GH-CNN: A new CNN for coherent hierarchical classification

Mona-Sabrine Mayouf[1] [0000−0001−8714−0038]
Florence Dupin de Saint-Cyr[1] [0000−0001−7891−9920]

IRIT, Université Paul Sabatier, Toulouse, France
{mouna-sabrine.mayouf,florence.bannay}@irit.fr
http://www.irit.fr

Abstract. Hierarchical multi-label classification is a challenging task implying the encoding of a high level constraint in the neural network model. Before the rise of this field, the classification was done without paying attention to the hierarchical links existing between data. Nevertheless, information relating the classes and subclasses may be very useful for improving the network performances. Recently, some works have integrated the hierarchy information by proposing new neural network architectures (called B-CNN or H-CNN), achieving promising results. However with these architectures, the network is separated into blocks where each block is responsible for predicting only the classes of a given level in the hierarchy. In this paper, we propose a novel architecture such that the whole network layers are involved in the prediction of the entire labels of a sample, i.e., from its class in the top level of the hierarchy to its class in the bottom level. The proposed solution is based on a Bayesian adjustment encoding the hierarchy in terms of conditional probabilities, together with a customized semantic loss function that penalizes drastically the hierarchy violation. A teacher forcing strategy learning is used to enhance the learning quality. Thanks to this approach, we could outperform the state of the art results in terms of accuracy (improved for all levels) and also in terms of hierarchy coherence.

Keywords: Multi-label classification · Artificial neural networks · hierarchical classification · semantic loss function

1 Introduction

Classification is a crucial task in everyday life, it is the first thing that is learned by any living being in order to survive. A smarter task is hierarchical classification since it involves high-level structural knowledge. As recalled in the survey [11], hierarchical classification is used in many applications such as text categorization (where knowing the hierarchy associated to a word may help a user to disambiguate polysemous terms), protein function prediction (where the functions are naturally organized into hierarchies like the Enzyme Commission class hierarchy and the Gene ontology), music genre classification, phoneme classification, 3D shape classification (where some semantic meaning can be assigned to geometry by using an existing class hierarchy like e.g. Princeton Shape Benchmark), classification of emotional speech (Berlin emotional speech database).

In this study we adopt the point of view defended by Silla and Freitas in their survey [11] which defines hierarchical classification as the process of doing classification under the guidance of a pre-established taxonomy, in the context of supervised learning. For instance hierarchical classification is a particular case of structured classification where meta-data is available about the classes organization. The approaches of hierarchical classification can be distinguished according to the depth where it is performed. For instance some works always classify at the level of leaves nodes, these approaches are called *mandatory leaf-node prediction* (MLNP) and others classify at any level of the hierarchy (non mandatory leaf-node prediction). In non-MLNP approaches, a sample can be assigned a label of any level in the hierarchy, it is often done by using a confidence threshold under which no further digging inside the sub-levels of the classification is done.

Another distinction is done about the way the classifier uses the hierarchy:

- *flat classifiers* only aim to give the leaf class, then the hierarchy maybe used a posteriori to deduce all the implicitly assigned ancestor. The drawback of this kind of approaches is that it requires to discriminate among a large number of classes (the leaves of the taxonomy), moreover, the hierarchy is not used to guide the learning.
- *local classifiers* are using the predicted upper-class to narrow the choices of the current class . A disadvantage of these approaches is that an error at a given level will propagate to the sub-levels.
- *global classifiers* are trained on the entire class hierarchy at once and do not perform local training.

In this paper, in order to overcome some limitations of local classifiers and based on the idea that conditional probabilities should play a role to constrain the links between a class and its subclasses, we propose a new architecture called "Globally Hierarchically Coherent"-CNN (GH-CNN) which exploits Bayes' rule and branching CNN yielding a powerful architecture with a well-designed semantic loss function that penalizes the hierarchy violation. The classifier can be considered as global since in the architecture that we propose the whole network is involved in the prediction of the entire label of a sample, i.e., from its class in the top level of the hierarchy to its class in the bottom level. A teacher forcing strategy learning (which uses the ground truth class in order to predict one of its subclass) is used to enhance the learning quality. Thanks to this approach, we could outperform the state of the art results in terms of accuracy and hierarchy coherence for both BreakHis and Fashion MNIST datasets.

2 State of the art about hierarchical classification

Hierarchical multi-label classification (HMC) aims at classifying objects with a set of labels that respects a given hierarchy constraint. In HMC, the classes of objects are organized as a tree where the edges correspond to superclass-subclass links. The goal of HMC is to assign to each object a set of labels corresponding to a path in this hierarchy. We expose in this state of the art some works dealing with the hierarchical classification. These approaches can be categorized into three sets: the branch based CNN approaches,

the local approaches, the approaches that translate the hierarchical constraint inside the loss function.

In the field of branch based CNN approaches, Zhu and Bain [16] introduce the Branch Convolutional Neural Network (B-CNN) which is a CNN with a particular architecture where the first layers are dedicated to coarse class predictions and the last layers to fine class prediction according to a given hierarchical structure of the target classes. The predictions of the different hierarchical levels are then aggregated with a weighted sum of the loss functions associated to each of them. Moreover the learning phase is done by following a curriculum incremental strategy (as in [5]) consisting in successively learning coarse to fine concepts. The authors experiments show that B-CNN improves over the corresponding baseline CNN on the benchmark datasets MNIST, CIFAR-10 and CIFAR-100. This approach is related to the one developed in this article because the structure of the network is a little bit similar, however the loss function used by Zhu and Bain is not used to adjust the output as we do. Similarly, Seo and Shin [10] are using hierarchical classification for recognizing and classifying people's clothing in apparel images. Their proposal is a VGG19 architecture with additional intermediary outputs: the network is able to give three predicted values for a given sample: one at the top level of the hierarchy ("coarse 1 level", "coarse 2 level" and "fine level"). In the approach of Kolisnik et al. [3]: a new architecture, called H-CNN, is introduced based on B-CNN and designed on VGG16 model to classify with hierarchy constraint the images of FashionImage dataset. The model is an extension of the solution proposed in Seo and Shin [10] which separates the neural network into connected blocks where each block is responsible for predicting a class at a given level. The novelty of this article is the conditional probability update where the probabilities of the super-classes are multiplied by a Conditional Probability Weight Matrix (CPWM) in order to guide the classification of the subclasses. The conditional probability update was previously used by Phan et al. [8] to highlight the relationships among diseases in classification of chest X-rays, and also by Taoufiq et al. [14] for urban structure classification. Most of these works exploit the hierarchy structure. The experiments done on Kaggle Fashion Product Images dataset have promising results and enhance the accuracy of fine-classes prediction compared to a simple model and to a B-CNN model without conditional probability adjustment. Note that the primary goal of these approaches is to fine-tune the prediction of the fine-granular classes, it contrasts with our own goal which is to guarantee a respect of the hierarchy and to obtain accuracy both on super-classes and subclasses ; moreover our method is different since we are not partitioning the network in blocks dedicated to some precise level of the hierarchy.

Concerning the local approaches, in [6], Murtaza et al. propose to use hierarchical classification on the BreakHis dataset (described in Example 1), for this aim, they build three classifiers: a binary classifier for predicting if the tissue is benign or malignant, a multi-class classifier for the benign subclasses (A, F, TA, PT) and a second multi-class classifier for the malignant ones (DL, LC, MC, PC). The architecture is a cascade network where the output of the binary classifier guides the choice of the second classifier to use. The approach is a local approach in the taxonomy of Silla and Freitas [11], it separates the network in three parts, our approach takes a different point of view since it uses a single global architecture for all levels which provides more accurate results.

Nevertheless, even if [6] makes mistake on some predictions, its results are completely coherent with the hierarchy (by construction).

Lots of articles deal with using semantic loss function for translating high level knowledge. Among them we can mention the work of Xu et al. [15] which proposes to integrate a Boolean logical constraint into a loss function, called semantic loss function. The article focuses on constraints expressing the exclusive membership to a unique class when using a layer of sigmoid activation functions. In this article, for each sample s, the network provides a vector of n probabilities $x = (\hat{y}_1, \ldots, \hat{y}_n)$ where $\hat{y}_k$ represents the predicted probability that the variables X_k is true for the input s. The semantic loss function $Loss^s$ associated to the constraint (called α) that only one variable X_k should be true (denoted $x \models \alpha$) is defined by $Loss^s = -\log \sum_{x \models \alpha} \prod_{k:x \models X_k} \hat{y}_k \prod_{k:x \models \neg X_k} (1 - \hat{y}_k)$. In the same vein, Giunchiglia and Lukasiewicz propose to impose a hierarchical constraint by designing an appropriate loss function in [1]. Their solution is based on enforcing inclusion between the objects of a class to its superclass: if an object is assigned to a category A, it should also be assigned to its supercategory B ($A \subseteq B$). For that, they adjust the output z_B of the superclass B wrt the output $\hat{y}_A$ of A, by defining $\hat{y}_B = \max(z_B, \hat{y}_A)$. The final loss function is a sum of the loss function concerning the output $\hat{y}_A$ and the loss function concerning the adjusted output $\hat{y}_B$. Even if [1] imposes the respect of the hierarchy, this approach is using a sigmoïd function which is incompatible with the intra-category exclusivity constraint (ICE) presented below.

3 Notations

We consider a dataset $D = \{s_1, s_2, ..., s_n\}$ of n samples, with a hierarchy of labels $\mathscr{C} = \mathscr{C}^1 \cup \mathscr{C}^2 \cup ... \cup \mathscr{C}^C$, the labels are organized in a tree of of depth C, where the more general labels are in the first stratum (or level) $\mathscr{C}^1$ and the most specific ones are in the stratum $\mathscr{C}^C$. Strata are called categories (or levels in the hierarchy). Each stratum $\mathscr{C}^i$ is composed of a number $\mathscr{N}_i$ of classes: $|\mathscr{C}^i| = \mathscr{N}_i$. The classes are uniquely identified by two numbers: the number i of the level and the absolute number of the class in this level (in $[1, \mathscr{N}_i]$): c^i_j denotes the jth class of the hierarchy level i. The hierarchical relations between classes are described by two functions ch (for children) and pa (for parent) where $ch(c^i_j)$ gives the list of the numbers of the classes of level $i+1$ that are subclasses of the class c^i_j, and $pa(c^i_j)$ is the number associated to the superclass of c^i_j in level $i-1$.

The aim of the classification task is to assign to each sample s a multi-label with C labels: $s.label = (c^1, \ldots, c^C)$ where for all level i, $c^i \in \mathscr{C}^i$. It means that the sample s is assigned to the class c^C which is a subtype of the class c^{C-1} which itself is a subtype of c^{C-2} and so on until c^2 is a subtype of c^1.

Example 1 *BreakHis [13] stands for "Breast Cancer Histopathological Images". It is a public dataset of histopathological biopsy images of breath observed by different microscopic magnifications. In BreakHis dataset, D is the set of histopathological images with $|D| = n = 7909$. Each sample s of this dataset is double-labeled. Benign subtypes are Adenosis (A), Fibro Adenoma (F), Tubular Adenoma (TA) and Phyllodes Tumor (PT). Malignant sub-types are Ductal Carcinoma (DC), Lobular Carcinoma (LC), Mucinous Carcinoma (MC) and Papillary Carcinoma (PC). The hierarchy has two levels:*

the category $\mathscr{C}^1$ which represents the tumor type, and the category $\mathscr{C}^2$ which is the category of the tumor subtype. More precisely, $\mathscr{C}^1 = \{B, M\}$ with $\mathscr{N}_1 = 2$ and $\mathscr{C}^2 = \{A, F, TA, PT, DC, LC, MC, PC\}$ with $\mathscr{N}_2 = 8$.

Here, $c_1^1 = B$ and $ch(c_1^1) = \{1, 2, 3, 4\}$, i.e., the subtypes of the benign class are the four first classes of category $\mathscr{C}^2$ namely: A, F, TA and PT respectively corresponding to c_1^2, c_2^2, c_3^2 and c_4^2. Similarly $ch(c_2^1) = \{5, 6, 7, 8\}$ contains the number of the classes of the malign subtypes. Moreover Fibroadenoma is a benign tumor, is translated into $pa(c_2^2) = 1$, while Lobular Carcinoma is malign is translated into $pa(LC) = pa(c_5^2) = 2$.



The following definition expresses two constraints that a hierarchical classifier should respect:

Definition 1 (ICE and ICH constraints) *A classifier is a function that maps a sample s to a vector of **sets** of classes $s.label = (sc^1, sc^2, \ldots, sc^C)$ where for all $i \in [1, C]$ the set $s.label(i) = sc^i$ is the set of classes of the category $\mathscr{C}^i$ that are assigned to s ($sc^i \subseteq \mathscr{C}^i$).*

A classifier complies with the intra-category exclusivity constraint *(ICE) if each sample s of the dataset D has a unique label per category:*

$$\forall s \in D, \forall i \in [1, C], \quad |s.label(i)| = 1, \tag{ICE}$$

In the following, we consider ICE classifiers, hence labels are vectors of singleton sets of classes, thus they are abbreviated into C-uplets of classes (with no curly brackets) instead of C-uplets of singletons of classes .

An ICE classifier complies with the inter-categories hierarchical constraint *(ICH) if for any sample s, its label represents a path from the root to a leaf in the hierarchy i.e. $s.label = (c^1_{j(1)}, c^2_{j(2)}, \ldots, c^C_{j(C)})$ is s.t.:*

$$c^1_{j(1)} \in \mathscr{C}^1 \text{ and } \forall i \in [2, C], j(i) \in ch(c^{i-1}_{j(i-1)}) \text{ and } c^i_{j(i)} \in \mathscr{C}^i \tag{ICH}$$

In other words, at each level i the label of the sample is a number which correspond to a subtype of the label of the sample at level $i-1$, hence this number is in the list of numbers $ch(c^{i-1}_{j(i-1)})$ associated to the children of $c^{i-1}_{j(i-1)}$.

4 An architecture compliant with ICE and ICH

The GH-CNN architecture (see Figure 1) is designed in order to ensure that the hierarchical constraint existing between classes and subclasses (ICH) and the exclusivity in the same category constraint (ICE) hold. The network is composed as follows:

– A set of hidden layers described in Section 4.1.

- A penultimate primary output layer (PNPO) with C outputs : $z^1, z^2, \ldots, z^C$, each output z^i is a vector of length $\mathscr{N}_i$ (containing the $\mathscr{N}_i$ membership probabilities for the sample to belong to each class of $\mathscr{C}^i$), See Section 4.2.
- The final adjusted finer output layer (FAFO) is a layer composed of $C-1$ outputs adjusted from the PNPO corresponding layers. This adjustment is done through a Bayesian update that encodes the hierarchical implication. Each output $\hat{y}^i$ ($i \in [2, C]$) of this layer is an adjustment via the Bayes' rule of the vector z^i from its parent class (see Section 4.3).
- A particular loss function that penalizes both hierarchy violation and classification errors (weighted wrt the depth in the hierarchy), see Section 4.4.

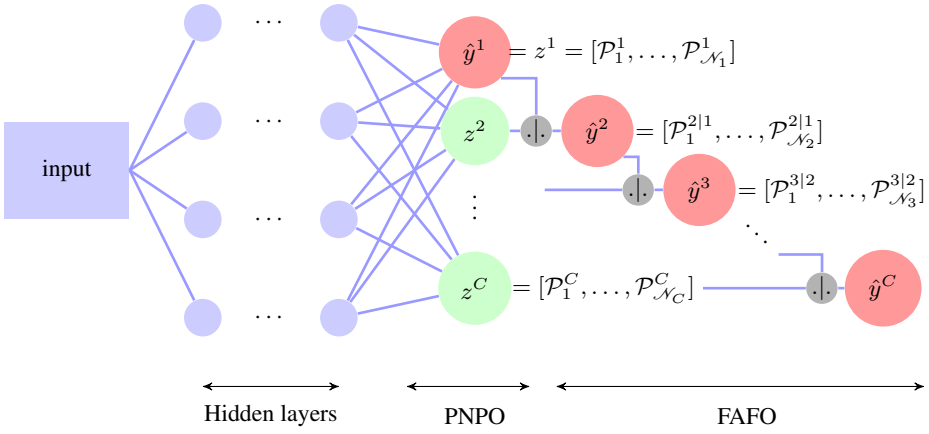


Fig. 1. The GH-CNN architecture. The nodes ⨀ represent the Bayesian adjustement defined by equation (3).

4.1 The hidden layers

Any neural network backbone can be used for the first layers, the particularity of GH-CNN architecture only appears at the penultimate and last layers of the network. In practice, we opted for the VGG19 model as a skeleton for the hidden layers. Our choice is justified by the great performances of this network in several classification tasks [12] (cited more than 70 000 times). This model is a pre-trained CNN, with 19 learnable layers: 16 convolutional layers followed by 3 fully connected layers, with a total of 144M parameters. It was primary used to classify images during the ImageNet competition [2]. The competition's aim is to truly classify an image among 1000 daily-object classes. The huge training set containing millions of images trained for a long training time, gave to VGG19 the powerful ability to recognize daily-objects achieving landmark results [9]. As recalled in Section 2, VGG was also used for fashion images hier-

archical classification by Kolisnik et al [3] and by Seo and Shin [10] with a promising learning behavior.

4.2 The penultimate layer of primary outputs (PNPO)

In the GH-CNN architecture, the PNPO layer produces C penultimate outputs called (y^i) with $i \in [1, C]$, one for each category $\mathscr{C}^i$ present in the dataset. The inputs of PNPO are activated with a softmax function[1]. Each obtained output z^i is a vector of probabilities of $\mathscr{N}_i$ values: $z^i = (\mathcal{P}^i_1, \mathcal{P}^i_2, \dots P^i_{\mathscr{N}_i})$ where $\mathcal{P}^i_j$ corresponds to the probability $\mathcal{P}(s \in c^i_j)$ for a sample s to belong to the j^{th} class of the level $\mathscr{C}^i$ (this class being called c^i_j).

4.3 Updating the probability of a subclass in the layer FAFO

The aim of the FAFO layer is to adjust the vector of probabilities z^i of classes at level i into an output vector of probabilities $\hat{y}^i$. This is done iteratively on the levels by stating that z^1 is already adjusted i.e., $\hat{y}^1 = z^1$ then the adjusted probabilities of level $i+1$: $\hat{y}^{i+1}$ are computed from z^{i+1} accordingly to the adjusted probabilities of the super classes (at level i) $\hat{y}^i$. Indeed, it is clear that the probability vectors z^i and z^{i+1} obtained in the penultimate layer which represent the probability for the sample to belong to classes of two consecutive levels i and $i+1$ in the hierarchy, do not take into account the fact that a class c^i_j of the level i is the parent of a set of classes at level $i+1$, and that this set of classes constitute a partition of the class c^i_j (i.e., $c^i_j = \bigcup_{k \in ch(c^i_j)} c^{i+1}_k$ and for all $k, k' \in ch(c^i_j)$ with $k \neq k'$, it holds that $c^{i+1}_k \cap c^{i+1}_{k'} = \emptyset$).

Let us denote by $\boxed{s^i_j}$ the event that the sample s is associated to the jth class of the level i in the hierarchy, i.e., $s.label(i) = c^i_j$. Due to the partition of the class into its subclasses, the events of associating a sample s to one **subclass of** the class c^i_j is an EME[2]. Hence, for coherence purpose, the layer FAFO is designed to adjust each probability vectors z^i in order to enforce that each adjusted probability vector $\hat{y}^i$ should verify the EME law. Intuitively, the knowledge of the superclass must condition the knowledge of its subclasses. In terms of probabilities, it translates into:

$$\begin{aligned} \mathcal{P}(s^i_j) &= \textstyle\sum_{k \in ch(c^i_j)} \mathcal{P}(s^i_j | s^{i+1}_k) \times \mathcal{P}(s^{i+1}_k) \\ &= \textstyle\sum_{k \in ch(c^i_j)} \mathcal{P}(s^{i+1}_k) \end{aligned} \tag{1}$$

Note that the second equality is due to $\mathcal{P}(s^i_j | s^{i+1}_k) = 1$ when $k \in ch(c^i_j)$, since when a sample belongs to a **subclass of** c^i_j, then this sample should belong to c^i_j itself, which is denoted s^i_j. Now, given a level of the hierarchy $i \in [2, C]$, let us consider the conditional probability, abbreviated $\mathcal{P}^{i|i-1}_j$, for a sample s to belong to a class c^i_j (for

[1] Softmax is an activation function that takes $\mathscr{N}$ inputs $(x_k)_{k=1..\mathscr{N}}$ and gives a probability vector z of dimension $\mathscr{N}$ s.t. its kth component is $z(k) = \frac{\exp^{x_k}}{\sum_{j=1}^{\mathscr{N}} \exp^{x_j}}$.

[2] EME=exhaustive mutually exclusive set of events

any $j \in [1, \mathscr{N}_i]$) knowing that this sample belongs to the parent class of c_j^i (indexed by the number $pa(c_j^i)$ in the level $i-1$). The knowledge about the fact that the sample belongs to the parent class $c^{i-1}_{pa(c_j^i)}$ is assumed to incompass the fact that the sample belongs also to the grand-parent class and to the great-grand-parent class and so on until the root of the hierarchy. Hence, this conditional probability can be expressed as follows:

$$\mathcal{P}_j^{i|i-1} = \mathcal{P}(s_j^i | s^{i-1}_{pa(c_j^i)}, s^{i-2}_{pa(pa(c_j^i))}, ..., s^1_{pa(pa(..(c_j^i))..))}) \tag{2}$$

According to the softmax function done in layer PNPO, $\forall j \in [1, \mathscr{N}_i]$, z_j^i can be viewed as the probability of s to be attributed a label for the level i equal to c_j^i, in short $z_j^i = \mathcal{P}(s_j^i)$, then due to Bayes theorem, we get the equation of the FAFO layer:

$$\hat{y}_j^i = P_j^{i|i-1} = \frac{\hat{y}^{i-1}_{pa(c_j^i)} \times z_j^i}{\sum_{t \in ch(pa(c_j^i))} z_t^i} \tag{3}$$

Finally, the output of FAFO is a vector $(\hat{y}^i)_{i \in [1..C]}$ of vectors such that for any i in $[2, C]$, $\hat{y}^i = [\mathcal{P}_1^{i|i-1}, \ldots, \mathcal{P}_{\mathscr{N}_i}^{i|i-1}]$ and $\hat{y}^1 = z^1 = [\mathcal{P}_1^1, \ldots \mathcal{P}_{\mathscr{N}_1}^1]$ (see Figure 1).

Example 2 *Suppose that, forwarding a sample through a network associated with the hierarchy of Example 1 yields to a probability distribution $\mathcal{P}^1$ of the tumor type equal to* 0.6 *for the Benign class and* 0.4 *for the malignant one: $\mathcal{P}^1 = (0.6, 0.4)$ and $\mathcal{P}^2 = (0.3, 0.025, 0.025, 0.05, 0.2, 0.2, 0.1, 0.1)$.*

After the Bayesian update of $\mathcal{P}^2$, we obtain $\mathcal{P}^{2|1}$= (0.45[3], 0.0375, 0.0375, 0.075, 0.13 , 0.14, 0.066, 0.068). The reader can check that $\mathcal{P}^{2|1}(1) + \mathcal{P}^{2|1}(2) + \mathcal{P}^{2|1}(3) + \mathcal{P}^{2|1}(4) = \mathcal{P}^1(1)$.

At the end of the forward pass through the network, the sample s is assigned to the predicted classes of each category having the maximal probability. The final predicted class inside category $\mathscr{C}^i$ is thus $\hat{c}^i = \text{argmax}(\hat{y}^i)$ where $\text{argmax}(v)$ selects the index i in the vector v such that $v(i)$ is the maximum value in v, and in case of equal maximal values in v, one index is chosen randomly among the maxima. Hence, the definition of $\hat{c}^i$ guarantees ICE.

4.4 Hierarchical loss function

In order to support the ICH constraints and to take into account the different levels of robustness required at different levels of the hierarchy, the loss hierarchical loss function $Loss^{h.v}$ is composed of two parts $Loss^h$ that penalizes the errors with respect to the ground truth, this penalty is weighted according to the hierarchy level, and $Loss^v$ that translates the semantic constraint ICH.

$$Loss^{hv} = Loss^h + Loss^v \tag{4}$$

[3] $0.45 = 0.3 \times 0.6/(0.3 + 0.025 + 0.025 + 0.05)$

Hierarchically weighted loss function: $Loss^h$ is the part that guarantees the learning of the classification at each level of the hierarchy. It is a linear combination of the cross-entropy distance between the prediction and the ground truth at each level:

$$Loss^h = \sum_{i=1}^{C} \alpha_i \times d(y^i, \hat{y}^i), \text{ with } \alpha_i \in \mathbb{N} \tag{5}$$

where $d(v, \hat{v})$ is the cross-entropy between the two vectors v and $\hat{v}$ defined by $d(v, \hat{v}) = \sum_k v(k) \log(\hat{v}(k)) + (1 - v(k)) \log(1 - \hat{v}(k))$. According to the nature of the hierarchy, several configurations of the weights α_i are worth noticing:

- **Egalitarian penalty**: all α_i are equal. The loss function considers equally important the errors done on superclasses or on subclasses (in Section 5, it is implemented with $\alpha_i = 1$ for all i).
- **Superclass/subclass enhanced penalty** These variants are proposed when superclasses (respectively subclasses) are considered as more important than the subclasses (respectively superclasses) for guiding the learning, the weights should be decreasing (respectively increasing) along the hierarchy. In Section 5, it is implemented with $\alpha_i = C - i + 1$ (or $\alpha_i = i$ respectively) for all i.
- **Finest/Coarsest basic model**: We remark that (without the Bayesian update) by setting $(\alpha_1, \ldots, \alpha_{C-1}, \alpha_C) = (0, \ldots, 0, 1)$ (or respectively by setting $(\alpha_1, \alpha_2, \ldots, \alpha_C) = (1, 0, \ldots, 0)$), we obtain the basic neural network that classifies the finest class (respectively the coarsest class) without taking into account its superclasses (respectively its subclasses)

Hierarchy violation loss function We introduce a loss term that penalizes the hierarchy violation: it is the greatest error done on a prediction at a level where the predicted class and subclass are not coherent, (the subclass is not a child of the class).

$$Loss^v = \max_{i \in [1,C] \text{ s.t. } \hat{c}^{i+1} \text{ not child of } \hat{c}^i} \max(d(y^i, \hat{y}^i), d(y^{i+1}, \hat{y}^{i+1})) \tag{6}$$

where $d(v, \hat{v})$ is the cross-entropy distance.

5 Experiments and Results

In this section we expose the experiments done on the BreakHis dataset and on Fashion MNIST. The BreakHis dataset is described in Example 1. The Kaggle Fashion MNIST, is one of the largest hierarchical dataset, with more than 40k images and 3 hierarchy levels. The coarsest category contains 4 classes, its subcategory contains 21 classes, and the finest category contains 45 classes.

For both datasets, the images were resized to 250x250, the chosen operators were label conservative (Horizontal and vertical flip, HSV coloration and color inversion). The datasets were split into 70% for training, 10% for validation and 20% for test.

The training phase is divided into three parts:

- a preliminary warm-up (during 15% of the training phase) where the model is only trained on the coarsest category to ensure a more accurate classification at this level (which will guide the next levels).

- a teacher forcing strategy (during 25% of the training phase) where the ground truth of the superclass is used to guide the learning of a subclass instead of using the predicted results. The teacher forcing strategy is commonly used with recurrent neural networks (RNNs) [4]. This enabled us to adapt the $Loss^{hv}$.
- a training with one variant of $Loss^{hv}$ (during the remaining time).

The algorithms were implemented using the Keras library with Python 3 on Osirim platform [7]. The training period of each experiment contains 1000 epochs, with Adam optimizer, and a train batch size of 128.

We define **the hierarchy violation rate metric (HV)** in order to evaluate the variants of GH-CNN: HV is the number of predicted samples disrespecting the hierarchy divided by the total number of samples in the test set.

Table 1 represents the results obtained using GH-CNN with the different loss functions, with teacher forcing strategy and Bayesian adjustment.

Dataset	**Loss variant**	**Acc**^{1}	**Acc**^{2}	**Acc**^{3}	**F1S**^{1}	**F1S**^{2}	**F1S**^{3}	**HV**	$(L_{\min}, L_{\max})_{\%}$
BreakHis	B-CNN	**98.46**	-	-	-	**95.45**	-	-	-
	Loss^h (1,0)	97.03	76.45*	-	98.69	70.64*	-	58.13	(**0.0035**,1.8744)
	Loss^h (0,1)	85.83*	95.49	-	83.54*	94.26	-	65.14	(0.0058,1.9201)
	Loss$^h(1,1)$	97.65	94.91	-	98.45	91.67	-	11.06	(0.0120,2.1352)
	Loss$^h(2,1)$	98.43	96.78	-	99.01	**95.57**	-	9.15	(0.0177,3.5138)
	Loss^v	67.03	58.91	-	64.30	52.13	-	66.25	(0.2344,**7.3807)**
	Loss$^{hv}(2,1)$	**98.46**	**96.81**	-	**99.11**	**95.45**	-	**4.39**	(0.0037,6.7671)
Fashion MNIST	Loss^h (1,0,0)	**99.98**	85.14*	69.03*	99.12	80.62*	65.64*	69.45	(0.0002,1.0322)
	Loss^h (0,0,1)	80.13*	78.02*	93.12	83.62*	75.64*	93.12	71.43	(0.0091,1.7610)
	Loss^h (1,1,1)	99.47.	86.63	91.79	93.12	99.53	84.35	18.67	(0.3546,6.2380)
	Loss^h (3,2,1)	99.81	88.95	94.61	99.81	86.11	95.41	10.19	(0.0031, 7.1092)
	Loss^v	71.15	58.21	69.43	69.61	60.03	68.53	78.84	(2.0451,8.2751)
	Loss^{hv} (3,2,1)	**99.31**	**98.74**	**95.06**	**99.62**	**99.01**	**94.64**	**3.45**	(0.01984,7.8924)

Table 1. Performances of GH-CNN, the parameters of the Loss functions are inside parenthesis, (α_1, α_2[, α_3]), **Acc**^{i} and **F1S**^{i} are the accuracy and F1 score percentages for classes of level i. **($\mathbf{L}_{\min}$,$\mathbf{L}_{\max}$)** are the minimal and maximal training loss values. The * means that the model was only tested (but not trained) for this level.

- For both datasets, the coarsest class only basic model (with Loss^h (1,0) or (1,0,0)) is less accurate for classifying in the finest class than the basic model trained only on the finest class (even if the first model has a greater performance on the coarsest level). It seems that the model needs to be fine-tuned on finer classes in order to be more efficient. Besides, for both basic approaches, the hierarchy violation rate HV is the highest compared to other loss function variants since there is no information learned about the hierarchical links between classes during the training.
- For both datasets, the egalitarian loss function increases the accuracy. However, in the first half of the training period the loss values are higher than for the basic models. It can be interpreted by saying that the model is learning a more difficult combined task. Note that the HV rate drastically decreases compared with the basic models, since the network is learning simultaneously superclasses and classes, an implicit link has to be discovered.

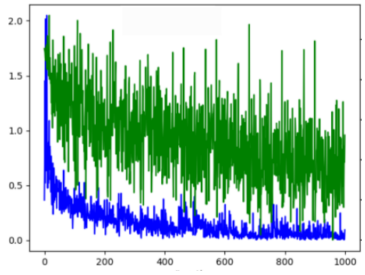


Fig. 2. Training loss curves (HG-CNN with/without Bayesain update in blue/green)

- The superclass enhanced penalty loss function improves the performances on the superclasses, then improves the performances on subclasses due to the Bayesian adjustment.
- The hierarchy violation loss function can only be used as a complement it is not useful per se in terms of classification because the model does not have any feedback about the accuracy of the class predictions. However, the model GH-CNN trained with $Loss^{hv}$ has the greatest performances in term of accuracy and F1-score for all the levels because the hierarchy violation loss forces the CNN to discover and respect the hierarchical link between labels (the hierarchy violation rate is the lowest reaching 4.39% ($< \alpha_{risk} = 5\%$). Hence $Loss^{hv}$ is achieving an accurate and confident classification with coherent hierarchical links.
- Experiments done without the warm-up phase showed unstable Loss values which leads us to believe that warm-up helps the network to find the right starting weights for classifying the coarsest classes, then the teacher forcing strategy improves the learning of the subclasses. We observe that the highest loss is reached with the $Loss^{hv}$ for both datasets, but this high error deceases during the last 25% training time attesting that the network manages the classification of each category separately and integrates the hierarchy between the levels.
- Experiments done without the Bayesian update using the $Loss^{hv}$ showed a very disturbed training loss curve (the green curve in Figure 2[4]) compared to training with the Bayesian update (the blue curve in 2) attesting that the Bayesian curve plays a crucial role in imposing the ICH.
- Comparing with state of the art works, we can remark that, for BreakHis dataset, almost all of the approaches did not pay attention to the hierarchical link between classes, except for [6] where the hierarchy question was addressed (achieving an accuracy of 95.48% of the tumors type detection and 94.62% for the subtypes), while we obtained with $Loss^{hv}$ 98.46% and 96.81% accuracy rates respectively. Concerning the Fashion MNIST dataset, in [10] a B-CNN model was used (giving an accuracy of 93.33% for the finest level). In [3], a conditional probability update was used also with a B-CNN (achieving an accuracy of 99.75%, 98.06% and 91.04%) for the three levels respectively. While we achieved, thanks to the global architecture of HG-CNN, to the Bayesian update and to the well designed loss function, a greater accuracy of 99.71%, 98.94% and 95.06% respectively.

6 Conclusion and perspectives

This paper presents the GH-CNN, a novel architecture, that encodes the labels hierarchy inside the network using both a Bayesian adjustment and a particular loss function penalizing hierarchy violations. GH-CNN outperforms the state of the art results for both BreakHis and Fashion

[4] The vertical axis of Figure 2 is scaled by 3 (e.g. at the first epoch the loss value is 6.76).

MNIST datasets. As a conclusion, GH-CNN is well designed such that all the layers of the network are involved in the determination of the classes at all the levels. Also, the hierarchical coherence is imposed by the Bayesian adjustment before the back-propagation and guarantied thanks to the well-designed loss function. An additional novelty of this paper is the flexibility of $Loss^{hv}$ which can be customized accordingly to the nature of the task. A first perspective of this work is to compare GH-CNN to a CNN where the loss function contains an encoding of the hierarchical constraint as proposed in [15] and a second perspective is about exploring the different combinations of $Loss^{h}$.

References

1. Giunchiglia, E., Lukasiewicz, T.: Coherent hierarchical multi-label classification networks. Advances in Neural Information Processing Systems **33**, 9662–9673 (2020)
2. Imagenet. `http://www.image-net.org`
3. Kolisnik, B., Hogan, I., Zulkernine, F.: Condition-CNN: A hierarchical multi-label fashion image classification model. Expert Systems with Applications **182**, 115195 (2021)
4. Lyu, H., Sha, N., Qin, S., Yan, M., Xie, Y., Wang, R.: Advances in neural information processing systems. Advances in neural information processing systems **32** (2019)
5. Mayouf, M.S., Dupin de Saint-Cyr, F.: Formalizing data preparation in curriculum incremental deep learning on breakhis dataset. Tech. rep., IRIT (2022)
6. Murtaza, G., Shuib, L., Mujtaba, G., Raza, G.: Breast cancer multi-classification through deep neural network and hierarchical classification approach. Multimedia Tools and Applications **79**(21), 15481–15511 (2020)
7. Osirim (observatory of systems information retrieval and indexing of multimedia contents) platform description. `https://osirim.irit.fr/site/`, accessed: 2020-06-18
8. Pham, H.H., Le, T.T., Tran, D.Q., Ngo, D.T., Nguyen, H.Q.: Interpreting chest x-rays via cnns that exploit hierarchical disease dependencies and uncertainty labels. Neurocomputing **437**, 186–194 (2021)
9. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
10. Seo, Y., Shin, K.s.: Hierarchical convolutional neural networks for fashion image classification. Expert systems with applications **116**, 328–339 (2019)
11. Silla, C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery **22**(1), 31–72 (2011)
12. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
13. Spanhol, F.A., Oliveira, L.S., Petitjean, C., Heutte, L.: A dataset for breast cancer histopathological image classification. IEEE Trans. on Biomed. Engineering **63**(7), 1455–1462 (2016)
14. Taoufiq, S., Nagy, B., Benedek, C.: Hierarchynet: Hierarchical CNN-based urban building classification. Remote Sensing **12**(22), 3794 (2020)
15. Xu, J., Zhang, Z., Friedman, T., Liang, Y., Broeck, G.: A semantic loss function for deep learning with symbolic knowledge. In: Int. conf. on machine learning. pp. 5502–5511 (2018)
16. Zhu, X., Bain, M.: B-CNN: branch convolutional neural network for hierarchical classification. arXiv preprint arXiv:1709.09890 (2017)