

Saliency for Fine-grained Object Recognition in Domains with Scarce Training Data

Carola Figueroa Flores^{a,b,*}, Abel Gonzalez-Garcia^a, Joost van de Weijer^a,
Bogdan Raducanu^a

^a*Computer Vision Center
Edifici "O" - Campus UAB
08193 Bellaterra (Barcelona), Spain*

^b*Department of Computer Science and Information Technology
Universidad del Bío Bío, Chile*

Abstract

This paper investigates the role of saliency to improve the classification accuracy of a Convolutional Neural Network (CNN) for the case when scarce training data is available. Our approach consists in adding a saliency branch to an existing CNN architecture which is used to modulate the standard bottom-up visual features from the original image input, acting as an attentional mechanism that guides the feature extraction process. The main aim of the proposed approach is to enable the effective training of a fine-grained recognition model with limited training samples and to improve the performance on the task, thereby alleviating the need to annotate a large dataset. The vast majority of saliency methods are evaluated on their ability to generate saliency maps, and not on their functionality in a complete vision pipeline. Our proposed pipeline allows to evaluate saliency methods for the high-level task of object recognition. We perform extensive experiments on various fine-grained datasets (Flowers, Birds, Cars, and Dogs) under different conditions and show that saliency can considerably improve the network's performance, especially for the case of scarce training data. Furthermore, our experiments show that saliency methods that obtain improved saliency maps (as measured by traditional saliency benchmarks) also translate

*Corresponding author

Email address: cafigueroa@cvc.uab.es (Carola Figueroa Flores)

to saliency methods that yield improved performance gains when applied in an object recognition pipeline.

Keywords: object recognition, fine-grained classification, saliency detection, scarce training data

1. Introduction

Fine-grained object recognition focuses on the classification of subclasses belonging to the same category. Examples of fine-grained datasets include natural categories such as flowers [1], birds [2], dogs [3] and man-made categories such as cars [4]. The problem of fine-grained object classification is difficult because the differences between subclasses are often subtle and expert labelers, with knowledge of the discriminating attributes, are needed for the collection of datasets. Therefore the collection of large datasets is expensive and the development of algorithms that only require few labeled examples is of special interest to the field.

Computational saliency estimation aims to identify to what extent regions or objects stand out with respect to their surroundings to human observers. Saliency methods can be divided into methods that aim to identify the salient object (or objects) and methods that aim to produce a saliency map that is in accordance to measurements of human eye-movements on the same image. Itti et al. [5] proposed one of the first computational saliency methods based on combining the saliency cues for color, orientation and luminance. Many works followed proposing a large variety of hand-crafted features for saliency [6, 7]. Similar as other fields in computer vision, computational saliency estimation has moved in recent years from hand-designed features to end-to-end learned deep features [8].

Saliency detection in human vision plays a role in the efficient extraction of information by placing the attention on those regions in the image that are most informative. However, the vast majority of saliency methods are not evaluated on their efficiency to improve object recognition but instead are evaluated on

the task of how accurate their generated saliency masks are. Given that saliency is only an intermediate step of the visual pipeline, evaluating the efficiency of saliency in terms of an improvement of the final task - here we consider fine-grained recognition - could be considered a more valuable evaluation. Therefore, in this paper we aim to evaluate the usefulness of saliency by directly evaluating its improvement on image classification.

Previous works have found that the incorporation of attention mechanisms in neural networks could be beneficial. This theory was subsequently extended to captioning methods where the attention highlights the part of the image that is currently being described by words. Similar to these methods we will incorporate a saliency model, which modulates the normal forward pipeline similarly as an attention model would, but now within the context of fine-grained image classification. Contrarily to these attention methods, we use a saliency network that is pretrained on the task of saliency estimation. Especially, we are interested in demonstrating its effectiveness in the case of scarce training data, a scenario where attending to the relevant information from the image can significantly reduce the danger of overfitting. The main underlying idea is that using saliency as an attention mechanism can help backpropagation to focus on the relevant image information; something which is especially important when only few training examples are available.

In this paper, we investigate to what extent saliency estimation can be exploited to improve the training of an object recognition model when scarce training data is available. For that purpose we design an image classification deep neural network that incorporates saliency information as input. This network processes the saliency map through a dedicated network branch and uses the resulting features to modulate the standard bottom-up visual features from the original image input. The main aim of the proposed method is to enable the effective training of a fine-grained recognition model with limited training samples and to improve the performance on the task, thereby alleviating the need to annotate a large dataset. We evaluate our method on different datasets and under different settings, achieving considerable performance improvements

when leveraging saliency data, especially when training data is scarce.

This paper is organized as follows. In section 2 we discuss the related work. Section 3 describes our method in detail, and we perform extensive experiments in section 4. Finally, section 5 presents the conclusions.

2. Related Work

Saliency estimation: The seminal work of Itti et al. [5] proposed one of the first biologically motivated computational models for saliency estimation. Their saliency map was inferred from multi-scale representations of color, orientation and intensity contrast. Saliency research was propelled further by the availability of large data sets which allowed for direct comparison of methods and enabled the use of data-driven methods based on machine learning algorithms. The question of whether saliency is important for object recognition has been raised in [9]. Using a biologically plausible mechanism, the authors demonstrated that indeed saliency (of a top-down nature) has a positive impact on classification. Besides object recognition, saliency has also been used for object tracking. In [10], the authors formulated discriminant tracking as a saliency problem and addressed it using a biologically inspired framework.

Recent methods in saliency are mostly based on deep learning networks. Initially, pretrained deep convolutional networks were used directly to extract features for saliency estimation. Afterwards, end-to-end networks that learn a mapping from the input image to the saliency map [8] were introduced. But like most previous work on saliency estimation, the main focus of these works is to estimate a saliency map, not how saliency could contribute in a object recognition pipeline. In this paper, we aim to investigate if saliency can improve the recognition of objects with deep neural networks.

Attention: The method proposed in this paper is partially based on insights gained from some recent work on attention in neural networks. In [11] the authors propose a method that incorporates attention branches within a feed-forward network for object classification. The attention map, which is repeated

for multiple layers in the network, learns to modulate the network features with an attention mechanism. Our saliency branch is similar to the proposed attention mechanism in [11]. In our work, however, we use a pretrained saliency branch that is optimized to return a saliency map in accordance with human vision. The fact that the network is pretrained is important because that allows it to be used even for object classification problems with very few training examples. In this case, the proposed method in [11] would probably fail because it would have to train additional parameters for the attention branch, which would be extremely challenging in the scarce data domain.

Zagoruyko and Komodakis [12] propose a method to train a student network from a teacher network. Their novelty with respect to earlier work is the usage of attention to guide the teaching of the student network. They construct a spatial attention map by considering the activations of an image in a teacher network, and mapping these activations to a single spatial attention map which reflects on what locations the hidden neuron activations were most prominent. They consider that this information is important and can help guide the training process once it is also transferred to the student network. They show that their approach significantly improves the learning of the student network. Concretely, they show that guiding the backpropagation of gradients by telling to what spatial coordinates to ‘attend’ can assist in the training process. Our paper supports this claim by showing that spatial guidance can help training, although within a different context as in our case we use saliency as attention map and train the network for a new task.

Fine-grained recognition: Most of the state of the art general object classification approaches [11, 13] have difficulties in the fine-grained recognition task, which is more challenging due to the fact that basic-level categories (e.g. different bird species or flowers) share similar shape and visual appearance. One reason for this could be attributed to the popular codebook-based image representation, often resulting in the loss of subtle image information that is critical for the fine-grained task.

Most current fine-grained recognition approaches operate on a two-stage pipeline, first detecting some object parts and then categorizing the objects using this information. The work of Huang et al. [14] first localizes a set of part keypoints, and then simultaneously processes part and object information to obtain highly descriptive representations. Mask-CNN [15] also aggregates descriptors for parts and objects simultaneously, but using pixel-level masks instead of keypoints. The main drawback of these models is the need of human annotation for the semantic parts in terms of keypoints or bounding boxes.

For this reason, several recent approaches perform fine-grained recognition without explicit part annotations. Some have attempted to detect semantic parts using co-segmentation, like in [16]. In [17], their framework first performs unsupervised part candidates discovery and global object discovery which are subsequently fed into a two-stream CNN in order to model jointly both the local and global features. Alternatively, [18] uses Fisher vectors for image representation and shows that larger codebooks are able to model subtle visual details without explicitly modeling parts, which leads to better classification accuracy compared to small codebooks. Regardless, most fine-grained approaches use the object ground-truth bounding box at test time, achieving a significantly lower performance when this information is not available. Moreover, automatically discovering discriminative parts might require large amounts of training images. Our approach is more general, as it only requires image level annotations at training time and could easily generalize to other recognition tasks.

Few-shot learning: Few-shot learning aims to create models for which very few labeled samples are available. Early work on this topic is attributed to Fei-Fei et al. [19], who showed that, taking advantage of previously learned categories, it is possible to learn new categories using one or very few samples per class. More recently, [20] proposed a conditional distance measure that takes into account how a particular appearance model varies with respect to every other model in a model database. The approach has been applied to one-shot gesture recognition. Nowadays, several deep learning-based approaches have

emerged to address the problem of few-shot learning. We can identify three main strategies. One family is based on metric learning. In [21], the authors propose a framework that trains a network to map a small labeled support set and an unlabeled example to its label. An extension of this idea is presented in Prototypical networks [22], but in this case each class in the support set has been substituted by a ‘prototype’ (computed as the mean of the samples in the corresponding class), to which each sample is compared.

A second family of approaches is based on meta-learning, i.e. learning a model that given a few training examples of a new task tries to quickly learn a learner model that solves this new task [23]. In [24], the authors propose an LSTM-based meta-learner that is trained to optimize a neural network classifier. The meta-learner captures both short-term knowledge within a task and long-term knowledge common among all the tasks.

Finally, the third family of approaches is based on data augmentation for data-starved classes. In [25], the authors propose a way to increase (“hallucinate”) the number of samples for the classes with limited data. Their method is based on the intuition that certain aspects of intra-class variation generalize across categories, like for instance pose transformations. In practice, for data-rich classes, they use a neural network to learn transformations between pairs of samples and this transformation is later on applied on the real samples from data-starved classes to generate synthetic ones, thus increasing the population of these classes. For the same purpose (i.e. data augmentation for data-starved classes), in [26] the authors propose an attributed-guided augmentation approach which learns a mapping that allows the creation of synthetic data by manipulating certain attributes of real data. Thus, the newly created data presents attributes based on user-defined criteria (values). Instead of performing the data augmentation in image space, they perform it in feature space. This idea is further extended in [27], where the authors use a deep encoder-decoder architecture to generate feature trajectories by exploiting the pose manifold in terms of pose and appearance.

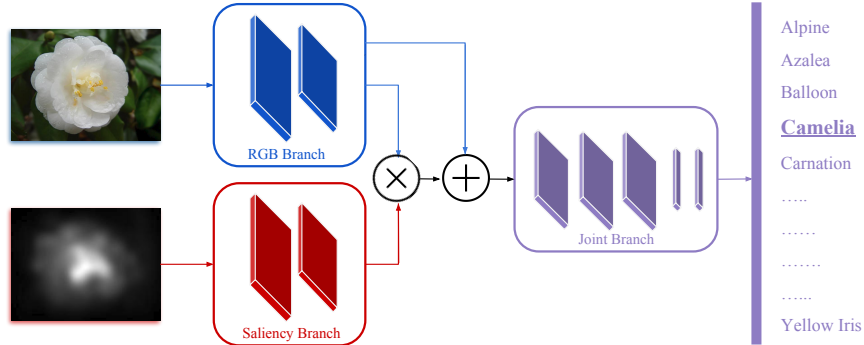


Figure 1: Overview of our fine-grained recognition model using saliency information. We process the two inputs, RGB and Saliency map, through two convolutional layers and then fuse the resulting features with a modulation layer. We then continue processing the fused features with three more convolutional layers and three fully connected layers, ending with the final classification layer.

3. Saliency Modulation for Scarce Data Object Classification

Image classification results have improved much since the advent of deep convolutional neural networks [13, 28] due to the excellent visual representations learned by these models. Given the great number of parameters of these networks, we require large datasets of labeled data to effectively train them. For example the popular ImageNet dataset has over 1M labeled images [29]. Once learned, these strong image representations can be transferred to other related tasks by a process called finetuning. This process allows to use deep learning on tasks for which significantly less labeled data is available. In some cases, however, the available data for the target task is so scarce that is still insufficient to finetune large networks and obtain satisfactory results.

Saliency is an attentional mechanism which allows humans to focus their limited resources to the most relevant information in the image. Since processing resources are limited, the data is processed in a serial manner, prioritizing those parts that are expected to have high information content. In this paper, we investigate another potential application of saliency, namely its function to facilitate the fast learning of new objects in the context of deep neural net-

works. Especially when only a few training examples are available, focusing on the relevant parts of the image could significantly improve the speed of learning, understanding speed as the number of example images required to learn a new class. Therefore, we seek to incorporate saliency estimation into an image classification pipeline, with the aim to decrease the data requirements for learning object categories.

Fig. 1 provides an overview of the proposed network architecture. Our network contains two branches: one to process the RGB images and one to process their corresponding saliency images, which are pre-computed and given as input. They are combined with a *modulation layer* (\times symbol) and further processed by several shared layers of the joint branch to finally end on a classification layer. Note how the RGB branch followed by the joint branch correspond to a standard image classification network. The novelty of our architecture is the introduction of the saliency branch, which transforms the saliency image to the *modulation image*. This modulation image is then used to modulate the features of the RGB branch, putting more emphasis on those features that are considered important for the fine-grained recognition task. In the following sections we provide the details of the network architecture, the functioning of the modulation layer, and the saliency methods used. We explain our model using AlexNet [13] as base classification network, but the theory could be applied to most convolutional neural network architectures. We also consider ResNet-50 and ResNet-152 [28] as base networks in our experiments (sec. 4.2).

3.1. Combining RGB with Saliency for Image Classification

Consider a saliency map $s(x, y)$ where x and y are the spatial coordinates. We will assume that saliency maps are of the same size as the original image $I(x, y, z)$, where $z = \{1, 2, 3\}$ indicate the three color channels of the image. A straightforward way to incorporate the saliency into the image classification network is by concatenating the image and the saliency map into an image with four channels such that $I(x, y, 4) = s(x, y)$. This strategy has been previously used by Murabito et al. [30] in a classification pipeline that combines two CNN

networks: one to compute top-down saliency maps from an RGB image, and a second network that appends the generated saliency map to the RGB image channels to perform image classification. In this case, the classification network only needs to train from scratch the weights of the first layer, the following layers can be initialized with a pretrained network. We call this approach *early fusion* of saliency and image content.

In this article we propose *delayed fusion* of saliency and image content, where we use the saliency map to modulate the features of an intermediate network layer. Consider the output of the i^{th} layer of the network, l^i , with dimension $w_i \times h_i \times z_i$. Then we define the modulation with a function $\hat{s}(x, y)$ as

$$\hat{l}^i(x, y, z) = l^i(x, y, z) \cdot \hat{s}(x, y), \quad (1)$$

yielding the saliency-modulated layer \hat{l}^i . Here the modulation image \hat{s} is the output of the saliency branch, which takes s as input (as depicted in Fig. 1). Note that we consider a single saliency map \hat{s} that is independent of the number of feature maps. To ensure that \hat{s} has the same spatial dimensions as l^i , we use a similar architecture for both the saliency branch and the RGB branch. Concretely, the main difference resides in the size of the channel dimension: the saliency branch takes an intensity image as input (instead of a 3-channel RGB image) and outputs a scalar modulation image of $w_i \times h_i \times 1$ (instead of a $w_i \times h_i \times c_i$ feature map). Moreover, we use a sigmoid activation function at the end of the saliency branch, as opposed to the ReLU non-linearity of the RGB branch. This ensures that $0 \leq \hat{s}(x, y) \leq 1$ and thus provides a suitable range for feature modulation.

In the original architecture, max pooling is performed right after the second convolutional layer. In our model, we postpone this max pooling to after the features from both branches are fused, i.e. we perform max pooling on the salience-modulated layer \hat{l}^i . The reasoning behind this choice is to leverage the greater modulation potential of higher resolution saliency features. We experimentally show (sec. 4.2) that this results in a small performance boost.

In addition to the formulation in Eq. (1) we also introduce a skip connection

from the RGB branch to the beginning of the joint branch, defined as

$$\hat{l}^i(x, y, z) = l^i(x, y, z) \cdot (\hat{s}(x, y) + 1). \quad (2)$$

This skip connection is depicted in Fig. 1 (+ symbol). It prevents the modulation layer from completely ignoring the features from the RGB branch. This is inspired by a previous work [12] that found this approach beneficial when using attention for network compression. We confirm the usefulness of the skip connection in the experiments section, sec. 4.2.

We train our architecture in an end-to-end manner. The backpropagated gradient from the modulation layer into the image classification branch is equal to

$$\frac{\partial L}{\partial l^i} = \frac{\partial L}{\partial \hat{l}^i} \cdot (s + 1), \quad (3)$$

where L is the loss function of the network. This shows that the saliency map not only modulates the forward pass (see Eq. (2)), but it also modulates the backward pass in exactly the same manner; in both cases putting more weight on the features that are on locations with high saliency, and putting less weight on the irrelevant features in the background on which the network could potentially overfit.

3.2. Training the Saliency Branch

The aim of the saliency branch is to process the saliency map $s(x, y)$ into effective modulation features $\hat{s}(x, y)$ that increase the classification performance when training with scarce data. The main intuition is that the saliency features \hat{s} will focus the backpropagated gradient to the relevant image features, thereby reducing the required data necessary to train the network. The additional saliency branch necessary to compute $\hat{s}(x, y)$ has its own set of parameters and could, in principle, increase the possibility of overfitting. We therefore consider two different scenarios to initialize this branch. In both cases, we start with an equivalent architecture to the one depicted in Fig. 1 but without the saliency branch. We pretrain this network for image classification on ImageNet [29]. Then, we add the saliency branch and apply either of the following options:

- *Initialization from scratch*: the weights of the saliency branch are randomly initialized using the Xavier method.
- *Initialization from pretrained*: the weights of the saliency branch are pretrained on an image classification network for which abundant training data is available. To do this, we first generate saliency images for the ImageNet validation dataset, which consists of 50K images (40K for training and 10K for validation) using the saliency method of choice. On this dataset we train our method, initializing the saliency branch from scratch. We now have a good pretrained model for the saliency branch too. Finally, we use this pretrained network (using both the saliency and RGB branch) to initialize all the weights of our network except the top classification layer.

3.3. Saliency input

The input to the saliency branch is a saliency map. Among the many saliency methods that provide satisfactory results [31], we perform most of our experiments using two of the top performing methods:

- iSEEL [32] leverages the inter-image similarities to train an ensemble of extreme learners. The predicted saliency of the input image is then calculated as the ensemble’s mean saliency value. Their approach is based on two aspects: (i) the contextual information of the scene and (ii) the influence of scene memorability (in terms of eye movement patterns by resemblance with past experiences). We use MATLAB code released by the authors.
- SALICON [33] exploits the power of high-level semantics encoded in a CNN pretrained on ImageNet. Their approach represents a breakthrough in saliency prediction, by reducing the semantic gap between the computational model and the human perception. Their method has two key elements: (i) an objective function based on saliency evaluation metrics

and (ii) integration of information at different image scales. We use the open source implementation provided by [34].

Besides these two methods, we also perform experiments with three other approaches for a more comprehensive comparison.

- Itti and Koch [5]: First, we consider the classical saliency model of Itti et al. Several activation maps, corresponding to multiscale image features (color, intensity and orientations) are generated from the visual input and combined into a single topographical saliency map. A neural network is used to select the most salient locations in order of decreasing magnitude, which could be subsequently analyzed by more complex, higher cognitive level processes.
- GBVS [35]: The Graph-based Visual Saliency (GBVS) is also a biologically-plausible bottom-up model following the approach proposed earlier by Itti et al., but improving the performance of the generation of activation maps and the normalization/combination step. They used the Markovian formalism to describe the dissimilarity and concentration of salient locations of the image seen as a graph.
- BMS [36]: Boolean Map based Saliency (BMS) approach computes saliency by analyzing the topological structure of the Boolean maps. These maps are generated by randomly thresholding the color channels. As topological element they choose ‘surroundedness’ because it better characterizes the image/background segregation.

Figure 2 depicts the estimated saliency maps for an example image using the five different saliency methods presented above. In addition to these methods, we consider two additional saliency map baselines. *White* regards all image pixels as equally salient, and thus the saliency maps are uniformly white. On the other hand, *Center* emulates a center prior by representing saliency as a centered 2-dimensional Gaussian distribution. These two baselines allow us to determine whether our model is actually leveraging the saliency information contained

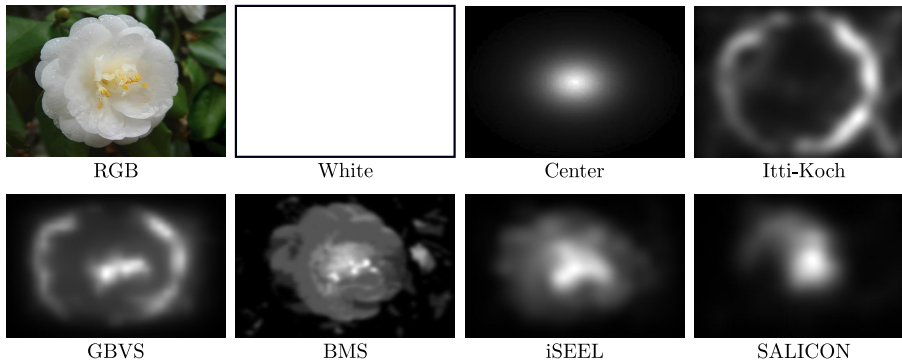


Figure 2: Saliency images generated with the different saliency estimation approaches considered, as well as the two baseline saliency maps evaluated, White and Center. We also include the original RGB image for reference.

in the maps, or it is simply adding a general image bias that is beneficial for recognition (e.g. center bias). We are especially interested in assessing whether saliency methods that obtain higher performance on saliency benchmarks also yield better performance when incorporated into our saliency pipeline.

4. Experiments

4.1. Experimental Setup

Datasets. We have performed the evaluation of our approach on four standard datasets used for fine-grained classification

- *Flowers*: Oxford Flower 102 dataset [1] consists of 8189 images of flowers grouped in 102 classes. Each class contains between 40 and 258 images.
- *Birds*: is a dataset consisting of 11,788 images of bird species divided in 200 categories [2]. Each image is annotated with its bounding box and the image coordinates of 15 keypoints. However, in our experiments we used the whole image.
- *Cars*: the dataset in [4] contains 16,185 images of 196 classes of cars. The data is split into 8,144 training images and 8,041 testing images, where each class has been separated roughly in a 50-50 split.

- *Dogs*: Stanford Dogs [3] consists of 20,580 images of different breeds of dogs from around the world grouped in 120 categories. Since some of these images appear also in Imagenet, in our experiments we have discarded the repeated ones.

Networks. Our base network is AlexNet [13], which consists of five convolutional layers followed by three fully connected layers. We used the pretrained network on ImageNet [29] and fine-tuned it for fine-grained recognition on each dataset for 70 epochs with a learning rate of 0.01 and a weight decay of 0.003. The top classification layer is randomly initialized using Xavier. We have attached a saliency branch to this network as shown in Figure 1.

For some experiments we have also used the ResNet-50 and ResNet-152 [28], consisting of 50 and 152 convolutional layers, respectively, organized in 5 residual blocks. The structure of the saliency branch has been kept the same as in Figure 1, i.e. consisting of two convolutional layers and having a ReLu activation function after the first one and a sigmoid function after the second.

Evaluation protocol. For all the above datasets, we randomly select and fix 5 images for test, 5 for validation, and keep the rest for training. We do this for each class in the dataset independently. In order to investigate different data scarcity levels, we train each model with subsets of k training images for $k \in \{1, 2, 3, 5, 10, 15, 20, 25, 30, K\}$, where K is the total number of training images for the class, which does not include the 10 held out images for validation and test. Contrarily to current few-shot approaches, this setting grants us a more complete disclosure of the results of our model under multiple limited-data scenarios. We use accuracy in terms of percentage of correctly classified samples as evaluation measure. We train and test each model five times with different random initializations, and show the average performance for the five runs.

4.2. Experimental Results

In the experimental section we evaluate the best strategies for fusing the saliency and RGB branches, compare several network architectures, evaluate various saliency methods as input to the saliency branch, and compare our results with state of the art on standard benchmark datasets for fine-grained object recognition.

Optimal architecture. In order to justify the design choices in our model, we present here multiple architectural variations to integrate saliency information into a neural network. We call *Baseline-RGB* to the original network model, which only contains the RGB branch and thus does not use any saliency information. We test an *Early fusion* model in which the saliency image is directly concatenated to the RGB input.

We consider several variants of our model in which delayed fusion is performed at different network levels, indicated as Fusion L1 for fusion after layer 1 (similarly for Fusion L2, L3, L4, and L5). In all cases, we use a two-layer saliency branch, indicated by S2. Moreover, we evaluate whether performing the fusion after the pooling layer is a better option than doing it before. Finally, we include a model without the skip connection from the RGB branch to the joint branch (No SC).

We evaluate all models on *Flowers* [1] with AlexNet [13] and using iSEEL [32] as the saliency method of choice. Table 1 shows the results for different number of training images. First, we observe how the performance of all methods steadily grows when increasing the number of training images. In general, incorporating saliency information helps when fused within the network, but damages the accuracy if concatenated to the input image. We attribute this to the need to learn a low-level filter from scratch, which in turn affects the feature representation at higher levels. Performing the fusion immediately after the second convolutional layer seems to be the best option. Fusing before or after the pooling layer leads to similar results, the advantage of fusing higher resolution saliency features gives only a marginal boost. Finally, the skip connection from

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB	31.8	45.8	53.1	63.6	72.4	76.9	81.2	85.1	87.2	88.0	68.5
Early Fusion	19.3	25.7	30.1	40.8	60.9	69.2	75.3	79.9	82.4	83.7	56.7
Fusion L1-S2	33.3	47.9	54.3	65.1	71.9	76.3	82.1	85.9	87.9	90.7	69.5
Fusion L2-S2	34.7	49.3	55.2	65.2	72.7	76.7	83.9	86.5	89.1	91.3	70.5
Fusion L3-S2	32.9	46.7	54.1	64.9	71.7	74.4	82.3	85.1	87.3	89.1	68.9
Fusion L4-S2	32.5	48.9	54.0	65.1	71.7	73.5	81.0	84.9	87.2	88.8	68.2
Fusion L5-S2	32.5	48.9	54.0	63.3	71.1	73.3	81.0	84.3	87.2	88.7	68.4
Fusion L2-S2 + After pool	34.3	49.1	55.5	66.0	72.1	77.5	83.6	85.6	88.9	90.2	70.2
Fusion L2-S2 + No SC	33.9	48.1	55.1	65.1	71.1	77.6	82.4	86.3	88.1	90.9	69.9

Table 1: Results for the baseline model and different variations of our architecture incorporating saliency information. The results correspond to the classification accuracy on the Flowers dataset [1] with AlexNet [13]. Each column indicates the number of training images used, and the rightmost column shows the average

the RGB branch to the joint branch is also beneficial.

We have also explored different architectures for the saliency branch. We first assess whether an additional convolutional layer in the saliency branch leads to better performance. Table 2 presents the comparison between a two-layer saliency branch (S2) and a three-layer version (S3). For completeness, we explore merging after the second layer of the RGB branch (L2) as in previous experiments, and merging after the third layer (L3). We observe how an extra layer does not further improve the model’s performance. Alternatively, we investigate whether having fewer parameters in the saliency branch achieves higher results. We evaluate with 75% and 50% fewer parameters by reducing the number of output channels in the first layer. Table 3 shows how reducing the number of parameters in the saliency branch slightly reduces the final performance.

Pretraining the saliency branch on ImageNet. As described in section 3, we consider two alternative ways of initializing the saliency branch: from scratch and pretrained on ImageNet [29]. In this section, we compare these two approaches with respect to the Baseline-RGB. The experiments are performed on *Flowers* dataset (see Figure 3a) and represent the classification accuracy ver-

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB	31.8	45.8	53.1	63.6	72.4	76.9	81.2	85.1	87.2	88.0	68.5
Fusion L2-S2	34.7	49.3	55.2	65.2	72.7	76.7	83.9	86.5	89.1	91.3	70.5
Fusion L3-S2	32.9	46.7	54.1	64.9	71.7	74.4	82.3	85.1	87.3	89.1	68.9
Fusion L2-S3	34.5	48.2	55.9	65.0	72.8	76.1	83.0	86.5	89.0	91.0	70.2
Fusion L3-S3	33.1	49.3	54.2	65.1	72.1	74.9	82.9	85.3	88.0	89.0	69.4

Table 2: Results on Flowers [1] with AlexNet [13] using two (S2) or three (S3) convolutional layers for the saliency branch.

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB	31.8	45.8	53.1	63.6	72.4	76.9	81.2	85.1	87.2	88.0	68.5
Fusion L2-S2 (100%)	34.7	49.3	55.2	65.2	72.7	76.7	83.9	86.5	89.1	91.3	70.5
Fusion L2-S2 (75%)	34.7	49.0	55.3	65.1	72.0	77.0	83.3	85.9	88.3	89.1	70.0
Fusion L2-S2 (50%)	34.7	49.1	55.9	65.1	71.8	77.1	83.5	86.2	88.0	89.0	70.0

Table 3: Results on Flowers [1] with AlexNet [13] when reducing the number of parameters of the saliency branch.

sus the number of training samples. Adding a saliency branch initialized from scratch already outperforms the baseline using only RGB (see also Tab. 1), and pretraining this branch with ImageNet further increases the performance in a systematic and substantial manner. Our method with pretraining is especially advantageous in the scarce-data domain (i.e. < 20 images per class). For example, we obtain a better performance than the baseline using half the data, 10 images/class vs. 20 images/class, respectively. Furthermore, in the very low-range of number of samples we obtain similar performance with only *one third* of the samples (3 images/class vs. 10 images/class). Finally, our saliency branch is still beneficial even when using all available training samples. In fact, our method trained with a limited number of samples (around 25 per class) already surpasses the final performance of baseline using all samples.

Figure 4 shows some qualitative results for the case when the pretrained version of our approach predicts the correct label, meanwhile the Baseline-RGB fails. Alternatively, figure 5 depicts the opposite case: the Baseline-RGB predicts the correct label of the test images, meanwhile the pretrained version of

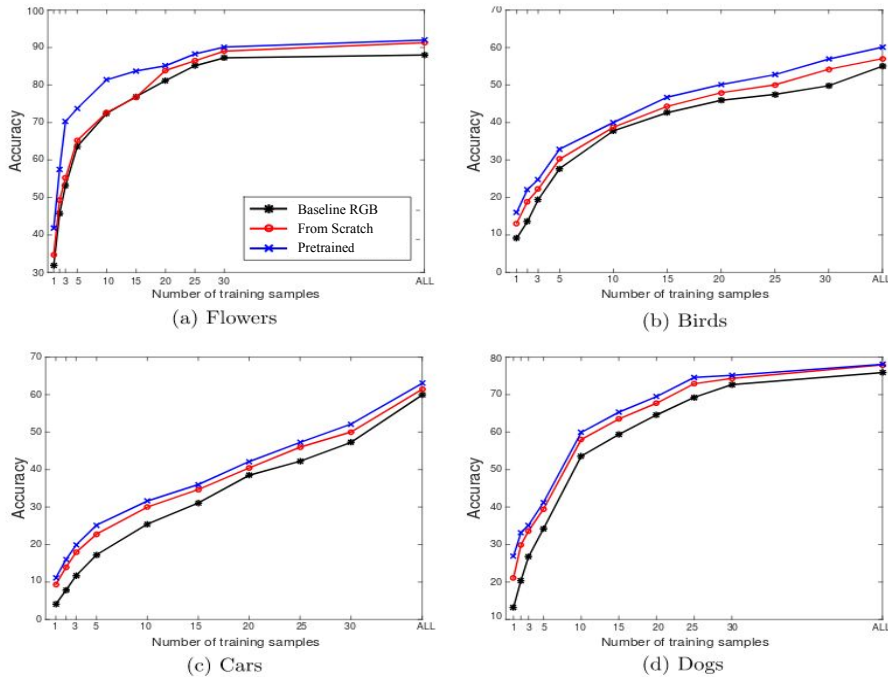


Figure 3: Experiments on four datasets using iSEEL [32] to generate the saliency maps. *Baseline-RGB* is compared against two different ways to initialize the saliency branch of our model: from scratch and pretrained on ImageNet [29].

our approach fails. In both cases, the saliency images have been generated using the iSEEL method. A possible explanation for the failures in this latter case could be that the saliency images are not able to capture the relevant region of the image for fine-grained discrimination. Thus, the salience-modulated layer focuses on the wrong features for the task.

Different datasets. Besides *Flowers* dataset, we validate our approach on three other datasets: *Birds*, *Cars* and *Dogs* (see figures 3b, c, and d, respectively). We follow the same experimental protocol as in the *Flowers* case. We can see how most trends observed in *Flowers* also apply to these datasets. For example, incorporating saliency information improves the classification accuracy, especially when data is scarce. Moreover, pretraining the saliency branch

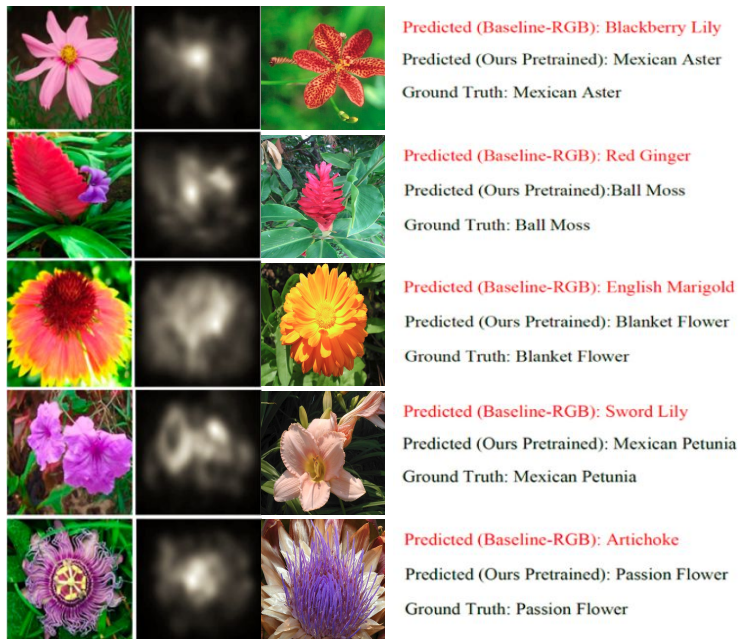


Figure 4: Some success examples on Flowers [1]: when the prediction done by Baseline-RGB fails to infer the right label for some test images, but the prediction by our approach is correct. From left to right: input image, saliency images generated with iSEEL [32], example image of the class with which the input image was wrongly predicted.

is beneficial for our method and leads to a further performance boost. Even when using all available samples, our method outperforms the baseline model. Therefore, we can claim that our approach successfully generalizes to other fine-grained datasets.

Confirmation of intuition. Our method is based on the idea that adding a saliency branch helps the network to focus on the relevant image regions during the training. To verify that this is actually happening we propose the following experiment: we measure if the percentage of backpropagated gradient magnitude which passes through the relevant image regions is increased by our proposed network architecture. We perform this experiment on the Birds dataset for which we have access to bounding box information of the birds (defining the relevant region). We measure the percentage of backpropagated gradient

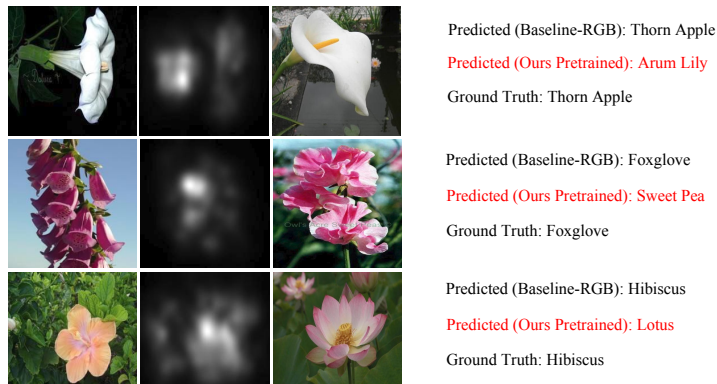


Figure 5: Some failure examples on Flowers [1]: when the prediction done by our method fails to infer the right label for some test images, but the prediction by Baseline-RGB is correct. From left to right: input image, saliency images generated with iSEEL [32], example image of the class with which the input image was wrongly predicted.

energy which is in the bounding box of the bird (this is computed by dividing the gradient magnitude in the bounding box by the gradient energy in the whole image). We measure this just before the third convolutional layer for AlexNet (which is just before the joint branch in Figure 1), and we measure this for both the network with and without saliency branch.

The results are presented in Figure 6. The results show that the percentage of backpropagated gradient that passes through the relevant image regions is higher for our approach. As expected it is even higher for the network with the pretrained saliency branch. However the gap with the network trained from scratch diminishes with the number of epochs. The fact that more backpropagated gradient energy goes through the relevant image regions may explain why our method obtains better results than the standard baseline method.

Different saliency methods. Table 4 presents results on the *Flowers* using our full AlexNet model combined with the different input saliency maps. We can observe how, instead of helping, the two saliency baselines are actually hurting the method performance with respect to the Baseline-RGB. We hypothesize that this is due to the noise introduced in the network’s internal

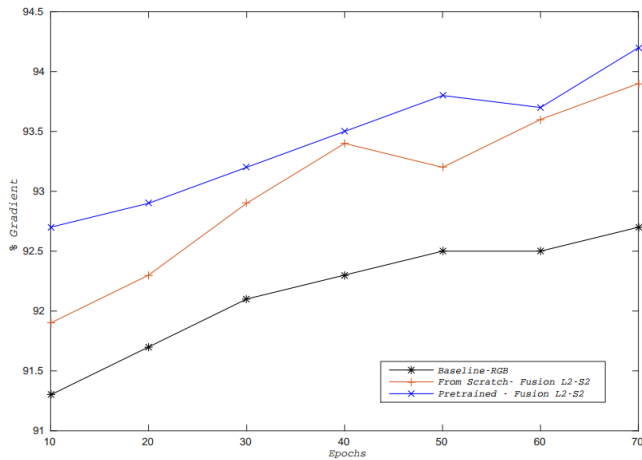


Figure 6: Average percentage of the total backpropagated gradient energy per epoch that is inside the bird bounding box. The graph shows that for our approach significantly more backpropagated gradient is on the relevant image region (for both the version trained from scratch and the version with pretrained saliency branch).

representation when the input saliency map is independent of the input image. On the other hand, all the saliency estimation methods increase the method performance, especially in the scarce-data range (i.e. < 10 images). Moreover, better saliency methods (e.g. iSEEL and SALICON) result in higher accuracies. In order to experimentally confirm this observation, we show in Fig. 7 the accuracy of our image classification model as a function of the saliency estimation performance of the corresponding method. We measure saliency estimation performance in terms of Normalized Scanpath Saliency (NSS), which is the official measure currently used by the popular MIT saliency benchmark [31] to sort all the participating methods. There is indeed a clear linear correlation, supported quantitatively by a Pearson product-moment correlation coefficient of 0.95. Therefore, we conclude that our model is agnostic to the saliency method employed. More importantly, it shows that better saliency methods (evaluated based on saliency estimation) actually lead to better image classification performance once integrated into an object recognition pipeline. This observation

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB	31.8	45.8	53.1	63.6	72.4	76.9	81.2	85.1	87.2	88.0	68.5
Baseline-White	23.1	29.7	37.2	55.1	66.9	73	82.5	84.8	86.6	87.9	62.7
Baseline-Center	24.3	30.3	39.2	55.7	68.3	74.1	82.7	84.5	86.8	87.8	63.4
Itti-Koch [5]	32.8	46.8	53.9	64.0	72.9	77.1	82.9	85.4	87.1	88.3	69.1
GBVS [35]	33.3	46.9	54.0	64.1	73.0	77.3	83.1	85.7	87.5	88.8	69.4
BMS [36]	34.2	47.3	54.9	64.8	73.3	77.8	83.4	86.1	88.1	90.1	70.0
iSEEL [32]	34.7	49.3	55.2	65.2	72.7	76.7	83.9	86.5	89.1	91.3	70.5
SALICON [37]	37.6	51.9	57.1	68.5	75.2	79.7	84.9	88.2	91.2	92.4	72.7

Table 4: Comparison of different saliency methods regarding the effect on our model. The results correspond to the classification accuracy on the Flowers dataset [1] when using our full model with AlexNet [13] as base network. Each column indicates the number of training images used, and the rightmost column shows the average.

can be a motivation for saliency research: it not only leads to better saliency estimation but indirectly also contributes to improved object recognition.

Different base networks. In order to evaluate the generality of our approach across different base networks, we have considered ResNet-50 and ResNet-152 as alternatives to AlexNet. We have tested several possible fusion architectures (Tables 5 and 6), but the optimal performance has been obtained when the fusion between the RGB and saliency branches takes place after the fourth residual block, with a two-layer saliency branch (Block4-S2). Results in Table 7 show the classification accuracy achieved on *Flowers* when using ResNet-50 and ResNet-152 with SALICON saliency maps. Furthermore, we compared our two initialization methods for the saliency branch (from scratch and pretrained on ImageNet) against the Baseline-RGB. Although under both initializations we obtained higher accuracy, the one that performs the best is the pretrained. These results confirm the trend already observed for AlexNet regarding the benefits of pretraining the saliency branch as shown in Fig. 3.

Comparison with standard dataset splits. All previous experiments use a custom data split consisting of a fixed test set of 5 images and a varying number of training images. In order to enable comparisons with published results by

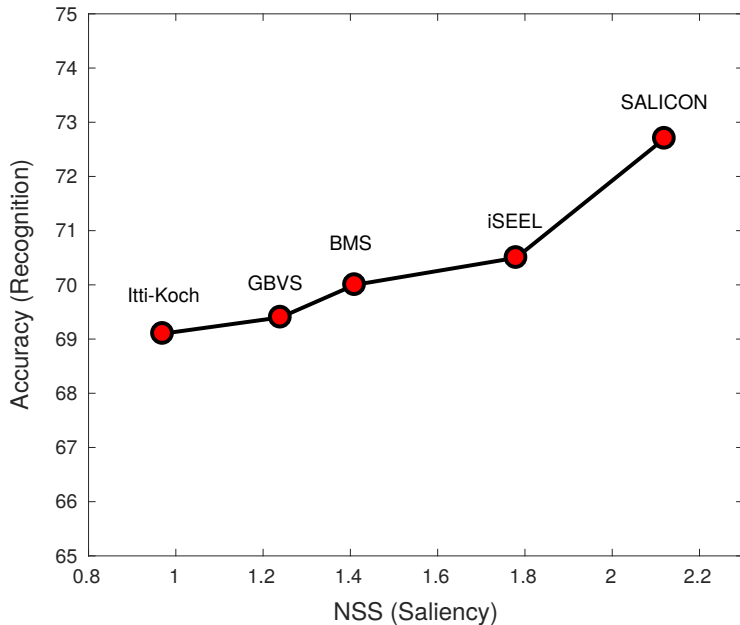


Figure 7: Correlation between the performance of the saliency method in terms of NSS and the fine-grained recognition accuracy of our method using the corresponding saliency model. Results with AlexNet [13] on Flowers [1].

other methods, we perform here experiments using the standard data split of each dataset, employing the entirety of the corresponding given sets for training and evaluation. Table 8 presents results for our approach and several state of the art fine-grained recognition approaches for Flowers, Birds, and Cars datasets. We discard Dogs dataset due to the overlap with the ImageNet images already used for pretraining the network, as they can no longer be ignored when using the full sets. Our approach uses SALICON saliency and ResNet152 as base network, which is equivalent to the networks used by the most recent works. Our method is competitive with specialized fine-grained approaches, despite the more sophisticated techniques included in those (e.g. part localization), some of which might be complementary to our saliency modulation. Moreover, our approach is especially beneficial in the scarce training data regime, whereas some of the state of the art methods may not work under these conditions.

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB	39.1	59.6	67.8	81.6	89.0	91.7	92.7	93.0	93.0	95.4	80.3
Block1-S2	38.0	59.2	68.0	80.7	88.8	91.0	91.9	92.0	92.1	94.8	79.6
Block2-S2	38.2	59.5	68.0	81.4	90.0	91.6	92.0	92.4	93.0	94.9	80.1
Block3-S2	39.3	62.9	68.5	83.0	90.0	92.1	93.5	94.9	93.4	95.9	81.4
Block4-S2	45.8	64.3	72.8	83.0	90.5	93.0	93.9	94.6	93.7	96.7	82.7
Block5-S2	38.2	57.9	65.9	80.8	87.1	90.9	91.1	91.2	91.9	92.6	78.8

Table 5: Results for the baseline model and different variations of our architecture incorporating saliency information in different blocks. The results correspond to the classification accuracy on the Flowers dataset [1] with ResNet-50 [28]. Each column indicates the number of training images used, and the rightmost column shows the average

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB	39.0	60.1	68.0	82.5	89.0	92.0	92.1	93.3	94.2	95.8	80.6
Block1-S2	39.0	59.9	68.0	82.1	88.6	91.9	92.2	93.0	94.2	95.1	80.4
Block2-S2	38.8	60.2	68.2	83.0	90.2	92.2	93.0	94.0	94.0	96.2	81.0
Block3-S2	43.0	63.7	68.9	83.1	90.2	92.1	93.1	94.3	96.1	96.3	82.1
Block4-S2	42.6	64.2	70.9	85.5	90.9	92.7	94.0	95.0	97.0	97.8	83.1
Block5-S2	39.0	58.0	65.8	80.3	87.1	90.8	91.5	92.0	92.3	92.7	79.0

Table 6: Results for the baseline model and different variations of our architecture incorporating saliency information in different blocks. The results correspond to the classification accuracy on the Flowers dataset [1] with ResNet-152 [28]. Each column indicates the number of training images used, and the rightmost column shows the average

Comparison with few-shot method. Our scarce-data approach is similar in spirit to the few-shot learning methods [21, 22, 23]. For this reason, we propose here a comparison with the state of the art method for few-shot classification, Prototypical networks [22]. In the standard few-shot protocol, the task is framed as N -way k -shot, i.e. provide each time a set of k labeled samples from each of N classes that have not previously been trained upon. The goal is then to classify a disjoint batch of unlabeled samples, known as ‘queries’, into one of these N classes. Therefore, some classes are used to train the few-shot method, while others are only used at test time. In our case, we do not require

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB Resnet-50	39.1	59.6	67.8	81.6	89.0	91.7	92.7	93.0	93.0	95.4	80.3
Resnet-50 Block4-S2 From Scratch	45.8	64.1	71.8	83.0	90.5	93.0	93.9	94.6	93.7	96.7	82.7
Resnet-50 Block4-S2 Pretrained	47.1	65.2	72.9	83.8	91.3	93.9	94.6	95.4	94.7	97.4	83.6
Baseline-RGB Resnet-152	39.0	60.1	68.0	82.5	89.0	92.0	92.1	93.3	94.2	95.8	80.6
Resnet-152 Block4-S2 From Scratch	42.6	64.2	70.9	85.5	90.9	92.7	94.0	95.0	97.0	97.8	83.1
Resnet-152 Block4-S2 Pretrained	46.9	65.5	73.0	84.7	92.0	94.2	95.3	95.8	97.3	98.1	84.3

Table 7: Results on Flowers [1] using ResNet-50 and Resnet-152 [28] as base networks and SALICON [37] as saliency method.

Method	Flowers	Birds	Cars
Krause et al. [16]	-	82.0	92.6
RA-CNN [38]	-	85.3	92.5
Bilinear-CNN [39]	-	84.1	91.3
Compact Bilinear Pooling [40]	-	84.3	91.2
Low-rank Bilinear Pooling [41]	-	84.2	90.9
Cui et al. (with Imagenet) [42]	96.3	82.8	91.3
MA-CNN [43]	-	86.5	92.8
Ge-Yu [44]	90.3	-	-
DLA [45]	-	85.1	94.1
Ours (Resnet152 Block4-S2 From Scratch)	96.4	85.6	92.1
Ours (Resnet152 Block4-S2 Pretrained)	97.8	86.1	92.4

Table 8: Comparison with state of the art methods for domain-specific fine-grained recognition using the standard data splits of Flowers [1], Birds [2] and Cars [4]. Our approach uses ResNet-152 [28] as base network and SALICON [37] saliency maps.

such split, as we can train and test the model in all classes simultaneously. Moreover, their test episodes are composed of only N classes at a time, where N is generally a small number (e.g. below 20). Contrarily, we follow a more general classification approach and test on all classes simultaneously, which is inherently more challenging as the misclassification probability increases.

We propose two different scenarios to compare our method to Prototypical networks on the task of Flower [1] classification. The first, *20-way 5-shot*, closely resembles the setting introduced by [21] and usually employed by few-shot ap-

Method	20-way 5-shot	102-way 5-shot
Prototypical networks [22]	53.8	26.2
Ours	81.0	73.8

Table 9: Results for few-shot classification on Flowers [1] when using our full model with AlexNet [13] as base network.

proaches. We split the set of classes in train and test, selecting 20 random classes for the testing phase. Then, we run Prototypical networks for the 20-way 5-shot classification task, following similar settings to those used in the mini-ImageNet experiment of [22]. We train until convergence using 100 training episodes and test using 5 episodes, with 5 queries per episode both during training and testing. The second scenario, *102-way 5-shot*, is more similar to the conventional classification task, in which all classes are used for training and testing. We maintain the training settings for this case, but remove from the ‘shot’ set those queries used at test time. Table 9 presents the results of these experiments. Our method leads to substantially superior performance in both cases, but the difference is especially remarkable for the 102-way setting. This demonstrates the limitations of this type of few-shot approaches when scaling to many classes, even when they are trained with the same set of classes used for test.

5. Conclusions

In this paper, we investigated the role of saliency in improving the classification accuracy of a CNN when the available training data is scarce. For that purpose we have considered adding a saliency branch to an existing CNN architecture, which is used to modulate the standard bottom-up visual features from the original input image. We have shown that the proposed approach leads to an improvement of the recognition accuracy with limited number of training data, when applied to the task of fine-grained object recognition.

Extensive evaluation has been performed on several datasets and under different settings, demonstrating the usefulness of saliency for fine-grained object

recognition, especially for the case of scarce training data. In addition, our approach allows to compare saliency methods on the high-level task of fine-grained object recognition. Traditionally, saliency methods are evaluated on their ability to generate saliency maps that indicate the relative relevance of regions for the human visual system. However, it remained unclear if these saliency methods would actually translate into improved high-level vision results for tasks such as object recognition. Our experiments show that there exists a clear correlation (Pearson product-moment correlation coefficient of 0.95) between the performance of saliency methods on standard saliency benchmarks and the performance gain that is obtained when incorporating them in a object recognition pipeline. Future work will be devoted to extend the current framework by proposing an end-to-end deep architecture that estimates automatically the saliency map, thus eliminating the need for pre-computing it off-line.

Acknowledgements

This work is partially funded by MINECO grant TIN2016-79717-R, Spain. Carola Figuroa is supported by a Ph.D. scholarship from CONICYT, Chile. We acknowledge the CERCA Programme of Generalitat de Catalunya. We also acknowledge the generous GPU support from NVIDIA.

References

References

- [1] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: Sixth Indian Conference on Computer Vision, Graphics & Image Processing, 2008, pp. 722–729.
- [2] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, P. Perona, Caltech-UCSD Birds 200 (2010) [Last accessed: July 10th, 2018]. URL <http://www.vision.caltech.edu/visipedia/CUB-200.html>

- [3] A. Khosla, N. Jayadevaprakash, B. Yao, L. Fei-Fei, Novel dataset for fine-grained image categorization, in: First Workshop on Fine-Grained Visual Categorization at CVPR, 2011.
- [4] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3d object representations for fine-grained categorization, in: 4th IEEE Workshop on 3D Representation and Recognition, at ICCV, 2013, pp. 1–8.
- [5] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (11) (1998) 1254–1259.
- [6] S. Ramanathan, H. Katti, N. Sebe, M. Kankanhalli, T.-S. Chua, *An Eye Fixation Database for Saliency Detection in Images*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 30–43.
- [7] A. Borji, D. N. Sihite, L. Itti, What/where to look next? modeling top-down visual attention in complex interactive environments, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44 (5) (2014) 523–538.
- [8] X. Li, L. Zhao, L. Wei, M. H. Yang, F. Wu, Y. Zhuang, H. Ling, J. Wang, Deepsaliency: Multi-task deep neural network model for salient object detection, *IEEE Transactions on Image Processing* 25 (8) (2016) 3919–3930.
- [9] S. Han, N. Vasconcelos, Biologically plausible saliency mechanisms improve feedforward object recognition, *Vision Research* 50 (2010) 2295–2307.
- [10] V. Mahadevan, N. Vasconcelos, Saliency-based discriminant tracking, in: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1007–1013.
- [11] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual attention network for image classification, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3156–3164.

- [12] S. Zagoruyko, N. Komodakis, Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, in: International Conference on Learning Representations, 2017.
- [13] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
- [14] S. Huang, Z. Xu, D. Tao, Y. Zhang, Part-stacked cnn for fine-grained visual categorization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1173–1182.
- [15] X.-S. Wei, C.-W. Xie, J. Wu, C. Shen, Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization, Pattern Recognition 76 (2018) 704 – 714.
- [16] J. Krause, H. Jin, J. Yang, L. Fei-Fei, Fine-grained recognition without part annotations, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5546–5555.
- [17] G.-S. Xie, X.-Y. Zhang, W. Yang, M. Xu, S. Yan, C.-L. Liu, Lg-cnn: From local parts to global discrimination for fine-grained recognition, Pattern Recognition 71 (2017) 118–131.
- [18] P.-H. Gosselin, N. Murray, H. Jégou, F. Perronnin, Revisiting the fisher vector for fine-grained classification, Pattern Recognition Letters 49 (2014) 92–98.
- [19] L. Fei-Fei, R. Fergus, P. Perona, One-shot learning of object categories, IEEE Transactions on Pattern Analysis Machine Intelligence 28 (4) (2006) 594–611.
- [20] R. Krishnan, S. Sarkar, Conditional distance based matching for one-shot gesture recognition, Pattern Recognition 48 (4) (2015) 1302 – 1314.

- [21] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, D. Wierstra, Matching networks for one shot learning, in: *Advances in Neural Information Processing Systems*, 2017.
- [22] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.
- [23] T. Munkhdalai, H. Yu, Meta networks, in: *International Conference on Machine Learning*, 2017, pp. 2554–2563.
- [24] S. Ravi, H. Larochelle, Optimization as a model for few-shot learning, in: *International Conference on Learning Representations*, 2017.
- [25] B. Hariharan, R. Girshick, Low-shot visual recognition by shrinking and hallucinating features, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3018–3027.
- [26] M. Dixit, R. Kwitt, M. Niethammer, N. Vasconcelos, Aga: Attribute-guided augmentation, in: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3328–3336.
- [27] B. Liu, X. Wang, M. Dixit, R. Kwitt, N. Vasconcelos, Feature space transfer for data augmentation, in: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9090–9098.
- [28] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, *International Journal of Computer Vision* 115 (3) (2015) 211–252.

- [30] F. Murabito, C. Spampinato, S. Palazzo, D. Giordano, K. Pogorelov, M. Riegler, Top-down saliency detection driven by visual classification (2018) [Last accessed: July 10th, 2018].
URL <https://doi.org/10.1016/j.cviu.2018.03.005>
- [31] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, A. Torralba, Mit saliency benchmark [Last accessed: July 10th, 2018].
URL <http://saliency.mit.edu/>
- [32] H. R. Tavakoli, A. Borji, J. Laaksonen, E. Rahtu, Exploiting inter-image similarity and ensemble of extreme learners for fixation prediction using deep features, *Neurocomputing* 244 (2017) 10–18.
- [33] X. Huang, C. Shen, X. Boix, Q. Zhao, Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks, in: *IEEE International Conference on Computer Vision*, 2015, pp. 262–270.
- [34] C. L. Thomas, Opensalicon: An open source implementation of the salicon saliency model, Tech. Rep. TR-2016-02, University of Pittsburgh (2016) [Last accessed: July 10th, 2018].
URL <https://arxiv.org/pdf/1606.00110.pdf>
- [35] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, in: *Advances in Neural Information Processing Systems*, 2006.
- [36] J. Zhang, S. Sclaroff, Saliency detection: A boolean map approach, in: *IEEE International Conference on Computer Vision*, 2013, pp. 153–160.
- [37] M. Jiang, S. Huang, J. Duan, Q. Zhao, Salicon: Saliency in context, in: *Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1072–1080.
- [38] J. Fu, H. Zheng, T. Mei, Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4476–4484. doi:10.1109/CVPR.2017.476.

- [39] T.-Y. Lin, A. RoyChowdhury, S. Maji, Bilinear cnn models for fine-grained visual recognition, in: International Conference on Computer Vision (ICCV), 2015.
- [40] Y. Gao, O. Beijbom, N. Zhang, T. Darrell, Compact bilinear pooling, CoRR abs/1511.06062. arXiv:1511.06062.
URL <http://arxiv.org/abs/1511.06062>
- [41] S. Kong, C. C. Fowlkes, Low-rank bilinear pooling for fine-grained classification, CoRR abs/1611.05109. arXiv:1611.05109.
URL <http://arxiv.org/abs/1611.05109>
- [42] Y. Cui, Y. Song, C. Sun, A. Howard, S. J. Belongie, Large scale fine-grained categorization and domain-specific transfer learning, CoRR abs/1806.06193. arXiv:1806.06193.
URL <http://arxiv.org/abs/1806.06193>
- [43] H. Zheng, J. Fu, T. Mei, J. Luo, Learning multi-attention convolutional neural network for fine-grained image recognition, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5219–5227. doi:10.1109/ICCV.2017.557.
- [44] W. Ge, Y. Yu, Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning, CoRR abs/1702.08690. arXiv:1702.08690.
URL <http://arxiv.org/abs/1702.08690>
- [45] F. Yu, D. Wang, T. Darrell, Deep layer aggregation, CoRR abs/1707.06484. arXiv:1707.06484.
URL <http://arxiv.org/abs/1707.06484>