PSEUDO-ACTIVE VISION FOR IMPROVING DEEP VISUAL PERCEPTION THROUGH NEURAL SENSORY REFINEMENT

Nikolaos Passalis and Anastasios Tefas

Dept. of Informatics, Aristotle University of Thessaloniki, Greece
Email: {passalis, tefas}@csd.auth.gr

2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506195

ABSTRACT

Active vision approaches hold the credentials for improving the accuracy of Deep Learning (DL) models for many challenging visual analysis tasks and varying environmental conditions. However, active vision approaches are typically closely tied to the underlying hardware, slowing down their adoption, while they typically increase the latency of perception systems, since sensory data must be recaptured. In this work, we propose a pseudo-active data refinement method that works by appropriately refining the sensory input, without having to reacquire the sensor data through traditional camera control approaches. The proposed method is fully differentiable and can be trained for the task at hand in an end-to-end fashion, while it can be directly deployed in a wide variety of systems, tasks and conditions. The effectiveness and robustness of the proposed method is demonstrated across a variety of tasks using two challenging datasets.

***Index Terms*—** Active Perception, Active Vision, Visual Perception, Deep Learning, Robotic Perception

1. INTRODUCTION

Deep Learning (DL) led to remarkable performance in many challenging computer vision tasks [1]. However, despite its success in such tasks, employing DL methods in real world applications, that often have different requirements than simply training a model on a typical computer vision dataset, pose significant challenges. For example, robotics typically require visual perception algorithms that can handle temporal and spatial embodiment [2], while also providing active perception capabilities [3], none of which is currently fully addressed by existing DL approaches to a satisfactory degree.

This paper focuses on active vision that allows for appropriately controlling the camera of a perception system in order to improve the perception accuracy [3]. It is worth noting that a camera can be controlled both regarding its external parameters, e.g., pan and tilt, as well as some of its internal parameters, e.g., exposure and color profile, etc. Even through there is an increasing amount of literature for handling the former [4, 5, 6, 7, 8], less focus has been given to the latter (with respect to the performance of DL models). Indeed, most DL algorithms implicitly assume that the heavy pre-processing that is involved most digital camera sensors, e.g., color constancy algorithms [9, 10, 11, 12], will mitigate the effect of varying illumination conditions. As a result, most DL models do not explicitly deal with these effects. However, as we experimentally demonstrate in this paper, DL models are especially prone to both varying illumination conditions, as well as changes in contrast, brightness, and slight color shifts. This is not a surprising finding, since adversarial attacks, as well as studies of the effect of color shifts on the accuracy of DL models, have indeed demonstrated the vulnerability of DL models to such transformations [13, 14].

Apart from distribution shifts that arise from the environmental factor described above, using different hardware for the deployment can also reduce the visual perception accuracy, if the sensors are not adequately calibrated. The typical solution to this problem is to manually tune the hardware, as well as employ the appropriate software pre-processing modules in order to ensure that the models will perform as expected. However, it is obvious that this process requires a significant amount of effort, often slowing down the integration in many real robotics applications.

Active perception algorithms can be employed to tackle the aforementioned challenges, i.e., to appropriately control the internal camera parameters in order to maximize perception accuracy for a given DL model. This process has many similarities with the visual perception systems developed in many biological organisms, such as many mammals. In these cases, different mechanisms exist for adjusting various parameters that affect the signal reaching to sensor cells, well before propagating the information to the visual cortex. Perhaps the most well-known example of such mechanism is the adaptive response of the iris to different illuminations conditions, altering the amount of light eventually reaching the retina [15]. Despite their potential, active vision methods come with two important drawbacks. First, they typically lead to the (complete or partial) loss of (at least) one frame acquired by the camera, which can negatively affect the latency of the system. Second, such algorithms are not easily deployed, since the output of the active perception algorithm must be first adequately translated in order to match the underlying hardware of each system, i.e., to translate the

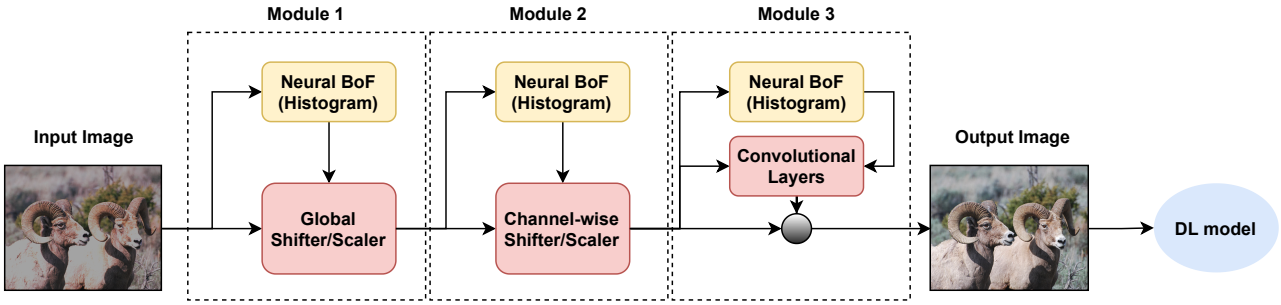

Fig. 1. The proposed method is composed of three separate neural modules. Each one is responsible for learning how to apply a different kind of image transformation in order to gradually refine the input image in order to maximize the accuracy of the subsequent DL model.

model's output to control signals.

To overcome these limitations in this work we propose a pseudo-active sensory refinement method that works by applying a number of neural transformation layers on the sensor data. This allows for refining the sensory input, without having to reacquire the sensor data. In contrast with traditional image processing operations, such as histogram equalization, contrast corrections, etc., the proposed method is *end-to-end* trainable and formulated as a series of neural layers. As a result, the proposed method can be fully integrated in DL end-to-end training pipelines. However, at the same time, it provides significant advantages, since a) it can be directly used with any DL model, without requiring any model-specific training or any platform-specific adjustments, b) it does not require support by the underlying hardware, and c) it allows for avoiding the need to reacquire a new frame for processing by the employed DL model. As a result, the proposed method provides a solid step towards developing practical and powerful tools that can be directly deployed in a wide variety of systems, tasks and conditions, increasing the perception accuracy. Indeed, we demonstrate the effectiveness of the proposed method using two different tasks and datasets, i.e., image recognition on ImageNet dataset (ILSVRC 2012) [16] and object detection on PASCAL VOC 2007 dataset [17].

The rest of the paper is structured as follows. First, we introduce the proposed method in Section 2. Then, we provide the experimental evaluation in Section 3. Finally, conclusions are drawn in Section 4.

2. PROPOSED METHOD

Let $\mathbf{x} \in \mathbb{R}^{W \times H \times C}$ be an input image, where W, H and C denote its width, height and number of channels respectively. Typically, the input image $\mathbf{x}$ is fed into a DL model $f(\mathbf{x})$ in order to perform visual information analysis, e.g., object recognition. The proposed method works by employing a series of neural transformation layers, as shown in Fig. 1, before feeding the input into the DL model. The proposed method is composed of three separate modules, each one performing a different *learnable* transformation on the input. Each of these modules operate on a dynamic differentiable color histogram compiled through the previous stage. Therefore, during the first stage, the image is globally shifted and scaled according to the input histogram. Then, the second stage performs the same transformation, but on channel level. These two modules combined aim at correcting global and per color channel brightness and contrast. Then, the third module is responsible for refining local regions of the input image, while also taking into account its global histogram, in order to recover information that is potentially lost during image acquisition, e.g., due to over-exposure.

More specifically, the proposed method works as follows. First, we compile a differentialable histogram with adaptive bins using a Neural Bag-of-Feature-based formulation [18]. To this end, we quantize the input image features $[\mathbf{x}]_{ij} \in \mathbb{R}^C$ using a codebook $\mathbf{V} = [\mathbf{v}_1; \dots; \mathbf{v}_{N_K}]^T \in \mathbb{R}^{N_K \times C}$ as:

$$\mathbf{h} = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} \mathbf{u}_{ij} \in \mathbb{R}^{N_K} \tag{1}$$

where N_K is the number of codewords, while the similarity vector $\mathbf{u}_{ij}$ for the codewords $\mathbf{v}_k$ and the input features $\mathbf{x}_{ij}$ is calculated as

$$[u_{ij}]_k = \text{sigm}(\mathbf{v}_k^T \mathbf{x}_{ij}), \tag{2}$$

where $\text{sigm}(\cdot) \in (0, 1)$ denotes the sigmoid function. Three different histograms are compiled, as shown in Fig. 1: a) one at the input ($\mathbf{h}_1$), b) one at the output of the first module ($\mathbf{h}_2$) and c) one at the end of the second module ($\mathbf{h}_3$). Note that the first histogram operate on the averaged color intensities, i.e., $C = 1$, instead of separate color channels, since its aim is to capture the global information regarding the brightness distribution in the image.

After compiling the first histogram, the proposed method employs two linear layers to appropriately shift and scale the

Table 1. Object recognition evaluation (recognition accuracy (%) is reported) on ImageNet dataset (ILSVRC 2012)

Method	Transformation	top-1	top-5	Method	Transformation	top-1	top-5
Baseline	No	69.76	89.08	Baseline	Brightness(-)	60.52	82.64
Baseline	Contrast (+)	62.18	83.94	Autocontrast	Brightness(-)	61.02	82.99
Hist. Equalization	Contrast (+)	54.69	77.74	Proposed	Brightness(-)	**61.67**	**83.58**
Autocontrast	Contrast (+)	62.18	83.93	Baseline	Brightness(–)	53.13	76.38
Proposed	Contrast (+)	**63.43**	**84.95**	Autocontrast	Brightness(–)	53.78	77.15
Baseline	Contrast (++)	57.49	80.23	Proposed	Brightness(–)	**54.78**	**78.01**
Hist. Equalization	Contrast (++)	50.00	73.48	Baseline	Hue (+)	62.11	84.38
Autocontrast	Contrast (++)	57.47	80.23	Autocontrast	Hue (+)	62.17	84.39
Proposed	Contrast (++)	**59.23**	**81.78**	Proposed	Hue (+)	**62.37**	**84.70**
Baseline	Brightness (+)	64.55	85.92	Baseline	Hue (-)	62.63	85.02
Autocontrast	Brightness (+)	66.81	87.26	Autocontrast	Hue (-)	62.69	85.05
Proposed	Brightness (+)	**67.03**	**87.35**	Proposed	Hue (-)	**63.41**	**85.46**
Baseline	Brightness(++)	60.09	82.48	Baseline	Combined	55.31	78.35
Autocontrast	Brightness(++)	64.10	85.14	Autocontrast	Combined	55.17	78.23
Proposed	Brightness(++)	**64.31**	**85.38**	Proposed	Combined	**55.60**	**78.80**

input:

$$\mathbf{x}' = (\mathbf{x} - \mathbf{h}_1^T \mathbf{W}_{11})/(1 + \mathbf{h}_1^T \mathbf{W}_{12}), \quad (3)$$

where $\mathbf{W}_{11} \in \mathbb{R}^{N_K \times 1}$ denotes the weights of the shifting sub-layer and $\mathbf{W}_{12} \in R^{N_K \times 1}$ denotes the weights of the scaling sub-layer. Then, this process is repeated in the second module, but the shifting and scaling is now applied per-channel:

$$[x'']_{ijk} = ([x']_{ijk} - [\mathbf{h}_2^T \mathbf{W}_{21}]_k)/(1 + [\mathbf{h}_2^T \mathbf{W}_{22}]_k), \quad (4)$$

where $\mathbf{W}_{21} \in \mathbb{R}^{N_K \times C}$ denotes the weights of the shifting sub-layer and $\mathbf{W}_{22} \in \mathbb{R}^{N_K \times C}$ denotes the weights of the scaling sub-layer.

Finally, the third module works by first projecting the third histogram into a lower dimensional space:

$$\mathbf{h}_x = \tanh(\mathbf{h}_3^T \mathbf{W}_p) \in \mathbb{R}^{N_P} \quad (5)$$

where $\mathbf{W}_p \in \mathbb{R}^{N_K \times N_P}$ and $\tanh(\cdot)$ denotes the hyperbolic tangent function. Then, the vector $\mathbf{h}_x$ is upsampled into a $W \times H \times N_P$ tensor and concatenated (channel-wise) with the output of the previous layer to form the $\mathbf{x}_p \in \mathbb{R}^{W \times H \times (C+N_P)}$ tensor. The output of the final layer is then calculated as:

$$\mathbf{x}''' = \mathbf{x}'' \odot (1 + \tanh(g(\mathbf{x}_p))), \quad (6)$$

where $g(\cdot) \in \mathbb{R}^{W \times H \times (C)}$ is a series of convolution and deconvolution layers and $\odot$ denotes the Hadamard (element-wise) product between two tensors. In this work we use two 5×5 convolution layers with 16 and 8 filters, followed by two symmetric deconvolution layers (the last layer has C filters). The $\tanh(\cdot)$ non-linearity is used for all the layers. Note that we assumed that the input is already normalized into a specific (limited) range. If this assumption does not hold, then the output of the last layer must be also scaled using a trainable parameter.

All the parameters of the proposed method are learned by first employing an image transformation on the original image $\mathbf{x}$, e.g., brightness, contrast or hue adjustment, leading to a *corrupted* image $\tilde{\mathbf{x}}$. Then, all the layers are trained using the back-propagation algorithm in order to recover the original image to the output of the proposed sequence of neural layers, denoted by $h(\cdot)$, by minimizing the following loss function:

$$\mathcal{L} = \frac{1}{WHC} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{C} ([\mathbf{x}''']_{ijk} - [\mathbf{x}]_{ijk})^2, \quad (7)$$

where $\mathbf{x}''' = h(\tilde{\mathbf{x}})$.

3. EXPERIMENTAL EVALUATION

The proposed method was first extensively evaluated on the ImageNet dataset (ILSVRC 2012) [16] using an image recognition setup and a ResNet-18 architecture [19]. The proposed method was trained on the ImageNet dataset using the Adam optimizer [20]. The proposed model was trained for 10,000 iterations with a learning rate of 0.001, followed by 2,000 iterations with a reduced learning of 0.0001. The batch size was set to 32, while the number of codewords was set to $N_K = 100$ and the number of projection dimensions for the last layer to $N_P = 8$. The proposed method was also compared to a) directly using the trained ResNet-18 architecture (denoted as "Baseline") with the default normalization scheme (global standardization), b) employing a histogram equalization approach (denoted as "Hist. Equalization") and c) maximizing the contrast (denoted as "Autocontrast").

We also used different approaches to deteriorate the original images: a) increasing the contrast by 30% (+) and 40% (++), b) increasing (+/++) or decreasing (-/–) the brightness by 30% and 40% respectively, c) increasing (+) or decreasing (-) the hue by 10%. Also, we evaluated the methods in a more challenging combined setup where the contrast was increased by 20%, brightness was decreased by 20% and hue was increased by 5%. The proposed model was trained by randomly

Table 2. Ablation study on the ImageNet dataset using the contrast (++) transformation (%)

Method	top-1	top-5
Baseline	57.49	80.23
Module 1	58.22	80.87
Module 1 + 2	58.42	80.99
Module 1 + 2 + 3	**59.23**	**81.78**

selecting one transformation and then recovering the original image.

The experimental results for the ImageNet dataset are reported in Table 1. Several interesting conclusions can be drawn from the reported results. First, note that all the applied transformations, i.e., increasing the contrast, increasing/reducing the brightness, increasing/decreasing the hue, and/or combining multiple transformations on the input image, lead to a significant reduction in recognition accuracy. For example, even a mild shift in the hue by 10% leads to a reduction in top-1 accuracy from 69.76% to 62.11%. At the same time, employing non-linear histogram equalization does not improve the recognition accuracy. In contrast, this type of image enhancement actually reduces the recognition accuracy, possibly by introducing a catastrophic distribution shift. Similar results were observed for the rest of transformations as well, but omitted from Table 1, due to lack of space. On the other hand, maximizing the contrast in the input image can lead to improved recognition accuracy in virtually all the evaluated cases. Employing the proposed method can further improve the recognition results, by being able to learn how to appropriately refine the original images in order to compile new transformed images that would be appropriate for the DL model at hand. It is worth noting that for some cases, e.g., for contrast transformations, using the proposed method can improve the top-1 recognition by over 3% (relative improvement). Furthermore, in order to evaluate the effect of the three different modules that are used by the proposed method, we also performed an ablation study, where each of these modules was separately evaluated. The experimental results are reported in Table 2, where we can observe that each of the employed modules further increases the recognition accuracy.

To further demonstrate the generality of the proposed method, we conducted additional experiments for one other task, i.e., object detection. To this end, we employed the Single Shot MultiBox Detector (SSD) [21], with a MobileNet backbone [22] for object detection. The proposed method was not trained on the corresponding detection dataset. Instead, we directly employed the model trained on the ImageNet dataset for a recognition task in order to evaluate how it generalizes on this related task. The experimental results are reported in Table 3. Indeed, using the proposed method again improves the perception accuracy over the evaluated baselines. Quite interestingly maximizing the contrast does not work so well for this task, while the proposed method still leads to impressive improvements (e.g., the mean Average Precision (mAP) increases by over 1.5% in some cases). These results highlight the ability of the proposed method to be directly employed and combined with virtually any DL model and further increase its robustness by providing an efficient pseudo-active vision approach.

Table 3. Evaluation on the VOC2007 (object detection) dataset (%)

Method	Transform.	VOC2007 mAP
Baseline	-	75.51
Baseline	Contrast (+)	62.24
Autocontrast	Contrast (+)	62.13
Proposed	Contrast (+)	**63.19**
Baseline	Contrast (++)	66.95
Autocontrast	Contrast (++)	66.86
Proposed	Contrast (++)	**67.44**
Baseline	Brightness (++)	70.30
Autocontrast	Brightness (++)	71.36
Proposed	Brightness (++)	**71.45**
Baseline	Brightness (–)	56.84
Autocontrast	Brightness (–)	55.81
Proposed	Brightness (–)	**56.85**
Baseline	Combined	56.64
Autocontrast	Combined	56.39
Proposed	Combined	**57.29**

4. CONCLUSIONS

In this work we presented a pseudo-active sensory refinement method that works by applying a number of neural transformation layers on the input, allowing for efficiently refining the sensory input, without having to re-acquire the sensor data. The proposed method is fully differentiable and can be trained for the task at hand in an end-to-end fashion. Furthermore, as demonstrated through the conducted experiments, the proposed method can perform well across a variety of tasks. In this way, it provides a solid step towards providing powerful tools that can be directly deployed in a wide variety of systems, tasks and conditions, increasing the perception accuracy, paving the way for developing more advanced active vision approaches for manipulating the camera parameters.

Acknowledgments: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871449 (OpenDR). This publication reflects the authors views only. The European Commission is not responsible for any use that may be made of the information it contains.

5. REFERENCES

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, et al., "The limits and potentials of deep learning for robotics," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 405–420, 2018.

[3] Ruzena Bajcsy, Yiannis Aloimonos, and John K Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, 2018.

[4] Nitin J Sanket, Chahat Deep Singh, Kanishka Ganguly, Cornelia Fermüller, and Yiannis Aloimonos, "Gapflyt: Active vision based minimalist structure-less gap detection for quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2799–2806, 2018.

[5] Anton Mitrokhin, P Sutor, Cornelia Fermüller, and Yiannis Aloimonos, "Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception," *Science Robotics*, vol. 4, no. 30, 2019.

[6] Nikolaos Passalis and Anastasios Tefas, "Leveraging active perception for improving embedding-based deep face recognition," in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing*, 2020, pp. 1–6.

[7] Theodoros Bozinis, Nikolaos Passalis, and Anastasios Tefas, "Improving visual question answering using active perception on static images," in *Proceedings of the International Conference on Pattern Recognition*, 2021, pp. 879–884.

[8] Weihao Yuan, Rui Fan, Michael Yu Wang, and Qifeng Chen, "Active perception with a monocular camera for multiscopic vision," *arXiv preprint arXiv:2001.08212*, 2020.

[9] Marc Ebner, *Color constancy*, vol. 7, John Wiley & Sons, 2007.

[10] Simone Bianco, Claudio Cusano, and Raimondo Schettini, "Color constancy using cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 81–89.

[11] Yuanming Hu, Baoyuan Wang, and Stephen Lin, "Fc4: Fully convolutional color constancy with confidence-weighted pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4085–4094.

[12] Firas Laakom, Nikolaos Passalis, Jenni Raitoharju, Jarno Nikkanen, Anastasios Tefas, Alexandros Iosifidis, and Moncef Gabbouj, "Bag of color features for color constancy," *IEEE Transactions on Image Processing*, vol. 29, pp. 7722–7734, 2020.

[13] Mahmoud Afifi and Michael S Brown, "What else can fool deep learning? addressing color constancy errors on deep neural network performance," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 243–252.

[14] Camilo Pestana, Naveed Akhtar, Wei Liu, David Glance, and Ajmal Mian, "Adversarial perturbations prevail in the y-channel of the ycbcr color space," *arXiv preprint arXiv:2003.00883*, 2020.

[15] Richard S Snell and Michael A Lemp, *Clinical anatomy of the eye*, John Wiley & Sons, 2013.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[17] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[18] Nikolaos Passalis and Anastasios Tefas, "Learning bag-of-features pooling for deep convolutional neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 21–37.

[22] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.