DP-NET: LEARNING DISCRIMINATIVE PARTS FOR IMAGE RECOGNITION

R. Sicre, H. Zhang, J. Dejasmin, C. Daaloul, S. Ayache, T. Artières

DP-NET: LEARNING DISCRIMINATIVE PARTS FOR IMAGE RECOGNITION

R. Sicre, H. Zhang, J. Dejasmin, C. Daaloul, S. Ayache, T. Artières

Centrale Marseille, Aix Marseille Univ, CNRS, LIS, Marseille, France

ABSTRACT

This paper[1] presents Discriminative Part Network (DP-Net), a deep architecture with strong interpretation capabilities, which exploits a pretrained Convolutional Neural Network (CNN) combined with a part-based recognition module. This system learns and detects parts in the images that are discriminative among categories, without the need for fine-tuning the CNN, making it more scalable than other part-based models. While part-based approaches naturally offer interpretable representations, we propose explanations at image and category levels and introduce specific constraints on the part learning process to make them more discrimative.

Index Terms— image classification, part-based models, interpretability.

1. INTRODUCTION

Since 2012, Deep Neural Networks (DNN) have been re-popularized in the fields of computer vision and machine learning. Deep learning methods are used to address almost every single computer vision problem such as image classification, retrieval, detection, segmentation, etc. The ability to transfer pre-trained representations learned on large annotated datasets led to large improvements.

With most efforts dedicated to improving further these methods, it is interesting to remember previous methods and concepts that occur prior to the deep learning tsunami. For instance, a large effort has been dedicated to Part-Based Models (PBM) starting with the deformable part model [1]. Based on these models, diverse strategies were later proposed to learn a collection of discriminative parts [2, 3]. Global image representations are further derived from the learned parts to perform recognition, while mainstream methods would aggregate dense local representations instead [4].

With the re-popularization of deep learning, links between part-based models and CNN architectures are discussed in [5]. First, PBM started using pre-trained network to replace previous feature extraction techniques [6, 7, 8]. Later, part inspired architectures are introduced to target the problem of fine-grained classification [9, 10, 11] or scene recognition [12], showing the benefit of such architectures on specific datasets and tasks. Furthermore, ProtoPNet [13] and its extensions [14, 15, 16] provide new interpretability ability by exploiting relations between categories and parts. However, these architectures still need multiple training stages; can not cope with large scale dataset as ImageNet [17]; require extra annotations such as bounding boxes or part annotations; or do not provide more interpretability than part-level visualization.

Our method is a part-based architecture that addresses these limitations. Learning discriminative parts is performed jointly with the classification. Unlike previous models, we exploit a pretrained DNN to compute representations on a number of random regions in images, in order to improve simplicity and scalability. Parts are learned solely based on image level labels, i.e. without any priors or bounding boxes. Moreover, our model provides explanations of parts and classification decision, for a specific input or more broadly for a category. Finally, we show that introducing a number of constraints on the parts in the objective function helps discovering more discriminative and relevant parts.

2. PREVIOUS WORKS

Early part-based approaches Most approaches define parts as image regions that can help differentiate between the categories. However, methods vary in how they select these regions. The constellation model [18] and deformable part model [1] are the original methods that model classes by parts and their positions. Later approaches [2, 3, 19, 20, 6] first learn the parts, then the decision function, based on parts response. Parts are learned by identifying candidate regions [2], using mean-shift algorithm [3], an iterative *softassign* matching algorithm [19, 20], or boosting [6]. However Parizi *et al* [7] jointly learns the parts and the category classifiers.

Our method also optimizes the final classification in a single-stage optimization. We further study constraints on parts inspired by these works [6, 20].

Part-inspired architecture Several recent part inspired methods focus on fine-grained image classification. Most of these methods follow detection architectures, as R-CNN [21, 22], where detection and classification are learned alternatively. Other works are based on attention models

[1]This work has received funding from the UnLIR ANR project (ANR-19-CE23-0009), the Excellence Initiative of Aix-Marseille Universite - A*Midex, a French "Investissements d'Avenir programme" (AMX-21-IET-017). Part of this work was performed using HPC resources from GENCI-IDRIS (Grant 2020-AD011011853 and 2020-AD011013110).

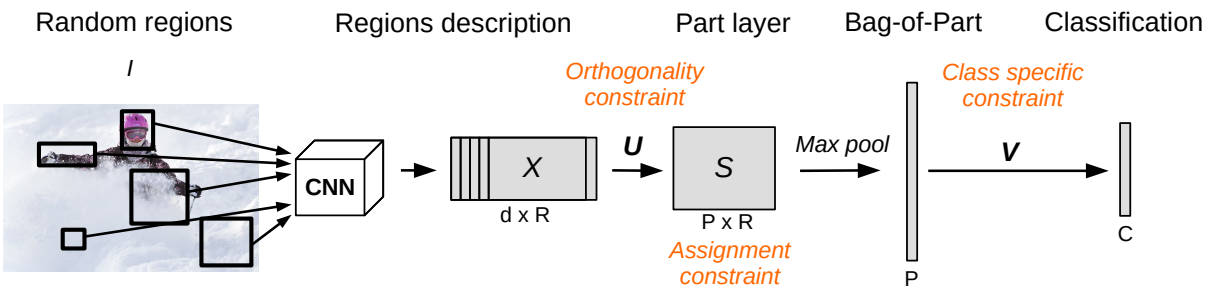

Fig. 1. Figure of the proposed architecture and its learned parameters U and V.

[23, 11, 24]. Recently, Chen *et al* [12] propose to build protoype-agnostic scene layout, using graph convolutions to encode parts and their relations. Also, Krause *et al* [25] generate parts using co-segmentation and alignment.

ProtoPNet [13] propose to learn parts prototype in three stages, which is soon improved by allowing shared parts [14], negative parts [15] and more effective optimization [16]. These recent methods are directly aligned with our objectives, but their optimization remain complex with fine-tuning of the backbone network, fixed size region, pruning requirements, etc. Our method thus offer a simpler adaptable architecture that can learn on large datasets.

Interpretability Interpretability recently receives a lot attention [26]. Numerous methods consider models as black boxes, and perform post-hoc interpretability. While other methods aim at changing models, or representations, to make them more transparent or easily understandable.

Some transparency can be obtained through parts. However, these methods can not easily link parts representation with the decision, except for ProtoPNet [13], which is the first work to propose visualizable parts and their contributions to a decision at inference.

In our work, we also provide explainability at inference, but at a lower computational cost. We also extend the posthoc method CAM [27] to part models, enabling various levels of interpretability. Finally, we study various constraints to provide both discriminative and interpretable parts.

3. DISCRIMINATIVE PARTS NN (DP-NET)

3.1. DP-net architecture

In a nutshell, the DP-net is a part-based architecture that includes: 1) a pretrained CNN, i.e. backbone, that produces high-level descriptions of numerous (randomly selected) regions in an input image, 2) a part layer that outputs a matching score for every part-region pair, 3) a max-pooling that encodes parts information in a dense vector representation, and 4) a final classification layer, see Fig. 1. Using a pretrained CNN (that is not refined) and randomly selected regions makes our method particularly simple, adaptable and scalable.

In more details, when processing an input image $I \in \mathcal{I}$, a fixed number R of rectangular regions, $\mathcal{R} = \{i_1, ..., i_R\}$, are extracted and then described by a $d-$dimensional descriptor vector output by a pretrained CNN, e.g. VGG19, ResNet50, etc. The resulting matrix is defined as $X \in \mathbb{R}^{d \times R}$, whose r^{th} column is the descriptor of the r^{th} region, $x^r = CNN(i^r) \in \mathbb{R}^d$.

The part layer computes a matching score between regions and parts that are represented as $d-$dimensional vectors living in the same space as region descriptors. We note $U \in \mathbb{R}^{P \times d}$ the part layer weights, with u_p the representation of the p^{th} part. The output of this layer is defined as the score matrix $S \in \mathbb{R}^{P \times R}$ with $S = U \times X = (s_{p,r})_{p,r \in [1,...P] \times [1,...R]}$ with $s_{p,r} = u_p \times x^r$. This score matrix is often referred to as the assignment matrix as it relates extracted regions to the learned parts. We further apply one-dimensional max-pooling, and $L2$ normalization, to obtain a final "bag-of-parts" (BOP) embedding, introduced in [2] and used in most part-based method. The bag-of-parts is a P-dimensional vector given as input to a final classification layer, i.e. a fully connected layer, with a softmax activation function, with weights $V \in \mathbb{R}^{C \times P}$ where C denotes the number of classes. To summarize, for an input image I, the model computes successively X, S, b and o, as:

$$X = [CNN(i_1(I)), ..., CNN(i_R(I))], \quad S = U \times X,$$
$$b = \left[\max_{r \in R}(s_{p,r})\right]_{p \in [1,...P]}, \qquad o = \text{softmax}(V \times b).$$

3.2. Learning

Our model simply learns the part matrix U and the classification layer V, through gradient descent, to minimize categorical cross-entropy noted as $C_{CCE}(U, V)$.

Actually the most important aspect of the learning lies in constraints that are added through additional terms to the objective function, to improve interpretability. Here is a list of desirable properties of the parts:

1) Parts should be complementary, i.e. parts should be different one from another.

2) Parts should cover as much as possible the diversity of regions extracted from images.

3) Parts should be discriminative with respect to classes.

4) Parts should be specific to categories.

These properties can be satisfied by including additional terms in the objective function. We note that property 3) is already addressed by $C_{CCE}(U, V)$.

Specifically, the first property is induced by enforcing parts to be *orthogonal* one to another, simply adding to the objective function a term as $C_{\perp}(U) = -\frac{1}{P^2}\sum_{i=1}^{P}\sum_{j=1,j\neq i}^{P}(u_i^T u_j)^2$.

The second property can be rewritten as: we want every region to be, as much as possible, *assigned* to one of the parts, e.g. following [20]. This can be easily encouraged by minimizing the entropy of each column of the score matrix S, matching regions to all parts. Thus, we define $C_{Assign}(U) = -\sum_{r=1}^{R}\sum_{p=1}^{P} s_{p,r} log(s_{p,r})$. Note that softmax is first applied on the columns of the matrix S and each part vector u_p is assumed to be $l2$-normalized for both $C_{\perp}(U)$ and $C_{Assign}(U)$.

Finally, the last property is enforced by adding a constraint on the parameters of the classification layer V, by minimizing the contribution of parts that are not assigned to the given class, as in [13]. Let q be the number of parts learned per class, our *class-specific (CS)* constraint can be computed as: $CS(V) = \frac{1}{P(C-1)}\sum_{i=1}^{C}\sum_{j=1,j\notin[q(i-1),qi]}^{P} V_{i,j}$

Finally the learning is performed through the minimization of the combined loss function:

$$\begin{aligned}\mathcal{C}(U, V, Z) = C_{CCE}(U, V) + \lambda_1 C_{\perp}(U) \\ + \lambda_2 C_{Assign}(U) + \lambda_3 C_{CS}(U, Z)\end{aligned} \quad (1)$$

3.3. Interpretability strategies

First of all, the learned part p can be simply characterized by the region r, when this region produces the highest match scores $s_{p,r}$ over the entire training set.

Secondly, we want to quantify the importance of the part p for the detection of a particular class c. This measurement can be obtained through the classification layers that takes as input the bag-of-parts representation b, which consist in a single score per part. Moreover, the output of the model is computed as $o = softmax(V \times b)$ so that the prediction for class c is $o_c = v_c \times b = \sum_p v_{c,p} * b_p$. The importance of part p when predicting class c for a particular input image can then be measured by $v_{c,p} * b_p$.

At the class level, [13] proposed to interpret categories by the participation of the learned parts in the decision. We follow this idea and adapt here the popular Class Activation Map (CAM) [27]. To measure the global importance of part p to recognize class c, we want to compute an average of this quantity over the training set. Yet, we found more informative to weigh this quantity by the popularity $f(p)$ of the part among all classes. This frequency term is similar to the $TF.IDF$ weighting scheme from the seminal Vector Space Model in text information retrieval. We thus compute the discriminative power of part p for class c as:

$$d(p, c) = \sum_{I\in\mathcal{I}, y_I=c} \frac{v_{c,p} * b_p(I)}{log(f(p))} \quad (2)$$

where $f(p)$ measures to which extent part p is frequently detected in images of all classes. $f(p)$ is computed as the number of classes having p in their most activated parts, measured by $v_{c,p} * b_p$. Given these statistics, given a class c, one can visualize the typical regions that are associated with the top parts maximizing $d(p, c)$, by finding regions maximizing $s_{p,r}$.

Finally, given an input image and a class c, one can use the presented statistics to infer how the image regions participate to the final classification. Specifically, we can first identify the top-N most discriminative parts for the class c according to Eq. (2). Then the top-M regions that activate these parts are selected and can be highlighted to compute a heatmap, to help explaining the decision of the model.

4. EXPERIMENTS

Datasets We study three datasets: **CUB-200-2011** [28] fine-grained classification of bird species, the **MIT 67 Scenes** [29] dataset of indoor scenes, and the large **ImageNet** [17] (ILSVRC 2012) dataset. Only image-level categories are used and no data augmentation is performed.

Implementation details Each image is resized to 544×544 then given as input to a CNN: VGG-19 or ResNet-50, pretrained on ImageNet. From the output of the last convolutional layer, we extract the feature maps covering R random regions and perform average pooling on these features. We set $R = 500$ for MIT and CUB and $R = 100$ for ImageNet.

To learn on MIT-67 and CUB-200, we perform 40 epochs with Adam optimizer and a learning rate of 10^{-3}. The learning rate is divided by 10 after each 10 epochs, reaching 10^{-6}. Since our input data is large, we build large batches of training data and perform 32 batch-level epochs. Concerning ImageNet, we similarly perform 10 batch-level epochs and 4 epochs with learning rate set to 10^{-3}.

Concerning constraints, after evaluation, the scaling constants are set to $\lambda_1 = 10^{-2}$, $\lambda_2 = 10^{-3}$, and $\lambda_3 = 10^{-3}$.

Model performance We first show the benefits of using the proposed DP-Net over global representation on the three datasets and two networks, see Table 1. We learn 20, resp. 10, parts per class for MIT and CUB, resp. ImageNet. The global representation model computes global average pooling after the last convolutional layer of the same network followed by

Method	ISA parts [20]		Our DP-Net		
Dataset	MIT		MIT		
Network	VGG	Places	VGG	Places	RN50
Global	73.3	78.5	76.2	79.8	78.1
Parts	75.1	81.6	76.9	82.0	79.7

Dataset	Birds		ImageNet	
Network	VGG	RN50	VGG	RN50
Global	66.4	81.5	61.0	70.8
Parts	76.1	84.9	69.0	74.6

Table 1. Tables comparing our DP-Net with global representations on MIT, Birds, and ImageNet datasets using VGG and ResNet. We note that parts are learned without constraints.

Dataset	Constraints			
	wo	$\perp$	Assign	CS
Birds	84.9	84.6	84.6	84.5
MIT	79.7	79.1	80.3	79.5
	$\perp$+Assign	CS+$\perp$	CS+Assign	CS+$\perp$+Assign
Birds	85.1	84.4	84.3	85.0
MIT	80.3	78.8	79.9	80.5

Table 2. Accuracy obtained with ResNet 50 when using the constraints defined in 3.2 (wo = without any constaint).

either a single, or two, FC layer. The best performing global model is presented, as adding the second layer allows to compare the model with the same number of parameters as DP-Net. We observe that part methods significantly outperform global models on most cases.The method is also compared to the part-based method proposed in [20]. Our model performs similarly but is simpler and more efficient.

Training on ImageNet is particularly interesting, as none of the previous part-based method can cope with such a large dataset. However, we note that the original models obtain better scores than DP-Net and our global representation: 71.3% and 74.9% accuracy for VGG-19 and ResNet-50. The performance drop for global representation is explained by the image dimension increase and global average pooling added in the case of VGG.

As mentioned earlier, several constraints on parts are evaluated: orthogonality, assignment between regions and parts, and class specific (CS) parts, see Table 2. These constraints allow better interpretation of parts but there is no significant alteration of the performance using these constraints.

Interpretability evaluation We illustrate how parts help interpreting the DNN decision. First, Figure 2 shows the three most discriminative parts for the "casino" class of the MIT dataset according to equation 2. We observe that constraints offer parts that look better aligned with categories.

Figure 3 shows examples of heatmaps, where the most

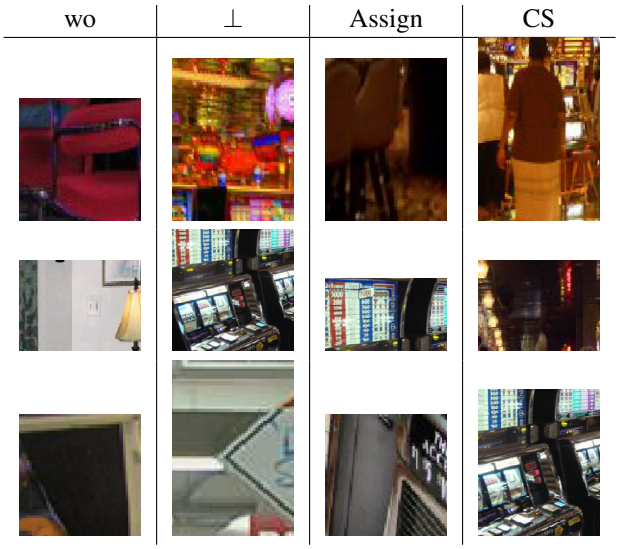

Fig. 2. Three most important parts for the class `Casino`.

Fig. 3. Heatmap illustrations using test images of classes `Artstudio`, `Computer room`, `Casino` and `Kindergarden` on top row and most discriminant region used to classify birds test images on second row.

discriminative regions are highlighted in yellow. We note that the model focuses on semantically meaningful regions to take its decision. Figure 3 also highlights the most discriminative region used by the model to classify birds species. It is interesting to see that our model finds distinctive characteristics of species to recognize them: the long tail of the Florida Jay or the grainy appearance of the Spotted Catbird's chest.

5. CONCLUSION

This paper presents DP-Net, an approach using a pretrained CNN to learn interpretable part representations. The neural architecture is accurate, scalable, and can deal with a variety of image recognition problems. We introduce a number of constraints to drive part learning to favour interpretability of the model decisions. Our experiments show interesting performance compared to standard global representations systems, across several networks and multiple datasets. We also provide evidence of the interpretability capabilities of our model by visualizing parts, their discriminative capabilities and their contribution in the decision.

6. REFERENCES

[1] Pedro Felzenszwalb, Ross Girshick, David McAllester, and Deva Ramanan, "Object detection with discriminatively trained part-based models," *Trans. PAMI*, 2010.

[2] Mayank Juneja, Andrea Vedaldi, C V Jawahar, and Andrew Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *CVPR*, 2013.

[3] Carl Doersch, Abhinav Gupta, and Alexei A Efros, "Mid-level visual element discovery as discriminative mode seeking," in *NeurIPS*, 2013.

[4] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," 2010, ECCV.

[5] Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik, "Deformable part models are convolutional neural networks," in *CVPR*, 2015.

[6] Pascal Mettes, Jan C. van Gemert, and Cees G. M. Snoek, "No spare parts: Sharing part detectors for image categorization," *CoRR*, vol. abs/1510.04908, 2015.

[7] S.N. Parizi, A. Vedaldi, A. Zisserman, and P. Felzenszwalb, "Automatic discovery and optimization of parts for image classification," in *ICLR*, 5 2015.

[8] Ronan Sicre and Frédéric Jurie, "Discriminative part model for visual recognition," *CVIU*, 2015.

[9] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang, "Part-stacked cnn for fine-grained visual categorization," in *CVPR*, 2016.

[10] Di Lin, Xiaoyong Shen, Cewu Lu, and Jiaya Jia, "Deep lac: Deep localization, alignment and classification for fine-grained recognition," in *CVPR*, 2015.

[11] Xiao Liu, Tian Xia, Jiang Wang, and Yuanqing Lin, "Fully convolutional attention localization networks: Efficient attention localization for fine-grained recognition," *arXiv preprint arXiv:1603.06765*, 2016.

[12] Gongwei Chen, Xinhang Song, Haitao Zeng, and Shuqiang Jiang, "Scene recognition with prototype-agnostic scene layout," *arXiv preprint arXiv:1909.03234*, 2019.

[13] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su, "This looks like that: deep learning for interpretable image recognition," in *NeurIPS*, 2019.

[14] Dawid Rymarczyk, Lukasz Struski, Jacek Tabor, and Bartosz Zieliński, "Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification," in *ACM SIGKDD*, 2021.

[15] Meike Nauta, Ron Van Bree, and Christin Seifert, "Neural prototype trees for interpretable fine-grained image recognition," in *CVPR*, 2021.

[16] Dawid Rymarczyk, Lukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński, "Interpretable image classification with differentiable prototypes assignment," in *ECCV*, 2022.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[18] M. Weber, M. Welling, and P. Perona, "Towards automatic discovery of object categories," in *CVPR*, 2010.

[19] Ronan Sicre, Julien Rabin, Yannis Avrithis, Teddy Furon, and Frederic Jurie, "Automatic discovery of discriminative parts as a quadratic assignment problem," *ICCV Work.*, 2017.

[20] Ronan Sicre, Yannis Avrithis, Ewa Kijak, and Frédéric Jurie, "Unsupervised part learning for visual recognition," in *CVPR*, 2017.

[21] Georgia Gkioxari, Ross Girshick, and Jitendra Malik, "Contextual action recognition with r* cnn," in *ICCV*, 2015.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.

[23] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, and al., "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *CVPR*, 2015.

[24] Jianlong Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *CVPR*, 2017.

[25] Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei, "Fine-grained recognition without part annotations," in *CVPR*, 2015.

[26] Zachary C Lipton, "The mythos of model interpretability," *arXiv preprint arXiv:1606.03490*, 2016.

[27] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, "Learning deep features for discriminative localization," in *CVPR*, 2016.

[28] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.

[29] A. Quattoni and A. Torralba., "Recognizing indoor scenes," in *CVPR*, 2009.