considered to be context. Based on this assumption, we cluster the context features using K-means++ (Arthur and Vassilvitskii 2007) to generate a dictionary of context feature centers $\mathcal{C} = \{\mathcal{C}_q \in \mathbb{R}^D | q = 1, \dots, Q\}$. We apply a threshold on the cosine similarity $s(\mathcal{C}, f_i) = \max_q [(\mathcal{C}_q^T f_i)/(\|\mathcal{C}_q\| \, \|f_i\|)]$ to segment the context and the object in any given training image (Fig. 6).

5 End-to-End Training of CompositionalNets

In the following, we show how inference in CompositionalNets can be formulated as feed-forward neural network (Sect. 5.1) and discuss how CompositionalNets can be trained end-to-end for image classification (Sect. 5.2) and object detection (Sect. 5.3).

5.1 Inference as Feed-Forward Neural Network

The computational graph of our proposed fully generative compositional model is acyclic. Hence, we can perform inference in a single forward pass as illustrated in Fig. 7. We use a standard DCNN backbone to extract a feature representation $F = \psi(I, \Omega) \in \mathbb{R}^{H \times W \times D}$ from the input image I, where Ω are the parameters of the feature extractor (purple tensor in Fig. 7). The pipeline after the feature extractor illustrates the computation of the model likelihood $p(F_i|\Theta)$ when the model is centered at position i in the feature map (illustrated by the dotted black square on the feature tensor F). For image classification, the model will always be positioned at the image center, whereas for object detection, the model will be evaluated at every position in F (as defined in Eq. 11). We omit the subscript in F_i in the following for notational clarity.

After feature extraction the model computes the vMF likelihood function $p(f_i|\lambda_k)$ (Eq. 4). The function is composed of two operations: An inner product $b_{i,k} = \mu_k^T f_i$ and a non-linear transformation $\mathcal{N} = \exp(\sigma_k b_{i,k})/Z(\sigma_k)$. Since μ_k is independent of the position i, computing $b_{i,k}$ is equivalent to a 1×1 convolution of F with μ_k. Hence, the vMF likelihood (Fig. 7 yellow tensor) can be computed by :

$$L = \{\mathcal{N}(F * \mu_k) | k = 1, \dots, K\} \in \mathbb{R}^{H \times W \times K}. \tag{13}$$

The mixture likelihoods $p(f_i|\mathcal{A}_{i,y}^m, \chi_{i,y}^m, \Lambda)$ (Eq. 12) are computed for every position i as a dot-product between the mixture coefficients $\{\mathcal{A}_{i,y}^m, \chi_{i,y}^m\}$ and the corresponding vector $l_i \in \mathbb{R}^K$ from the vMF likelihood tensor L:

$$E_y^m = \{(1-\omega) l_i^T \mathcal{A}_{i,y}^m + \omega l_i^T \chi_{i,y}^m | \forall i \in \mathcal{P}\} \in \mathbb{R}^{H \times W}, \tag{14}$$

(Fig. 7 blue planes). Similarly, the occlusion likelihood can be computed as $O = \{\max_n l_i^T \beta_n | \forall i \in \mathcal{P}\} \in \mathbb{R}^{H \times W}$ (Fig. 7 red plane). Together, the occlusion likelihood O and the mixture likelihoods $\{E_y^m\}$ are used to estimate the overall likelihood of the individual mixtures as $s_y^m = \log p(F|\theta_y^m, \beta) = \sum_i \max(E_{i,y}^m, O_i)$. The final model likelihood is computed as $s_y = \log p(F|\Theta_y) = \max_m s_y^m$ and the final occlusion map is selected accordingly as $\mathcal{Z}_y = \mathcal{Z}_y^{\bar{m}} \in \mathbb{R}^{H \times W}$ where $\bar{m} = \text{argmax}_m s_y^m$.

5.2 End-to-end Training for Image Classification

Our model is fully differentiable and can be trained end-to-end using backpropagation. In our image classification experiments, context-awareness does not have a significant influence as the background is largely cropped out. Therefore, we use CompositionalNets as defined in Sect. 3 for image classification. Hence, the trainable parameters of a CompositionalNet are $T = \{\Lambda, \mathcal{A}_y\}$. We optimize those parameters jointly using stochastic gradient descent. The loss function is composed of three terms:

$$\begin{aligned} \mathcal{L}(y, y', F, T) = \mathcal{L}_{class}(y, y') + \gamma_1 \mathcal{L}_{vmf}(F, \Lambda) \qquad (15) \\ + \gamma_2 \mathcal{L}_{mix}(F, \mathcal{A}_y). \qquad (16) \end{aligned}$$

$\mathcal{L}_{class}(y, y')$ is the cross-entropy loss between the network output y' and the true class label y. We use a temperature $\mathcal{T}$ in the softmax classifier: $f(y)_i = \frac{e^{y_i \cdot \mathcal{T}}}{\Sigma_i e^{y_i \cdot \mathcal{T}}}$, with $\mathcal{T} = 2$. $\mathcal{L}_{vmf}$ and $\mathcal{L}_{mix}$ regularize the parameters of the compositional model to have maximal likelihood for the features in F. The parameters $\{\gamma_1, \gamma_2\}$ control the trade-off between the loss terms.

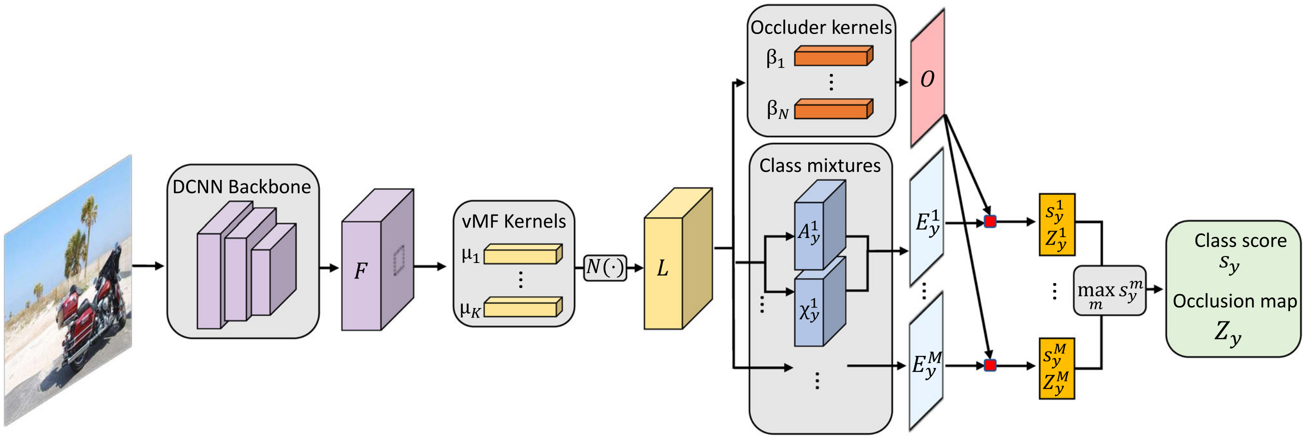


Fig. 7 Feed-forward inference with a context-aware CompositionalNet. A DCNN backbone is used to extract the feature map F, followed by a convolution with the vMF kernels $\{\mu_k\}$ and a non-linear vMF activation function $\mathcal{N}(\cdot)$. The resulting vMF likelihood L is used to compute the occlusion likelihood O using the occluder kernels $\{\beta_n\}$. Furthermore, L is used to compute the context-aware mixture likelihoods $\{E_y^m\}$ using the mixture models of the object $\{A_y^m\}$ and the context $\{\chi_y^m\}$. O and $\{E_y^m\}$ compete in explaining L (red box) and are combined to compute an occlusion robust score $\{s_y^m\}$. The binary occlusion maps $\{Z_y^m\}$ indicate which positions in L are occluded. The final class score s_y is computed as $s_y = \max_m s_y^m$ and the occlusion map Z_y is selected accordingly

The vMF cluster centers μ_k are learned by maximizing the vMF-likelihoods (Eq. 4) for the feature vectors f_i in the training images. We keep the vMF variance σ_k constant, which also reduces the normalization term $Z(\sigma_k)$ to a constant. We assume a hard assignment of the feature vectors f_i to the vMF clusters during training. Hence, the free energy to be minimized for maximizing the vMF likelihood (Wang et al. 2017) is:

$$\mathcal{L}_{vmf}(F, \Lambda) = -\sum_i \max_k \log p(f_i|\mu_k) \tag{17}$$

$$= C\sum_i \max_k \mu_k^T f_i, \tag{18}$$

where C is a constant. Intuitively, this loss encourages the cluster centers μ_k to be similar to the feature vectors f_i.

In order to learn the mixture coefficients $\mathcal{A}_y^m$ we need to maximize the model likelihood (Eq. 5). We can avoid an iterative EM-type learning procedure by making use of the fact that the the mixture assignment ν_m and the occlusion variables z_i have been inferred in the forward inference process. Furthermore, the parameters of the occluder model are learned a-priori and then fixed. Hence the energy to be minimized for learning the mixture coefficients is:

$$\mathcal{L}_{mix}(F, \mathcal{A}_y) = -\sum_i (1 - z_i^{\uparrow}) \log\Big[\sum_k \alpha_{i,k,y}^{m\uparrow} p(f_i|\lambda_k)\Big] \tag{19}$$

Here, $z_i^{\uparrow}$ and $m^{\uparrow}$ denote the variables that were inferred in the forward process (Fig. 7).

5.3 Training CompositionalNets for Object Detection

For object detection, we train context-aware CompositionalNets as proposed in Sect. 4. The trainable parameters of the model are $T^c = \{\Lambda, \{\mathcal{A}_y^c\}, \{\chi_y^c\}\}$ where $c \in \{ct, br, tl\}$. The loss function has three main objectives. The model should explain data with maximal likelihood ($\mathcal{L}_g$), while also localizing ($\mathcal{L}_{detect}$) and classifying ($\mathcal{L}_{class}$) the object accurately in the training images.:

$$\mathcal{L} = \mathcal{L}_{class}(y, y') + \epsilon_1 \sum_c \big(\mathcal{L}_g(F^c, T^c) \tag{20}$$

$$+ \epsilon_2 \mathcal{L}_{detect}(\hat{S}^c, \bar{S}^c, F, T^c)\big), \tag{21}$$

where ϵ_1, ϵ_2 control the trade-off between the loss terms. While $\mathcal{L}_g$ is learned from images $\hat{I}^c$ with feature maps F^c that are centered at $c \in \{c, bl, tr\}$, the other losses are learned from unaligned training images I with feature maps F.

Generative Regularization The model is regularized to be generative in terms of the neural feature activations by optimizing:

$$\mathcal{L}_g(F^c, T) = \mathcal{L}_{vmf}(F^c, \Lambda) \tag{22}$$

$$+ \sum_i \mathcal{L}_{con}(f_i^c, \mathcal{A}_{i,y}^c, \chi_{i,y}^c), \tag{23}$$

where $\mathcal{L}_{vmf}(F^c, \Lambda)$ is defined in Eq. 17 and the parameters of the context-aware model $\mathcal{A}_y^c$ and χ_y^c are learned by optimizing the context loss:

$$\mathcal{L}_{con}(f_i^c, \mathcal{A}_{i,y}^c, \chi_{i,y}^c) = \pi_i \mathcal{L}_{mix}(f_i^c, \mathcal{A}_{i,y}^c) \tag{24}$$

$$+ (1 - \pi_i)\mathcal{L}_{mix}(f_i^c, \chi_{i,y}^c). \tag{25}$$

Here, $\pi_i \in \{0, 1\}$ is a context assignment variable that indicates if a feature vector f_i belongs to the context or to the object model and $\mathcal{L}_{mix}$ is defined in Eq. 19. We estimate the context assignments a-priori using a simple binary segmentation as described in Sect. 4.3.

Localization and Bounding Box Estimation We denote the normalized response map of the ground truth class as $\hat{S}^c \in \mathbb{R}^{H\times W}$ and the ground truth annotation as $\bar{S}^c \in \mathbb{R}^{H\times W}$. The elements of the response map are computed as:

$$\hat{s}_i^c = \frac{s_{i,\hat{m}}}{\sum_i s_{i,\hat{m}}}, \hat{m} = \underset{m}{\text{argmax}} \max_i p(f_i|\mathcal{A}_{i,y}^m, \chi_{i,y}^m, \Lambda). \quad (26)$$

The ground truth map $\bar{S}^c$ is a binary map where the ground truth position $\bar{i}$ is set to $\bar{S}^c(\bar{i}) = 1$ and all other entries are set to zero. The detection loss is then defined as:

$$\mathcal{L}_{detect}(\hat{S}^c, \bar{S}^c, F, T^c) = 1 - \frac{2 \cdot \Sigma_i(\hat{s}_i^c \cdot \bar{s}_i^c)}{\sum_i \hat{s}_i^c + \sum_i \bar{s}_i^c}. \quad (27)$$

6 Experiments

We give an overview of the datasets used for evaluation and the training setup in Sect. 6.1. Subsequently, we extensively evaluate our model at image classification and object detection of partially occluded objects in Sects. 6.2 & 6.3. Finally, we show that CompositionalNets are human interpretable, as their predictions can be understood in terms of occluder localization (Sect. 6.4.1), object part detection (Sect. 6.4.2) and object pose estimation (Sect. 6.4.3). The code for basic CompositionalNets as originally introduced in Kortylewski et al. (2020a) is publicly available [1].

6.1 Datasets and Training Setup

Datasets for Image Classification We evaluate our model on the *Occluded-P3D+-Vehicles* dataset as proposed in Wang et al. (2017) and extended in Kortylewski et al. (2020b). The dataset is based on images of vehicles from the PASCAL3D+ dataset (Xiang et al. 2014) that were synthetically occluded with four different types of occluders: segmented *objects* as well as patches with *constant white color*, *random noise* and *textures* (see Fig. 8a for examples). The amount of partial occlusion of the object varies in four different levels: 0% (L0), 20-40% (L1), 40-60% (L2), 60-80% (L3).

Furthermore, we introduce a dataset with images of real occlusions which we term *Occluded-COCO-Vehicles*. It contains the same classes as the Occluded-P3D+-Vehicles dataset. The dataset was generated by categorizing objects from the MS-COCO (Lin et al. 2014) into the four occlusion levels as defined in the Occluded-P3D+-Vehicles dataset. To achieve this, we relate the amount of object that is visible in the MS-COCO images to the one from the Occluded-P3D+-Vehicles based on the segmentation masks that are available in both datasets. The numbers of test images for each occlusion level are: 2036 (L0), 768 (L1), 306 (L2), 73 (L3). For training purpose, we define a separate training dataset of 2036 images from level L0. Figure 8b illustrates some example images from this dataset.

While the current generation of CompositionalNets has been primarily developed for recognizing a smaller number of vehicle-type objects, we also want to study if CompositionalNets retain their robustness to partial occlusion even when tested at a larger scale with non-vehicle type objects. For this, we introduce the *Occluded-ImageNet* dataset. In particular, we select different numbers of classes {25, 50, 100} from the ImageNet dataset (Deng et al. 2009) to study the influence of the dataset scale on the performance. We crop the objects from the training images with available annotation to make our results comparable to the Occluded-P3D+-Vehicles and Occluded-COCO-Vehicles data. We randomly split the data into 50 test images per class and assign the remaining images to the training data. This results in {12241, 25120, 49263} training and {1250, 2500, 5000} test images for the different sized subsets. We artificially occlude the test images by using segmented objects from the MS-COCO dataset. Note that to simulate occlusion, we only use those object classes from MS-COCO that do not overlap with the ImageNet classes. The amount of partial occlusion of the object varies in four different levels (L0, L1, L2, L3) as in the Occluded-P3D+-Vehicles dataset.

Datasets for Object Detection The object detection datasets are defined in a similar way as the classification data. We synthesize an *Occluded-P3D+-Vehicles-Detection* dataset, which contains vehicles at a certain scale and various levels of occlusion. The occluders, which include humans, animals and plants, are cropped from the MS-COCO dataset (Lin et al. 2014). In an effort to accurately depict real-world occlusions, we superimpose the occluders onto the object, such that the occluders are placed not only inside the bounding box of the objects but also on the background. We generate the dataset in a total of 9 occlusion levels along two occlusion dimensions: We define three levels of occlusion of the object (FG-L1: 20–40%, FG-L2:40–60% and FG-L3:60–80% of the object area is occluded). Furthermore, we define three levels of occlusion of the context around the object (BG-L1: 0–20%, BG-L2:20–40% and BG-L3:40–60% of the context area is occluded). Example images are shown in Fig. 8c. In order to evaluate the tested models on real-world occlusions, we test them on the subset of the MS-COCO dataset as defined for classification (Fig. 8d).

Model Initialization We initialize the vMF kernels $\{\mu_k\}$ and the mixture components $\{\mathcal{A}_y, \chi_y\}$ by maximum like-

[1] https://github.com/AdamKortylewski/CompositionalNets

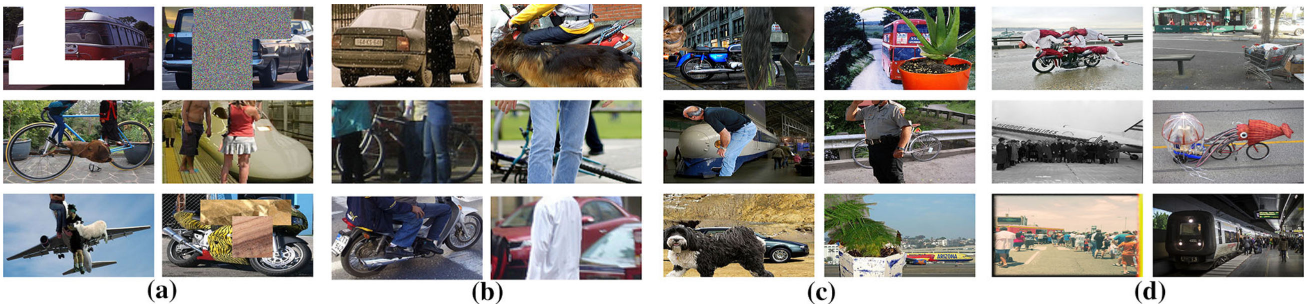


Fig. 8 Example images of partially occluded objects for image classification (**a**, **b**) and object detection (**c**, **d**), with artificial occlusion (**a**, **c**) and real occlusion (**b**, **d**)

lihood estimation after clustering the training data. We compute the mixture assignments using spectral clustering with the hamming distance between vMF kernel activations in the pool4 layer in VGG-16 of all training images. The intuition is that objects with a similar viewpoint and 3D structure will have similar vMF activation patterns, and thus should be assigned to the same mixture component.

Training Setup for Classification We train CompositionalNets from the feature activations of the layer before the classifier for several popular DCNNs: VGG-16 (Simonyan and Zisserman 2014), ResNet50 (He et al. 2016) and ResNext (Xie et al. 2017). All models were pretrained on ImageNet(Deng et al. 2009). We set the vMF variance to $\sigma_k = 30, \forall k \in \{1, \ldots, K\}$. We train the parameters of the generative model $\{\{\mu_k\}, \{\mathcal{A}_y, \chi_y\}\}$ using backpropagation and keep the parameters of the backbone Ω fixed (training Ω did not have significant effects on the results). We learn the parameters of $n = 5$ occluder models $\{\beta_1, \ldots, \beta_n\}$ in an unsupervised manner as described in Sect. 3.1 and keep them fixed throughout the experiments. We set the number of mixture components to $M = 4$. The mixing weights of the loss are set to: $\gamma_1 = 3.0$, $\gamma_2 = 3.0$. We train for 50 epochs using stochastic gradient descent with momentum $r = 0.9$ and a learning rate of $lr = 0.01$. Our reported parameter settings are very general as they are fixed in all our experiments even for different datasets.

Training Setup for Object Detection We optimize the loss described in Eq. 20, with $\epsilon_1 = 0.2$ and $\epsilon_2 = 0.4$. We apply the Adam Optimizer (Kingma and Ba 2014) with different learning rates on different parts of the network: $lr_{vMF} = 2 \cdot 10^{-5}$, $lr_{mix\ model} = lr_{corner\ model} = 5 \cdot 10^{-5}$. The model is trained for a total of 2 epochs with 10,600 iteration per epoch. The training takes three hours in total on a machine with 4 Nvidia TITAN Xp GPUs.

Runtime Empirically, we find that for classification the training and inference time increase by a factor of 2–3 depending on the backbone when compared to a standard network with fully-connected classification head. However, note that we have not invested into optimizing the training or inference time of our model, whereas the runtime of standard models has been extensively optimized by hardware companies and the vision community.

6.2 Image Classification Under Occlusion

Occluded-P3D+ Table 1 reports the results of VGG-16, ResNet50 and ResNext that were fine-tuned with the respective training data. Furthermore, we show the results of dictionary-based compositional models (CoD) (Kortylewski et al. 2020b) and TDAPNet (Xiao et al. 2019). We also report CompositionalNets learned from the pool4 and pool5 layer of the VGG-16 network respectively (CompNet-VGG-[p4 & p5]) and CompositionalNets learned from features of the last residual block (RB4) of ResNet50 (CompNet-Res50) and both the last (RB4) and second last (RB3) residual block of ResNext (CompNet-RXT). In this setup, all models are trained with non-occluded images ($L0$), while at test time the models are exposed to images with different amount of partial occlusion ($L0$-$L3$).

Overall, we observe that **DCNNs do not generalize well to out-of-distribution examples in terms of partial occlusion**. They perform well on the type of data that they were exposed to during training (L0) and generalize to a limited extent away from it (L1). However, their performance collapses when objects are strongly occluded (L3).

In contrast, we observe that CompositionalNets generalize very well even under strong occlusion. Moreover, using the higher layers of more powerful residual backbones also significantly increases the performance of CompositionalNets. We also observe that CompNets learned from the RB3 layer of ResNext perform significantly worse, compared to those learned from RB4 (−5.3% on average). In contrast, CompNets learned from pool4 of VGG still give comparable results pool5. We conjecture from this observation that the additional convolutional and residual layers in RB4 are of critical importance for the ResNext backbone.

Occluded MS-COCO Table 2 shows classification results under a realistic occlusion scenario by testing on the

Table 1 Classification results for vehicles of PASCAL3D+ with different levels of artificial occlusion (0%, 20–40%, 40–60%, 60–80% of the object are occluded) and different types of occlusion (w=white boxes, n=noise boxes, t=textured boxes, o=natural objects)

PASCAL3D+ Vehicles Classification under Occlusion

Occ. Area	L0: 0%	L1: 20–40%				L2: 40–60%				L3: 60–80%				Mean
Occ. Type	–	w	n	t	o	w	n	t	o	w	n	t	o	
VGG	99.2	96.9	97.0	96.5	93.8	92.0	90.3	89.9	79.6	67.9	62.1	59.5	62.2	83.6
Resnet50	99.1	96.8	96.8	96.8	91.0	91.6	91.5	91.8	73.4	66.1	69.2	71.4	58.3	84.1
ResNext	**99.7**	98.7	98.5	98.4	94.5	94.5	93.4	92.3	76.7	68.0	62.8	51.2	55.9	83.4
CoD(Kortylewski et al. 2020b)	92.1	92.7	92.3	91.7	92.3	87.4	89.5	88.7	90.6	70.2	80.3	76.9	87.1	87.1
TDAPNet (Xiao et al. 2019)	99.3	98.4	98.9	98.5	97.4	96.1	97.5	96.6	91.6	82.1	88.1	82.7	79.8	92.8
CompNet-VGG-p4	97.4	96.7	96.0	95.9	95.5	95.8	94.3	93.8	92.5	86.3	84.4	82.1	**88.1**	92.2
CompNet-VGG-p5	99.3	98.4	98.6	98.4	96.9	**98.2**	**98.3**	97.3	88.1	90.1	**89.1**	83.0	72.8	93.0
CompNet-Res50-RB4	99.3	98.7	98.7	98.5	96.9	96.9	97.2	96.9	88.7	86.1	83.6	79.4	72.9	91.8
CompNet-RXT-RB3	98.1	95.9	96.3	96.3	94.6	92.1	93.5	92.7	87.4	76.1	80.1	75.2	76.4	88.8
CompNet-RXT-RB4	**99.7**	**99.3**	**99.0**	**99.2**	**98.0**	**98.2**	97.0	**97.5**	**93.0**	**90.3**	83.8	**84.7**	83.3	**94.1**

The best performing method for each experiment is highlighted in bold

CompositionalNets learned from several DCNN backbones (VGG-16, ResNet50,ResNext) outperform related approaches and their non-compositional versions significantly

Table 2 Classification results for vehicles of MS-COCO with different levels of real occlusion (L0: 0%,L1: 20–40%,L2 40–60%, L3:60–80% of the object are occluded)

MS-COCO Vehicles Classification under Occlusion

Train Data	PASCAL3D+					MS-COCO					MS-COCO + CutPaste				
Occ. Area	L0	L1	L2	L3	Avg	L0	L1	L2	L3	Avg	L0	L1	L2	L3	Avg
VGG	97.8	86.8	79.1	60.3	81.0	99.1	88.7	78.8	63.0	82.4	99.3	92.3	89.9	80.8	90.6
ResNet50	98.5	89.6	84.9	71.2	86.1	**99.6**	92.7	86.9	67.1	86.6	**99.6**	94.1	92.5	84.9	92.8
ResNext	98.7	90.7	85.9	75.3	87.7	99.5	92.9	87.6	73.9	88.5	**99.6**	94.5	93.1	89.0	94.1
CoD	91.8	82.7	83.3	76.7	83.6	–	–	–	–	–	–	–	–	–	–
TDAPNet	98.0	88.5	85.0	74.0	86.4	99.4	88.8	87.9	69.9	86.5	98.1	89.2	90.5	79.5	89.3
CompNet-VGG-p4	96.6	91.8	85.6	76.7	87.7	97.7	92.2	86.6	82.2	89.7	98.3	93.8	88.6	84.9	91.4
CompNet-VGG-p5	98.2	89.1	84.3	78.1	87.5	99.1	92.5	87.3	82.2	90.3	99.4	93.9	90.6	90.4	93.5
CompNet-Res50-RB4	98.5	92.6	88.9	83.6	90.9	99.2	**95.2**	91.8	89.0	93.8	99.0	**95.2**	93.4	89.0	94.2
CompNet-RXT-RB3	97.5	91.7	82.3	73.2	86.2	98.2	93.1	84.1	83.6	89.8	98.7	93.8	87.3	84.9	91.2
CompNet-RXT-RB4	**98.8**	**94.0**	**93.5**	**87.7**	**93.5**	99.0	94.8	**93.5**	**91.8**	**94.8**	99.0	95.0	**94.1**	**91.8**	**95.0**

The best performing method for each experiment is highlighted in bold

The training data consists of images from: PASCAL3D+, MS-COCO as well as data from MS-COCO that was augmented with data augmentation in terms of partial occlusion. On average, CompositionalNets outperform their non-compositional versions and related work in all test cases

Occluded-COCO-Vehicles dataset. The models in the first part of the Table (PASCAL3D+) are trained solely on non-occluded images of the PASCAL3D+ data. We can observe that from all standard DCNNs VGG-16 transfers the worst to the MS-COCO data, while ResNet50 and ResNext generalize better, in particular to strongly occluded objects. ResNext even outperforms the recently proposed TDAPNet that was specifically designed for image classification under occlusion. **CompositionalNets consistently outperform non-compositional DCNNs** by a large margin, highlighting the generality of our approach. In particular, CompNet-ResNext-RB4 can generalize very strongly from training on non-occluded PASCAL3D+ data to strongly occluded objects from MS-COCO.

The second part of the table (MS-COCO) shows the classification performance after fine-tuning on the non-occluded training set of the Occluded-COCO-Vehicles dataset. A common pattern of all three standard DCNNs is that their performance increases for low occlusion ($L0 - 2$) while it decreases for stronger occlusion ($L3$). In contrast, the performance of the CompositionalNets increases substantially after fine-tuning for all occlusion levels.

Table 3 Classification results for different number of classes from ImageNet and different levels of artificial occlusion (L0: 0%,L1: 20–40%,L2 40–60%, L3:60–80% of the object are occluded)

ImageNet Classification under Occlusion

Number of classes	25					50					100				
Occ. Area	L0	L1	L2	L3	Avg	L0	L1	L2	L3	Avg	L0	L1	L2	L3	Avg
ResNext + CutOut	**98.6**	69.2	40.8	20.7	57.3	**97.7**	58.6	28.4	13.5	49.6	**97.1**	53.0	25.5	10.5	46.5
CompNet-ResNext-RB4-512	98.1	**70.6**	**46.9**	**31.3**	**61.7**	95.1	60.7	36.5	19.8	53.0	92.9	53.7	30.1	14.7	47.9
CompNet-ResNext-RB4-1024	97.2	69.8	45.9	29.1	60.5	95.4	**61.3**	**37.0**	**21.2**	**53.7**	94.2	**56.1**	**32.6**	**16.0**	**49.7**

The best performing method for each experiment is highlighted in bold

We compare a standard ResNext model trained with strong data augmentation in terms of CutOut with CompositionalNets that are learned from a ResNext backbone with 512 and 1024 vMF kernels. CompositionalNets are more robust under strong occlusion in all experiments and stay competitive in low occlusion scenarios

The third part of Table 2 (MS-COCO-CutPaste) shows classification results after training with strong data augmentation in terms of partial occlusion. In particular, we artificially occlude the non-occluded training images used in the previous experiment with all four types of artificial occluders used in the Occluded-P3D+-Vehicles dataset. This data augmentation increases the dataset by a factor of 5. From the classification results, we can observe that data augmentation increases the performance of the DCNNs significantly. VGG-16 still suffers from strong occlusions. However, ResNet50 and particularly ResNext become significantly more robust to occlusion. The performance of the CompositionalNets learned from VGG-16 increases further when trained with augmented data, whereas the ones learned from ResNet50 and ResNext only benefit slightly, while still outperforming their non-compositional counterparts. Similar to the results in Table 1, CompNet-ResNext-RB3 performs significantly worse compared to CompNet-ResNext-RB4, highlighting the importance of the last layers in the backbone of ResNext. Interestingly, CompNet-ResNext-RB3 performs on par to CompNet-ResNext-p4 under real occlusion, whereas it is less robust to the artificial occluders (Table 1). This suggests that the ResNext features have developed an invariance to partial occlusion from the ImageNet pre-training that does not transfer as well to artificial occlusion. We discuss this in more detail in Sect. 6.4.1.

Occluded ImageNet Table 3 shows classification results for larger scale experiments with non-vehicle objects on the Occluded-ImageNet dataset. We compare a standard ResNext model trained with strong data augmentation using CutOut (DeVries and Taylor 2017) with CompositionalNets learned from a ResNext backbone that were trained with non-augmented images only. We also test CompositionalNets with 512 and 1024 vMF kernels. We observe that CompositionalNets are more robust under strong occlusion in all experiments and stay competitive in low occlusion scenarios. The CompositionalNet performance decreases slightly on non-occluded images in the 100 class experiment (ResNext+CutOut: 97.1; CompNet-ResNext-1024: 94.2). However, CompositionalNets perform better under any amount of partial occlusion even for this large set of non-vehicle objects.

Furthermore, we observe that the number of vMF kernels does not have a critical influence on the performance. This highlights the advantage of the compositional representation which enables an efficient sharing of the vMF kernels among the different classes. Nevertheless, we can observe that increasing the number of vMF kernels has a positive effect on the performance for higher number classes, because the model can represent the objects more accurately when more parts are available.

Overall, we can observe that CompositionalNets are already capable to generalize to large amounts of non-vehicle classes, while retaining their robustness to partial occlusion. This is a very promising result considering that the design of our probabilistic compositional model is particularly suited for the robust analysis of rigid objects such as vehicles. We expect improvements by investigating how CompositionalNets could better model articulated objects such as animals.

Summary of Classification Experiments All classification experiments clearly highlight the robustness of CompositionalNets at classifying partially occluded objects, while also being highly discriminative when objects are not occluded. Overall, CompositionalNets learned from several popular DCNNs show a significantly improved generalization performance compared to non-compositional DCNNs under artificial as well as real occlusion. Notably, CompNet-ResNext trained from non-occluded PASCAL3D+ data performs essentially on par with the best DCNN (ResNext) trained from strongly augmented MS-COCO images. This highlights the data efficiency and strong generalization ability of CompositionalNets. Furthermore, we observe that CompositionalNets have a lot of potential for large-scale classification tasks with non-vehicle type objects.

Table 4 Ablation study for the generative regularization of a CompositionalNet learned from the features of the last residual block (RB4) of ResNext

Ablation Study on PASCAL3D+ Vehicles Classification under Occlusion

Occ. Area	L0: 0%	L1: 20–40%				L2: 40–60%				L3: 60–80%				Mean
Occ. Type	–	w	n	t	o	w	n	t	o	w	n	t	o	
CompNet-ResNext initialization	98.1	95.8	96.0	95.6	91.3	91.7	91.8	91.7	80.8	75.7	77.2	77.1	67.9	87.0
CompNet-ResNext $\gamma_1 = 0, \gamma_2 = 0$	99.5	98.3	97.8	98.0	95.0	94.5	92.9	92.6	81.7	71.0	59.0	59.0	58.3	84.4
CompNet-ResNext $\gamma_1 = 0, \gamma_2 = 3$	99.6	97.7	96.5	97.4	93.6	95.0	91.4	93.8	86.8	87.0	75.3	80.0	77.6	90.1
CompNet-ResNext $\gamma_1 = 3, \gamma_2 = 0$	**99.9**	99.2	98.8	98.9	97.5	97.5	95.6	96.5	92.8	87.8	78.4	81.9	**83.9**	93.0
CompNet-ResNext $\gamma_1 = 3, \gamma_2 = 3$	99.7	**99.3**	**99.0**	**99.2**	**98.0**	**98.2**	**97.0**	**97.5**	**93.0**	**90.3**	**83.8**	**84.7**	83.3	**94.1**

The best performing method for each experiment is highlighted in bold
The results highlight the importance of maximum likelihood regularization of both the vMF kernels and the mixture models

6.2.1 Ablation Study

In Table 4 we show the results of an ablation study using a CompositionalNet learned from a ResNext backbone. After initialization, the CompositionalNet achieves an average performance of 87% and hence already outperforms a standard ResNext architecture by 3.6% on average. Note that the CompositionalNet at this point is not as discriminative at $L0$ and performs on par at $L1$, while significantly outperforming ResNext at $L2$ and $L3$.

When the parameters of the vMF kernels and mixture components are not regularized during training ($\gamma_1 = 0, \gamma_2 = 0$) the overall performance decreases. In particular, the CompositionalNet becomes more discriminative for the type of data it has observed at training time ($L0$) and cannot generalize well to stronger occlusion. Hence, it behaves as one would expect from a standard DCNN.

Regularizing only the mixture components to maximize the likelihood of the vMF kernel activations ($\gamma_1 = 0, \gamma_2 = 3$) increases the performance in all experiments. In particular, the end-to-end training makes CompositionalNets highly discriminative for the within-distribution-data ($L0$) while preserving strong generalization performance for out-of-distribution data ($L1 - L3$). Similarly, regularizing only the vMF kernels to maximize the likelihood of the ResNext features ($\gamma_1 = 3, \gamma_2 = 0$) also increases performance for all experiments significantly for non-occluded as well as strongly occluded data. The best performance is achieved with a joint regularization of the vMF kernels and mixture components ($\gamma_1 = 3, \gamma_2 = 3$).

Table 5 Effect of data augmentation on the classification performance of a ResNext model for vehicles of MS-COCO with different levels of real occlusion (L0: 0%,L1: 20–40%,L2 40–60%, L3:60–80% of the object are occluded)

Data Augmentation for MS-COCO Occluded Vehicles

Occ. Area	L0	L1	L2	L3	Avg
No Augmentation	99.5	92.9	87.6	73.9	88.5
CutMix	99.2	93.2	86.3	75.3	88.5
CutOut	99.3	94.6	89.9	79.5	90.8
CutPaste	**99.6**	94.5	93.1	89.0	94.1
AutoAugment+CutMix	99.2	93.8	87.3	78.8	89.8
AutoAugment+CutOut	99.3	95.2	91.5	83.6	92.4
AutoAugment+CutPaste	99.3	**95.3**	93.2	90.4	94.6
CompNet No Augmentation	99.0	94.8	**93.5**	**91.8**	**94.8**

The best performing method for each experiment is highlighted in bold
Data augmentation with partial occlusion is done with CutOut (DeVries and Taylor 2017), CutMix (Yun et al. 2019), and our proposed CutPaste. We further evaluate the effect of Auto-Augmentation (Cubuk et al. 2018). Note that data augmentation can enhance the robustness to partial occlusion significantly. However, a plain CompositionalNet with ResNext backbone and no data augmentation still slightly outperforms models trained with massive data augmentation

6.2.2 Robustness through Massive Data Augmentation

In Table 5 we evaluate the effect of data augmentation on the robustness of standard neural networks to partial occlusion. In particular, we train a ResNext model and evaluate it under real occlusion on the occluded MS-COCO dataset. We use Auto-Augmentation (Cubuk et al. 2018) with the ImageNet policy that was found in the original paper. Furthermore, we test data augmentation in terms of partial occlusion using CutOut (DeVries and Taylor 2017), CutMix (Yun et al. 2019), and our proposed CutPaste augmentation. Note that CutOut and CutPaste actively remove parts of the image and replace it with *irrelevant* information (black patches in CutOut, and segmented objects in CutPaste). Importantly, the labelling of the image remains the same. In contrast, CutMix generates images of hybrid objects by composing patches of images of two different classes. The labelling of the resulting image in the cross-entropy loss is changed accordingly to be a mixture of the two classes. Note that CutOut and CutPaste simulate partial occlusions more realistically because they add irrele-

Table 6 Detection results on the Occluded-P3D+-Vehicles-Detection dataset under different levels of occlusions. All models were trained on non-occluded images of the PASCAL3D+ dataset except Faster R-CNN with reg., which was trained with CutOut. The results are measured by correct AP(%) @IoU0.5, which means only correct classified images with $IoU > 0.5$ of the predicted bounding box are treated as true-positive. Notice with $\omega = 0.5$ the context-aware model reduces to a CompositionalNet as proposed in Sect. 3

PASCAL3D+ Vehicles Detection under Occlusion

Occ. Area Foreground	FG L0	FG L1			FG L2			FG L3			Mean
Occ. Area Background	BG L0	BG L1	BG L2	BG L3	BG L1	BG L2	BG L3	BG L1	BG L2	BG L3	–
Faster R-CNN	**98.0**	88.8	85.8	83.6	72.9	66.0	60.7	46.3	36.1	27.0	66.5
Faster R-CNN with reg.	97.4	**89.5**	86.3	89.2	76.7	70.6	67.8	54.2	45.0	37.5	71.1
CompNet-VGG-p4-RPN $\omega = 0.5$	74.2	68.2	67.6	67.2	61.4	60.3	59.6	46.2	48.0	46.9	60.0
CompNet-VGG-p4-RPN $\omega = 0$	73.1	67.0	66.3	66.1	59.4	60.6	58.6	47.9	49.9	46.5	59.6
CompNet-VGG-p4 $\omega = 0.5$	91.7	85.8	86.5	86.5	78	77.2	77.9	61.8	61.2	59.8	76.6
CompNet-VGG-p4 $\omega = 0.2$	92.6	87.9	88.5	**88.6**	82.2	**82.2**	**81.1**	71.5	**69.9**	**68.2**	81.3
CompNet-VGG-p4 $\omega = 0$	94.0	89.2	**89.0**	88.4	**82.5**	81.6	80.7	**72.0**	69.8	66.8	**81.4**
CompNet-ResNext-RB3 $\omega = 0.2$	94.6	85.5	85.3	85.4	76.4	74.7	74.7	62.4	60.7	58.0	75.8

The best performing method for each experiment is highlighted in bold

vant clutter to an image that has no influence on the overall image labeling. In contrast, CutMix generates unrealistic artificial images of hybrid objects and labels.

From the results in Table 5 we observe that CutPaste, CutOut and CutMix all enhance the robustness to partial occlusion. However, the methods have large differences in terms of effectiveness. In the strongest level of occlusion (L3), CutMix only gives small improvements of 1.4% over a plain ResNext model, whereas CutOut is much more effective with a 5.4% improvement. Our proposed CutPaste augmentation is by far the most effective with 16.1% improvement in absolute performance in the strongest occlusion level. Hence we observe that CutOut and CutPaste are more effective in enhancing occlusion robustness because they simulate partial occlusion more realistically compared to CutMix. We also observe that the performance of all models improves with additional Auto-Augmentation. However, even with these massive data augmentation techniques all models are still slightly outperformed by a plain CompositionalNet that was trained from non-occluded objects without any augmentation. Note that in addition to the strong robustness to partial occlusion, CompositionalNets can localize occluders accurately and are much more interpretable compared to standard deep networks (Sect. 6.4).

6.3 Object Detection Under Occlusion

In this Section, we evaluate CompositionalNets at the task of object detection. In the experiments, we use a context-aware CompNet-VGG16-pool4 and a CompNet-ResNext-RB3 because of their higher resolution compared to the other types of backbones tested in image classification.

Occluded-P3D+ In Table 6, we report the results of Faster-RCNN and CompositionalNets on the Occluded-P3D+-Vehicles-Detection dataset. The models are trained on the images from the original PASCAL3D+ dataset with non-occluded objects. Qualitative detection results are illustrated in Fig. 9.

We observe that **Faster R-CNN fails to detect strongly occluded objects reliably**, while it performs well under low levels of occlusion. When trained with strong data augmentation in terms of partial occlusion using CutOut (DeVries and Taylor 2017), the detection performance increases under strong occlusion. However, the model still suffers from a 59.9% drop in performance on strong occlusion, compared to the non-occlusion setup. We suspect that the inaccurate prediction is because of two major factors. (1) The Region Proposal Network (RPN) in the Faster R-CNN is not able to predict accurate proposals of objects that are heavily occluded. (2) The VGG-16 classifier cannot successfully classify valid object regions under heavy occlusion.

We proceed to investigate the performance of the region proposals on occluded images. We conduct this experiment by replacing the VGG-16 classifier in the Faster R-CNN with a CompositionalNet classifier that is learned from the pool4 layer of VGG, which is expected to be more robust to occlusion. Based on the results in Table 6 (CompNet-VGG-p4-RPN), we observe two phenomena. (1) In high levels of occlusion, the performance is better than Faster R-CNN. Thus, the CompositionalNet generalizes to heavy occlusions better than the VGG-16 classifier. (2) In low levels of occlusion, the performance is worse than Faster R-CNN.

The Importance of Spatial Alignment in compositionalNets Recall that in the classification experiments, we observed that CompositionalNets are robust to occlusion

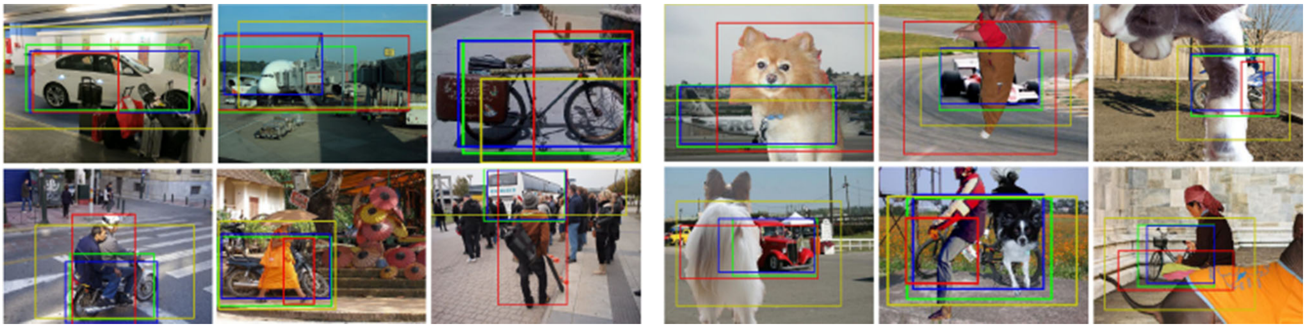


Fig. 9 Qualitative detection results on images with synthetic and real occlusion. Blue box: Ground truth. Red box: Bounding box by Faster R-CNN. Yellow Box: RPN+CompositionalNet. Green box: CompositionalNet + robust bounding-box voting

Table 7 Detection results on the Occluded-COCO-Vehicles-Detection dataset, measured by AP(%) @IoU0.5

MS-COCO Vehicles Detection under Occlusion					
Occ. Area	L0	L1	L2	L3	Avg
Faster R-CNN	77.2	59.0	40.8	24.6	50.4
Faster R-CNN with reg.	80.7	63.3	45.0	33.3	55.6
Faster R-CNN with occ.	82.5	66.0	50.7	45.6	61.2
CompNet-VGG-p4-RPN	60.0	49.7	45.4	38.6	48.4
CompNet-VGG-p4 $\omega = 0.5$	81.6	70.8	51.7	40.4	61.1
CompNet-VGG-p4 $\omega = 0.2$	86.8	**77.8**	65.4	**59.6**	**72.4**
CompNet-VGG-p4 $\omega = 0$	**89.4**	76.2	61.1	54.4	70.3
CompNet-RXT-RB3 $\omega = 0.2$	85.7	72.5	**65.9**	**59.6**	70.9

The best performing method for each experiment is highlighted in bold All models are trained on non-occluded images of the PASCAL3D+ dataset, except Faster R-CNN with reg. is trained with cutout and Faster R-CNN with occ. is trained with images that were artificially occluded using segmented objects. CompNet-VGG-p4-RPN has been evaluated with $\omega = 0$. Note how the CompositionalNets are significantly more robust to partial occlusion compared to Faster R-CNN

because they can roughly localize occluders and subsequently focus on the non-occluded object parts for classification. They can localize occluders because the different components in the mixture models explicitly represent the *spatial distribution* of features of an object in a certain pose. To be successful, the localization process requires the features of the object in the image to be roughly aligned with the mixture models. Therefore, CompositionalNets require bounding box proposals in which the center of the object is roughly aligned to the center of the bounding box, independent of whether the object is occluded or not. In this sense, CompositionalNets are rather high precision models which require spatial alignment between image and model. This is in contrast to standard deep networks, which are observed to not use spatial information extensively and behave more like bag-of-words type models (Brendel and Bethge 2019). The results in Sects. 6.2 and 6.4.1 show that the spatial distribution of object parts in an image is very important for computer vision models, because it enables the rough localization of occluders and thus a robust classification and detection under occlusion. One problem with the RPN proposals is that they mostly cover the visible part of the object only and often do not align the object center to the center of the bounding box.

Effect of Robust Bounding Box Voting Our approach of accurately estimating the corners of the bounding box substantially improves the performance of the CompositionalNet, in comparison with the RPN. This further validates our conclusion that the CompositionalNet classifier requires precise proposals to classify objects correctly with partial occlusions.

Effect of Context-Aware Representation We observe that models with a reduced influence of the context ($\omega = 0.2, \omega = 0.0$) outperform models with equal weight on context and object representation ($\omega = 0.5$). Hence, **context-aware CompositionalNets are more robust to partial occlusion than unaware models**. Furthermore, the performance of all models follows a similar trend over all three levels of foreground occlusions: the performance decreases as the level of background occlusion increases from BG-L1 to BG-L3. This further confirms our understanding of the effects of the context as a valuable source of information in object detection.

Occluded-MS-COCO As show in Table 7 and Fig. 9, the context-aware CompositionalNet with robust bounding box voting outperforms Faster R-CNN and CompNet+RPN significantly. Furthermore, the quantitative results clearly show the benefit of the context-awareness ($\omega = 0.2$) over unaware CompositionalNets ($\omega = 0.5$). While fully deactivating the context ($\omega = 0$) slightly decreases the performance, controlling the prior of the context model to $\omega = 0.2$ reaches a sweet spot where the context is helpful but does not have an overwhelming influence as the in the original CompositionalNet. Similar as observed in the classification experiments, CompNet-RXT-RB3 performs significantly worse compared to its pool4 variant for artificial occluders (Table 6), whereas it performs similarly under real occlusion (Table 7). We will discuss this phenomenon in more detail in Sect. 6.4.1.

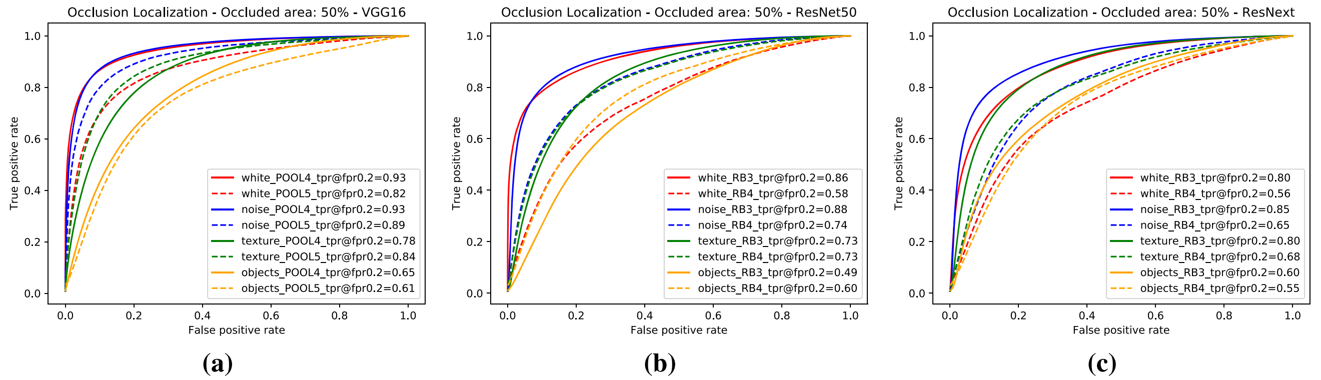


Fig. 10 ROC curves measuring occlusion localization scores in image classification with CompositionalNets learned from different DCNN backbones: **a** pool4 and pool5 layer of VGG16, and features after the residual block 3 (RB3) and residual block 4 (RB4) of **b** ResNet50 and **c** ResNext. The objects in the test data are on average 50% occluded. Note how all models can localize occluders well. The CompNets learned from VGG16 significantly outperform the backbones with residual connections

6.4 Model Interpretability

While it is important that computer vision systems can robustly generalize to out-of-distribution examples in terms of partial occlusion, in real-world applications it is equally important that their prediction result is human interpretable. In this section, we show that CompositionalNets are highly interpretable models. We demonstrate that they can localize occluders accurately in image classification and object detection (Sect. 6.4.1), while being trained with class-level supervision only. Furthermore, we show that the predictions of CompositionalNets can be understood in terms of detecting object parts (Sect. 6.4.2) and estimating the objects' viewpoint (Sect. 6.4.3).

6.4.1 Occluder Localization

A successful localization of occluders increases the robustness of a model to partial occlusion and also enables a human observer to better understand a models' underlying reasoning process. In the following, we test the ability of CompositionalNets at occluder localization. We compute the occlusion score as the log-likelihood ratio between the occluder model and the object model: $\log p(f_p|z^m_p = 1)/p(f_p|z^m_p = 0)$, where $m = \text{argmax}_m\, p(F|\theta^m_y)$ is the mixture component that explains the feature activations of the DCNN backbone the best.

Occluder Localization in Image Classification We study occluder localization quantitatively (Fig. 10) for correctly classified images from the Occluded-P3D+-Vehicles dataset using the ground truth segmentation masks of the occluders and the objects. We show qualitative results in Fig. 12b. The evaluation is done on the occlusion level $L2$, hence a pixel on the object will be occluded with 50% chance. We evaluate CompositionalNets learned from different DCNN backbones (VGG-16, ResNet50, ResNext) from the last and second-last layer before the classifier. All models were trained for classifying non-occluded vehicles of the PASCAL3D+ dataset (classification performance can be seen in Table 1).

In general, we can observe from the ROC curves in Fig. 10 that **CompositionalNets can localize occluders accurately**, although there are differences in terms of their performance. Furthermore, we observe that it is more difficult for the models to localize natural object occluders compared to box-shaped occluders, probably because of the fine-grained irregular shape of objects.

Insights into Robustness to Partial Occlusion in Neural Networks When comparing our experimental results at image classification and occluder localization in more detail, we make four interesting observations:

1. CompositionalNets learned from the lower layers of a backbone (pool4 and RB3) can consistently localize occluders more accurately when compared to models learned from higher layers of the same backbone.
2. The CompositionalNets learned from the VGG backbone (Fig. 10a) are more successful at localizing occluders, compared to models learned from layers with a similar resolution of ResNet50 and ResNext (Fig. 10b, c).
3. The performance of CompositionalNets with residual backbones is higher on images with real occlusion, compared to those with artificial object occluders. In contrast, we observe the opposite for CompositionalNets learned from the VGG backbone.
4. The ability to localize occluders more accurately does not directly translate into a superior performance at classifying partially occluded objects (Table 1). For example all three high-level models perform similarly at localizing "object" occluders, nevertheless CompNet-ResNext-RB4 performs more than 10% better at clas-

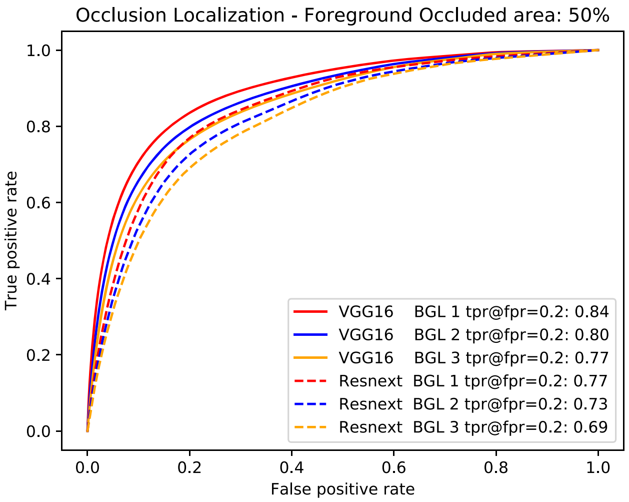


Fig. 11 ROC curves measuring occlusion localization scores in object detection with context-aware CompositionalNets learned from pool4 of VGG (solid lines) and RB3 of ResNext using $\omega = 0.2$. We the object is on average 50% occluded at each level of background occlusion (colored lines). Note that context-aware CompositionalNets can predict the occluded regions of the objects accurately at object detection

sifying images with these types of occluders at level $L3$ compared to the other models. This phenomenon can also be observed for the “white box“ occluders, which CompNet-ResNext cannot localize as accurately as CompNet-VGG16-pool5, while their performance is on par at level $L3$.

In general, **neural networks can exploit two complementary approaches to achieve robustness to occlusion**, depending on the backbone architecture: (A1) When the backbone is powerful enough, the features can become robust to occlusion, in a similar way as they are robust to illumination or viewpoint changes. Hence, by using a powerful feature extractor, residual models can achieve a very high classification performance even when using a rather simple classifier (global average pooling and one fully-connected layer. (A2) When the backbone cannot learn features that are robust to occlusion, then a more complex classifier is required, such as the multiple fully-connected layers in the VGG network.

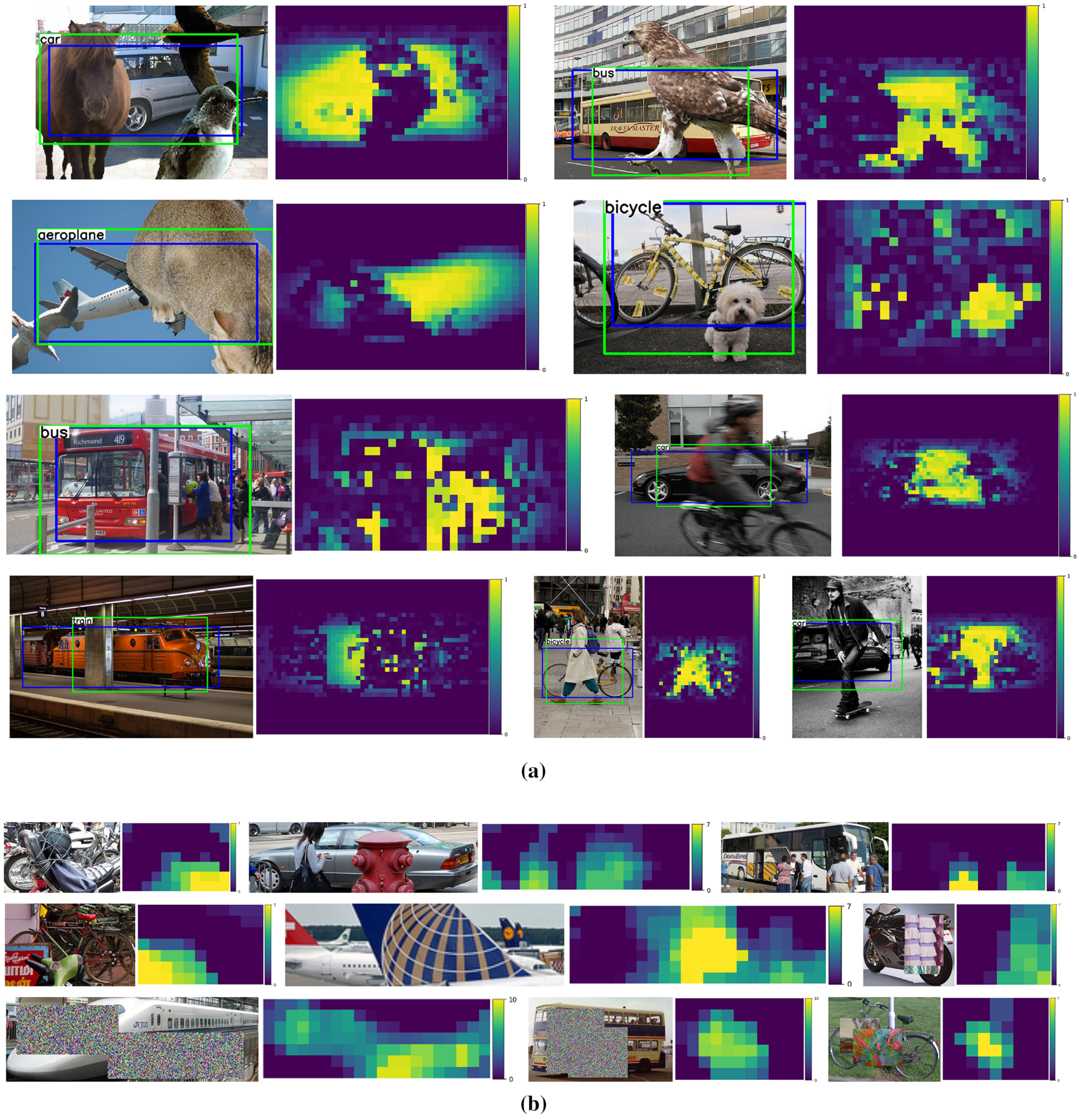
Based on this intuition, our experimental results for occluder localization and object recognition lead us to the following conjecture: Using powerful residual backbones, CompositionalNets achieve high classification performance, because of the highly discriminative features. However, they cannot localize the occluders as accurately, because the robustness to occlusion makes it difficult to distinguish between occluded and non-occluded features. In contrast, CompositionalNets based on the VGG backbone can localize the occluders well, because the features are not robust to occlusion. However, those CompositionalNets do not achieve the highest classification performance because the features are less discriminative compared to those of the residual backbones. Furthermore, the lower layers in neural networks typically exhibit less invariance. Therefore, CompositionalNets learned from those layer can consistently achieve better localization performance compared to those learned from higher layers of the same backbone. Finally, CompositionalNets with residual backbones rely more on invariance then on occluder localization. They perform better on real data, because the they rely on invariant features that were learned during ImageNet pre-training. This invariance does do not generalize well to the artificial occluders, therefore their performance is lower compared to the real occlusion scenario. In contrast, CompositionalNets learned from VGG rely less on invariance to occlusion and more on occluder localization. As the artificial occluders are easier to localize, their performance is higher on artificially generated occlusions compared to the real data.

In general, replacing the standard classifiers with compositional models increases the classification performance under occlusion for all models and also enables them to localize occluders. However, the ability to localize occluders and the final classification performance are at odds with each other, depending on the extent to which the features are invariant to occlusion.

Occluder Localization in Object Detection Figure 11 illustrates the ROC curve of a context-aware CompNet-VGG16-pool4 and CompNet-RXT-RB3 for successfully detected objects. We can observe that they can predict occluded regions accurately. Furthermore, the performance increases compared to the classification experiments as the context-awareness reduces false-positive detections in the background regions. In Fig. 12a, we show qualitative results of a context-aware CompNet-VGG16-pool4 at occluder localization in object detection. We illustrate results for artificially occluded objects and real occlusions from the MS-COCO dataset, in which the CompositionalNet could successfully locate the objects. Overall, the model can locate occluders with high accuracy, despite their large variability in terms of appearance and shape. Note how the shape of the occluders is outlined accurately, although the localization is done for each pixel in the feature map independently. In summary, we observe that the occluder localization results for object detection are consistent with those for image classification, in that they confirm the ability of CompositionalNets at localizing occluders accurately.

6.4.2 Interpretation of vMF Kernels

We further investigate the interpretability of our CompositionalNets using network dissection as proposed by Bau et al. (2017). In short, network dissection looks at the top activation of the hidden units and correlates them with a large range of human labeled visual concepts in the Broden dataset. Most of



(a)

(b)

Fig. 12 Qualitative results for occluder localization in: **a** Object detection with a context-aware CompNet-VGG16-pool4. **b** Image classification with a CompNet-VGG16-pool5. CompositionalNets can localize occluders accurately for different objects under real and artificial occlusions, despite the high variability of the occluders in terms of shape and appearance. Note that occluder localization is performed independently per pixel in the feature maps

the concepts are annotated as segmentation mask with input resolution and the activation maps are up-scaled to the same size to calculate the intersection over union (IoU) scores. By setting a threshold for the best matched score, Network Dissection studies the latent representations of various layers in a network. Since CompositionalNets are unifying part-based compositional models with deep neural networks, intuitively we expect to see the hidden units in CompositionalNets to be more correlated with visual concepts of parts, since the models evaluated are trained for classifying vehicles in PASCAL3D+, we expect to see more vehicle related parts emerge.

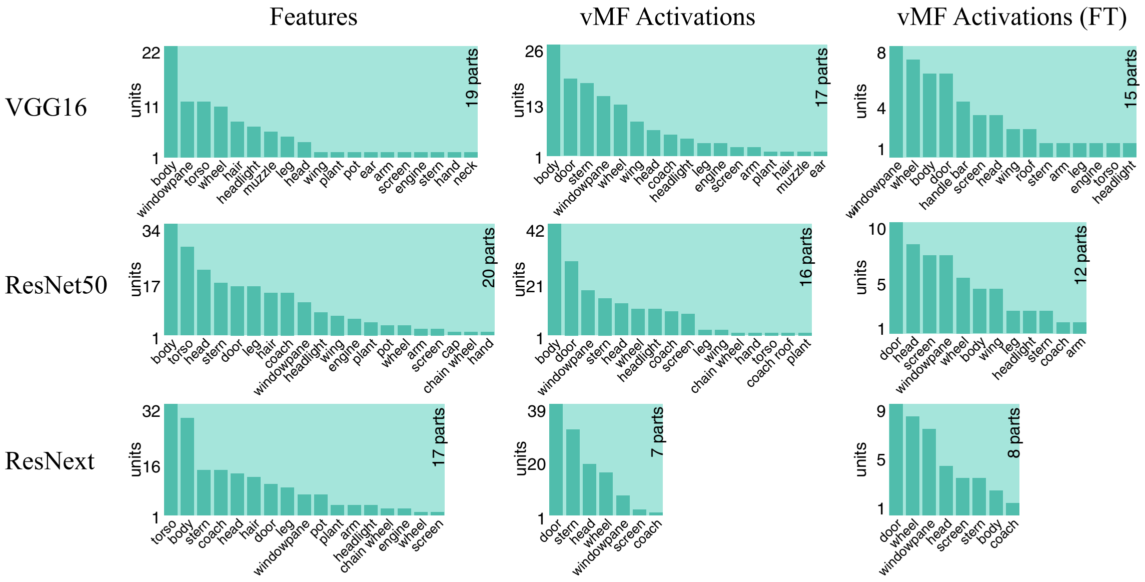


Fig. 13 Network dissection results for different backbone architectures and different activation layers on the Broden dataset "Part" category. The horizontal axis of the bar-plots represents the parts with above-threshold correlation score, and the vertical axis represents the number of hidden units that are correlated with the specific part. Note that the vMF kernel activations are more focused on the vehicle related parts, while after fine-tuning the number of correlated units for each parts is reduced, hence avoiding redundancies in the representation

For the experiment, we adopt the code from the authors of Bau et al. (2017) and use most default settings, except that we change the testing categories into "part" only. This means that during the hidden unit and visual concepts correlation test, only classes from the part category are involved. Note that these classes include both vehicle parts (e.g., windowpane, wheel, stern, etc.) and non-vehicle ones (e.g., hair, torso, muzzle, etc.).

As aforementioned, we expect to see the hidden units in CompositionalNets to be highly correlated with vehicle related parts. More specifically, we are interested in studying (1) what part concepts are correlated with the units before and after vMF kernels, and (2) how the end-to-end training affects these correlations. In Fig. 13, we examine three different backbone architectures, one in each row, and three different types of hidden units, one in each column. "Features" are the hidden units from the layer before the vMF kernel, which is pool5 for VGG16 and the residual block four (RB4) layer for ResNet50 and ResNext, all with weights initialized from ImageNet pretrained models. "vMF Activations" are the units right after the vMF kernel, where the kernel weights are initialized by clustering as described in Sect. 3. "vMF Activations (FT)" shows the same units after the end-to-end fine-tuning. In the barplots, the horizontal axis lists the parts with scores above the threshold, while the vertical axis shows the number of hidden units that are correlated with a specific part.

Comparing the "Features" versus the "vMF Activations", we observe that the vMF kernels indeed help the hidden units to be more concentrated on the vehicle related parts. The non-vehicle parts, such as "pot" and "cap", are removed while more units become correlated with vehicle ones, like "door" and "stern". However, this might also introduce a lot of redundancy in the representation and hence a waste of computational resources. When comparing "vMF Activations" with "vMF Activations (FT)", we find that not only some non-vehicle parts are further removed, the number of units correlated to each vehicle part is also reduced. This indicates the training may help the vMF kernels to recognize more diverse part representations and reduce the redundancy in the representation.

It is worth mentioning that the annotation of parts in the Broden dataset is coarse and not specific for different object classes, e.g., windowpane is shared for car, airplane, and house, etc.. Therefore, it may not match the internal representations learned by the CompositionalNets. Nevertheless, the results support our conjecture observed in Fig. 2 that the vMF kernels work as part detectors and CompositionalNets learn part representations without supervision.

6.4.3 Interpretation of Mixture Components

In an effort to better understand the inner workings of CompositionalNets we study what the individual mixture components have learned to represent during training. We study a CompNet-VGG16-pool5 with $M = 4$ mixture components, but the following analysis gives very similar results for other CompositionalNet architectures.

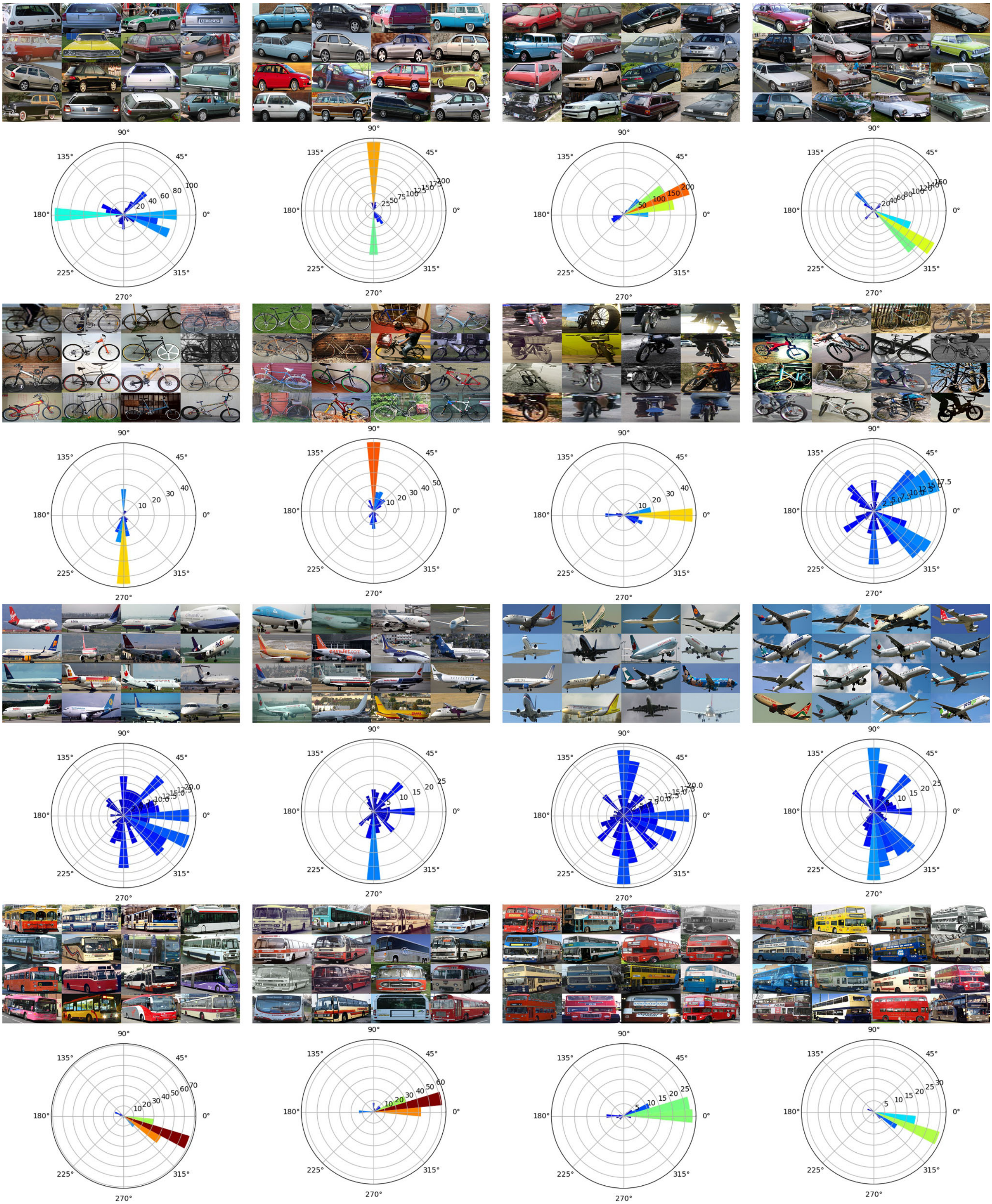


Fig. 14 Visualization of mixture components $p(F|\theta_y^m)$ for $M = 4$ components learned from the `pool5` layer of a VGG-16 networks and corresponding azimuth pose distribution (below each visualization). Note how images with different 3D viewpoint are approximately separated into different components

Each mixture component is by design of the training process specific to a particular object class. We have already shown that the vMF kernels are responsive to individual object parts (see Sect. 6.4.2 and Fig. 2). During training the mixture components are learned by clustering of the spatial activation patterns of vMF kernel activations. As the spatial distribution of parts of an object varies significantly with changes in the object pose, the mixture components become specific to certain viewpoints of the objects. We illustrate this property of CompositionalNets in Fig. 14 by showing the training images with highest likelihood in each mixture. Furthermore, we illustrate the histogram of poses for all training images in each mixture components as bar plots (using the pose annotations in the PASCAL3D+ dataset). As vehicles in natural images mostly vary in terms of their azimuth angle, we restrict the plots to this angle only. Each bar plot shows the distributions of azimuth angles in the range of $[0° - 360°]$ degrees. The length of the bars is normalized such that the longest bar indicates the most frequent azimuth angle within each mixture. The color of the bar is selected from the colormap "jet" and normalized such that the maximal value (dark red) is equivalent to more than 10% of the total number of training images for a particular class. We can observe from Fig. 14 that for the classes "car", "bicycle" and "bus" the mixture components are very specific to objects in a certain azimuth angle. Note that this happens despite a significant variability in the objects' appearance and backgrounds.

In contrast, for the class "airplane" the pose distribution is less viewpoint specific. We think that the reason is that airplanes naturally vary in terms of several pose angles, in contrast to vehicles on the street which vary mostly in terms of their azimuth angle w.r.t. the camera. As the number of mixture components is fixed a-priori it is difficult for the model to become specific to a certain pose during maximum likelihood learning from the data. In future work, it would therefore be useful to explore unsupervised strategies for determining the number of mixture components per class based on the training data.

7 Conclusion

In this work, we studied the problem of generalizing beyond the training data in terms of partial occlusion. We showed that current approaches to computer vision based on deep learning fail to generalize to out-of-distribution examples in terms of partial occlusion. In an effort to overcome this fundamental limitation we made several important contributions:

- We introduced **Compositional Convolutional Neural Networks**—a deep model that unifies compositional part-based representations and deep convolutional neural networks (DCNNs). In particular we replace the fully connected classification head of DCNNs with a differentiable generative compositional model.
- We demonstrated that CompositionalNets built from a variety of popular DCNN architectures (VGG16, ResNet50 and ResNext) have a significantly increased ability to **robustly classify out-of-distribution data** in terms of partial occlusion compared to their non-compositional counterparts.
- We found that a robust detection of partially occluded objects requires a separation of the representation of the objects' context from that of the object itself. We proposed **context-aware CompositionalNets** that are learned from image segmentations obtained using bounding box annotations and showed that context-awareness increases robustness to partial occlusion in object detection.
- We found that **CompositionalNets can exploit two complementary approaches to achieve robustness to partial occlusion**: learning features that are invariant to occlusion, or localizing and discarding occluders during classification. We showed that CompositionalNets that combine both approaches achieve the highest robustness to partial occlusion.

Furthermore, we showed that CompositionalNets learned from *class-label supervision only* develop a number of intriguing properties in terms of model interpretability:

- We showed that **CompositionalNets can localize occluders accurately** and that the DCNN backbone has a significant influence on this ability.
- Qualitative and quantitative results show that **CompositionalNets learn meaningful part representations**. This enables them to recognize objects based on the spatial configuration of a few visible parts.
- Qualitative and quantitative results show that **the mixture components in CompositionalNets are viewpoint specific**.
- In summary, the **predictions of CompositionalNets are highly interpretable** in terms of where the model thinks the object is occluded and where the model perceives the individual object parts as well as the objects viewpoint.

Our experimental results also hint at important future research directions. We observed that a good occluder localization is add odds with classification performance, because classification benefits from features that are invariant to occlusion, whereas occluder localization requires features to be sensitive to occlusion. We believe that it is important to resolve this trade-off with new types of models that achieve high classification performance while also being able to localize occluders accurately.

Acknowledgements Funding was provided by the Swiss National Science Foundation (P2BSP2.181713) and the Office of Naval Research (Grant No. N00014-18-1-2119 and N00014-20-1-2206).

References

Alain, G., & Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. arXiv preprint arXiv:1610.01644.

Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*.

Banerjee, A., Dhillon, I. S., Ghosh, J., & Sra, S. (2005). Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, *6*(Sep), 1345–1382.

Bau, D., Zhou, B., Khosla, A., Oliva, A., & Torralba, A. (2017). Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6541–6549).

Brendel, W., & Bethge, M. (2019). Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. arXiv preprint arXiv:1904.00760.

Cai, Z., & Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6154–6162).

Chen, Y., Zhu, L., Lin, C., Zhang, H., & Yuille, A. L. (2008). Rapid inference on a novel and/or graph for object detection, segmentation and parsing. In *Advances in neural information processing systems* (pp. 289–296).

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2018). Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501.

Dai, J., Hong, Y., Hu, W., Zhu, S. C., & Nian Wu, Y. (2014) Unsupervised learning of dictionaries of hierarchical compositional models. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2505–2512).

Dechter, R., & Mateescu, R. (2007). And/or search spaces for graphical models. *Artificial Intelligence*, *171*(2–3), 73–106.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). IEEE.

DeVries, T., & Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552.

Economist, T. (2017). Why uber's self-driving car killed a pedestrian.

Fawzi, A., & Frossard, P. (2016). Measuring the effect of nuisance variables on classifiers. Technical report.

Fidler, S., Boben, M., & Leonardis, A. (2014). Learning a hierarchical compositional shape vocabulary for multi-class object representation. arXiv preprint arXiv:1408.5516.

Fong, R., & Vedaldi, A. (2018). Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8730–8738).

George, D., Lehrach, W., Kansky, K., Lázaro-Gredilla, M., Laan, C., Marthi, B., et al. (2017). A generative vision model that trains with high data efficiency and breaks text-based captchas. *Science*, *358*(6368), eaag2612.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580–587).

Girshick, R., Iandola, F., Darrell, T., & Malik, J. (2015). Deformable part models are convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 437–446).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hu, Z., Ma, X., Liu, Z., Hovy, E., & Xing, E. (2016). Harnessing deep neural networks with logic rules. arXiv preprint arXiv:1603.06318.

Huber, P. J. (2011). *Robust statistics*. Berlin: Springer.

Jian Sun, Y. L., & Kang, S. B. (2018). Symmetric stereo matching for occlusion handling. In *IEEE conference on computer vision and pattern recognition*.

Jin, Y., & Geman, S. (2006). Context and hierarchy in a probabilistic image model. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)* (vol. 2, pp. 2145–2152). IEEE.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kortylewski, A. (2017). Model-based image analysis for forensic shoe print recognition. Ph.D. thesis, University_of_Basel.

Kortylewski, A., He, J., Liu, Q., & Yuille, A. (2020a). Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion. In *Proceedings of the IEEE conference on computer vision and pattern recognition.*

Kortylewski, A., Liu, Q., Wang, H., Zhang, Z., & Yuille, A. (2020b). Combining compositional models and deep networks for robust object classification under occlusion. In *The IEEE Winter Conference on Applications of Computer Vision*.

Kortylewski, A., & Vetter, T. (2016). Probabilistic compositional active basis models for robust pattern recognition. In *British machine vision conference*.

Kortylewski, A., Wieczorek, A., Wieser, M., Blumer, C., Parbhoo, S., Morel-Forster, A., Roth, V., & Vetter, T. (2019). Greedy structure learning of hierarchical compositional models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 11612–11621).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Lampert, C. H., Blaschko, M. B., & Hofmann, T. (2008). Beyond sliding windows: Object localization by efficient subwindow search. In *2008 IEEE conference on computer vision and pattern recognition* (pp. 1–8). IEEE.

Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 8595–8598). IEEE.

Li, A., & Yuan, Z. (2018). Symmnet: A symmetric convolutional neural network for occlusion detection. In *British machine vision conference*.

Li, X., Song, X., & Wu, T. (2019). Aognets: Compositional grammatical architectures for deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6220–6230).

Li, Y., Li, B., Tian, B., & Yao, Q. (2013). Vehicle detection based on the and-or graph for congested traffic conditions. *IEEE Transactions on Intelligent Transportation Systems*, *14*(2), 984–993.

Liao, R., Schwing, A., Zemel, R., & Urtasun, R. (2016). Learning deep parsimonious representations. In *Advances in neural information processing systems* (pp. 5076–5084).

Lin, L., Wang, X., Yang, W., & Lai, J. H. (2014). Discriminatively trained and-or graph models for object shape detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(5), 959–972.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In: *European conference on computer vision* (pp. 740–755). Springer.

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

Montavon, G., Samek, W., & Müller, K. R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, *73*, 1–15.

Narasimhan, N. D. R. M. V. S. G. (2019). Occlusion-net: 2d/3d occluded keypoint localization using graph networks. *IEEE conference on computer vision and pattern recognition*.

Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., & Clune, J. (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in neural information processing systems* (pp. 3387–3395).

Nilsson, N. J., et al. (1980). Principles of artificial intelligence.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).

Ross, A. S., Hughes, M. C., & Doshi-Velez, F. (2017) Right for the right reasons: Training differentiable models by constraining their explanations. arXiv preprint arXiv:1703.03717.

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems* (pp. 3856–3866).

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Song, X., Wu, T., Jia, Y., & Zhu, S. C. (2013). Discriminatively trained and-or tree models for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3278–3285).

Stone, A., Wang, H., Stark, M., Liu, Y., Scott Phoenix, D., & George, D. (2017). Teaching compositionality to cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5058–5067).

Tabernik, D., Kristan, M., Wyatt, J. L., & Leonardis, A. (2016). Towards deep compositional networks. In *2016 23rd international conference on pattern recognition (ICPR)* (pp. 3470–3475). IEEE.

Tang, W., Yu, P., & Wu, Y. (2018). Deeply learned compositional models for human pose estimation. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 190–206).

Tang, W., Yu, P., Zhou, J., & Wu, Y. (2017). Towards a unified compositional model for visual pattern modeling. In *Proceedings of the IEEE international conference on computer vision* (pp. 2784–2793).

Wang, A., Sun, Y., Kortylewski, A., & Yuille, A. (2020). Robust object detection under occlusion with context-aware compositionalnets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Wang, J., Xie, C., Zhang, Z., Zhu, J., Xie, L., & Yuille, A. (2017). Detecting semantic parts on partially occluded objects. arXiv preprint arXiv:1707.07819.

Wang, J., Zhang, Z., Xie, C., Premachandran, V., & Yuille, A. (2015) Unsupervised learning of object semantic parts from internal states of cnns by population encoding. arXiv preprint arXiv:1511.06855.

Wang, J., Zhang, Z., Xie, C., Zhou, Y., Premachandran, V., Zhu, J., Xie, L., & Yuille, A. (2017). Visual concepts and compositional voting. arXiv preprint arXiv:1711.04451.

Wu, T., Li, B., & Zhu, S. C. (2015). Learning and-or model to represent context and occlusion for car detection and viewpoint estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(9), 1829–1843.

Xia, F., Zhu, J., Wang, P., & Yuille, A. L. (2016). Pose-guided human parsing by an and/or graph using pose-context features. In *Thirtieth AAAI conference on artificial intelligence*.

Xiang, Y., Mottaghi, R., & Savarese, S. (2014). Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision* (pp. 75–820). IEEE.

Xiang, Y., & Savarese, S. (2013). Object detection by 3d aspectlets and occlusion reasoning.

Xiao, M., Kortylewski, A., Wu, R., Qiao, S., Shen, W., & Yuille, A. (2019). Tdapnet: Prototype network with recurrent top-down attention for robust object classification under partial occlusion. arXiv preprint arXiv:1909.03879.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1492–1500).

Yan, S., & Liu, Q. (2015). Inferring occluded features for fast object detection. In *Signal processing* (Vol 110).

Yuille, A. L., & Liu, C. (2018). Deep nets: What have they ever done for vision? arXiv preprint arXiv:1805.04025.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. arXiv preprint arXiv:1905.04899.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833). Springer.

Zhang, Q., Nian Wu, Y., & Zhu, S. C. (2018a). Interpretable convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8827–8836).

Zhang, Q. S., & Zhu, S. C. (2018). Visual interpretability for deep learning: A survey. *Frontiers of Information Technology and Electronic Engineering*, *19*(1), 27–39.

Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2018b). Occlusion-aware r-cnn: detecting pedestrians in a crowd, pp. 637–653.

Zhang, Z., Xie, C., Wang, J., Xie, L., & Yuille, A. L. (2018c). Deepvoting: A robust and explainable deep network for semantic part detection under partial occlusion. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1372–1380).

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2015). Object detectors emerge in deep scene cnns. In *ICLR*.

Zhu, H., Tang, P., Park, J., Park, S., & Yuille, A. (2019). Robustness of object recognition under extreme occlusion in humans and computational models. In *CogSci conference*.

Zhu, L., Chen, Y., Lu, Y., Lin, C., & Yuille, A. (2008a). Max margin and/or graph learning for parsing the human body. In *2008 IEEE conference on computer vision and pattern recognition* (pp. 1–8). IEEE.

Zhu, L. L., Lin, C., Huang, H., Chen, Y., & Yuille, A. (2008). Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *Computer vision–eccv 2008* (pp. 759–773). Springer.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.