ELSEVIER

Computer Vision and Image Understanding

journal homepage: www.elsevier.com

CompactNets: Compact Hierarchical Compositional Networks for Visual Recognition

Hans Lobel[a,**], René Vidal[b], Alvaro Soto[a]

[a]*Pontificia Universidad Católica de Chile, Vicuña Mackenna 4860, Santiago 7820436, Chile*
[b]*The Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21218, USA*

ABSTRACT

CNN-based models currently provide state-of-the-art performance in image categorization tasks. While these methods are powerful in terms of representational capacity, they are generally not conceived with explicit means to control complexity. This might lead to scenarios where resources are used in a non-optimal manner, increasing the number of unspecialized or repeated neurons, and overfitting to data. In this work we propose CompactNets, a new approach to visual recognition that learns a hierarchy of shared, discriminative, specialized, and compact representations. CompactNets naturally capture the notion of *compositional compactness*, a characterization of complexity in compositional models, consisting on using the smallest number of patterns to build a suitable visual representation. We employ a structural regularizer with group-sparse terms in the objective function, that induces on each layer, an efficient and effective use of elements from the layer below. In particular, this allows groups of top-level features to be specialized based on category information. We evaluate CompactNets on the ILSVRC12 dataset, obtaining compact representations and competitive performance, using an order of magnitude less parameters than common CNN-based approaches. We show that CompactNets are able to outperform other group-sparse-based approaches, in terms of performance and compactness. Finally, transfer-learning experiments on small-scale datasets demonstrate high generalization power, providing remarkable categorization performance with respect to alternative approaches.

1. Introduction

In the past few years, deep learning techniques have achieved state-of-the-art performance in most, if not all of visual recognition tasks, ranging from image classification (Krizhevsky et al., 2012), to semantic segmentation (Badrinarayanan et al., 2017), to visual question answering (Jahangiri et al., 2017). At the heart of these impressive results lies their deep hierarchical compositional structure, which allows them to learn powerful mid-level representations that serve as a shared and flexible set of building blocks for modeling a large variety of visual categories.

Several works have studied the hierarchical structure of deep neural networks (Giryes et al., 2015; Haeffele and Vidal, 2015; Montufar et al., 2014; Bruna and Mallat, 2013; Choromanska et al., 2015), showing among other results, that depth and compositionality play a critical role to learn suitable representations. However, as noted in (Geman et al., 2002; Zhu and Mumford, 2006; Jin and Geman, 2006), in order to take full advantage of compositionality, models should control their complexity, specifically the compositional construction of new elements in the hierarchy. We can characterize this complexity as the number of lower-level elements used to compose new ones. In the case of modeling object categories, controlling complexity

**Corresponding author:
e-mail: halobel@ing.puc.cl (Hans Lobel)

translates into using the smallest number of building blocks to compose an object, without sacrificing categorization performance. We call this property the **compositional compactness** of a hierarchical compositional model.

Generally, visual recognition models based on CNNs do not explicitly consider compositional compactness. Specifically, this type of models do not constrain the information coming from the channels of the preceding layers. Most previous research has focused on the raw capacity of the models, exploring their width and depth, giving place to architectures with vast representational power, but with a massive amount of parameters (Simonyan and Zisserman, 2015). As shown in (Zeiler, 2012), this unnecessarily increases the size of parameter space, by generating repeated, unspecialized, or even dead units, and also increasing the risk of overfitting (Cogswell et al., 2016). A few works include special constrains to control complexity, such as (Scardapane et al., 2017; Alvarez and Salzmann, 2016; Wen et al., 2016), however, their main goal is only limited to achieve network compression. In this work, we argue that a mechanism to induce compositional compactness into deep hierarchical models should lead to more specialized, less redundant, and therefore, more semantically meaningful representations. Furthermore, a more compact representation should also lead to a less complex model with improved categorization performance and lower risk of overfitting.

Paper Contributions: In this work we present **CompactNets**, a deep hierarchical compositional model that explicitly takes into account compositional compactness at all levels of the hierarchy, effectively specializing network units. Our main contribution is a scheme to structurally induce compositional compactness, by controlling the complexity and shape of the parameter space for each network unit independently. The approach is based on a hierarchical variation of group-sparsity-based regularization, that is able to regulate parameter learning at all abstraction levels of the model. This allows us to structurally induce compositional compactness, by controlling the complexity and shape of the parameter space for each network unit independently. As a result, the resulting models present compact and specialized representations at all levels of abstraction and with only a fraction of the number of active parameters of an unconstrained approach. Structural regularization is a topic of increasing relevance for the deep learning community, consequently we believe that our approach constitutes a relevant contribution towards more robust and efficient visual recognition systems.

In terms of experimental evaluation, the CompactNet model shows superior performance when compared with related approaches in the ILSVRC12 dataset. In particular, the proposed method is able to generate representations that are more compact and transferable than the ones obtained by competing approaches, which include group-sparse regularization-based methods and standard CNNs such as AlexNet (Krizhevsky et al., 2012). Regarding efficiency, CompactNets have an extremely small footprint, without sacrificing performance. By increasing the strength of the regularization, we are able to match the performance of AlexNet, using 25x less parameters. As the proposed methodology is orthogonal to methods that seek network size reduction, such as (Iandola et al., 2016), we argue that it can be jointly applied to obtain even further size reductions. Finally, based on qualitative analyses, we show that the learned units are able to capture human-interpretable visual patterns in a more specialized and compact manner than alternative approaches. Also, as expected, we observe that CompactNets capture the compositional compactness property with different strength depending on the depth, with a higher degree of specialization in deeper layers. We argue that this behavior relates to the natural growth in the number of visual patterns needed to model more complex visual abstraction levels.

Paper Outline: The remainder of this article is organized as follows: Section 2 reviews relevant previous works. Section 3 presents the basic elements of the proposed model. Section 4 describes the proposed learning algorithm which incorporates compositional compactness. Section 5 presents the qualitative and quantitative evaluation performed on different datasets. Finally, Section 6 presents the conclusions and open research questions.

2. Related Work

The importance of compact compositional models has been noticed in the early visual recognition literature (Potter, 1999; Geman et al., 2002). A fruitful line of work is that of stochastic visual grammars (Zhu and Mumford, 2006; Tu et al., 2013). In these works, grammars able to to learn compact hierarchical models of visual elements, generally using a relatively small vocabulary and a few compositional rules, are used to represent an exponentially large number of different configurations. These techniques have been used in various visual recognition tasks to model elements such as faces (Suo et al., 2010), objects (Girshick et al., 2011), and actions (Albanese et al., 2010). A different approach is taken in (Lobel et al., 2015), where a shallow dictionary learning-based recognition model uses a group-sparse regularization term to effectively induce a behavior akin to compositional compactness, but only at the top-level classification.

Compact models have also been explored in the context of CNN-based visual recognition models. Most related approaches focus on reducing the size of the parameter space, by avoiding redundancies given by highly correlated coefficients (Denil et al., 2013; Ioannou et al., 2016; Tai et al., 2016). Conversely, in (Wen et al., 2017; Alvarez and Salzmann, 2017), explicit regularization is used in order to generate highly correlated and redundant filters in each layer, that are then efficiently compressed by means of a post-training low-rank approximation. Another popular approach is the use of parameter pruning (Han et al., 2015; Li et al., 2017; Zhu and Gupta, 2018). In these works, a previously trained CNN is pruned, by removing weights that show low contribution to the overall prediction. Results show that pruned networks present almost no performance loss when compared to the dense versions. Recent works based on explicit parameter regularization show the relevance of introducing compact CNN-based models (Zhou et al., 2016; Alvarez and Salzmann, 2016; Wen et al., 2016; Scardapane et al., 2017). By using group-sparsity inducing terms in the loss function, these works are able to dynamically limit the number of units or channels used by each of the network's layers, leading to a dramatic reduction on the effective number of used parameters, while keeping performance competitive. Although these works also use group-sparse regularization, the scheme we propose gives rise to different regularization terms. Specifically, our focus is related to enhancing the learning capabilities of deep hierarchical models by independently exploiting the *compositional compactness* present in each unit of a hierarchical model, while theirs is mainly on network compression.

Another line of research related to compact models has focused on architectural improvements. An example of this is (Springenberg et al., 2015), where all pooling and fully-connected layers are removed. This modification results in a simpler network with competitive performance, showing that some of the elements of traditional CNNs are not mandatory to obtain reasonable performance. In (Changpinyo et al., 2017), a pre-defined fraction of connections between input and output channels is deactivated in convolutional layers, such that the number of parameters can be drastically reduced. Although similar in spirit, the key difference with respect to our approach is the fact that connections are discarded before training, limiting the ability to adapt to data complexity. Finally, the SqueezeNet architecture (Iandola et al., 2016) takes the notion of dimensionality reduction layers of (Szegedy et al., 2015) a step further, reducing the number of parameters 50x when compared to the AlexNet architecture, while maintaining similar performance.

3. Model Description

Here we describe the proposed CompactNet architecture. The model is divided into two types of layer, namely feature extraction and classification. An L layer hierarchy is composed of $L-1$ feature extraction layers and a final classification layer. Figure 1 shows a high-level description of these components.

3.1. Feature extraction layer

Each feature extraction layer consists of four modules. Input feature maps (or image pixels in the first extraction layer) are convolved with K convolutional filters of size 3×3 (7×7 in the

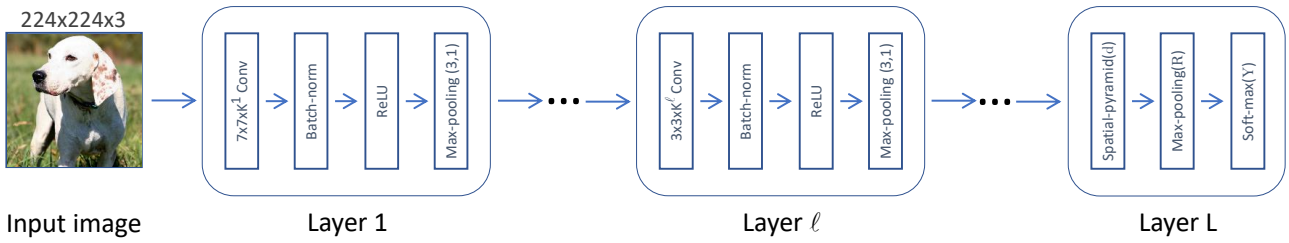

Fig. 1: A high-level description of the proposed hierarchical structure. We differentiate between feature extraction layers and a classification layer. Each of them is formed by several components.

first layer). After this, a batch-normalization module is applied, followed by a ReLU nonlinearity. Finally, a max-pooling operator with receptive field of 3 pixels and a stride of 1 pixel is applied, in order to keep only the most salient features. To keep computational load bounded, every time the number of convolutional filters is doubled, the stride in the max-pooling operation is also doubled. Finally, on each feature extraction layer, except from the first and the ones where the number of filters is doubled, the dimensionality of input and output feature maps is the same.

The set of parameters of the K^l convolution filters in layer l, are contained in the visual dictionary Θ^l, modeled as a tensor with dimensions $K^l \times K^{l-1} \times S$, where S is the *size* of the filters ($7 \times 7 = 49$ for the first module, $3 \times 3 = 9$ for the rest).

3.2. Classification layer

The classification layer consists of three modules. It initially processes input feature maps by means of a spatial pyramidal decomposition (Lazebnik et al., 2006) of depth d, which generates $R = \sum_1^d 2^{2(d-1)}$ spatial regions, each containing K^{L-1} feature maps. After this, each region is processed by a max-pooling operator, producing a feature map of size $R \times K^{L-1}$. This feature map is finally processed by a linear soft-max module, which outputs the probabilities for each of the Y possible visual categories.

Similar to the feature extraction layers, the parameters of the soft-max module are contained in tensor W, with dimensions $Y \times K^{l-1} \times R$.

4. Learning Compact Compositional Models

In this section we present the proposed regularized learning scheme to estimate the CompactNet model parameters. We take ideas from the shallow model described in (Lobel et al., 2015) and generalize them to deep visual hierarchies.

4.1. Learning Problem

Given a set of training examples $\{x_i, y_i\}_{i=1}^N$, where x_i refers to the i-th image and y_i to its corresponding visual category, we propose to obtain estimates for the parameters Θ^l and W by solving a regularized learning problem:

$$\underset{W,\Theta^l}{\operatorname{argmin}} \; C_W \Omega(W) + C_\Theta \sum_{l=1}^{L-1} \Gamma_l(\Theta^l) + \frac{1}{N} \sum_{i=1}^{N} L(x_i, y_i; W, \Theta), \quad (1)$$

where L is the standard cross-entropy loss (we also evaluated a hinge-loss, showing no considerable differences). In this formulation, Ω and Γ_l are convex regularizers and $C_W, C_\Theta > 0$ are regularization constants. We describe next the details behind the terms of the learning problem.

4.2. Compositional Compactness

To fully specify the formulation in Eq. (1), we need to select appropriate regularizer. As previously described, the main objective is to capture the property of compositional compactness by structurally controlling the complexity of the model. Our main goal is to increase specialization and decrease the redundancy of convolutional filters, leading to a more efficient and meaningful representation.

As shown in (Lobel et al., 2015), on a shallow model a powerful and efficient way to achieve the previous goal is to use

a group-sparse regularizer on the top-level classifiers weights. Such a regularizer enforces each category classifier to use only a selected subset of feature maps (convolutional layers in the previous layer), while maintaining recognition performance. The effects of this regularization strategy are twofold. First, it generates a behavior where subsets of the learned filters are shared by only a small number of categories, making them more specialized. Second, if a filter is not part of any subset, it is discarded from the model, as it is not used by any category. We use a similar approach here for the top-level classifier, generating pruning, specialization, and category-level sharing in filters from the layer immediately below $(L-1)$. More formally, we define the regularizer $\Omega(W)$ as follows:

$$\Omega(W) = (1-\alpha)\frac{1}{2}\|W\|_F^2 + \alpha \sum_{y=1}^{Y}\sum_{k=1}^{K^{L-1}} \|W_{y,k,\cdot}\|. \qquad (2)$$

First term in Eq. (2), corresponds to a common ℓ_2-norm regularizer used in SVMs and CNNs (weight decay), which seeks to reduce the risk of overfitting. Second term, $\sum_{y=1}^{Y}\sum_{k=1}^{K^{L-1}}\|W_{y,k,\cdot}\|$, uses an $\ell_{1,1,2}$-norm, *i.e.* a sum of of euclidean norms of certain groups in the last dimension of the tensor, to independently penalize the number of feature maps used by each classifier. Both terms are additively combined using the α constant, which controls their relative importance. To understand how this regularizer accomplishes this goal, consider that if filter k contributes to the classification of instances from class y, then we must have $W_{y,k,r} \neq 0$ for some $r \in [1, R]$, hence $\|W_{y,k,\cdot}\| = \sqrt{\sum_r W_{y,k,r}^2} \neq 0$. Thus, the total number of filters (feature maps) used by all classes is given by:

$$\sum_{y=1}^{Y}\sum_{k=1}^{K^{L-1}} \delta\Big(\sqrt{\sum_{r=1}^{R} W_{y,k,r}^2} > 0\Big), \qquad (3)$$

where δ denotes the Kronecker delta function. As common in sparse representation-based methods, we replace the counting norm by the ℓ_1-norm and approximate the total number of filters used by the classifier as:

$$\sum_{y=1}^{Y}\sum_{k=1}^{K^{L-1}} C(y,k) = \sum_{y=1}^{Y}\sum_{k=1}^{K^{L-1}} \|W_{y,k,\cdot}\|, \qquad (4)$$

where $C(y,k)$ denotes the contribution of filter k to the classification of instances from class y.

The expression in Eq. (4) exactly coincides with the second term of Eq. (2), which can be interpreted as a convex relaxation for the total number of filters used by all classes (Eq. (3)). In terms of the effects of the group-sparse term of the regularizer, as classifiers use as few feature maps as possible, higher-level filters are specialized to categories where they contribute. Moreover, this also means that if the contribution of a filter to all classes is equal to zero, then its weights will be zero, effectively discarding it from the model and reducing the effective size of the visual dictionary.

As noted in (Zou and Hastie, 2005; Wang et al., 2006), by mixing an ℓ_2-norm and a sparsity-inducing norm, we are keeping the best of both regularizers. In our case, this means obtaining classifiers with high generalization power (ℓ_2-norm), with compact representations ($\ell_{1,1,2}$-norm), exactly what compositional compactness seeks. An illustration of the operation of regularizer Ω on tensor W can be seen in Figure 2.

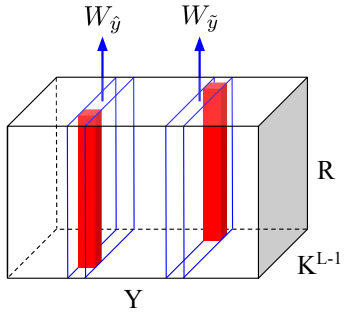

Fig. 2: An illustration of the operation of regularizer $\Omega(W)$. Tensor W is represented as a cuboid of dimensions $Y \times K^{L-1} \times R$, while the planes $W_{\hat{y}}$ and $W_{\tilde{y}}$ represent the classifiers of categories $\hat{y}$ and $\tilde{y}$, respectively. Red bars, of dimensions $1 \times 1 \times R$, represent groups of zeroed parameters, consisting of all the coefficients from a given classifier that act on the feature map generated by a given filter k in layer $L-1$. Notice that in this case, $W_{\hat{y}}$ and $W_{\tilde{y}}$ discard different feature maps, meaning the associated units are being specialized on different categories.

Based on the regularization strategy for classifiers (Eq. (2)) and the representation described in Section 3, we propose a strategy to regularize the remaining layers of the hierarchical model by inducing a compositional compactness behavior, where filters from a given layer minimize the number of active features maps from the previous layer. Specifically, we define regularizers Γ_l for $l \in [1, L-1]$. This strategy allows features

in all layers to be specialized to visual patterns, that share representations at certain levels of abstraction.

We define the *contribution* of a filter t in layer $l-1$ to a filter k in layer l as:

$$C^l(k,t) = \sqrt{\sum_{s=1}^{S} {\Theta^l_{k,t,s}}^2} = \|\Theta^l_{k,t,\cdot}\|. \quad (5)$$

Using $C^l(k,t)$ and similarly to Eq. (2), we formally define regularizer $\Gamma_l(\Theta^l)$ as follows:

$$\Gamma_l(\Theta^l) = (1-\beta^l)\frac{1}{2}\|\Theta^l\|_F^2 + \beta^l \sum_{k=1}^{K^l}\sum_{t=1}^{K^{l-1}} \|\Theta^l_{k,t,\cdot}\|. \quad (6)$$

As in Eq. (2), the two terms of the regularizer are mixed by a constant, but now this constant depends on the depth of the dictionary, providing a method to control the level of sparsity of each layer. This is a non-trivial aspect of our model, as it allows us to increase or decrease the degree of specialization of the dictionaries. In particular, as one would expect more specific visual patterns, and a greater number of them, when abstraction level increases, this mechanism of control allows CompactNets to capture this situation by means of increasing specialization with depth.

The first term (ℓ_2-norm) of Eq. (6) remains the same, while the second one ($\ell_{1,1,2}$-norm) focuses on groups of size S within dictionary filters. Each of these groups is formed by the coefficients that act on features generated using a certain filter t from the layer below. Thus, as the $\ell_{1,1,2}$-norm seeks to set to zero all the elements in a group, filters use effectively as few feature maps from the previous layer as possible.

4.3. Learning Scheme

To solve the learning problem in Eq. (1), we use stochastic gradient descent with momentum. It is important to mention that the group-sparse regularizers used to induce compositional compactness are non-differentiable at some points, thus forcing the use of sub-gradients (Shor, 1985). As sparse solutions generally lie on points of non-differentiability and sub-gradients rarely lead accurately to such points (Shalev-Shwartz and Tewari, 2011), we employ a thresholding strategy after training in order to generate sparse solutions (see details in Section 5).

5. Experimental evaluation

This section presents a comprehensive evaluation of the CompactNet model. The main focus is to analyze the main effects of our group-sparse regularization strategy in terms of compositional compactness and recognition accuracy. In particular, we analyze the impact of our strategy with respect to the depth of each layer in the network. We base our analysis mainly on the ILSVRC12 dataset, that we used to compare our approach with respect to related approaches.

5.1. Implementation details

Evaluated Architectures: To highlight the advantages of the CompactNet architecture with respect to models based on alternative regularization schemes that have appeared in the literature (Springenberg et al., 2015; Zhou et al., 2016; Alvarez and Salzmann, 2016; Wen et al., 2016; Scardapane et al., 2017), we implement two competing models that we refer as SimpleNet and CompressedNet. SimpleNet consists of only the application of an ℓ_2-norm term to the parameters, effectively eliminating any regularizer-induced sparsity, while keeping the rest of the architecture the same. This model is similar to (Springenberg et al., 2015). More formally, for the SimpleNet scheme we set $\alpha = 0$ in Eq. (2) and $\beta_l = 0$ in Eq. (6).

CompressedNet focuses on discarding as many units in every layer as possible, akin to the models described in (Zhou et al., 2016; Alvarez and Salzmann, 2016; Wen et al., 2016; Scardapane et al., 2017), representing a relevant comparison point, as CompactNets seek to discard groups of connections between layers, but not whole units. More formally, the CompressedNet architecture sets $\alpha = 0$ in Eq. (2) and swaps the Γ regularizer for a new Ψ regularizer, defined as follows:

$$\Psi_l(\Theta^l) = (1-\beta)\frac{1}{2}\|\Theta^l\|_F^2 + \beta \sum_{k=1}^{K^l} \|\Theta^l_{k,\cdot,\cdot}\|. \quad (7)$$

The regularizer defined in Eq. (7) tries to shrink the ℓ_2-norm of the weights of each filter as far as possible, effectively leaving filters that are not needed with a zero ℓ_2-norm. A depiction of the operation of regularizer Ψ on tensors Θ^l can be seen in Figure 3.

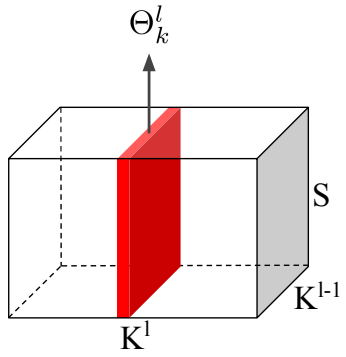

Fig. 3: An illustration of the operation of regularizer $\Psi_l(\Theta^l)$. Tensor Θ^l is represented as a cuboid of dimensions $K^l \times K^{l-1} \times S$, where S is the dimensionality of filters in layer $l-1$. A red plane represents groups of zeroed parameters, consisting of all the coefficients from a given filter. Notice that in this case, only filter Θ_k^l is discarded.

Weight initialization: We use the procedure described in (He et al., 2015b). This consists of initializing the parameters of each layer l using random coefficients with a normal distribution with zero mean and standard deviation set to $\sqrt{\frac{2}{S\,K^{l-1}}}$, where S is the dimensionality of filters in layer $l-1$, and K^{l-1} is the number of filters in layer $l-1$. The classification layer is initialized using a normal distribution with zero mean and standard deviation set to $\sqrt{\frac{2}{21K^L}}$.

Dataset Details: We use the ILSVRC12 benchmark dataset (Russakovsky et al., 2015). This large-scale set contains 1.35M color images (1.2M training, 50K validation, 100K testing), evenly divided into 1000 different object categories. We use the standard evaluation procedure, with 1.2K images per class for training and 50 for validation.

Optimization: As states in Section 4, we solve the optimization problem in Eq. 1 using SGD with momentum. We use batches of 256 elements, a momentum factor of 0.9, and initial learning rate of 0.1, including a scheduled reduction with a factor of 0.1 every 30 epochs. We train the models for 90 epochs.

Data Augmentation: We follow the procedure in (Krizhevsky et al., 2012). First, we downsize all images to 256×256 pixels. During training, we randomly extract one 224×224 crop from each image, a different crop each time the image is processed, and use it to compute the loss and the derivative. When testing, we extract five 224×224 patches, four corner patches and the center patch, as well as their horizontal reflections, obtaining 10 different crops. Test-time categorization is performed by averaging the predictions made for each crop and reporting the class with the highest average score.

Source code: We will make available a PyTorch implementation of the CompactNet in `https://github.com/halobel/CompactNet/`. The implementation will also include implementations of SimpleNet and CompressedNet architectures as well.

5.2. *Effect of Regularization on the Parameters*

In this section we analyze how the group-sparse regularization schemes, CompactNet and CompressedNet, affect the parameter space. We compare it to a case where no group-sparse regularization is applied. More specifically, our intention is twofold, namely i) to assess which and how many of the connections between layers are discarded (zeroed), and ii) how this varies depending on the layer and overall depth of the model.

All instantiations in this section share the following set of common characteristics. In terms of dictionary size, we keep the number of filters fixed at 256 for each layer. This minimizes the impact of other structural elements, other than the regularization scheme being used. Regularization constants C_W and C_Θ are both set to $1e^{-4}$ in Eq. (1), while for the CompactNet model, α and β^l are set to 0.01 in Eqs. (2) and (6), respectively (β^1 is set to 0, as the first dictionary acts on pixels). For the CompressedNet model, $\beta = 0.1$. For a more in-depth analysis of regularization constants values, see Section 5.3.2.

5.2.1. *Sparsity of the parameters*

We assess the sparsity of the parameters, *i.e.*, the ratio between the zeroed parameters and the total number of parameters, and relate it to the categorization performance of CompactNet, CompressedNet, and SimpleNet models. We use different instantiations of the models with increasing depth. In all the tests, we compute sparsity by zeroing all weights with an absolute value less than 1×10^{-5}. This value was experimentally selected as the largest one that does not decrease performance. We use top-1 error-rate as metric and report performance on

the validation set, as test set labels are not publicly available. Figure 4 and Table 1 present the results of the experiments.

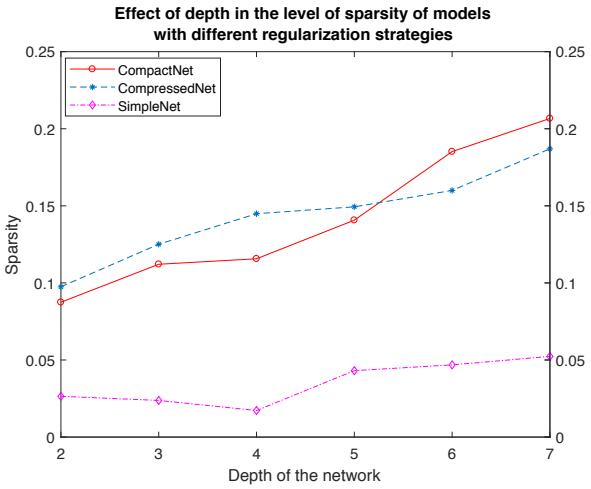

Fig. 4: Sparsity of different instantiations of all three models, with depths ranging from 2 to 7 layers. As depth increases, it is possible to observe a significant increase in the sparsity of the parameters of CompactNet and CompressedNet models. In contrast, SimpleNet models show only a slight increase. This indicates that the total number of parameters used on deeper structures can be drastically reduced.

Figure 4 shows that as the number of layers increases, the sparsity of the parameters also increases in the group-sparse regularized models (CompactNet and CompressedNet), *i.e.* the number of inactive (zeroed) connections between layers increases. This is to be expected, given that shallower models lack the complexity to correctly classify most examples, thus discarding weights is counterproductive. However, deeper models with a larger parameter space are more susceptible to discard superfluous parameters. This indicates that on deep models, with high expressive power, adaptive complexity control strategies based on group-sparse regularization, provide a suitable mechanism to structurally adjust the number of active parameters, validating our intuition regarding filter compactness.

Regarding categorization performance, Table 1 shows top-1 error-rate for the same models presented in Figure 4. As expected, all methods show a steep descent in error-rate as depth increases. However, the remarkable element about this behavior is that the three methods present similar performance, with a slight advantage to CompactNet as depth increases, even

Performance as a function depth						
Method	**2**	**3**	**4**	**5**	**6**	**7**
SimpleNet	74.6	64.2	55.7	48.8	43.8	41.0
CompressedNet	76.0	65.6	56.3	47.2	43.0	40.3
CompactNet	75.5	66.1	57.2	47.6	42.1	40.2

Table 1: Top-1 error error rate for different instantiations of all three models, with depths ranging from 2 to 7 layers. Performance for all methods is similar and increases with depth, showing subtle signs of advantage to CompactNets in the deepest instantiations.

though the CompactNet and CompressedNet models need approximately 20% less parameters than SimpleNet. This means that the effects of capturing compositional compactness not only helps in reducing the model footprint, but also in removing elements that end up degrading categorization performance.

In summary, these results show that a commonly dense model can be trained to be more compact, with the consequent advantages in terms of performance and storage requirements.

5.2.2. Structure of the discarded parameters of CompactNets

Another aspect worth exploring of the CompactNet architecture is the structure of the discarded parameters, as it can provide valuable information about the characteristics of the learned representations, in terms of filter compactness, complexity, and specialization of the different layers. To achieve this, we define three metrics to quantify the sparsity of the parameters at each layer of a model.

- Dictionary sparsity: Defined for a layer l as the ratio between the number of filters with all weights set to zero, and the total number of filters in dictionary Θ^l. In other words, the fraction of filters that are discarded in a layer.

- Average compactness: Defined for a layer l as the ratio between the average number of filters from layer $l-1$ with contribution equal to zero, to filters/classifiers in layer l, and the total number of filters in Θ^{l-1}, the dictionary of layer $l-1$. In other words, the fraction of channels that filters in layer l discard on average. See Eqs. (4) and (5) for more details regarding filter contribution.

- Layer sparsity: Defined for a layer l as the ratio between the number of zeroed parameters and the total number of parameters in the layer. For any given layer, the sum of dictionary sparsity and average compactness is equal or less than the layer sparsity, as there are coefficients that are zeroed in an unstructured manner in filters/classifiers.

Intuitively, dictionary sparsity captures the complexity of a layer in terms of the number of different visual patterns that it needs to model. Average compactness captures a notion of the complexity and specialization of filters/classifiers in a layer, as it quantifies the average number of filters that are not used by a given classifier at the next hierarchy level (group-based sparsity). The layer sparsity is included in order to explicitly compare the contribution of dictionary sparsity and average compactness to the overall sparsity of the layer. Figure 5 shows the three metrics for each of the layers of a CompactNet-7 model (7 layers) trained on the ILSVRC12 dataset. Notice that the structure of the network follows the one described in 5.2 therefore each layer has 256 filters available.

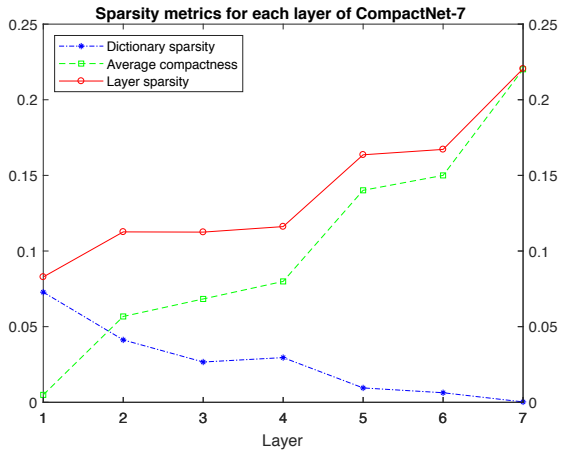

Fig. 5: Dictionary sparsity, average compactness, and layer sparsity metrics for each layer of a CompactNet-7 model. Most of the structurally discarded parameters come from the classification layer, and from the higher-level feature learning layers, as illustrated by the narrow gap between average compactness and layer sparsity. This can be intuitively explained by the fact that the captured visual patterns present a high level of abstraction, thus having higher specialization to certain categories.

As Figure 5 illustrates, the contribution of the dictionary sparsity and average compactness to the sparsity level of a layer strongly differ in their behaviors. In lower layers, the level of dictionary sparsity indicates that a significant number of filters is dropped from the model, thus reducing the complexity of the layers. This means that lower layers need less than the available 256 filters to provide a suitable representation of the training data. An observation that agrees with the typical architecture used in most current CNN models.

In terms of the level of average compactness observed in lower layers, the low value indicates that the model learns filters with low specialization. This means that lower layers learns highly general filters that are useful to most filters in the next layer. This observation also agrees with the typical features learned by CNN models (Zeiler and Fergus, 2014).

As one goes up in the hierarchy, dictionary sparsity falls to a negligible level, indicating that higher layers need to capture a larger number of diverse visual patterns than lower ones, and that most likely need dictionaries larger than 256 filters. This comes tied with the fact that on higher layers, the average compactness is notably higher than in lower layers, a sign of a higher degree of filter specialization. Furthermore, average compactness almost reaches the layer sparsity, showing that most of the structurally discarded parameters on these layers come from the specialization of filters/classifiers. From the point of view of a compositional model, this makes sense, as the more specific a pattern is, it will be more compact and will need fewer patterns from the layer below to be composed.

Finally, based on the analysis of Figure 5, we can conclude that layers in a CompactNet model smoothly transition from low abstraction (filters with low specialization) to high abstraction (filters with high specialization). Although CNNs are also capable of capturing this, CompactNet additionally captures the natural increase in dictionary size associated with increasing specialization, reinforcing its effects. It is worth noting that CompactNets are able to capture this important insight without any prior information. This highlights the good qualities of the proposed model, as it is able to capture and exploit intuitive concepts about visual pattern complexity and compositionality, in order to build a highly compact and specialized composi-

tional hierarchical model.

5.3. Categorization performance analysis

This section analyses whether the sparsity results and insights from Section 5.2.2 have a measurable impact in terms of categorization performance.

5.3.1. Network structure, regularization and training

Previous section highlights two key properties about the structural compositionality of a deep model, namely i) higher layers capture a larger number of patterns than lower ones, and ii) filters in higher layers show higher specialization than the ones in lower layers. In this section we adapt the network architecture of CompactNet to take full advantage of these properties. Specifically, we introduce the following changes with respect to the procedure presented in Section 5.2:

- Dictionary size is increased with depth in order to capture the need for more visual pattern in deeper layers. We use three dictionary sizes: 128, 256 and 512 filters.
- As depth increases, we linearly augment the value of the respective regularization constant β^l. This helps to enforce that higher layers should increase pattern specialization. We start with $\beta^1 = 0$ in layer 1, up to a constant call $\rho = \alpha = \beta^{L-1}$, for the last feature extraction and classification layers.
- Models are trained for 120 epochs using the same conditions as in the previous section.

In order to keep comparison fair, we also introduce similar modifications to SimpleNet and CompressedNet models, where applicable.

5.3.2. Sensitivity analysis

During preliminary evaluations, we identified two factors as critical to achieving good performance with CompactNets: depth, and regularization strength. To assess the interplay between these two factors, we perform an extensive sensitivity analysis, measuring how performance behaves as these factors change. We evaluated several networks on the ILSVRC12 validation dataset, varying depth between 2 and 18 layers, and $\rho \in [0, 1]$. Notice that indirectly, this analysis also covers the sensitivity of the SimpleNet model when $\rho = 0$. Results are presented in Figure 6.

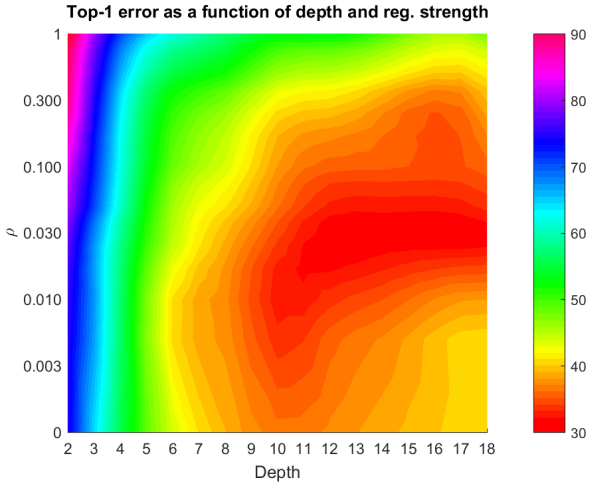

Fig. 6: Performance sensitivity as a function of depth and regularization strength. The best performance is achieved at depth 12 with $\rho = 0.03$. Notice that the lowest error-rate of the SimpleNet model ($\rho = 0$, depth 10) is worst than CompactNet's, and is achieved with a shallower network, probably due to overfitting.

Results show that performance depends more on depth than ρ, up to networks with eight layers, as performance remains approximately constant along the vertical axis. This situation is similar to the one presented in Figure 4, where shallower models are not powerful enough to capture relevant features for classification. From networks with right or more layers, regularization becomes a more significant parameter, as the number of parameters grows. CompactNets present an advantage in performance when ρ is nearer to zero than to one, showing that deeper structures tend to benefit from a certain level of redundancy in the learned representations (less specialization), although most of the performance can be delivered by a much smaller set of parameters (larger ρ). The highest-performance zone (dark red) can be identified to the right of the heatmap, centered on $\rho = 0.03$, covering networks with depth between 11 and 18 layers. The lowest top-1 error rate is 31.2, and is achieved with a network of depth 12 and $\rho = 0.025$. By re-

laxing the compositional compactness regularizer (SimpleNet, $\rho = 0$), we see an evident increase in error-rate as depth increases beyond 10 layers. The lowest error rate of SimpleNet is 36.5 and is achieved by a 10-layer network (5.3 points worse than the one obtained with the best configuration for a CompactNet). This provides evidence that capturing compositional compactness by means of a group-sparse regularizer might be effective in improving (or removing) representations that might otherwise degrade performance due to overfitting. For CompressedNet, we perform a similar analysis, and found the best configuration also at 12 layers, with a top-1 error rate of 34.5. A more in thorough performance comparison is presented in Figure 7, where the lowest top-1 error-rate obtained is presented for each (depth, model) combination, by tuning the value of ρ.

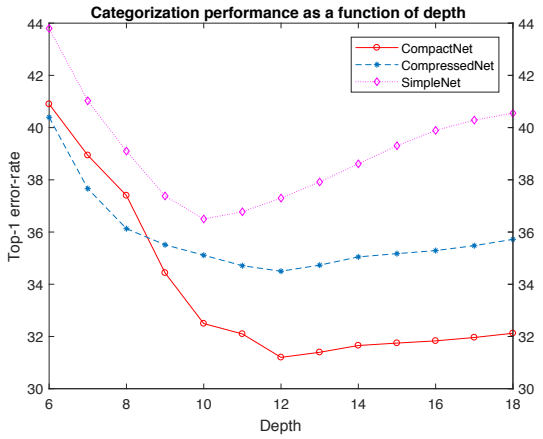

Fig. 7: Classification performance on the ILSVRC12 validation dataset, for models with different depths. Overall, CompactNets present the best performance in deeper models, while in shallower instantiations, CompressedNets shows an advantage.

We omit the results for instantiations with depths between 2 and 5 layers, as performance is very similar for all methods. Starting from depth 6, CompactNet shows an increasing advantage over the other architectures. This indicates the convenience, in terms of performance, of inducing group-sparse-based compositional compactness during training, instead of network compression by unit elimination. In Figure 7, we can also notice that the SimpleNet curve shows a sharp increase in error-rate with deeper networks. In contrast, CompactNets and CompressedNets behave differently, with only a minor performance decrease, overfitting might be present when ρ is low and the architecture is too deep. To asses this, in Figure 8 we plot the performance gap between the training and validation sets (validation - training), for all three models, using for each depth the value of ρ that generates the best performance in the validation set.

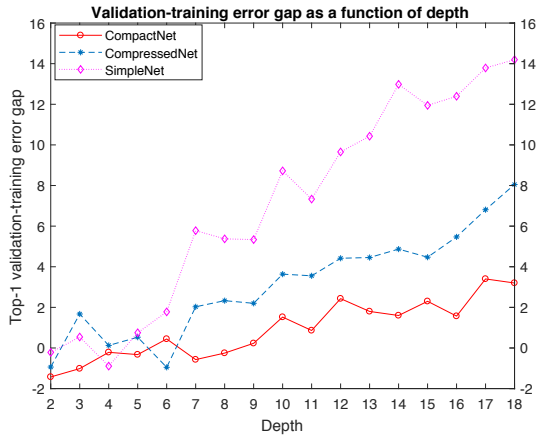

Fig. 8: Interestingly, as depth increases, the SimpleNet model shows the best performance on the training set, showing clear signs of overfitting. On the other hand, the other network models only shows a subtler performance increase, controlling overfitting in a more effective manner.

As expected, all methods show an increasing validation-training gap with deeper architectures, although with very different limits and local behavior. CompactNet only shows a subtler increase in the gap, with a maximum of approximately 3.5, controlling overfitting in a more effective manner. CompressedNets also show a similar behavior, but with a larger gap (≈ 8.0). This allows us to conclude that the group-sparse regularization terms, specially the ones related to compositional compactness, foster the learning of filters that are less prone to capture noise or non-discriminative visual patterns.

By varying the value of ρ, besides controlling the complexity of the models, we can also define different operation modes for the networks. Specifically, we identify three interesting scenarios for each group-regularized model:

- Compressed (C): the network with the smallest number of parameters that reaches near-AlexNet performance.
- Best performance (B): the network that presented the lowest top-1 error rate.

- Performance match (M): the sparsest networks (less parameters) that match the performance of the remaining models (when possible), e.g. the sparsest CompactNets that match de performance of CompressedNet and SimpleNet.

We plot these networks in Figure 9 as a function of the top-1 error rate and the number of effective parameters. We include all possible scenarios described before for CompactNets, CompressedNets and SimpleNets.

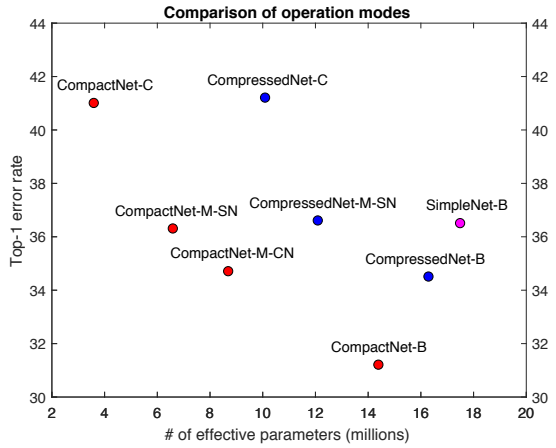

Fig. 9: Comparison of different operation modes for CompactNet, CompressedNet, and SimpleNet. The total number of parameters of all models lies between 19 and 22 millions.

As an overall impression, we can conclude that CompactNets always achieve better performance than the other models, using less parameters. In particular, the best performing CompactNet, besides beating CompressedNets by a 3.0 point margin, does it using almost two million less parameters. Comparatively, when performance is the same (M scenario), CompactNets use half the parameters of CompressedNet and a third of the parameters of SimpleNet. Perhaps the most surprising results is the number of effective parameters of CompactNet-C, achieving AlexNet performance level with 20x less parameters. As expected, the SimpleNet architecture discards a very small number of parameters when compared with CompactNet and CompressedNet, showing also lower performance.

Regarding the difference of CompactNets over CompressedNets, either in sparsity and in performance, we argue that this is mainly due to the fact that specialization naturally behaves more aggressively in deeper layers than in the initial ones, because of the higher level of abstraction these features show. As CompactNets are exactly based on these concept, they are able to take full advantage and discard a large number of parameters, without suffering performance loss. On the other hand, as CompressedNets are based on the idea of discarding units, they are more useful when abstraction is low and less convolutional filters are needed. This means that on higher layers, where more filters are needed, they are not as useful.

5.3.3. Generalization capabilities of the learned features

In this section, we show that by reducing the effect of overfitting, the group-sparse regularization term included in our model leads to an improved transfer-learning performance compared with an unregularized deep learning approach (Yang and Ramanan, 2015).

We evaluate our method by transferring the features learned on the ILSVRC12 dataset to the 15 Scene Categories (Lazebnik et al., 2006) and the MIT67 Indoor (Quattoni and Torralba, 2009) datasets. The 15 Scene Categories is a small-scale dataset containing 4485 natural scene images divided into 15 categories, while the MIT67 Indoor is a mid-scale dataset containing 6700 indoor scene images divided into 67 categories. These two datasets present important differences with the ILSVRC12 dataset, either in terms of size and image content, which poses an interesting setting to evaluate the transferability of the learned representations.

In order to provide a more complete analysis, we also explore the use of low-level features other than pixels, specifically the HoG+LBP (Wang et al., 2009) handcrafted features, commonly used to train shallow models from scratch on these datasets.

We evaluate the following mechanisms to generate and transfer features:

- **CNN features**: FC6 feature from an ImageNet trained AlexNet are fed to a multiclass linear SVM trained on the target datasets. Results are based on the experiments described in (Yang and Ramanan, 2015).

- **CompactNet/CompressedNet features**: Same as the

previous mechanism, but using a CompactNet-12 or CompressedNet-12 model instead of AlexNet.

- **CompactNet/CompressedNet fine-tunning**: A CompactNet-12 or CompressedNet-12 model is first trained on the ILSVRC12 dataset and then fine-tunned on the target dataset, dropping the group-sparse regularization terms. The classification layer is trained from scratch using a larger learning-rate than the rest of the layers.

- **CompactNet/CompressedNet from scratch**: A CompactNet or CompressedNet model is trained from scratch on the target dataset, using raw image pixels or HOG+LBP as low-level features. The reported depth is the one with best performance.

In terms of training and evaluation protocols, we use data augmentation in all datasets in the same manner as in previous experiments. For the 15 Scene Categories dataset we use five random splits of the data, with 100 images per class for training and the rest for testing. For MIT67 Indoor we use with 80 images per class for training and 20 for testing. For the sake of completeness, we also include performance numbers on the ILSVRC12 dataset, in order to assess the use of hand-crafted features on large-scale datasets. Table 2 shows the results for the aforementioned feature generating mechanisms, using the top-1 error rate as performance metric.

Although performance varies strongly with the different combinations of dataset, method, and feature type, we can establish some interesting relations among them. On 15 Scene Categories, results suggest that handcrafted features (HoG+LBP) with a somewhat shallow architecture (4 layers) trained from scratch are as good as CompactNet or CompressedNet transferred features. Moreover, performance using CompactNet or CompressedNet transferred features is superior than the one obtained by CNN transferred features, acknowledging the usefulness of group-sparse regularization for transfer-learning tasks. However, using pixels with a model trained from scratch delivers poorer performance than any other configuration by a large margin.

ILSVRC12		
Method	**Pixels**	**HoG+LBP**
CompressedNet scratch (12 layers)	38.7	46.3
CompactNet scratch (12 layers)	**37.6**	45.6

15 Scene Categories		
Method	**Pixels**	**HoG+LBP**
CNN features	13.2	-
CompressedNet features	**11.8**	15.0
CompactNet features	11.9	15.1
CompressedNet fine-tunning	12.9	14.2
CompactNet fine-tunning	**12.3**	14.0
CompressedNet scratch (3 layers)	27.6	12.0
CompactNet scratch (3 layers)	27.5	**11.7**

MIT67 Indoor		
Method	**Pixels**	**HoG+LBP**
CNN features	40.5	-
CompressedNet features	40.8	42.0
CompactNet features	**40.2**	42.2
CompressedNet fine-tunning	**38.6**	39.3
CompactNet fine-tunning	**38.6**	39.5
CompressedNet scratch (4 layers)	60.8	41.2
CompactNet scratch (4 layers)	61.1	**40.8**

Table 2: Performance of different mechanisms for generating features on a transfer-learning setting. Features generated using a CompactNet model show remarkable performance on all tested datasets.

On MIT67 Indoor, the difference in performance between configurations is smaller than in 15 Scene Categories, showing the best performance when features are fine-tunned, with the same error-rate for the CompactNet and CompressedNet models. Performance of transferred features without fine-tunning is similar for all methods, while performance of models trained from scratch using pixels keeps delivering a worse error-rate.

Finally, based on the results in the ILSVRC12 dataset, pixels as low-level features outperform handcrafted features when using deep structures on large-scale datasets, opposed to the situation in small- and mid-scale datasets. This is something to be

expected, as the large amount of data in the ILSVRC12 dataset allows deep models to discover patterns in raw pixels that are discarded by engineered features, because of their handcrafted nature.

5.3.4. Performance comparison with other methods

In this section, we compare the results obtained by our approach with several related methods. We focus our comparison on approaches that have a similar structure and/or number of parameters, in order to be as fair as possible. We add to this comparison methods that include some form of complexity control, either by means of architecture or regularization. We also include state-of-the-art approaches that deliver high recognition performance, at the cost of a large number of parameters, or sophisticated architectures. Finally, As our approach is orthogonal to these, we include preliminar results combining our regularization approach with advanced architectures, namely VGG and Resnet, in order to provide insights regarding the potential benefits of a joint application.

Table 3 shows performance results on the large-scale ILSVRC12 dataset. Compared with previous analyses, we add two new elements. First, we include the top-5 error-rate performance metric, which is part of the standard procedure when comparing performance on this dataset. Second, we include an extra column indicating the effective number of parameters used by the methods. We sort methods into five groups: Basic, Advanced, Architectural compression, Group-Sparse (GS) compression and Compositional Compactness (CC).

When comparing the CompactNet model with basic methods, we see that despite the reduced number of parameters effectively used, our approach is able to outperform AlexNet and its highly tuned version, ZNet (Zeiler and Fergus, 2014). Every instantiation of CompactNet is able to clearly surpass or reach the performance of AlexNet, and CompactNet-M-SN surpasses the performance of ZNet, using an order of magnitude less parameters. We argue that representations learned by traditional hierarchical techniques like AlexNet or ZNet, present a significant level of redundancy, and our approach is able to compress, fuse or even discarded them. The CompressedNet architecture also displays superior performance when compared to basic architectures, indicating that group-sparse regularization can generate improvements in performance, no matter the form of its application. Finally, the SimpleNet architecture is able to reach near AlexNet performance, using around 25% of the parameters. This result is in line with other works (Springenberg et al., 2015), that report gains or no performance loss, when removing fully-connected layers.

Next, if we focus on methods that change the structure of

ILSVRC12				
Type	**Method**	**Top-1**	**Top-5**	**# Params.**
Basic	AlexNet (Krizhevsky et al., 2012)	40.7	18.2	62M
	ZNet (Zeiler and Fergus, 2014)	38.4	16.5	62M
Arch. compr.	ZNet no FC (Zeiler and Fergus, 2014)	44.8	22.4	8M
	All conv. (Springenberg et al., 2015)	41.2	-	10M
	Low-rank (Tai et al., 2016)	-	19.4	12.5M
	SqueezeNet (Iandola et al., 2016)	40.0	18.2	1.3M
GS compr.	AlexNet$_{GS}$ (Zhou et al., 2016)	46.1	-	21.4M
	VGG13$_{GS}$ (Zhou et al., 2016)	39.3	-	51.5M
	VGG13$^{C}_{GS}$ (Alvarez and Salzmann, 2016)	37.3	-	25.0M
Advanced	VGG11 (Simonyan and Zisserman, 2015)	29.4	10.1	132.9M
	VGG16 (Simonyan and Zisserman, 2015)	26.5	8.5	138.4M
	Inception v3 (Szegedy et al., 2017)	20.8	5.4	27.2M
	ResNet18 (He et al., 2015a)	28.2	9.4	11.7M
	ResNet152 (He et al., 2015a)	20.3	5.1	60.2M
	DenseNet161 (Huang et al., 2017)	21.3	5.6	28.7M
Ours basic	SimpleNet-B	36.5	15.7	17.5M
Ours GS compr.	CompressedNet-B	34.5	14.8	16.3M
	CompressedNet-C	41.2	18.0	10.1M
	CompressedNet-M-SN	36.6	15.2	12.1M
Ours CC	CompactNet-B	31.2	11.1	14.4M
	CompactNet-C	41.0	17.9	3.6M
	CompactNet-M-SN	36.3	15.3	6.6M
	CompactNet-M-CN	34.7	14.4	8.7M
Advanced CC	VGG11$_{CC}$	27.9	9.0	63M
	ResNet18$_{CC}$	26.5	8.2	9.7M

Table 3: Categorization results comparison of different related hierarchical compositional methods on the ILSVRC12 dataset. Overall, our method is able to deliver competitive performance, using only a fraction of the parameters of other deep hierarchical methods. It is important to note the when compared to closely related work, *i.e.*, other group-sparse regularized methods, CompactNets show a clear advantage, in terms of efficiency and effectiveness. Moreover, when mixed with state-of-the-art architectures, our scheme is able to improve upon their original performance numbers, using fewer parameters.

traditional CNN models in order to reduce the total number of parameters, we can see that while their number of parameters is relatively similar to ours, the error-rates of CompactNet and CompressedNet is lower, validating the importance of our proposed approach to shape the parameter space in a meaningful way. An interesting comparison can be made with SqueezeNet, which is an architecture that relies heavily on 1×1 convolutional filters to reduce the number of active parameters. The only configuration that is able to reach a comparable result is CompactNet-C, with 3.6M effective parameters. Although SqueezeNet remains the most efficient method by a clear margin, it is interesting to notice that by only using group-sparse regularization, the size of parameter space can be shrinked almost to the level of a highly tuned architecture.

A relevant comparison is with methods that use some form of group-sparse-based regularization. All methods use a strategy akin to CompressedNets, trying to compress the network by means of unit elimination, with subtle differences in the form of the regularizer and the architecture. Results indicate that in all cases, CompactNets obtain better performance and with fewer parameters, validating our intuition that the proposed strategy generates an efficient way to shape the parameter space. It is worth noting that also CompressedNets perform better that all the other group-sparse regularization-based works. This difference might be explained by the different underlying architectures or by a different tuning of regularization constants, which play a critical role in performance, as described in Section 5.3.2. As a final remark, these results prove that in terms of performance, it is more important to specialize filters than to discard them, and that this also generates a larger reductions in dimensionality.

Finally, when compared with advanced architectures (Simonyan and Zisserman, 2015; Szegedy et al., 2017; He et al., 2015a; Huang et al., 2017), CompactNet-B performance is marginally lower than state-of-the-art deep models with a similar number of layers and/or parameters (VGG11 and ResNet18). This confirms the relevance of group-sparse-based compositional compactness to generate high quality representations. However, the difference in performance with the rest of the advanced methods is larger and indicates that in order to reach lower error-rates with the CompactNet architecture, structural improvements are needed. We assess this hypothesis with a preliminar set of experiments, that incorporate the compositional compactness regularization into VGG and ResNet, generating two new network models, namely VGG11_{CC} and ResNet18_{CC}. Specifically, we apply the Ω and Γ regularizers in the same fashion as in CompactNets, except for layers whose inputs are not convolutional feature maps or a spatial-pyramid decomposition. In these cases. the value of R and S in Eqs. 2 and 6 is set to 1.

Results show that our approach is able to enhance the recognition performance, while considerably reducing the number of effective parameters. Specifically, for VGG11_{CC}, we decrease top-1 and top-5 errors between 1 and 2 performance points, while reducing the number of effective parameters to less than 50% of the original VGG11. This shows that our approach is also effective for the representations learned in fully-connected layers. In the case of ResNet18_{CC}, top-1 and top-5 errors are also decreased similarly, and the number of effective parameters is reduced to 17%. While this decrease is smaller that the one observed in VGG_{CC}, it is still important, as the the ResNet architecture limits the use of fully-connected layers just for classification. These results indicate that the proposed scheme can be successfully applied to different architectures, effectively biasing the learning process to generate compact visual representations with high discriminative power. Also, this opens the door to further include novel architectures into our framework or vice versa.

5.4. Interpretability and specialization of units

Besides promoting a compact representation, compositional compactness can provide a mean to specialize units in CNNs, by either i) forwarding categorical information from classifiers to selected higher-layer units, or by ii) selecting only a subset of the available feature maps to generate features. Both of these elements can be captured by the proposed group-sparse regularizers, Ω, case i), and Γ, case ii).

To qualitatively assess the aforementioned effects of the regularizers, we employ the recently proposed Network Dissection technique (Bau et al., 2017), and analyze the characteristics of the semantically interpretable units detected by this method. In short, the Network Dissection technique computes the correlation between the activation of units from a given layer, and a series of human-interpretable concepts of different abstraction levels, that are present in a fully annotated dataset. Using this information and a dynamic thresholding scheme, one can compute the number of units that are highly associated with concepts, and how many units are associated with each of these.

As shown in (Bau et al., 2017), interpretability and performance are not competing objectives, meaning that high performance can be achieved by either highly interpretable networks, or by networks that capture only abstract representations. Taking this into account, one can think that at a similar performance level, a way to measure specialization of units of different networks is not by counting how many interpretable units exists, but by counting how many units are needed on average to capture a visual concept. To this end, we define the specialization coefficient, $\hat{s}$, as the ratio between the number of unique interpretable visual concepts captured by a model, and the number of interpretable units. The higher this ratio is, the more specialized units are, as less of them are required to capture the visual concepts, while maintaining a similar level of performance.

We compute the specialization coefficient of the last layer before the classification layer (or fully-connected layers where applicable) of several network architectures trained on the ILSVRC12 dataset. Notice that by the effect of the Ω regularizer in the CompactNet model, this layer's units receive categorical information only from few classifiers, providing semantically-driven specialization. To keep comparison fair, the number of parameters and of units per layer is kept the same whenever possible. Also, when available, we take results from the network dissection project website [1]. Table 4 presents our results.

When inspecting the values for the specialization coefficient $\hat{s}$ of the different architectures, we can notice that the variants of the CompactNet model obtain a higher value than all other architectures by a large margin, with the exception of the CompressedNet architecture. The CompactNet-C presents the larger value for the coefficient, which coincides with the strong regularization strategy that it uses. This provides more evidence that our proposed group-sparse regularization strategy is effectively capturing unit specialization by categorical information forwarding of selected classifiers, and by means of compositional compactness. The results of CompressedNet-C are also interesting, as they indicate that one way to make an efficient use of a limited number of filters is by focusing on specializing them.

When comparing with off-the-shelf CNN architectures without group-sparse regularization, like SimpleNet, AlexNet or VGG16, one can observe that specialization decreases noticeably. VGG16 is an interesting case to analyze, as it uses the same filter size and number of filters per layer as our approach, but with a much larger number of parameters. This abundance of parameters produces many redundant units that capture the same visual concept, wasting representational power that could

Specialization coefficients			
Method	**# units**	**concepts**	$\hat{s}$
AlexNet (Krizhevsky et al., 2012)	256	47	0.36
VGG16 (Simonyan and Zisserman, 2015)	512	99	0.31
Inception v3 (Szegedy et al., 2017)	1024	108	0.17
ResNet152 (He et al., 2015a)	2048	195	0.10
DenseNet161 (Huang et al., 2017)	2208	166	0.19
SimpleNet-B	512	87	0.38
CompressedNet-C	512	42	0.53
CompressedNet-M-SN	512	85	0.50
CompressedNet-B	512	93	0.41
CompactNet-C	512	48	0.57
CompactNet-M-CN	512	84	0.48
CompactNet-B	512	104	0.44

Table 4: Specialization coefficients for different network architectures. CompactNet models, followed by CompressedNet models, obtain the larger value of $\hat{s}$ by a considerable margin, indicating that group-sparse regularization is an affective approach to increase unit specialization in neural networks.

[1] http://netdissect.csail.mit.edu/

be used to capture different visual patterns.

Something similar happens with more complex or deeper architectures, such as Inception, ResNet and DenseNet, where the difference in specialization is even larger. Although the number of parameters of these networks is not as large as in VGG16, they are considerably deeper and employ a significantly larger number of filters in deeper layers, than the one used by our approach. We argue that these structural differences induce the learning of a large number of filters, that jointly capture the appearance of a certain visual pattern with high precision.

6. Conclusions and Future Work

In this article, we have proposed the CompactNet model, a novel method for visual recognition that uses explicit group-sparse terms in the objective function, in order to induce compositional compactness on the learned representation. As a main contribution, our experiments provide evidence that the proposed method learns specialized and compact representations, relevant properties for the effectiveness and scalability of deep hierarchical compositional models. Moreover, extensive experimental validation performed on various datasets and architectures allowed us to demonstrate the generalization power of the visual patterns captured by the mid-level layers. Finally, we proved that specialization is highly effective to manage model complexity, leading to compact representations that achieve remarkable performance using a fraction of the parameters used by previous approaches.

In future work, we plan to investigate the effect of different regularization strategies, that might be better suited to deep structures. Also, we will further integrate our regularization strategy with modern deep architectures.

Acknowledgment

This work was partially funded by FONDECYT grant 1151018, NSF grants 1218709 and 1527340, ONR grant N000141310116, and the Millennium Institute for Foundational Research on Data (IMFD).

References

Albanese, M., Chellappa, R., Cuntoor, N., Moscato, V., Picariello, A., Subrahmanian, V., Udrea, O., 2010. PADS: A Probabilistic Activity Detection Framework for Video Data. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 2246–2261.

Alvarez, J., Salzmann, M., 2016. Learning the Number of Neurons in Deep Networks, in: Conference on Neural Information Processing Systems (NIPS).

Alvarez, J., Salzmann, M., 2017. Compression-aware training of deep networks, in: Conference on Neural Information Processing Systems (NIPS).

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A., 2017. Network dissection: Quantifying interpretability of deep visual representations, in: Conference on Computer Vision and Pattern Recognition (CVPR).

Bruna, J., Mallat, S., 2013. Invariant Scattering Convolution Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1872–1886.

Changpinyo, S., Sandler, M., Zhmoginov, A., 2017. The power of sparsity in convolutional neural networks. arXiv:1702.06257 .

Choromanska, A., Henaff, M., Mathieu, M., Ben Arous, G., LeCun, Y., 2015. The Loss Surfaces of Multilayer Networks, in: International Conference on Artificial Intelligence and Statistics.

Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., Batra, D., 2016. Reducing Overfitting in Deep Networks by Decorrelating Representations, in: International Conference on Learning Representations (ICLR).

Denil, M., Shakibi, B., Dinh, L., Ranzato, M., de Freitas, N., 2013. Predicting Parameters in Deep Learning, in: Conference on Neural Information Processing Systems (NIPS).

Geman, S., Potter, D., , Chi, Z., 2002. Composition Systems. Quarterly of Applied Mathematics 60, 707–736.

Girshick, R., Felzenszwalb, P., McAllester, D., 2011. Object detection with grammar models, in: Conference on Neural Information Processing Systems (NIPS).

Giryes, R., Sapiro, G., Bronstein, A., 2015. Deep Neural Networks with Random Gaussian Weights: A Universal Classification Strategy? CoRR abs/1504.08291.

Haeffele, B., Vidal, R., 2015. Global Optimality in Tensor Factorization, Deep Learning, and Beyond. CoRR abs/1506.07540.

Han, S., Pool, J., Tran, J., Dally, W., 2015. Learning both weights and connections for efficient neural networks, in: Conference on Neural Information Processing Systems (NIPS).

He, K., Zhang, X., Ren, S., Sun, J., 2015a. Deep Residual Learning for Image Recognition, in: ICCV ImageNet and MS COCO Visual Recognition Challenges Joint Workshop.

He, K., Zhang, X., Ren, S., Sun, J., 2015b. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, in: International Conference on Computer Vision 2015 (ICCV).

Huang, G., Liu, Z., van der Maaten, L., Weinberger, K., 2017. Densely con-

nected convolutional networks, in: Conference on Computer Vision and Pattern Recognition (CVPR).

Iandola, F., Moskewicz, M., Ashraf, K., Han, S., Dally, W., Keutzer, K., 2016. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. arXiv:1602.07360 .

Ioannou, Y., Robertson, D., Shotton, J., Cipolla, R., Criminisi, A., 2016. Training CNNs with Low-Rank Filters for Efficient Image Classification, in: International Conference on Learning Representations (ICLR).

Jahangiri, E., Yoruk, E., Vidal, R., Younes, L., Geman, D., 2017. Information Pursuit: A Bayesian Framework for Sequential Scene Parsing. CoRR abs/1701.02343.

Jin, Y., Geman, S., 2006. Context and Hierarchy in a Probabilistic Image Model, in: Conference on Computer Vision and Pattern Recognition (CVPR).

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. ImageNet Classification with Deep Convolutional Neural Networks, in: Conference on Neural Information Processing Systems (NIPS).

Lazebnik, S., Schmid, C., Ponce, J., 2006. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, in: Conference on Computer Vision and Pattern Recognition (CVPR).

Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H., 2017. Pruning Filters for Efficient ConvNets, in: International Conference on Learning Representations (ICLR).

Lobel, H., Vidal, R., Soto, A., 2015. Learning Shared, Discriminative, and Compact Representations for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 37, 2218–2231.

Montufar, G., Pascanu, R., Cho, K., Bengio, Y., 2014. On the Number of Linear Regions of Deep Neural Networks, in: Conference on Neural Information Processing Systems (NIPS).

Potter, D., 1999. Compositional Pattern Recognition. Ph.D. thesis. Division of Applied Mathematics, Brown University.

Quattoni, A., Torralba, A., 2009. Recognizing indoor scenes, in: Conference on Computer Vision and Pattern Recognition (CVPR).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision 115, 211–252.

Scardapane, S., Comminiello, D., Hussain, A., Uncini, A., 2017. Group sparse regularization for deep neural networks. Neurocomputing 241, 81–89.

Shalev-Shwartz, S., Tewari, A., 2011. Stochastic Methods for L1-regularized Loss Minimization. Journal of Machine Learning Research 12, 1865–1892.

Shor, N., 1985. Minimization Methods for Non-Differentiable Functions. Springer-Verlag.

Simonyan, K., Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition, in: International Conference on Learning Representations (ICLR).

Springenberg, J., Dosovitskiy, A., Brox, T., Riedmiller, M., 2015. Striving for Simplicity: The All Convolutional Net, in: International Conference on Learning Representations (ICLR).

Suo, J., Zhu, S., Shan, S., Chen, X., 2010. A Compositional and Dynamic Model for Face Aging. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 385–401.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going Deeper with Convolutions, in: Conference on Computer Vision and Pattern Recognition (CVPR).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2017. Rethinking the inception architecture for computer vision, in: Conference on Computer Vision and Pattern Recognition (CVPR).

Tai, C., Xiao, T., Wang, X., E, W., 2016. Convolutional neural networks with low-rank regularization, in: International Conference on Learning Representations (ICLR).

Tu, K., Pavlovskaia, M., Zhu, S., 2013. Unsupervised Structure Learning of Stochastic And-Or Grammars, in: Conference on Neural Information Processing Systems (NIPS).

Wang, L., Zhu, J., Zou, H., 2006. The Doubly Regularized Support Vector Machine. Statistica Sinica 16, 589–616.

Wang, X., Han, T., Yan, S., 2009. An HOG-LBP human detector with partial occlusion handling, in: International Conference on Computer Vision (ICCV).

Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H., 2016. Learning Structured Sparsity in Deep Neural Networks, in: Conference on Neural Information Processing Systems (NIPS).

Wen, W., Xu, C., Wu, C., Wang, Y., Chen, Y., Li, H., 2017. Coordinating filters for faster deep neural networks, in: International Conference on Computer Vision (ICCV).

Yang, S., Ramanan, D., 2015. Multi-scale Recognition with DAG-CNNs, in: International Conference on Computer Vision (ICCV).

Zeiler, M., 2012. ADADELTA: An Adaptive Learning Rate Method. CoRR abs/1212.5701.

Zeiler, M., Fergus, R., 2014. Visualizing and Understanding Convolutional Networks, in: European Conference on Computer Vision (ECCV).

Zhou, H., Alvarez, J.M., Porikli, F., 2016. Less Is More: Towards Compact CNNs, in: European Conference on Computer Vision (ECCV).

Zhu, M., Gupta, S., 2018. To Prune, or Not to Prune: Exploring the Efficacy of Pruning for Model Compression, in: International Conference on Learning Representations (ICLR).

Zhu, S., Mumford, D., 2006. A Stochastic Grammar of Images. Foundations and Trends in Computer Graphics and Vision 2, 259–362.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67, 301–320.