# In the Depths of Hyponymy: A Step Towards Lifelong Learning

Tommaso Boccato*, Timothy Patten†, Markus Vincze† and Stefano Ghidoni*
*Department of Information Engineering, Università degli Studi di Padova, Padova, Italy
Email: tommaso.boccato@studenti.unipd.it, stefano.ghidoni@unipd.it
†Automation and Control Institute, TU Wien, Vienna, Austria
Email: {patten, vincze}@acin.tuwien.ac.at

***Abstract*—This paper proposes a novel framework for lifelong learning of semantic classes in order to extend the operational time of robots deployed in real-world and uncontrolled environments. In contrast to the common approach that assumes fixed object classes, the proposed framework keeps track of the intra-class variability over time in order to refine the class definition encoded into a classifier. A carefully designed metric is also presented to quantify the intra-class variability, which leads to automatic triggering of the class restructuring. Experiments performed with the CIFAR-100 dataset validate the framework and the measure of intra-class variability.**

***Keywords–Classification; Lifelong learning; Open set learning.***

## I. INTRODUCTION

The applications in which a robot should be able to understand what it sees are countless: human-robot interaction, healthcare, service robotics, industrial robotics, logistics, connected and autonomous vehicles. A deep knowledge of the visual properties and functionalities that characterize the objects is vital in the application of the robot itself, allowing for better manipulation, navigation or exploration. Very often, this knowledge is manually encoded into the deployed computer vision algorithms during their training process. Lifelong learning capabilities [1], however, represent a desirable feature.

The last decade of advancements in deep learning have led to astonishing results in the applications that respond to the so called closed-world assumption (i.e., the assumption that the object classes encountered during the operational life of a robot are known and fixed a-priori) [2]. Robots, however, operate in dynamic and uncontrolled environments. As such, the use of standard approaches in these environments usually reveals performance drops. A continuous update of the semantic structure on which a classifier works requires the introduction of additional complexity in the system [3] [4] [5]. Moreover, the update should be efficient and downtimes minimized.

In the presented work, with reference to the classification task, a step is taken towards relaxing the aforementioned assumption by introducing a novel framework capable of allowing the refinement of the classes encoded into a classifier during its operational life. Specifically, the framework keeps track of the intra-class variability temporal evolution linked to the various categories in such a way as to trigger meaningful class reconfiguration. In other words, classes characterized by high intra-class variabilities should be divided into sets of subclasses whose labels are related to the original ones through hyponymy relationships (i.e., words of more specific meaning than general or superordinate terms applicable to them). An example of such a scenario is shown in Figure 1. Clearly, a metric capable of quantifying the abstract intra-class variability concept plays a key role within the framework. Therefore, we also propose a suitable metric design.

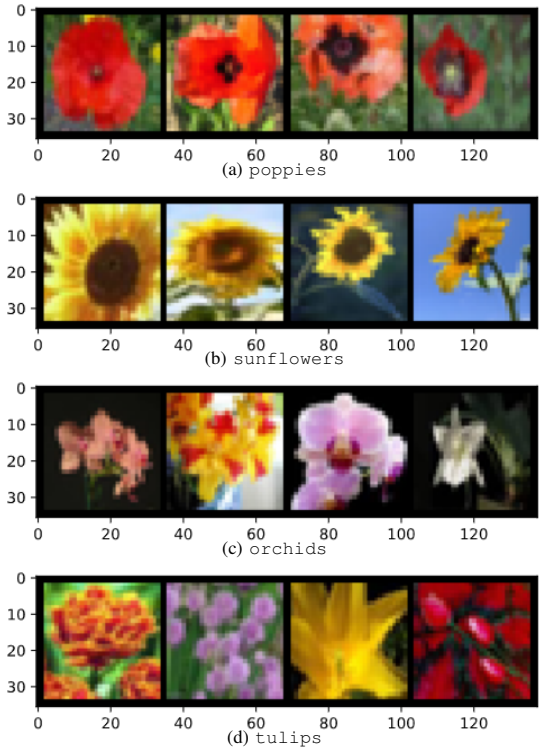



Figure 1. Randomly sampled batches extracted from 4 different CIFAR-100 [6] classes (poppies, sunflowers, orchids, tulips). All classes belong to the same super-class of flowers. Images in the top rows show homogeneous visual properties while images in the bottom rows are characterized by very different visual properties. Yet, all the batches belong to specific categories. A question arises: *How does the intra-class variability impact a classifier, and how can an agent (e.g., a robot) recognize and exploit this phenomenon?*

The remainder of the paper is organized as follows. Section II discusses related work. Section III outlines the lifelong learning framework and Section IV describes the metric for intra-class variability. Section V presents the experiments and results. Section VI concludes and discusses future work.

## II. RELATED WORK

According to survey [7], the classification task introduced within the proposed framework can be categorized as hierarchical. In particular, the so called "flat classification approach" is pursued. The class hierarchy, a tree data structure representing hyponymy relationships, is indeed ignored by the classifier,

 ISBN: 978-1-61208-787-0 

that only acts on its leaves. A possible class hierarchy implementation can follow the WordNet [8] [9] hyponymy network. However, most of the hierarchical classification literature assumes subsets of directed acyclic graphs to be trees in order to simplify their manipulation. We are currently unaware of works that exploit incremental or online hierarchical classification for lifelong learning purposes. On the other hand, parallel paths have been explored in robotics. This section continues with an overview of relevant literature addressing the open set recognition problem as well as measuring intra-class variability.

The Open World Recognition (OWR) framework is formally defined in [3], with the introduction of the Nearest Non-Outlier algorithm and the design of a suitable evaluation protocol; the algorithm is able to incrementally add object categories while detecting outliers. The OWR framework represents a starting point for [4] that proposes a deep extension of a non-parametric model that learns additional categories without retraining the whole system from scratch. The possibility of retrieving annotated images by autonomously mining the web constitutes a major contribution of the work. An attempt to extract label uncertainty from state-of-the-art object detection systems via dropout sampling is performed in [10]. Novel objects are also introduced to robots by means of pointing gestures and verbal communication [11]. Finally, an incremental version of the Regularized Least Squares for Classification algorithm is tested in [12]. The authors also address the problem of having an unbalanced proportion of training samples during the algorithm operational life. The work addressing the open set recognition problem all assume a definitive set of training classes. In contrast, we propose a framework capable of managing concept drifts introduced in all the classes encoded into the considered classifier.

The treatment of intra-class variability in the literature is scattered across several diverse fields, none that are specific to robotics or machine vision. For example, the intra-class variability affecting winter wheat mapping from multi-temporal Moderate Resolution Imaging Spectroradiometer (MODIS) Enhanced Vegetation Index (EVI) images is addressed by generating multiple training sub-classes to decrease the intra-class differences for the crop type detection [13]. The separability of the generated sub-classes exploits the Jeffries-Matusita (JM) Distance; such separability reflects the intra-class variability of the associated original class. A similar approach is used for liver lesion detection [14] where a multi-class convolutional neural network (CNN) categorizes image patches into sub-categories, which are then fused to obtain a binary lesion/non-lesion classification. A novel offline approach, instead, is proposed in [15] to model biometric data intra-class variability and typicality. The method consists of a two stage algorithm: the former is represented by the clustering of the input images while the latter performs a template extraction from the clustered data. Finally, [16] reports a few functions to represent the covariance matrix of a multi-variate distribution as a scalar. While these works consider intra-class variability, it has mainly been investigated from a qualitative and high-level point of view. Additionally, the concept is applied in domains different to our study: they do not specifically address lifelong learning for a robotic system.

## III. Lifelong Learning Framework

Our work builds on the *Open Set Learning* paradigm and its framework [3] [4] in order to explore an alternative path towards the development of an agent characterized by lifelong learning capabilities. The objective pursued by the definition of the framework is to theoretically describe the operational life of a classifier trained on a set of semantic categories or classes labeled by the positive integers $\mathcal{K}_1 = \{1, \ldots, N_1\}$, with $|\mathcal{K}_1| = N_1$. The considered model thus refines its semantic categories every time the intra-class variability associated to a specific category proves to be sufficiently high according to a pre-defined criterion; this concept, as well as the whole framework definition, is presented generically in order to allow the framework to enclose a large variety of future works. It is therefore natural to define $\mathcal{K}_t \subseteq \mathbb{N}^+$ as the set of classes encoded into the classifier at time $t$. Moreover, $|\mathcal{K}_i| = N_i \leq |\mathcal{K}_j| = N_j$ when $i < j$. An example of class structure temporal evolution is shown in Figure 2.

Let $\boldsymbol{x} \in \mathbb{R}^d$ be the features associated to a new sample seen by the classifier. Let $\mathcal{T}_t \subseteq \mathbb{R}^d \times \bigcup_{j=1}^{t} \mathcal{K}_j$ be the set containing all the samples, with the respective labels, seen by the classifier up to time $t$ (the definition of $\mathcal{T}_t$ does not allow the repetition of a specific pair, but such scenario can be verified in the operational life of a real classifier; the problem can be overcome by adding an auxiliary dimension to the space of features used to enumerate the samples). The set cardinality can be expressed as $|\mathcal{T}_t| = M_t + t$: the former term refers to the model training (ground truth labels) while the latter refers to the model operational life (labels provided by the classifier). A model, to function within the defined framework, must be characterized by the following main ingredients.

### A. Multi-class Recognition Function

The *multi-class recognition function* $F_t : \mathbb{R}^d \longrightarrow \mathcal{K}_t$ exploits the vector function

$$\psi_t(\boldsymbol{x}) = [f_t^i(\boldsymbol{x})], \ \forall i \in \mathcal{K}_t, \tag{1}$$

where the generic *per-class recognition function* $f_t^i : \mathbb{R}^d \longrightarrow \mathbb{R}$ belongs to a suitable space $\mathcal{H}$. Typically, $f_t^i(\boldsymbol{x})$ reports the likelihood of being in class $i$, the values of $f_t^i(\boldsymbol{x})$ are normalized across the respective semantic categories and the multi-class recognition function is implemented as:

$$F_t(\boldsymbol{x}) = \arg\max_{i \in \mathcal{K}_t} f_t^i(\boldsymbol{x}). \tag{2}$$

### B. State Update Function

For each semantic category, the corresponding element of the set should contain all the necessary information to compute its intra-class variability after the classification performed in the previous time step. The nature of the generic element $s_t^i$ is not specified: it could represent a scalar, a matrix or any other kind of data structure depending on the needs. Every time a new sample is classified, the *state* $\mathcal{S}_t = \{s_t^i\}, \ \forall i \in \mathcal{K}_t$ must be updated accordingly. The *state update function* $U : \mathcal{S}_t \times \mathbb{R}^d \longrightarrow \mathcal{S}_{t+1}$ is exploited for the purpose. Specifically,

$$s_{t+1}^i = U(s_t^i, \boldsymbol{x}), \tag{3}$$

if $\boldsymbol{x}$ is recognized as belonging to class $i$. Clearly, $s_{t+1}^j = s_t^j, \ \forall j \neq i$.

At this point, the intra-class variability computation can finally be formalized through the function $V : \mathcal{S}_{t+1} \longrightarrow \mathbb{R}$. Intuitively, the intra-class variability of class $i$ at time $t$ should depend on $\mathcal{T}_t^i = \{(\boldsymbol{x}, k) \text{ s.t. } k = i\}$; $s_{t+1}^i$ encapsulates this information allowing an efficient sequential update of the

metric. Indeed, it could not be feasible to store the entire $\mathcal{T}_t^i$ or to use the set for a direct intra-class variability computation. The additional state $\mathcal{S}_t$ is also motivated by the fact that the $V$ function, in general, is not invertible; this means that $V(s_{t+1}^i)$ may not be obtainable starting from $V(s_t^i)$.

Hence, a *trigger* $T : \mathbb{R} \longrightarrow \{0, 1\}$ is defined in accordance with a criterion selected by the designer in order to establish whether class $i$ needs to be split or not; it returns 1 if the considered semantic category has to be replaced by more specific sub-classes, 0 otherwise.

### C. Labeling Process and Data Retrieval Functions

The *labeling process function* $L_t : \mathcal{P}(\mathcal{T}_t^i) \longrightarrow \mathcal{P}(\mathbb{N}^+ \setminus \bigcup_{j=1}^{t} \mathcal{K}_j)$, where $\mathcal{P}(\bullet)$ denotes the power set, aims to retrieve the sub-class labels of class $i$ when its split is triggered (i.e., $T(V(s_{t+1}^i)) = 1$). It is important to remember that the used labels are excluded from the function codomain. Again, a subset $\mathcal{T}_t^i$ can be exploited to overcome possible limitations in the available spatial and temporal computational resources.

Once the new categories are collected, the classifier class structure has to be updated. The following rule is exploited:

$$\mathcal{K}_{t+1} = \mathcal{K}_t \setminus i \cup \mathcal{N}_{t+1}, \tag{4}$$

where $i$ is the label of the considered class and $\mathcal{N}_{t+1}$ represents the set of labels returned by the labeling process after the classification of the $t$-th sample.

The *data retrieval function* $R : \mathcal{P}(\mathcal{K}_{t+1}) \longrightarrow \mathcal{P}(\mathbb{R}^d \times \mathcal{K}_{t+1})$ is responsible for retrieving the new data $\mathcal{D}_{t+1} \in \mathcal{P}(\mathbb{R}^d \times \mathcal{K}_{t+1})$ for the incremental training of the model. The function domain is chosen as to allow approaches capable of mitigating the effect of catastrophic forgetting [5]. Additionally, it is worth noting that the $L_t$ and $R$ functions must rely on an external source of information (e.g., the web) and the performance of their implementations could not be error free.

### D. Incremental Learning Function

The *incremental learning function* is defined as $I_t : \mathcal{P}(\mathbb{R}^d \times \mathcal{K}_{t+1}) \times \mathcal{H}^{N_t} \longrightarrow \mathcal{H}^{N_{t+1}}$, where $N_{t+1} - N_t = |\mathcal{N}_{t+1}| - 1$. The objective of the function is to incrementally update the model by replacing the obsolete per-class recognition function $f_t^i(\boldsymbol{x})$ with the ones related to the new $|\mathcal{N}_{t+1}|$ semantic categories. The retrieved data $\mathcal{D}_{t+1}$ is exploited for the purpose. Hence, the state $\mathcal{S}_{t+1}$ has to be expanded and the added entries must be initialized properly. If possible, the model should gradually adapt to the new class structures without completely retraining.

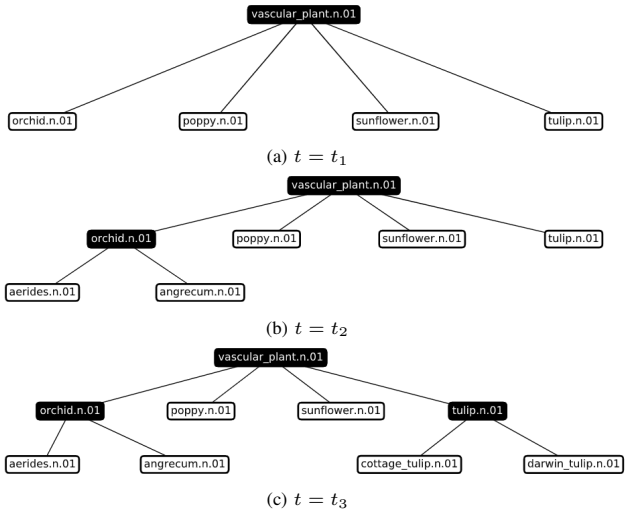
Every time $T(V(s_{t+1}^i)) = 0$, a simple implicit update of the $\mathcal{K}_t,\ F_t,\ f_t^i$ subscripts (time steps) has to be performed.

## IV. Metric for Intra-class Variability

This section describes the design of a suitable metric for quantifying the intra-class variability. This can then be used to trigger the splitting event and therefore the update of the classification model.

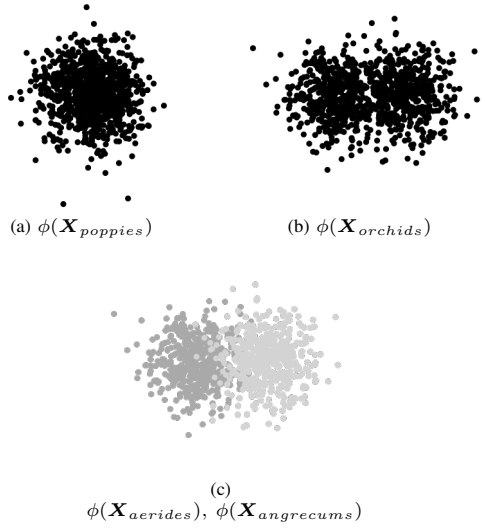
Let $\boldsymbol{X}$ be the matrix whose columns are the vectors belonging to the set $\{\boldsymbol{x}$ s.t. $(\boldsymbol{x}, k) \in \mathcal{T}_t^i\}$. In other words, $\boldsymbol{X}$ contains all the samples, belonging to or classified as belonging to class $i \in \mathcal{K}_t$, seen by the considered model up to time $t$. The matrix can be thought of as the repeated sampling of a probability distribution over $\mathbb{R}^d$ associated with the environment in which the model is immersed (when the $d$-th dimension is reserved for the sample enumeration, the underlying probability distribution should be defined over the first $d-1$ dimensions).

If the used classifier belongs to the category of deep models, $\phi : \mathbb{R}^d \longrightarrow \mathbb{R}^n$ can be defined as the function responsible for extracting deep representations (e.g., the output of the last layer before the linear ones in ResNet [17] or VGG [18]) from the generic sample features $\boldsymbol{x} \in \mathbb{R}^d$. For simplicity, the $\phi$ notation is overloaded by defining $\phi(\boldsymbol{X})$ as the matrix obtained applying function $\phi$ to $\boldsymbol{X}$ columnwise. Also, $\phi(\boldsymbol{X})$ can be thought of as the repeated sampling of a new probability distribution derived from the original one by applying $\phi$ to the multivariate random variable $\boldsymbol{x}$ (depending on the context, $\boldsymbol{x}$




(a) $t = t_1$ (b) $t = t_2$ (c) $t = t_3$

Figure 2. Example of class structure temporal evolution for the semantic categories in Figure 1. The leaves (i.e., white nodes) of the trees represent the classes encoded into the classifier at the considered time steps, where $t_1 < t_2 < t_3$. Clearly, classes that are present at time $t$ are labeled by the elements of $\mathcal{K}_t$. The `orchid` class is the first to be split ($t = t_2$), the `tulip` class follows ($t = t_3$). Trees follow the hyponymy in [8] [9].



(a) $\phi(\boldsymbol{X}_{poppies})$ (b) $\phi(\boldsymbol{X}_{orchids})$ (c) $\phi(\boldsymbol{X}_{aerides})$, $\phi(\boldsymbol{X}_{angrecums})$

Figure 3. Our formulation of intra-class variability. The setup is the same reported in Figure 1 and 2. The intra-class variability of the category shown in (a) is low while the intra-class variability of the category shown in (b) is high (sub-classes are shown in (c) for comparison). The shape of the deep representations reflects the hypothesis: $\phi(\boldsymbol{X}_{poppies})$ approximates a hyperball better than $\phi(\boldsymbol{X}_{orchids})$.

can be regarded as the features of a generic image sample or the associated random vector).

The intuition, therefore, is to link the abstract concept of intra-class variability to the shape of the $\phi(\boldsymbol{X})$ sampling in the space of the deep representations. The formulated hypothesis follows: *The lower the intra-class variability of class $i$, the better the sampling $\phi(\boldsymbol{X})$ approximates a hyperball.* Given the metric space $(\mathbb{R}^n, d)$, with the distance function set to be

$$\begin{aligned} d : \mathbb{R}^n \times \mathbb{R}^n &\longrightarrow \mathbb{R}^+ \cup \{0\} \\ (\boldsymbol{x}, \boldsymbol{y}) &\longmapsto d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|, \end{aligned} \tag{5}$$

the hyperball of radius $r > 0$ centered in $\boldsymbol{p}$ is defined as $B_r(\boldsymbol{p}) = \{\boldsymbol{x} \in \mathbb{R}^n \text{ s.t. } d(\boldsymbol{x}, \boldsymbol{p}) < r\}$. Figure 3 provides a visual explanation of this hypothesis. It is worth noting that the sampling shape depends on key important elements: the original probability distribution of the sample features, the sampling $\boldsymbol{X}$ and, consequently, the exploited dataset; and function $\phi$, hence, the considered model. Clearly, the concept of approximation introduced in the formulated hypothesis needs to be formalized.

A first proposal consists of analyzing the per-component variances of the random vector $\phi(\boldsymbol{x})$. Assuming that $\phi(\boldsymbol{x})$ is a zero mean vector (otherwise, the mean can be subtracted), its (sample) covariance matrix can be computed as

$$\boldsymbol{C}_{\phi(\boldsymbol{X})} = \frac{1}{|\mathcal{T}_t^i| - 1} \phi(\boldsymbol{X}) \phi(\boldsymbol{X})^T. \tag{6}$$

Hence, the considered variances can be identified in the diagonal terms of $\boldsymbol{C}_{\phi(\boldsymbol{X})}$; let

$$\boldsymbol{\sigma} = [\sigma_1^2, \ldots, \sigma_n^2], \tag{7}$$

be the vector containing these terms and

$$\tilde{\boldsymbol{\sigma}} = [\tilde{\sigma}_1^2, \ldots, \tilde{\sigma}_n^2] \tag{8}$$

$$= \left[ \frac{\sigma_1^2}{\sum_{i=1}^n \sigma_i^2}, \ldots, \frac{\sigma_n^2}{\sum_{i=1}^n \sigma_i^2} \right], \tag{9}$$

be its normalized counterpart. Two borderline cases can therefore emerge from the analysis of $\tilde{\boldsymbol{\sigma}}$:

$$\tilde{\sigma}_i^2 = \frac{1}{n}, \ \forall i \in [1, n], \tag{10}$$

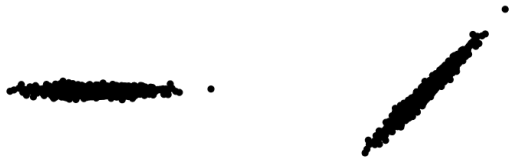
is the best approximation of the introduced hyperball and

$$\exists i \in [1, n] \text{ s.t. } \tilde{\sigma}_j^2 = \begin{cases} 1 \text{ if } j = i, \\ 0 \text{ otherwise}, \end{cases} \tag{11}$$

is the worst one. The former case characterizes samples that are homogeneously spread across the $n$ dimensions while the latter characterizes samples that are spread along a preferential dimension. At this point, an aggregate score of the $\tilde{\boldsymbol{\sigma}}$ terms needs to be computed in accordance with the approximation introduced in (10) and (11). Consequently, the concept of entropy is borrowed from Information Theory for the purpose. Let $H(\boldsymbol{p}) = -\sum_{i=1}^n p_i \log_2 p_i$ be the entropy of the generic distribution $\boldsymbol{p} = [p_1, \ldots, p_n]$. With reference to the framework of Section III, the proposed metric is defined to be

$$V(\boldsymbol{C}_{\phi(\boldsymbol{X})}) = H(\tilde{\boldsymbol{\sigma}}), \tag{12}$$

where the state $s_{t+1}^i$ is set to be $\boldsymbol{C}_{\phi(\boldsymbol{X})}$ and the vector $\tilde{\boldsymbol{\sigma}}$ can be straightforwardly obtained from the diagonal of matrix



(a) $\boldsymbol{C}_{\phi(\boldsymbol{X})_{0^\circ}} = \begin{bmatrix} 1 & 0 \\ 0 & 0.01 \end{bmatrix}$ (b) $\boldsymbol{C}_{\phi(\boldsymbol{X})_{45^\circ}} = \begin{bmatrix} 0.505 & 0.495 \\ 0.495 & 0.505 \end{bmatrix}$

Figure 4. Rotated versions of the same set of samples. The two cases lead to different aggregated scores.

$\boldsymbol{C}_{\phi(\boldsymbol{X})}$. It is easy to prove that (10) leads to the maximum value reachable by metric (12),

$$H\left(\left[\frac{1}{n}, \ldots, \frac{1}{n}\right]\right) = -\sum_{i=1}^n \frac{1}{n} \log_2 \frac{1}{n} \tag{13}$$

$$= \sum_{i=1}^n \frac{1}{n} \log_2 n \tag{14}$$

$$= \log_2 n, \tag{15}$$

while (11) leads to the minimum one, 0. Note that, in the presented scenario, the original entropy meaning is abandoned. The measure, indeed, is only exploited in order to quantitatively describe the shape of the considered samples.

Here, a subtle problem arises. The basis in which the set of deep representations is expressed could not be the most meaningful one according to the way in which the proposed metric is computed. In other words, rotated versions of the same sampling could lead to different aggregated scores; certainly, such behavior is not desired. Figure 4 shows a concrete example of the mentioned scenario. The samples in Figure 4b, $\phi(\boldsymbol{X})_{45^\circ}$, are obtained from the ones in Figure 4a, $\phi(\boldsymbol{X})_{0^\circ}$, through $\phi(\boldsymbol{X})_{45^\circ} = \boldsymbol{R}_{45^\circ} \phi(\boldsymbol{X})_{0^\circ}$, with

$$\boldsymbol{R}_{45^\circ} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \tag{16}$$

Consequently, $\boldsymbol{C}_{\phi(\boldsymbol{X})_{45^\circ}}$ can be computed as $\boldsymbol{C}_{\phi(\boldsymbol{X})_{45^\circ}} = \boldsymbol{R}_{45^\circ} \boldsymbol{C}_{\phi(\boldsymbol{X})_{0^\circ}} \boldsymbol{R}_{45^\circ}^T$. As reported by the captions, ${\sigma_{0^\circ}}_x^2 \gg {\sigma_{0^\circ}}_y^2$ while ${\sigma_{45^\circ}}_x^2 = {\sigma_{45^\circ}}_y^2$ leading to two different aggregated scores.

A possible solution to overcome the issue is inspired by Principal Component Analysis (PCA) [19]. The linear relationship shown in Figure 4b is measured by the off-diagonal terms of $\boldsymbol{C}_{\phi(\boldsymbol{X})}$ (i.e., the covariances). The larger the magnitudes of the terms, the higher the redundancy associated to the data. The goal, therefore, becomes to re-express the original sampling $\phi(\boldsymbol{X})$ into $\boldsymbol{Y} = \boldsymbol{R}\phi(\boldsymbol{X})$ according to a new orthonormal basis (i.e., a rotation) in which the covariance magnitudes related to $\boldsymbol{C}_{\boldsymbol{Y}}$ are minimized: matrix $\boldsymbol{C}_{\boldsymbol{Y}}$ should be diagonal. For a symmetric matrix $\boldsymbol{A}$, the following decomposition holds [19]:

$$\boldsymbol{A} = \boldsymbol{E} \boldsymbol{\Lambda} \boldsymbol{E}^T, \tag{17}$$

where $\boldsymbol{E}$ is a matrix whose columns are the orthogonal eigenvectors of $\boldsymbol{A}$ and $\boldsymbol{\Lambda}$ is a diagonal matrix. Recognizing that $\boldsymbol{C}_{\phi(\boldsymbol{X})}$ is symmetric [19] and setting $\boldsymbol{A} = \boldsymbol{C}_{\phi(\boldsymbol{X})}$, $\boldsymbol{R} = \boldsymbol{E}^T$ can be identified as the required solution (the orthogonal

eigenvectors stored in $\boldsymbol{E}$ can always be normalized in order to obtain an orthonormal change of basis):

$$\boldsymbol{C_Y} = \frac{1}{|\mathcal{T}_t^i| - 1}\boldsymbol{Y}\boldsymbol{Y}^T \tag{18}$$
$$= \frac{1}{|\mathcal{T}_t^i| - 1}(\boldsymbol{E}^T\phi(\boldsymbol{X}))(\boldsymbol{E}^T\phi(\boldsymbol{X}))^T \tag{19}$$
$$= \frac{1}{|\mathcal{T}_t^i| - 1}\boldsymbol{E}^T\phi(\boldsymbol{X})\phi(\boldsymbol{X})^T\boldsymbol{E} \tag{20}$$
$$= \boldsymbol{E}^T\left(\frac{1}{|\mathcal{T}_t^i| - 1}\phi(\boldsymbol{X})\phi(\boldsymbol{X})^T\right)\boldsymbol{E} \tag{21}$$
$$= \boldsymbol{E}^T\boldsymbol{C}_{\phi(\boldsymbol{X})}\boldsymbol{E} \tag{22}$$
$$= \boldsymbol{E}^T(\boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^T)\boldsymbol{E} \tag{23}$$
$$= (\boldsymbol{E}^T\boldsymbol{E})\boldsymbol{\Lambda}(\boldsymbol{E}^T\boldsymbol{E}) \tag{24}$$
$$= \boldsymbol{\Lambda}, \tag{25}$$

where (17) is exploited in (23). It is important to highlight how the diagonal terms of $\boldsymbol{\Lambda}$ (i.e., the eigenvalues of $\boldsymbol{C}_{\phi(\boldsymbol{X})}$), denoted as

$$\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n], \tag{26}$$

represent the variances associated to the sampling $\phi(\boldsymbol{X})$ expressed in the new selected basis.

Let

$$\tilde{\boldsymbol{\lambda}} = [\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n] \tag{27}$$
$$= \left[\frac{\lambda_1}{\sum_{i=1}^n \lambda_i}, \ldots, \frac{\lambda_n}{\sum_{i=1}^n \lambda_i}\right], \tag{28}$$

be the distribution extracted from $\boldsymbol{\lambda}$. The final proposal, therefore, consists of modifying (12) into

$$V(\boldsymbol{C}_{\phi(\boldsymbol{X})}) = H(\tilde{\boldsymbol{\lambda}}). \tag{29}$$

Again, the borderline cases (10) and (11) can be trivially translated into the new setup, as well as the metric minimum and maximum values.

## V. Experiments & Results

The presented experiments, and the respective results, aim to verify the hypothesis formulated in Section IV, and preliminarily investigate the employability of the defined metric in a real application scenario.

### A. Dataset

The experiments exploit the CIFAR-100 dataset [6], a popular benchmark for testing Computer Vision algorithms. The dataset consists of 100 "fine" classes (or sub-classes) containing 600 $32 \times 32$ pixel color images each. All the sub-classes are grouped into 20 "coarse" classes (or super-classes). Moreover, CIFAR-100 is divided into 50000 training images and 10000 testing images.

### B. Classifier

The DeepNCM classifier [20] is selected for the experiments. The model is a distance-based classifier that assigns a sample to the class with the closest mean:

$$F_t(\boldsymbol{x}) = \arg\max_{i \in \mathcal{K}_t} -d(\phi(\boldsymbol{x}), \boldsymbol{\mu}_{t-1}^i) \tag{30}$$
$$= \arg\min_{i \in \mathcal{K}_t} d(\phi(\boldsymbol{x}), \boldsymbol{\mu}_{t-1}^i), \tag{31}$$

where

$$d(\phi(\boldsymbol{x}), \boldsymbol{\mu}_{t-1}^i) = (\phi(\boldsymbol{x}) - \boldsymbol{\mu}_{t-1}^i)^T(\phi(\boldsymbol{x}) - \boldsymbol{\mu}_{t-1}^i), \tag{32}$$

and

$$\boldsymbol{\mu}_{t-1}^i = \frac{1}{|\mathcal{T}_{t-1}^i|}\sum_{\boldsymbol{x} \text{ s.t. } (\boldsymbol{x},i) \in \mathcal{T}_{t-1}^i} \phi(\boldsymbol{x}). \tag{33}$$

The incremental update of the model is granted by (33). The exploited implementation of DeepNCM relies on ResNet for the extraction of the deep representations. Hence, function $\phi$ corresponds to the network layers that precede the classification one, as anticipated in Section IV.

It is important to highlight that the class means $\{\boldsymbol{\mu}_t^i\}$ and covariance matrices $\{\boldsymbol{C}_{\phi(\boldsymbol{X}_t^i)}\}$ can be updated sequentially [21] according to:

$$\boldsymbol{\mu}_t^i = \frac{|\mathcal{T}_{t-1}^i|\boldsymbol{\mu}_{t-1}^i + \boldsymbol{x}}{|\mathcal{T}_{t-1}^i| + 1}, \tag{34}$$

$$\boldsymbol{C}_{\phi(\boldsymbol{X}_t^i)} = \frac{|\mathcal{T}_{t-1}^i| - 1}{|\mathcal{T}_{t-1}^i|}\boldsymbol{C}_{\phi(\boldsymbol{X}_{t-1}^i)} + \frac{1}{|\mathcal{T}_{t-1}^i| + 1}(\boldsymbol{x} - \boldsymbol{\mu}_{t-1}^i)(\boldsymbol{x} - \boldsymbol{\mu}_{t-1}^i)^T. \tag{35}$$

Therefore, with reference to the framework of Section III, the additional state information of the model can naturally be set to $s_{t+1}^i = \boldsymbol{C}_{\phi(\boldsymbol{X}_t^i)} = U(\boldsymbol{C}_{\phi(\boldsymbol{X}_{t-1}^i)}, \boldsymbol{x}) = U(s_t^i, \boldsymbol{x})$. Re-computing class means and covariance matrices by scratch, indeed, is prohibitively computationally expensive for large amounts of samples.

Hence, the choice of the classifier is motivated by the ease with which the DeepNCM framework can be extended in order to incorporate $\mathcal{S}_t$, $U$ and $V$.

### C. Qualitative Hypothesis Verification

To verify the presented hypothesis, DeepNCM is trained (200 epochs, further details on the training procedure can be found in [20]) on 20 modified CIFAR-100 super-classes, 250 samples per super-class, made of only one randomly selected sub-class. This change is introduced to start the metric computation from an initial set of super-classes that have a low intra-class variability. Subsequently, 5000 unseen samples belonging to the same sub-classes exploited during the model training (i.e., 250 samples per super-class) are supplied to the classifier. After each classification, the experiment assigns the samples to the respective ground truth categories in order to evaluate the metric regardless of the accuracy achieved during the classifier training. The model state is updated and the score produced by the metric computation is stored. Then, the model state is re-initialized. Again, 5000 unseen samples (i.e., 250 per super-class), from randomly chosen sub-classes, different from the ones of the training phase, are supplied to the classifier and the corresponding metric scores are computed and stored.

Note that misclassifications can impact the intra-class variability. The consequence, however, could be mitigated by the labeling process function $L_t$. For example, the function might be able to recognize if the images in $\mathcal{T}_t^i$ belong to the hyponyms of the considered super-class label $i$ in accordance with the exploited external source of information.

The first part of the experiment analyzes the metric behavior in a scenario in which the intra-class variability is expected to remain constant (referred to as "constant"), while the second

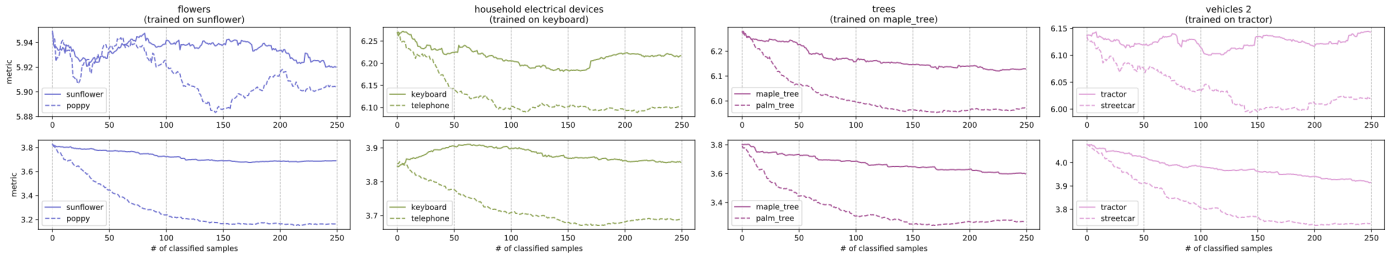



Figure 5. Metric scores for 4 randomly chosen example classes. Top row reports computations with the $V(\boldsymbol{C}_{\phi(\boldsymbol{X})}) = H(\tilde{\boldsymbol{\sigma}})$ definition while the bottom row reports computations with the $V(\boldsymbol{C}_{\phi(\boldsymbol{X})}) = H(\tilde{\boldsymbol{\lambda}})$ definition. Solid lines show the "constant" scenario and dashed lines show the "drift" scenario.

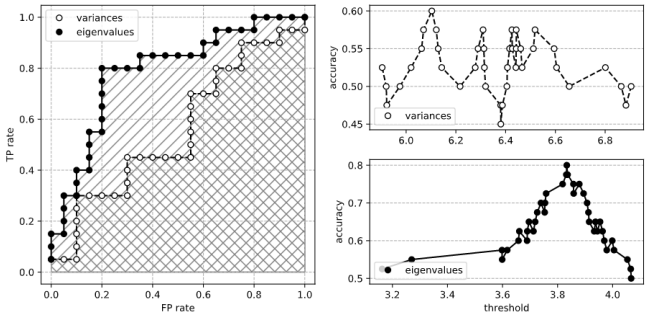



Figure 6. Quantitative evaluation of the considered scores/trigger pairs. The plot on the left reports the produced ROC curves while the plots on the right report the computed accuracies. White dotted lines refer to the $V(\boldsymbol{C}_{\phi(\boldsymbol{X})}) = H(\tilde{\boldsymbol{\sigma}})$ definition while black dotted lines refer to the $V(\boldsymbol{C}_{\phi(\boldsymbol{X})}) = H(\tilde{\boldsymbol{\lambda}})$ definition.

part investigates a scenario in which the intra-class variability is expected to increase (referred to as "drift"). Moreover, the metric scores are computed in accordance with both definition (12) and (29); this is necessary to understand the benefits introduced with the computation of the eigenvalues.

Figure 5 shows the obtained results for some example classes. Considering each super-class separately, most cases present lower metric values, under the same number of classified samples, for the "drift" scenario confirming the correctness of the formulated hypothesis with respect to the considered dataset/classifier pair. With reference to the $V(\boldsymbol{C}_{\phi(\boldsymbol{X})}) = H(\tilde{\boldsymbol{\lambda}})$ final metric definition, it is interesting to notice that the "constant" scenario is also characterized by slightly decreasing trends. The limited amount of training samples, indeed, leads to an adjustment of the scores during the testing phase. However, the initial values of the metric are spread into a large interval resulting in a partial overlapping of the curves related to the different tested scenarios; such data represents the legacy of the criterion with which the CIFAR-100 authors decided to collect the images in the different classes. Additionally, the initial scores assume intermediate values between 0 and $\log_2 n = 9$. Hence, it is immediate to infer that the considered initial configuration is placed in an intermediate position between the borderline cases described in (10) and (11).

### D. Quantitative Metric Evaluation

In order to quantitatively evaluate the performance of the defined metric, the separability of the scores associated to the "constant" and "drift" scenarios is investigated. Defining

$$T(V(s^i_{10001})) = \begin{cases} 0 \text{ if } V(s^i_{10001}) \geq \theta \\ 1 \text{ if } V(s^i_{10001}) < \theta, \end{cases} \tag{36}$$

as the family of threshold triggers acting on the metric scores after the 10000 sample classifications of the experiment, with $i \in \mathcal{K}_{10000} = \mathcal{K}^{const} \cup \mathcal{K}^{drift}$ (the super-class labels must be doubled in order to keep track of the model states deleted with the re-initialization performed after the first experiment part) and $\theta \in \mathbb{R}$, a True Positive (TP) is denoted as $V(s^i_{10001})$ s.t. $i \in \mathcal{K}^{drift} \wedge T(V(s^i_{10001})) = 1$ while a True Negative (TN) as $V(s^i_{10001})$ s.t. $i \in \mathcal{K}^{const} \wedge T(V(s^i_{10001})) = 0$. The False Positive (FP) and False Negative (FN) definitions immediately follow. Hence, the investigation is performed by computing the Receiver Operating Characteristic (ROC) curves for both the $V(\boldsymbol{C}_{\phi(\boldsymbol{X})}) = H(\tilde{\boldsymbol{\sigma}})$ and $V(\boldsymbol{C}_{\phi(\boldsymbol{X})}) = H(\tilde{\boldsymbol{\lambda}})$ definitions, and the respective Area Under the ROC Curve (AUC) integrals.

Figure 6 shows the produced ROCs and two additional accuracy evaluations. The computation of the eigenvalues reveals to be necessary with a final AUC of 0.79, a net improvement over the direct use of the per-component variances, characterized by an AUC of 0.56. The statement is also confirmed by the binary accuracy plots, with an accuracy peek of 0.80 for the $V(\boldsymbol{C}_{\phi(\boldsymbol{X})}) = H(\tilde{\boldsymbol{\lambda}})$ definition. Definition $V(\boldsymbol{C}_{\phi(\boldsymbol{X})}) = H(\tilde{\boldsymbol{\sigma}})$, instead, reveals a performance similar to that of a random trigger.

It is important to emphasize the naivety of the family of triggers considered in the evaluation process. Therefore, the presented results leave room for a promising future application of the metric in a real scenario.

## VI. Conclusion

This paper presented a novel lifelong learning framework and metric in order to manage and quantify the intra-class variability of a trained classifier. The proposed work is an important step to extend the life of robots, thus enabling them to operate longer in real uncontrolled environments without the luxury of the closed-world assumption. For future work, we intend to fully implement the introduced framework (i.e., $F_t$, $\mathcal{S}_t$, $U$, $V$, $T$, $L_t$, $R$ and $I_t$) and test the full framework's real-world performance on a robot platform.

## Acknowledgment

The research leading to these results has received funding from the Austrian Science Fund (FWF) under grant agreement No. I3969-N30 (InDex). This work was partly supported by MIUR (Italian Minister for Education) under the initiative Departments of Excellence (Law 232/2016).

## References

[1] S. Thrun and T. M. Mitchell, "Lifelong robot learning," in The Biology and Technology of Intelligent Autonomous Agents, 1995, pp. 165–196.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25, 2012, pp. 1097–1105.

[3] A. Bendale and T. Boult, "Towards open world recognition," in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1893–1902.

[4] M. Mancini, H. Karaoguz, E. Ricci, P. Jensfelt, and B. Caputo, "Knowledge is never enough: Towards web aided deep open world recognition," in Proc. of IEEE International Conference on Robotics and Automation, 2019, pp. 9537–9543.

[5] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," Neurocomputing, vol. 275, 2018, pp. 1261–1274.

[6] A. Krizhevsky, "Learning multiple layers of features from tiny images," Department of Computer Science University of Toronto, Tech. Rep., 2009.

[7] C. Silla and A. Freitas, "A survey of hierarchical classification across different application domains," Data Mining and Knowledge Discovery, vol. 22, 2011, pp. 31–72.

[8] G. A. Miller, "WordNet: A lexical database for english," Communications of the ACM, vol. 38, no. 11, 1995, pp. 39–41.

[9] D. Jurafsky and J. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Upper Saddle River, N.J.: Pearson Prentice Hall, 2008.

[10] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf, "Dropout sampling for robust object detection in open-set conditions," in Proc. of IEEE International Conference on Robotics and Automation, 2017, pp. 3243–3249.

[11] S. Valipour, C. Perez, and M. Jagersand, "Incremental learning for robot perception through hri," in Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017, pp. 2772–2777.

[12] R. Camoriano *et al.*, "Incremental robot learning of new objects with fixed update time," in Proc. of IEEE International Conference on Robotics and Automation, 2017, pp. 3207–3214.

[13] Y. Yang *et al.*, "An improved approach considering intraclass variability for mapping winter wheat using multitemporal MODIS EVI images," Remote Sensing, vol. 11, no. 10, 2019, p. 1191.

[14] M. Frid-Adar *et al.*, "Modeling the intra-class variability for liver lesion detection using a multi-class patch-based CNN," in Proc. of International Workshop on Patch-based Techniques in Medical Imaging, 2017, pp. 129–137.

[15] A. J. Abboud and S. A. Jassim, "Image quality guided approach for adaptive modelling of biometric intra-class variations," in Proc. of SPIE 7708, Mobile Multimedia/Image Processing, Security, and Applications, 2010, pp. 189–198.

[16] D. Paindaveine, "A canonical definition of shape," Statistics Probability Letters, vol. 78, no. 14, 2008, pp. 2240–2247.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. of International Conference on Learning Representations, 2015.

[19] J. Shlens, "A tutorial on principal component analysis," ArXiv, vol. abs/1404.1100, 2014.

[20] S. Guerriero, B. Caputo, and T. Mensink, "DeepNCM: Deep nearest class mean classifiers," in Proc. of International Conference on Learning Representations - Workshop, 2018.

[21] D. Savransky, "Sequential covariance calculation for exoplanet image processing," The Astrophysical Journal, vol. 800, no. 2, 2015.

 ISBN: 978-1-61208-787-0 