Understanding Deep Architectures by Visual Summaries

Marco Godi[1]
marco.godi@univr.it
Marco Carletti[1]
marco.carletti@univr.it
Maedeh Aghaei[2]
aghaei.maya@gmail.com
Francesco Giuliari[1]
francesco.giuliari@studenti.univr.it
Marco Cristani[1]
marco.cristani@univr.it

[1] University of Verona
Italy
[2] University of Barcelona
Spain

Abstract

In deep learning, visualization techniques extract the salient patterns exploited by deep networks for image classification, focusing on single images; no effort has been spent in investigating whether these patterns are systematically related to precise semantic entities over multiple images belonging to a same class, thus failing to capture the very understanding of the image class the network has realized. This paper goes in this direction, presenting a visualization framework which produces a group of clusters or *summaries*, each one formed by crisp salient image regions focusing on a particular part that the network has exploited with high regularity to decide for a given class. The approach is based on a sparse optimization step providing sharp image saliency masks that are clustered together by means of a semantic flow similarity measure. The summaries communicate clearly what a network has exploited of a particular image class, and this is proved through automatic image tagging and with a user study. Beyond the deep network understanding, summaries are also useful for many quantitative reasons: their number is correlated with ability of a network to classify (more summaries, better performances), and they can be used to improve the classification accuracy of a network through summary-driven specializations.

1 Introduction

Individuating the visual regions exploited by a deep network for making decisions is important: this allows to foresee potential failures and highlight differences among diverse network architectures [23, 25, 27, 29]. This is the goal of the *visualization* strategies: early work [2, 23, 24, 27] individuate those images which activate a certain neuron the most; other approaches consider the network as a whole, generating dreamlike images bringing the classifier to high classification scores [14, 18, 25]. The most studied type of visualization

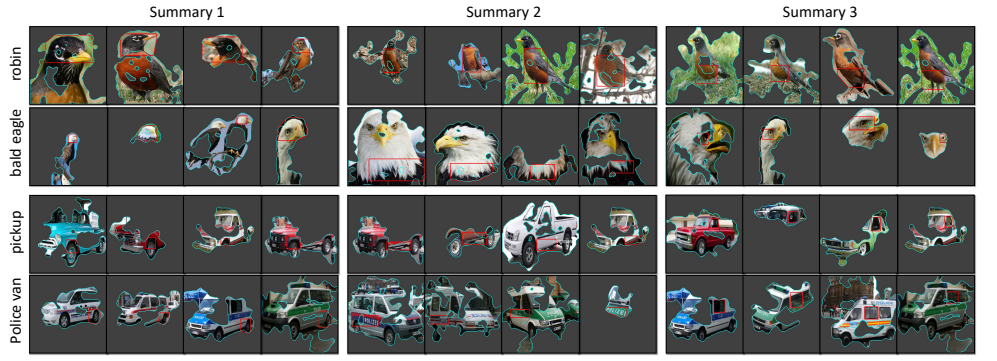

Figure 1: Visual summaries for AlexNet [9]. Each summary contains crisp salient regions, where common semantic parts are highlighted in red. It is easy to see that, i.e. in the *robin* class, the network systematically considers the head (Summary 1), the body (Summary 2), the legs and the lower body (Summary 3). Best seen in color.

techniques however, highlights those salient patterns which drive a classifier toward a class [3, 4, 11, 16, 26, 28] or against it [29] through smooth saliency maps.

However, no prior study investigated whether these salient patterns are systematically related to precise semantic entities to describe an object class. In fact, the previous visualization systems analyze single images independently, and no reasoning on multiple images from the same class is carried out. In other words, these approaches are not able to reveal if a network has captured an object class in all of its local aspects. It would be of great importance for interpretation of deep-architectures to be able to understand for example, that AlexNet when classifying the class "golden retriever" is systematically very sensible to the visual patterns representing the nose, the eye and the mouth, so that the absence of one or all of these patterns in an image will most probably bring to a failure. At the same time, knowing that GoogleNet has understood also the tail (in addition to the previous parts) can add a semantic explanation of its superiority w.r.t. AlexNet.

In this work, we present the first visualization approach which employs analysis of multiple images within an object class to provide an explanation on what has been understood by a network in terms of *visual parts* to form an object class. In practice, our approach takes as input a trained deep network and a set of images, and provides as output a set of image clusters, or *summaries*, where each cluster is representative of an object visual part.

Our visualization approach is composed by two phases. In the first phase, a crisp image saliency map is extracted from each test image, indicating the most important visual patterns for a given class. Important visual patterns are those that if perturbed in an image, lead to a high classification loss. The perturbation masks are found by an optimization process borrowed from [4] and made sparse to provide binary values which results to a so called crisp mask. In facts, most literature on visualization provide smooth masks where higher values mean higher importance in the region [2, 4, 11, 16, 26, 27, 28, 29]. In this work however, we empirically demonstrate that our proposed crisp mask brings to higher classification loss w.r.t. smooth mask by incorporating a model to remove noisy patterns. Crisp mask on the other hand, facilitates further computations in the formation of the summaries.

In the second phase, the connected components, i.e. *regions*, of the crisp masks are grouped across the image employing the affinity propagation algorithm [5], where the sim-

ilarity measure is borrowed from the proposal flow algorithm [6]. This allows for example to cluster together the wheel regions of different images from the car class, which together with other region clusters, facilitate interpretation of the class.

In the experiments, we show that our summaries capture clear visual semantics of an object class, by means of an automatic tagger and a user study. In addition, we show that the number of summaries produced by our approach is correlated with the classification accuracy of a deep network: the more the summaries, the higher the classification accuracy as demonstrated for AlexNet, VGG, GoogleNet, and ResNet in our experiments. Finally, we demonstrate that the summaries may improve the classification ability of a network, by adopting multiple, specific specialization procedures with the images of each summary.

The main contributions of this paper are as follows:

- Introduction of the first deep network saliency visualization approach to offer an understanding of the visual parts of an object class which are used for classification.
- Proposal of a model for crisp saliency mask extraction built upon the proposed model by [4].
- Generation of visual summaries by grouping together crisp salient regions of commonly repetitive salient visual parts among multiple images within a same object class.
- Presentation a comprehensive quantitative, qualitative, and human-based evaluation measures to demonstrate the advantages of visual summaries in terms of interpretability and possible applications.

2 Related Work

Visualization approaches can be categorized mainly into the *local* and *global* techniques. Local techniques focus on the understanding of single neurons by showing the filters or the activations [23]. Under this umbrella, *input-dependent* approaches select the images which activate a neuron the most [2, 25, 27]. *Global* approaches however, capture some general property of the network, as like the tendency in focusing on some parts of the images for the classification [4, 12, 16, 18, 28, 29]. These approaches are given a single image as input, and output a smooth saliency map in which the areas important for classification into a certain class are highlighted. Global approaches are mostly *gradient-based*, computing the gradient of the class score with respect to the input image [2, 12, 16, 25]. Our approach fall into the global category. Some other types of gradient-based approaches adds activations to the analysis, obtaining edge-based images with edges highlighted in correspondence of salient parts [16]. Notably, the technique of [29] individuates also the pixels which are *against* a certain class. *Generative* approaches generate dreamlike images bringing the classifier to high classification scores [13, 14, 18]. In particular, the work of [14] is heavily built on generative-based *local* representations, which are somewhat difficult to interpret, making the forecasting of the performance of the network against new data particularly complicated. *Perturbation-based* approaches edit an input image and observe its effect on the output [29]. In this case, the general output of the model is a saliency map showing how crucial is the covering of a particular area, that can be a pixel [4, 27] or superpixel-level map [15]. In all of the previous cases, the outputs are single masked images. Our approach is also perturbation based, since it looks for crisp portions of images that if perturbed, maximally distract the

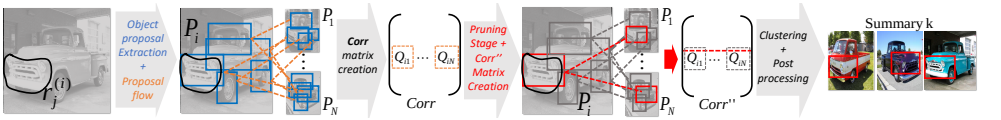

Figure 2: Sketch of the clustering phase of our proposed method (Sec. 3.2). The pipeline starts with region proposal computation and Proposal Flow-based matching. The region proposals are pruned using overlap measurement on the saliency maps. The resulting matrix of compatibility values is then used as input for a clustering algorithm.

classifier. However, unlike aforementioned models where the user has to interpret multiple saliency maps to explain the behavior of a particular classifier on a particular class, our proposed approach by providing visual summaries from the saliency maps, facilitates the interpretation task for the user.

3 Method

Our method is composed by two phases, *mask extraction* and *clustering*. The former captures what visual patterns are maximally important for the classifier, and the latter organizes the visual patterns into summaries.

3.1 Mask Extraction

Let us define a classifier as a function $y = f(x)$ where x is the input image and y is the classification score vector, in our case the softmax output of the last layer of a deep network. generating an output image in a global fashion.
Our starting point is the gradient-based optimization of [4]. In that method, the output of the optimization is a mask $m : \Lambda \to [0,1]$ with the same resolution of x, in which higher values mean higher saliency. The original optimization equation (Eq. (3) of [4]) is

$$m = \underset{m \in [0,1]^{\Lambda}}{\operatorname{argmin}} f_c(\Phi(x;m)) + \lambda_1 \|1 - m\|_1 \tag{1}$$

where $\Phi(x;m)$ is a perturbed version of x in correspondence of the non-zero pixels of m, in which the perturbation function Φ does blurring: $[\Phi(x;m)](u) = \int g_{\sigma_0 m(u)}(v-u)x(v)dv$ with u a pixel location, $m(u)$ the mask value at u and σ_0 the maximum isotropic standard deviation of the Gaussian blur kernel g_{σ_0}, $\sigma_0 = 10$. The function $f_c(\cdot)$ is the classification score of the model for the class c: the idea is to find a mask that perturbs the original image in a way that the classifier gets maximally confused, rejecting the sample for that class. The second member of Eq. (1) is a L1-regularizer with strength λ_1, which guides the optimization to minimally perturb the pixels of the input image. The authors of [4] suggested also a total variation (TV) regularizer $\sum_{u \in \Lambda} \|\nabla m(u)\|_{\beta}^{\beta}$, in which the sum operates on the β-normed partial derivatives on m, calculated as the difference of the values of two contiguous pixels according to the direction.

We contribute here by adding a sparsity regularizer $\sum_{u \in \Lambda} |1 - m(u)| m(u)$ enforcing sparsity [20] in the values of the mask m, making it binary. This regularizer has been designed to start working after a certain number of iterations, so we can get a rough version of the mask

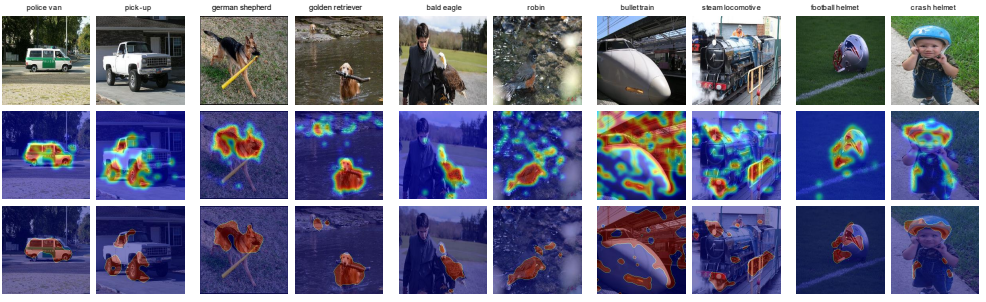

Figure 3: Qualitative analysis of the masks. First row, original image from different Imagenet classes. Second line, heatmaps computed with the method proposed by [4]. Third line, crisp masks computed with our optimization procedure. Best in colors.

before starting to optimize its crisp version, in line with the MacKay's scheduler of [10]. The final version of the optimization is thus:

$$m = \underset{m\in[0,1]^{\Lambda}}{\operatorname{argmin}} f_c(\Phi(x;m)) + \lambda_1 \|1-m\|_1 + \lambda_2 \sum_{u\in\Lambda} \|\nabla m(u)\|_{\beta}^{\beta} + \lambda_3 \sum_{u\in\Lambda} |1-m(u)|m(u) \quad (2)$$

With λs and β values set to $\lambda_1 = 0.01$, $\lambda_2 = 0.0001$, $\lambda_3 = 0$ and $\beta = 3$ during the first 300 iterations. We then modified the parameters to $\lambda_2 = 1$, $\lambda_3 = 2$ for the next 150 iterations. At the end of the mask extraction stage, each image x_i, $i = 1...N$ of a given class becomes associated to the corresponding mask m_i.

3.2 Clustering

Each saliency mask m_i can be analyzed by considering its connected components $\{r_j^{(i)}\}_{j=1...J_i}$ called here *regions*. Some of the regions are to be clustered together across multiple images of the same class to form the visual summaries of that class. The idea is that each region represents an articulated visual item composed by *parts*, and a summary is an ensemble of regions exhibiting at least a common part. A graphical sketch of the procedure is shown in Fig. 2.

In our implementation, object proposal technique [21] is employed to extract the parts of the regions. Next, the proposal flow technique [6] is incorporated to cluster the regions. Indeed, object proposals have been found well-suited for matching, with the proposal flow exploiting local and geometrical constraints to compare structured objects exhibiting sufficiently diverse poses [6].

Our procedure begins by considering the whole images of a class without resorting to the regions, in order to account as much as possible of the context where regions are merged. Given a class, all of its N images are processed; from image x_i, the set of object proposals P_i is extracted. Next, all of the images are pairwise matched adopting the proposal flow algorithm. Each pair of images $< x_i, x_j >$ will thus produce a $M_i \times M_j$ matrix Q_{ij}, with M_i indicating the number of object proposals found in image x_i. Each entry of the matrix $Q_{ij}(k,l)$ contains the matching compatibility between the k-th and the l-th object proposal of the images x_i and x_j, respectively.

After this step, all the object proposals of all the pairs of images are combined together into a $N_P \times N_P$ matrix $Corr$, where $N_P = \sum_{i=1...N} M_i$ is the total number of object proposals.

A given row of *Corr* will contain the matching score of a particular object proposal with all the remaining object proposals. *Corr* could be very large but can made easily sparse by thresholding the minimal admissible matching score.

At this point, we refer to the image regions $\{r_j^{(i)}\}$ extracted earlier and select from *Corr* all of the object proposals that overlap sufficiently with a region (overlap ratio higher than 75%). In the case of two overlapping proposals, one of them is removed if the ratio between the two areas is less than a certain threshold (2 in this work). The pruning stage leads to the $Corr''$ matrix.

The matrix $Corr''$ is considered as a similarity matrix, and the Affinity Propagation clustering algorithm is applied [5] on top of it. Affinity Propagation requires only one parameter to be set (making parameter selection easier) and it is able to discover the number of clusters by itself. The resulting clusters are ensembles of parts which, thanks to the proposal flow algorithm, should consistently identify a particular portion of an articulated object, thus carrying a clear visual semantics. Next, post-processing is carried out to prune out unreliable clusters. To this end, Structural Similarity Index (SSIM) [22] is applied to all the pairs of a cluster, discarding it as inconsistent if the median value of SSIM for that cluster is lower than a threshold based on the global median of SSIM within the whole class (90% in this work). This has the purpose of removing obvious mistakes in the clusters, caused by the variety of different poses that the proposal flow has not been able to deal with[1].

All the parts of a valid cluster are highlighted in red and shown surrounded by the regions they belong to; this eases the human interpretation and provides a summary (see an excerpt in Fig. 1). An explanation is provided for each image class using a different number of summaries, depending on the number of valid clusters that have been kept.

4 Experiments

For our experiments, we focus on 18 classes of Imagenet. These classes are selected considering the constraint of being adjacent in a *dense* [1] semantic space. In Table 1, adjacent classes are in subsequent rows with same background color. This constraint, brings together those classes that are adjacent to each other which provides the possibility of comparing *similar* classes along different experiments.

The set of experiments to validate our proposal is organized as follows: Sec. 4.1 is dedicated to show the superiority of our proposed crisp mask w.r.t. the original smooth mask [4] in terms of conciseness and expressiveness, providing higher classification drop. Sec. 4.2 is focused on the semantics of the summaries, showing that automatic taggers as well as humans, individuate a precise type of parts for each summary. Sec. 4.3 shows that the number of summaries is proportional to the classification ability of a deep architecture: the higher the number of classes the higher the classification accuracy. In Sec. 4.4 it is showed that summaries can be used to specialize the classifier on the visual summaries and improve the classification results.

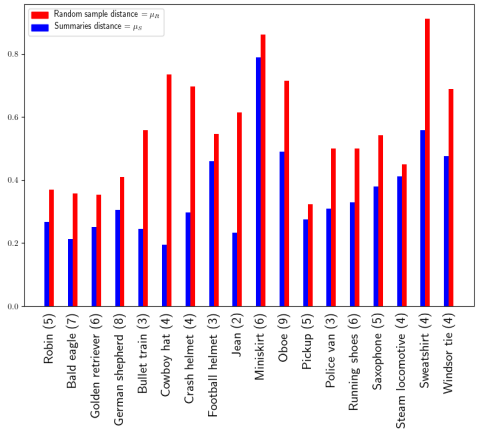

Figure 4: Coherency in terms of average Jaccard distance (y-axis) among the tags found with the automatic tagger, within the summaries (blue $= \mu_S$), and within a random sample of the class (red $= \mu_R$). *Lower is better*. The class labels come with the number of summaries found.

Class Name	μ_U	Most Proposed Tag per Summary
Robin	0.12	Head, Body, Legs, Wings, Tail
Bald eagle	0.23	Head, Neck border, Eye, Beak Face, Wing
Golden retriever	0.31	Nose, Eye, Ear, Mouth, Face, Legs, Head
German shepherd	0.22	Eye, Leg, Neck, Body, Ear, Nose, Face, Feather
Bullet train	0.38	Front train, Front glass, Train, Rails, Lights, Train body
Steam locomotive	0.56	Chimney, Front train, Wheels, Engine, Side, Window
Pick-up	0.19	Mudguard, Step bumpers, Side window, Windshield,Back, Wheel
Police van	0.17	Wheel, Police flag, Side window, Light, Rear window, Vehicle, Capote, Bumpers, Mudguard
Oboe	0.01	Body, Buttons
Saxophone	0.68	Body, Buttons, Bell
Crash helmet	0.36	Base, Side, Front, Logo
Football helmet	0.48	Front grids, Logo, Side, People
Jeans	0.01	Crotch, Pocket, Legs, Waistband
Miniskirt	0.12	Face, Waistband, Leg, Head
Cowboy hat	0.32	Ear, Face, Chin
Windsor tie	0.13	Pattern, Knot, Collar, Neck
Sweatshirt	0.31	Hoodie, Face, Arm, Laces, Wrinkles, Neck
Running shoes	0.38	Laces, Logo, Shoe side

Table 1: Classes from ImageNet, coherency of the summaries in terms of average Jaccard distance ($= \mu_U$) among the tags found with the user study and the set of tags collected during the user study with our approach.

4.1 Masks analysis

In this experiment the masks obtained by our approach are compared with those of the smooth mask a.k.a. IEBB [4] method employing the protocol as proposed by the authors. Given an image, the classification confidence associated to it w.r.t the ground truth class is measured. In the case of a deep network, the classification confidence for the i-th object class is the softmax output in the i-th entry. Afterwards, the image x is blurred as explained in Sec. 3.1 by using the corresponding mask m (either the one produced by our proposed approach or the one produced by the IEBB approach). The classification score is then recomputed after perturbation and the difference w.r.t. the score for the original image is computed. The average classification drop of a method is computed as the average score drop over the entire test set. We compare our proposal solely with IEBB, which is shown to be the state-of-the-art [4]. In addition, we compare with IEBB *thresh*, in which the smooth mask generated by IEBB is made crisp by a thresholding operation over the mask intensities. On each image the threshold is independently set to make the mask as big as the one produced by our proposed technique to ensure a fair comparison. The third column of Table 2 shows the classification loss of the two approaches. Notably, we succeed in improving the results, closely reaching the saturation. Interestingly, with IEBB *thresh*, the overall performance diminishes, with higher variance.

In Fig. 3, examples of the obtained masks using our approach and IEBB are shown. From our observations, the sparse optimization producing mask which are similar to the IEBB one. In fact, IEBB finds masks which cause a nearly complete loss. Nonetheless,

[1] Experimentally we found that in some cases of objects oriented in opposite directions, like cars towards right and left, proposal flow did not work properly providing erroneously high matching scores, as for some complex not rigid objects like animals in drastically different poses.

Method	Ref.	%Drop (Var)
IEBB	ICCV17[1]	99.738365 (8.13e-4)
IEBB *thresh.*	ICCV17[1]	97.703865 (5.758e-3)
Ours		**99.964912**(< 10e-6)

Table 2: Mask analysis results.

Model	Summaries	Acc.
AlexNet	5	57.1%
VGG16	5.5	72.4%
GoogleNet	6	74.5%
Resnet50	6.33	76.2 %

Table 3: Average number of summaries for each different architecture and top-1 accuracy.

our improvement gives the same importance to all of the pixels which leads to a higher classification drop, while facilitating the clustering step and consequently the final human interpretation of the summaries.

4.2 Analysis of the summaries

In this section of the experiments, we make use of an automated tagger [8] to show whether each summary individuates a visual semantic. For each object class, the n_i images of each single summary S_i, $i = 1,...,K$ are tagged, providing n_i lists of textual tags (only nouns are allowed). For convenience, the tagger is constrained to provide only 8 tag for each image. This procedure is repeated on K sets R_i, $i = 1,...,K$ of c_i random images taken from that class.

After tagging, the set of all the given tags is used to extract a one-hot vector for each image. The entry of the vector is 1 if a particular tag is given, and 0 otherwise. Synonyms tags were fused together by checking synsets of WordNet. This results to a vector of an average length of 28 entries. At this point, the n_i tag vectors of the summary S_i are pairwise compared with the Jaccard distance, and the average intra-summary distance is computed. This is computed for each summary, and the K average intra-summary distances are further averaged, obtaining the summary distance μ_S. This process is repeated for each class. In the same way, we compute the average distance obtained with the random image subsets R_i, getting a μ_R for each class. Results are shown in Fig. 4. As it can be seen, on average images belonging to the same summary are closer in semantic content (i.e. lower Jaccard distance) than random images of the same class.

Since the automated tagger could only work on the entire image, we expect to have much finer grained results by focusing on the parts highlighted by the summaries. To this end we organize a user study, with the goal of giving a precise name to each of the summary, by considering the parts highlighted within. We hire a total of 50 people (35 male, 15 female subjects) with an the average age of 33 (std:8.4). Each of the users was asked to give a set of (noun) tags to each summary, by considering the entire set of regions and parts contained within. Next we check the inter/rater reliability among users toward the same summary by computing the average pairwise Jaccard distance among the obtained sets of tag. The distances over the different summaries are averaged, thus obtaining for each class μ_U which is a measure of the agreement between users expressed as the average . To name each summary, we select the tag more used among the users. Table 1 report on the right these tags (one for each summary), together with the μ_U value. Interesting observations can be assessed: in some cases, the μ_U values are very small, but at the same time many tags are definitely more specific than those provided by the automatic tagger, indicating that the summaries individuate finer grained visual semantics that users have captured. Then, adjacent classes exhibit

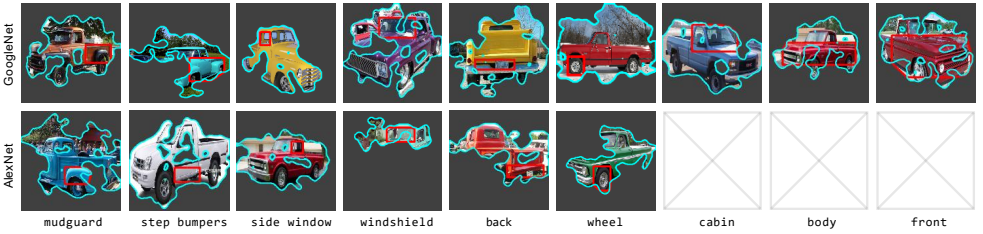

Figure 5: Motivating the superiority of GoogleNet against AlexNet. focusing on the *pick-up* class, our approach finds 9 summaries for the former architecture, 6 for the latter, showing that GoogleNet is capable of capturing more semantics. Best seen in color.

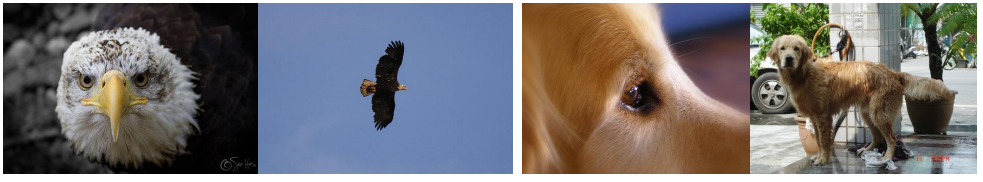

Figure 6: Examples of images two classes that were misclassified by the AlexNet but correctly classified by specializing the classification using SVMs trained on the summaries. The labels below are the class names and the tags associated with the summary that contributed the most to correcting the classification of each image.

many common visual summaries (*german shepard, golden retriever*).

4.3 Number of summaries and classification accuracy

Another interesting question to be answered is whether the number of summaries has a role in the general classification skill of a network. To this end, we analyze four famous architectures as, AlexNet [9], VGG [17], GoogleNet [19], and ResNet [7]. For each of these architectures, the average number of summaries over the 18 chosen classes for the analysis is computed. This value is later compared with the average classification ability of each architecture in terms of accuracy over ImageNet validation dataset. The comparison results are shown in Table 3. Notably, from AlexNet to ResNet, as the classification accuracy rate increases, the number of summaries also rises. From this observation, we can conclude that the network classification ability is related to the the number of discriminant patterns that the network is able to recognize. This has been shown qualitatively in Fig. 5. We obtained similar observations with other classes and other architectures.

4.4 Specializing classification with the summaries

The proposed idea in this section is to improve the classification results using the images belonging to the summaries. Due to the low number of images per summary (average of 32.25), we propose to employ a linear SVM per summary instead of explicitly fine-tuning the network itself. Positive examples to train each SVM are the images belonging to that summary, and negative examples are images from other classes or from other summaries

within the same class. The features used for classification are extracted from the first fully connected layer of the network. Given an image to classify, it is evaluated by all of the previously trained SVMs. The class scores vector is then obtained by selecting the highest score among the SVMs for each class. The obtained scores are used to improve the classification accuracy for a desired class by means of a convex weighted sum between the neural network classification softmax vectors and the resulting SVM class scores (normalized to sum to unity). Our experiments show that employing this approach, primarily designed to improve the classification of all the 18 classes chosen for the experiments on the AlexNet architecture, the overall classification accuracy score over all the 1000 ImageNet classes increases by 1.08% on the ImageNet validation set. Some examples of images that are classified correctly thanks to this boosting technique can be seen in Fig. 6.

5 Conclusion

Our approach is the first visualization system which considers multiple images at the same time, generalizing about the visual semantic entities captured by a deep network. Contrarily to the standard visualization tools, advantages of our proposed approach can be measured quantitatively, the most important of them is that of improving the original network by training additional classifiers specialized on recognizing the visual summaries. The future perspective is to inject the analysis of the summary in the early training of the deep network, and not only as a post processing boosting procedure.

References

[1] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV*, pages 71–84. Springer, 2010.

[2] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *CVPR*, pages 4829–4837, 2016.

[3] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341:3, 2009.

[4] Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, Oct 2017.

[5] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[6] Bumsub Ham, Minsu Cho, , Cordelia Schmid, and Jean Ponce. Proposal flow. In *CVPR*, 2016.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE CVPR*, pages 770–778, 2016.

[8] G. Kadrev, G. Kostadinov, and P. Ruskov. Expansion of a cnn-based image classifier's scope using transfer learning and k-nn. technical report. In *2016 IEEE 8th International Conference on Intelligent Systems (IS)*, pages 764–770, Sept 2016. doi: 10.1109/IS.2016.7737399.

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

[10] David JC MacKay. Probable networks and plausible predictionsâĂŤa review of practical bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505, 1995.

[11] A. Mahendran and A. Vedaldi. Salient deconvolutional networks. In *Proceedings of the ECCV*, 2016.

[12] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, pages 5188–5196, 2015.

[13] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, pages 427–436, 2015.

[14] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. doi: 10.23915/distill.00010. https://distill.pub/2018/building-blocks.

[15] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[16] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[20] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[21] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[22] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

[23] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[24] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014.

[25] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014.

[26] Jianming Zhang, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. In *ECCV*, pages 543–559. Springer, 2016.

[27] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

[28] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016.

[29] Luisa M Zintgracef, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.