Applied Intelligence (2021) 51:6622–6632
https://doi.org/10.1007/s10489-020-02103-6

Learn class hierarchy using convolutional neural networks

Riccardo La Grassa[1] · **Ignazio Gallo**[1] · **Nicola Landro**[1]

Accepted: 27 November 2020 / Published online: 8 February 2021

Abstract

A large amount of research on Convolutional Neural Networks (CNN) has focused on flat Classification in the multi-class domain. In the real world, many problems are naturally expressed as hierarchical classification problems, in which the classes to be predicted are organized in a hierarchy of classes. In this paper, we propose a new architecture for hierarchical classification, introducing a stack of deep linear layers using cross-entropy loss functions combined to a center loss function. The proposed architecture can extend any neural network model and simultaneously optimizes loss functions to discover local hierarchical class relationships and a loss function to discover global information from the whole class hierarchy while penalizing class hierarchy violations. We experimentally show that our hierarchical classifier presents advantages to the traditional classification approaches finding application in computer vision tasks. The same approach can also be applied to some CNN for text classification.

Keywords Convolutional neural network · Hierarchical deep learning · Image classification

1 Introduction

In recent years researchers have become increasingly interested in the multi-label and hierarchical learning approaches, finding many applications to several domains, including classification [1, 2], image annotation [3], bioinformatics [4–7]. Nowadays, machine learning is commonly used to solve complex problems, where for example an object is classified by assigning a label based on the rules learned from the model used. However, classes are not always disjoint from others and objects within them can be related to others as a hierarchical structure [8]. Human beings perceive the world with different types of granularity and can translate information from coarse-grained to fine-grained and on the contrary, perceiving different levels of abstraction of the information acquired [9, 10]. This concept is reflected in the taxonomy of the multi-label general approaches under the idea of structured output prediction [11].

In terms of neural models, the main difference between the prediction of structured output and flat multi-label classification lies in the level of neurons that contains the label prediction. In fact, in the presence of a structured output, the information is based on a different level of abstraction, while with the multi-label flat approach it is based on a single level.

Hierarchical multi-label classification (HMC) is a variant of the classification task where instances may belong to multiple classes at the same time and classes are organized in a hierarchy. In HMC approaches a relationship among classes and can be formalized by a tree or directed acyclic graph (DAG).

Our approach to HMC exploits the annotation hierarchy by building a single neural network that can simultaneously predict all categorization of an input source exploiting multiple layers of a neural model. For example, considering the class label prediction for an image containing a tiger, the proposed system can simultaneously predict that a "tiger" has been found but at the same time the same object is also a "feline" and a "mammal".

In literature exists two main approaches to HMC problem, known as local and global [8, 12, 13]. In the global approach, the output of the final layer predicts the test instance in which only one classifier sees information globally without having local information. In the local

✉ Riccardo La Grassa
rlagrassa@uninsubria.it

Ignazio Gallo
ignazio.gallo@uninsubria.it

Nicola Landro
nlandro@uninsubria.it

[1] University of Insubria, Varese, Italy

approach, there is a set of trained classifiers that follows a top-down strategy, in particular, the training process is independently for each base classifier.

Different local approaches have been proposed in the literature, like Local classifier per Node (LCN) [4], Local classifier per parent node (LCPN), Local classifier per level (LCL) [14]. LCN strategy trains a local classifier for each node of a graph providing a local decision to make predictions. LCPN uses a multi-class classifier for each internal class to recognize classes from its sub-classes and LCL methods train a multi-class classifier per hierarchical level. In contrast with local (LCN, LCL, LCPN) and global approaches, we use a single trained model and a single back-propagation error with many different layers fully connected, responsible to synchronize with a concept linked to a given hierarchical structure.

A recent paper [1] describes a novel method to solve HMC problem, that preserves local and global information simultaneously to discover the local hierarchical relationship among classes. Unlike this paper, our architecture exploits recent neural network potentialities and facilitates the multi-class prediction for each deep layers to capture local context following the hierarchical structure of the information. In our approach, we have a cascade of fully connected linear layers each one with softmax plus cross-entropy, where the output of a layer $l-1$ is the input of layer l; instead, in [1] the model has ReLu activation functions on two different layers fully connected with softmax and binary cross-entropy per block. Another difference with [1] is that the input of each layer l fuse with the input, instead, in our approach the input per layer is the output of the previous layer. The last difference is that our model uses local classification as final prediction in according to hierarchical multi-label classification task, instead of in HMCN-F the final layer is used as flat layer plus another layer that uses jointly local and global output information to obtain the final prediction.

In [15] the authors propose an approach called Hierarchical Complement Objective Training (HCOT) which exploits information extracted from the label hierarchy. Their approach maximizes the probability of the correct class and, at the same time, penalizes the probability of the other classes by using the respective class hierarchy as truth. However, they do not consider additional hyper-parameters in the training phase and apply their loss function in the last layer along with the cross-entropy loss function without changes for the CNN topology.

Our method can be summarized in the following key contributions:

- We propose a new Hierarchical Deep Loss (HDL) function as an extension of convolutional neural networks to assign hierarchical multi-labels to images.
- Our extension can be adapted to any CNN designed for classification by modifying its output layer.
- We have conducted experiments on many image classification problems but the same approach can be applied to other classification problems, such as text documents classification.
- We created and released two simple benchmark datasets for hierarchical multi-label image classification.
- To prove the effectiveness of our hierarchical classification approach we conduct empirical studies on many different datasets reporting results in terms of hierarchy recognition and the final accuracy. You can find full code on Gitlab repository [16].

2 The proposed approach

As mentioned above, our solution is an architectural extension that can be adapted to a generic neural network. In this paper, we used a standard Convolutional Neural Network, the ResNet18, as a base model to which we added our solution to solve a hierarchical images classification problem. As graphically represented in Figure 1, what we do is to extend the output layer with a set of new neural layers equal to the number of levels available in the class hierarchy tree of the problem to be solved, and to associate a

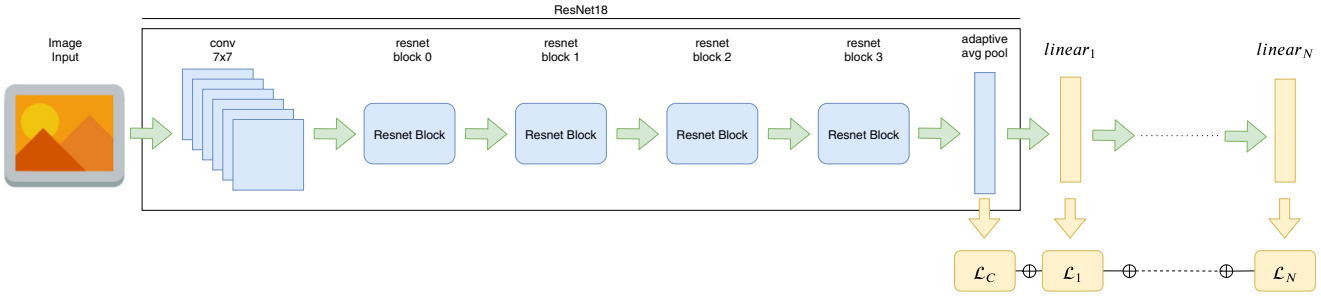


Fig. 1 A graphical representation of the proposed model. The output layer of a CNN (represented here by a Resnet-18) is replaced by N fully connected layers equal to the number of levels available in the class hierarchy tree of the problem to be solved. Each of the N added layers $linear_i$ is associated with a $\mathcal{L}_i$ cross-entropy loss function. $\mathcal{L}_C$ is the center loss function

loss function to each of these new layers added. In practice, we construct a mapping between the layers $1L, \ldots, NL$ of a class hierarchy and the new layers $linear_1, \ldots, linear_N$ of the neural network, as shown in Fig. 1. In this way, the network can learn to discriminate between all class labels belonging to a given layer of the hierarchy. The new layers $linear_1, \ldots, linear_N$ added to the neural network The new added neural layers $linear_1, \ldots, linear_N$ must necessarily contain fully connected layers containing a number of neurons equal to the dimension of the level of the class hierarchy to which it is associated, but they can also be preceded by other neural layers to avoid bottlenecks for very large problems to learn. To minimize the intra-class variance and at the same time to keep the features among different classes separated we compute the Center Loss function [17] on each training mini-batch and update all class centers after each training epoch. More formally we compute the center loss $\mathcal{L}_C$ as follow:

$$\mathcal{L}_C = \sum_{i=1}^{m} \| x_i - c_{y_i} \|_2^2 \tag{1}$$

where $c_{y_i} \in \mathbb{R}^d$ denotes the center for the class y_i in the features space of the deep model. In our experiments, we chose a Resnet-18 as a general model and apply Center Loss after the adaptive pooling layer. More formally, let $linear_g$ be a new added layer with a size equal to the number of classes available at the gL level of the class hierarchy, so we can describe this layer with the following formula

$$linear_g = \phi(W_g x + b_g) \tag{2}$$

where $W_g \in \mathbb{R}^{|linear_g| \times |d|}$, $b_g \in \mathbb{R}^{|l_g| \times 1}$ is the bias vector with ϕ linear activation function and d be the number of features. Then, we add a neural layer $linear_g$ for each hierarchical level gL of a generic dataset in order to perform the cross-entropy loss function to maximize the inter-class variance. More precisely, we apply softmax function from logits of layer $linear_g$ and use the cross-entropy loss function described below in the equation (3).

$$\mathcal{L}_g = -\sum_{i=1}^{m} log \frac{e^{W_{y_i}^T x_i + b_{yi}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \tag{3}$$

where g is the layer g-th, m and n are the mini-batch size and number of classes respectively, $x_i \in \mathbb{R}^d$ denotes the i-th deep feature, belonging to the y_i-th class and b is the bias.

Finally, the total loss function computed in our proposed approach is:

$$\mathcal{L} = \lambda_0 \cdot \mathcal{L}_C + \lambda_1 \cdot \mathcal{L}_1 + \cdots + \lambda_N \cdot \mathcal{L}_N \tag{4}$$

where $\lambda_{0,1,\ldots,N} = 1$, $\mathcal{L}_C$ is the center loss function and $\mathcal{L}_{1,\ldots,N}$ are the cross-entropy loss functions for each level $\{1L, ..., NL\}$ of the class hierarchy. the generic formula of the complete loss function that we propose here is the following

$$\mathcal{L} = \lambda_0 \cdot \mathcal{L}_C + \sum_{l=1}^{N} \lambda_l \cdot \mathcal{L}_l \tag{5}$$

3 Datasets

In this section, we briefly describe all the datasets used in this paper and available in the literature. We also introduce two new datasets useful for further evaluating the proposed approach.

Medical Visual Question Answering task (VQA-Med 2019) [7] is focused on radiology images (some examples are showed in Fig. 2) grouped in four main classes: Modality, Plane, Organ system, Abnormality. The original challenge of this dataset is to classify an image starting from a question connected to it, in fact for each image of the training set we have a combined question. In this paper, our focus is on the multi-label hierarchical classification of images, so in our experiments, we ignore the text classification task. We use all images in the training set to train the model while we use the validation set as a test set because the test

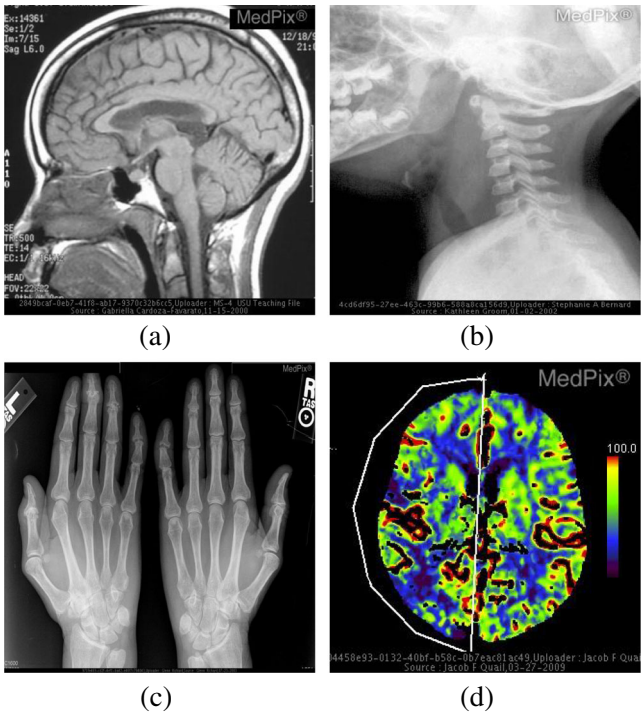


Fig. 2 Four images extracted from VQA-Med 2019 [7] dataset (synpic371, synpic10103, synpic16486, synpic48315). The labels separated by comma belong to the sub-categories of the three main classes following the order: Modality, Plain, Organ

set is not labelled with all labels. The training set contains 2816 images while the validation set contains 340 images. In total, we consider three levels of hierarchy (Modality Class, Plane Class, Organ Class) with their related different types of concepts. These three levels of class hierarchy are respectively 44, 15, 10 in size. In these experiments, our goal is to experimentally prove the effectiveness and robustness of our model to discriminate different concepts even in the case in which we have few examples per class on the training set.

We have created a **synthetic geometric shapes dataset** which contains images of two different geometric shapes: triangle and square, representing the first level of the class hierarchy. The next level of the class hierarchy contains $c_1, c_2, \ldots, c_6$, which are six different colours used as the fill colour of the geometric figure. The last level of the hierarchy contains six other different colours $s_1, s_2, \ldots, s_6$, used to draw a coloured outline for the geometric figure. The Fig. 3) shows some examples of images of this dataset, together with its class hierarchy.

Consequently, the different possible configurations that can be obtained from this class hierarchy is equal to 72. The dataset contains 20,000 training images and 6,000 images used for testing. The images size is 128x128x3. In these experiments, we want to answer the question "Which kind of shape is this? What is the fill colour? and the out fill colour?.

AnimalsTaxonomy8 is a dataset created by us starting from images of animals downloaded from Flickr, the hierarchy represents a small taxonomy with class, family, and species, as shown in Fig. 4. The two **classes** that we selected at the first level of the hierarchy are *mammalia* and *reptilia*. In the second level of the hierarchy (**family**), we selected *felidae* and *ursidae* for mammalia and *crocodyle*, *iguanidae*, *emydidae* and *pythonidae* for reptilia. The last hierarchical level represents **species** as *malaysia tiger*, *felis catus* known as cat, *ailuropoda melanoleuca* known as giant panda, *ursus maritimus* known as polar bear, *python molurus* known as green python, *trachemys scripta* as small turtle, *iguana iguana* and *crocodylus niloticus* known as crocodile nilus. An entire schematic representation of the dataset can be seen in Fig. 5 while some examples of images contained in the dataset are shown in Fig. 6.

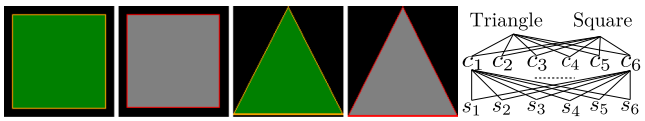


Fig. 3 Some examples of images belonging to the synthetic geometric shapes dataset. On the left, two examples of squares, followed by two examples of triangles with different colours used for the filling of the figure and for the outline of the same. On the right, the class hierarchy of this dataset. $c_1, c_2, \ldots, c_6$ and $s_1, s_2, \ldots, s_6$ are two sets of different colors

ImageNet is the dataset most widely used for images classification. The main reason is due to the high number of classes and the inhomogeneous of the images. We conduct an experiment using the *ILSVRC2012* [18] version of this dataset, containing 1000 classes and 1.2 million images (see Fig. 7). This is a huge hierarchical dataset available in the literature where all the annotation labels are extracted using WordNet [19]. WordNet is a large lexical database of English who contains Nouns, verbs, adjectives, adverbs, and they are grouped into sets of cognitive synonyms called synsets. This lasts are linked using conceptual-semantic and lexical relations.

Cifar100 [20] is a dataset commonly used in the literature as a benchmark for image classification. It contains 60,000 training samples and 10,000 test samples, each sample is a 32 × 32 RGB image and it is divided into 100 different classes (see Fig. 7). We conduct experiments on this dataset because is possible to extract different hierarchies on many levels of depth (see results in experiment 6).

4 Experiments

To evaluate various aspects of our approach to hierarchical classification, we conducted the following experiments on different datasets:

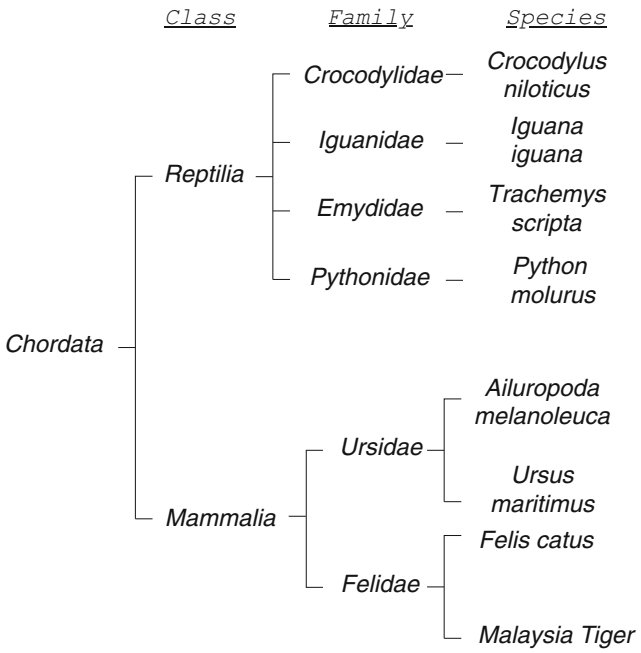


Fig. 4 Class hierarchy of the *Animals_Taxonomy8* dataset we created

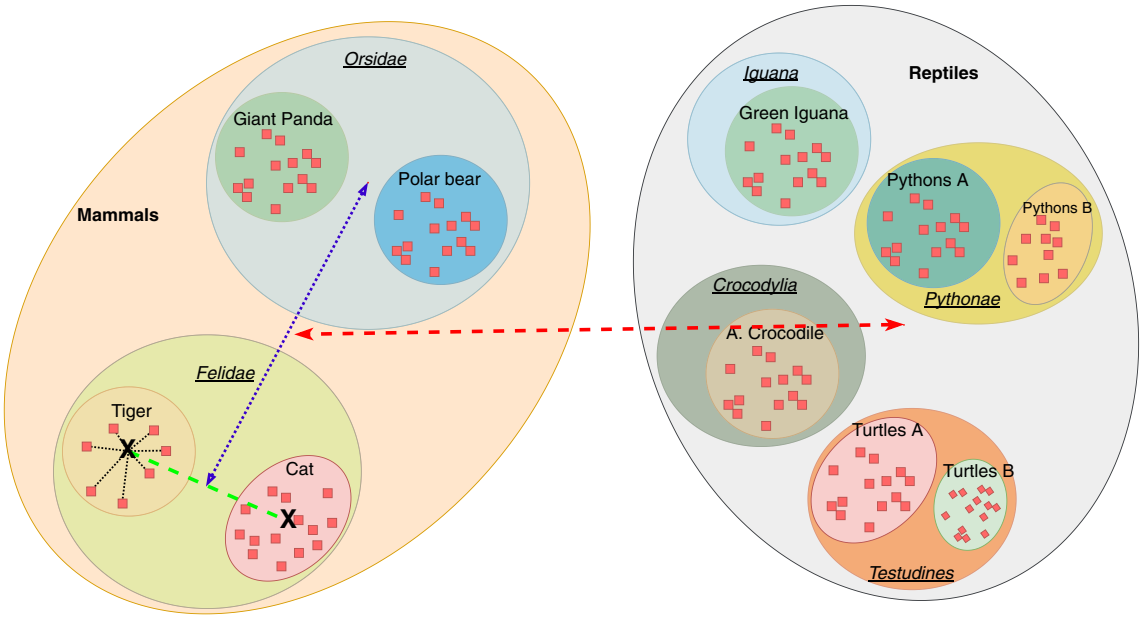


Fig. 5 Hierarchy of categories used in *Animals_Taxonomy8*

1. To analyze the behaviour of our method under conditions of a low amount of data available, in the first experiment we use the well-known dataset VQA-Med 2019 containing biomedical real images and the proposed *Animals_Taxonomy8* dataset.
2. In the second experiment we test the abstraction capacity of our approach on large datasets and at the same time we try to estimate the best learning rate to train the proposed model.
3. In the third, we extract hierarchical structure on the real-images dataset contains images of three types of animal taxonomy levels (Class-Family-Species) and prove the robustness of our HDL in the case which images are hard to recognize and they contain noise.
4. In the four experiments, we compare our HDL with a ResNet18 proving the effectiveness of our approach.
5. In the fifth experiment, we use WordNet to extract all category annotations from the ImageNet dataset to investigate the ability of our approach to obtain a correct hierarchy classification in case a non-hierarchical model makes a mistake in a way to do not lose the entire information. We also want to demonstrate that if the dataset is huge, we can always leverage the levels for the hierarchical classification proposed in this paper, using a pre-trained model to reduce training times. Furthermore we report the precision per class showing an high ability to hierarchy recognition.
6. We conduct experiments on Cifar-100 using HDL on variants of Resnet and compare with non-hierarchical models. We use WordNet to extract three different levels of hierarchy reporting the accuracy.

Experiment 1 In this experiment, we test our model in a situation where we have a few instances with high image complexity. We are interested in analyzing the performance in terms of the accuracy of the various $linear_1, \ldots linear_n$

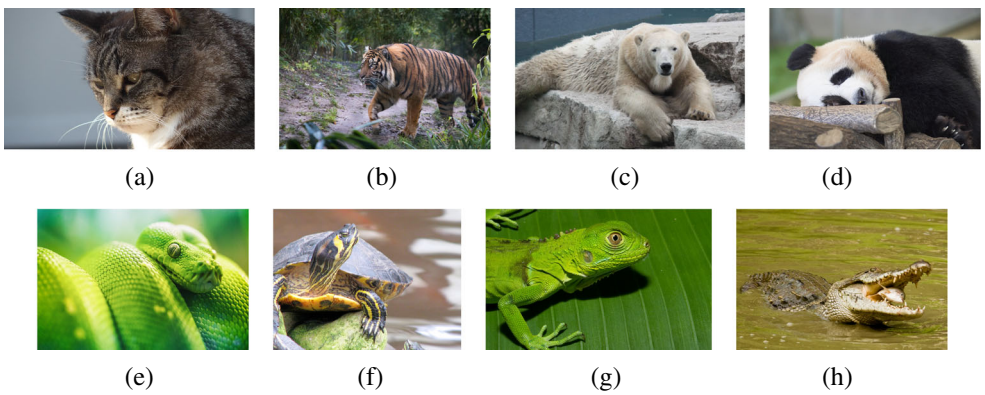


Fig. 6 An example image for each species of the class hierarchy represented in Fig. 4 and extracted from the dataset *Animals_Taxonomy8* proposed by us

Fig. 7 Some sample images extracted from the Cifar-100 dataset (on the left) and from the ImageNet dataset (on the right)

layers introduced. The hypothesis we are interested in verifying is that the classification accuracy is greater when the number of different concepts to be distinguished is lower. To do this we use the VQA-Med 2019 and Animals_Taxonomy8 datasets. As shown in the Table 4 in the first row (VQA-Med 2019), we have an accuracy of 38.05%, 74.04% and 66.66% for the levels having 44, 15 and 10 classes respectively. We can, therefore, observe that the accuracy of the first level is 1.94 times lower than the second level and 1.75 times lower than the third level, this demonstrates that our model offers better scalability when we have few concepts per level to learn. Similar results can be found in Animals_Taxonomy8, where the greater accuracy of level 3L, shown on the third row of the Table 4, compared to others, is because we only have two concepts (mammals or reptiles) to be distinguished compared to the 8 concepts of level 2L (see Figs. 8 and 9 for details).

Experiment 2 In a second experiment, we use our *synthetic geometric shapes* dataset containing several instances. In fact, the size of this dataset is 7.10 times greater than the VQA-Med 2019 dataset. We want to show that using more instances per class can improve the accuracy of our

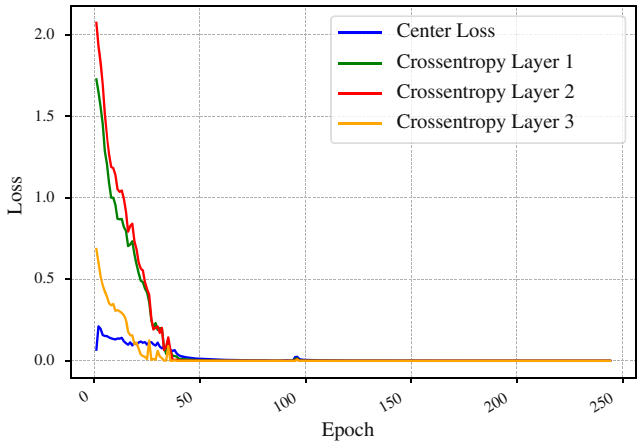


Fig. 9 Training losses *Animals_Taxonomy8* with $lr = 0.005$. We emphasize the different descent of losses. This is due to the number of concepts to distinguish from each layer. Each line represents the loss for each layer. For this dataset, we design our model with a shape of 6, 8, 2 to distinguish Family, Species and Classes respectively. As we show also in 4, the line yellow that represents linear layer 3 with 2 concepts (Mammals or Reptiles) has more descent power, indicating that our model quickly learns a few concepts rather than many as red line or green

model and subsequently, and then we can obtain better performance compared with a configuration like the first experiment. To prove that, we train with 20K instances the proposed HDL model and test it with 6k instances. The results are shown in Table 4 and confirm our expectations. The higher number of instances jointly with the simplicity of images allows the model to reach high accuracy starting from the first ten epochs. Furthermore, we conduct three different runs of this experiment jointly with the experiment conducted in Experiment 1, changing the learning rate from 0.005 to 0.01 and using a batch-size of 64. As can be observed from Table 4, there is no particular rule on the choice of the learning rate.

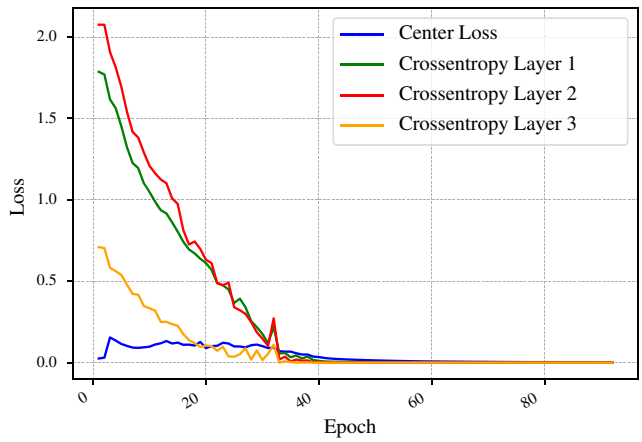


Fig. 8 Training losses *Animals_Taxonomy8* with $lr = 0.01$

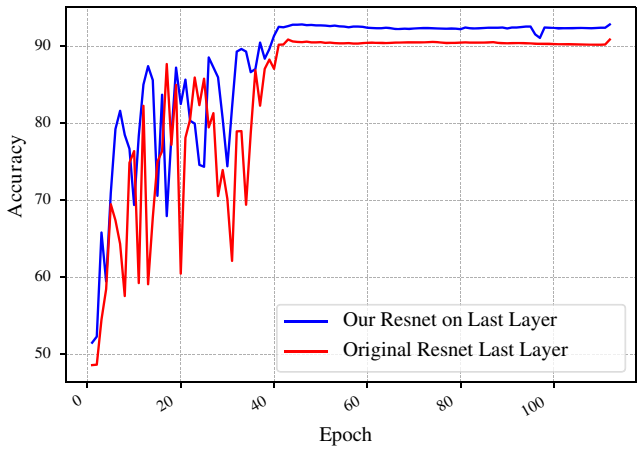


Fig. 10 HDL vs original Resnet18 with $lr = 0.005$

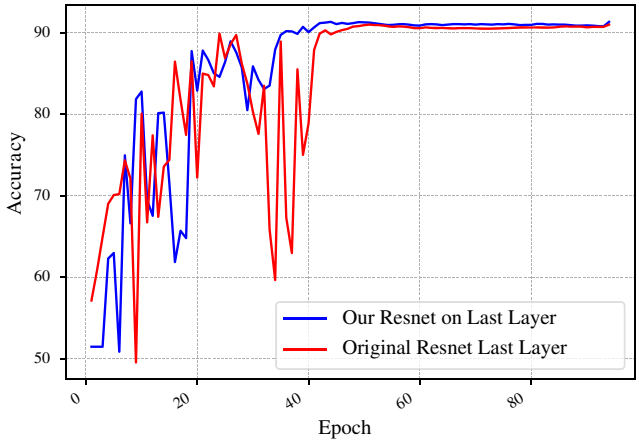


Fig. 11 HDL vs original Resnet18 with $lr = 0.01$

Experiment 3 In these experiments, we test our model using more instances than the first experiment and with images of animal (*Animals_Taxonomy8*) that contains noise. In particular, our model offers good performance also in the case the images are not simple as in the second experiments and when they contain noise or offers little comprehensibility, indeed many images are not clear, like for example a snake completely hidden by forest or a bar sign with a panda logo. However, as we show in Table 4, the accuracy of the third level 3L, responsible to recognize mammals or reptile is very high. We conclude that considering the poor understanding of images, noise and hard images to recognize, experimental results prove the robustness of our model.

Experiment 4 HDL is designed to maximize the learning capacity and to extract the hierarchical structure from the labelled data. Our intuition is that our model, lead to different losses at any level, with the power to reduce intra-variance and to maximize inter-variance, can obtain better accuracy than a classical convolutional neural network. To prove this, we conduct six different experiments using a classic ResNet18 and our HDL on *Animals_Taxonomy8* using two learning rate and a batch size of 64. The results showed in Table 5 and Figs. 10 and 11, clearly confirm our expectations. In all cases, the accuracy is higher than a classical ResNet18, this experiment proves the effectiveness of our proposed model.

Experiment 5 In this experiment, we use the output of a pre-trained Resnet18 on ImageNet extended with the deep stack of fully connected layers. Since the ImageNet class hierarchy was extracted from WordNet, we directly use it to generate the class hierarchy for all ImageNet samples.

Then, from each class of 1000 available on ImageNet, we find the path of its hypernyms. Starting from the root, the size of the first five levels in the class hierarchy obtained from ImageNet-2012 is [2, 11, 28, 61, 107]. In this experiment we chose only the fifth level with 107 classes, some of which are for example: organism, living being, sound, flesh, physical condition, color, instrumentality, commodity, fish, district, boxer, European and so on. The goal of this experiment is to count how many instances are misclassified by the pre-trained model considering the 1000 classes of the dataset, but at the same time, thanks to our deep stack hierarchy levels, to count how many of these samples are correctly classified using the fifth level of the class hierarchy.

In the Table 1 we report the ability of our approach to extract the right predictions from the fifth level of the class hierarchy even if the pre-trained model makes misclassifications. In the first column, we show the correct prediction value in case the pre-trained model makes a wrong classification. So 3306 is the number of instances that the pretrained Resnet18 misses but with our approach we are able to recognize the correct category among the 107 classes of the fifth level of the hierarchy. (see Table 1 for the details and descriptions of the experiment). This experiment demonstrates the ability of our hierarchical approach to recognize the correct categories of a class hierarchy, even when a classification model fails to recognize the correct

Table 1 Results obtained on the ImageNet-2012 dataset

Correct	Wrong	Other	Pre-trained Acc	HDL Acc
3306	803	16996	76.45% (1000 classes)	88.31% (107 classes)

The leftmost column reports the number of samples misclassified by the pre-trained Resnet-18 model using the 1000 classes of the dataset while for the same 3306 instances our hierarchical model has correctly predicted the class among the 107 available in the fifth level of the hierarchy. The column labeled "Wrong" contains the number of samples correctly classified by Resnet-18 (1000 classes) but where at the same time we are unable to recognize the category (107 classes). The "Other" column reports the number of instances that are correctly/incorrectly classified both on the 1000 leaf classes and on the 107 classes of the hierarchy. In the last column, we show the accuracy calculated in the fifth level of the hierarchy, while in the second-last column we report the accuracy (1000 classes) from the pre-trained model

class. Furthermore, to check the ability to class hierarchy recognition, thanks to the deep stack layers introduced, we compute the precision metric per class (see Table 2),

Table 2 Accuracy per class reported using our hierarchical level of 107 ImageNet classes

Class	Precision	Class	Precision
0	0.914893617021277	1	0.94286154320037
2	0.709677419354839	3	0.888888888888889
4	0.923076923076923	5	0.888888888888889
8	0.892263759086189	9	0.815719947159841
10	0.845528455284553	12	0.96969696969697
13	0.75	14	0.96875
16	0.828571428571429	17	0.869158878504673
18	0.931818181818182	19	0.933333333333333
20	0.91304347826087	21	0.857142857142857
22	0.941176470588235	23	0.91666666666666
24	0.793103448275862	25	0.90625
26	0.796875	27	0.830527497194164
28	0.807692307692308	29	0.733333333333333
30	0.761904761904762	31	0.936708860759494
32	0.948717948717949	33	0.846715328467153
34	0.815384615384615	35	0.875
37	1	38	0.823529411764706
39	0.777777777777778	40	0.705882352941176
42	0.607142857142857	43	0.933333333333333
44	0.735294117647059	45	0.757575757575758
46	0.645161290322581	47	0.772727272727273
50	0.827586206896552	51	0.8125
52	0.852941176470588	53	0.814814814814815
54	0.818181818181818	55	0.807692307692308
56	0.870967741935484	58	0.909090909090909
59	0.84	60	0.909090909090909
61	0.814814814814815	62	0.871794871794872
64	0.858974358974359	65	0.761904761904762
67	0.6875	68	0.857142857142857
69	0.772727272727273	73	0.717948717948718
74	0.941176470588235	76	0.689655172413793
79	0.971428571428571	80	0.758620689655172
81	0.828571428571429	82	0.888888888888889
83	1	84	0.708333333333333
85	0.882352941176471	86	0.827586206896552
88	0.863636363636364	91	0.807017543859649
92	0.525423728813559	93	0.955555555555556
94	0.470588235294118	96	0.612903225806452
97	0.727272727272727	98	0.480769230769231
99	0.761904761904762	100	0.904761904761905
105	0.65		

Some classes have been hidden because, in the randomly drawn test set, we don't have any samples for those classes

showing a high ability of our model to recognize the single categories introduced with the class hierarchy.

In conclusion, we demonstrate the capability of our approach to does not lose the entire information in the case the main model makes a misclassification in a way to preserve the truth of the categories from fine-grained to coarse-grained. For instance, if the main model makes an error to classify a frog instead of a green t-shirt, with our approach, we can recognize the hypernyms of t-shirt like for example, physical entity, object, artifact, covering, clothing, garment. In the last example, the dimension of our deep stack layers is 6 and in general, the dimension depends on all categories we can find at the same level on WordNet.

Experiment 6 We investigate and analyze the behavior of our model on Cifar100, using the same WordNet-based approach of experiment 5, to create the useful annotations to build the categories of the class hierarchy. Differently by [15], that groups the 100 classes of Cifar100 into 20 coarse classes, we extract the class hierarchy automatically using WordNet at different levels of depth. In our experiment we extract the following categories:

- Category 1L: whole, person, phenomenon, body of water, arrangement, substance, solid, geological formation, part, collection, location, land.
- Category 2L: object, causal agent, process, thing, group, matter.
- Category 3L: physical entity, abstraction.

Respectively, we extracted {12, 6, 2} classes per layer from Cifar-100.

In this experiment, we do not use a pre-trained model and we train the whole Resnet model with a deep hierarchical stack. We use three different levels of abstraction from fine-grained to coarse-grained inserted after the final output layer of the baseline model used. Specifically, our deep stack is composed by three different layers $linear_i$, and in particular $linear_1$ contains two fully connected layers of size 100, 12 respectively; $linear_2$ contains two fully connected layers of size 10, 6 respectively; and $linear_3$ contains two fully connected layers of size 4, 2 respectively.

We use ReLu activation function (except for the layers of size 12, 6, 2). In Table 3 we compare variants of ResNet used as baseline with our deep stack layers (1L, 2L, 3L). From this experiment we can conclude that our hierarchical approach, under the conditions of this experiment, does not improve in classification accuracy (100 classes of Cifar-100), however, it can be noted that the introduction of hierarchical stacks leads to a considerably improved classification accuracy for individual layers of the classes hierarchy.

Table 3 Error rates (%) on CIFAR-100 using ResNet and its variants

Model	Baseline	HDL	1L (12 cls)	2L (6 cls)	3L (2 cls)
Resnet-18 [21]	35.70	35.70	12.23	11.43	1.20
PreAct ResNet-18 [22]	27.40	27.40	9.68	9.18	1.0
PreAct ResNet-101 [22]	24.39	24.70	8.79	8.46	1.17
PreAct ResNet-152 [22]	26.18	25.90	9.59	8.95	1.01

1L, 2L, 3L represent 3 hierarchical categories level (coarse-grained) chosen for this experiment

4.1 Experiments settings

We build our hierarchical multi-label classifier model as an extension on a Resnet-18, but is it possible to apply to any Convolutional Neural networks. We implement our extension in Python using Pytorch framework. Figure 1 shows the architecture used for experiments. The size of the input images is re-scaled to 64x64x3 for Geometry dataset and 256x256x3 for *VQA-Med 2019* and *Animals_Taxonomy8* datasets. We do not apply any preprocessing of images as data augmentation, rotation or normalization. The kernel size of the first convolutional layers is 7x7 with a stride of 2 pixels, followed by a normalization of layer and a non-linear layer with ReLu activation function. A max-pooling operation over 3x3 regions and a stride of 2 pixels. Then, we have four blocks of Convolution, with 64, 128, 256, 512 numbers of plans respectively and apply an adaptive average pooling over 1x1 region. Finally, we add three fully connected linear layer, where each layer corresponding to the total number of concepts in our hierarchical dataset. In the forward process, we take the output after the adaptive average pooling and apply Center loss function and for each linear layers we apply softmax function and then cross-entropy loss. The total loss will be the sum of the local loss per layers. Our network was trained with Adam optimizer [23]. The batch-size used, learning rate, epochs are described jointly with the results for each dataset. In the fifth experiment, we use a Resnet18 pre-trained on ImageNet. Then, we introduce two fully connected layers of size 1000 and 107 (categories at fifth level extract by WordNet) and trained only these last two layers. We use the same configuration described before, except for the size of the input images changed to $256 \times 256 \times 3$.

In experiment 6, we use variants of Resnet (see references available in Table 3), a learning rate equals to 0.001, random crop, random horizontal flip as data augmentation and run all the experiments for 800 epochs.

Table 4 Accuracies comparison using three different datasets on different learning rate

	lr=0.005		
Datasets	1L	2L	3L
VQA-Med	38.05% (44)	74.04% (15)	66.66% (10)
Shapes	100% (6)	100% (6)	100% (2)
Animals_Taxonomy8	71.98% (8)	69.07% (6)	92.82% (2)
	lr=0.001		
VQA-Med	35.98% (44)	70.20% (15)	67.84% (10)
Shapes	100% (6)	100% (6)	100% (2)
Animals_Taxonomy8	72% (8)	69.12% (6)	92.89% (2)
	lr=0.01		
VQA-Med	34.8% (44)	71.97% (15)	69.61% (10)
Shapes	100% (6)	100% (6)	100% (2)
Animals_Taxonomy8	69.2% (8)	66.53% (6)	91.32% (2)

We use a batch size of 32 on VQA-MED and 64 on the other datasets. In round brackets the number of classes for that level of the class hierarchy

5 Results and discussion

This study is placed in the sub-category of multi-label classification called Structure output learning. In according with experimental results at Tables 1, 3, 4 and 5 we achieved good results on five different datasets finding the way to exploit the dependency among classes and make accurate categories predictions, reducing the misclassification than a main model. We obtain good results in categories

Table 5 Accuracies comparison of our model with a original ResNet from coarse-grained (1L) to fine-grained (3L)

	lr=0.005			
Our model	1L	2L	3L	ResNet18
	71.98%	-	-	71.19%
Animals_Taxonomy8	-	**69.07%**	-	68.58%
	-	-	**92.82%**	90.86%
	lr=0.01			
Our Model	1L	2L	3L	ResNet18
	69.2%	-	-	68.34%
Animals_Taxonomy8	-	**66.53%**	-	65.36%
	-	-	**91.32%**	90.98%

recognition task on ImageNet dataset showing the flexibility of our approach to be adapted also in a pre-trained network. We get the same results on Cifar-100 using variants of Resnet models on the "leaf" node of the classes hierarchy, however, we show high accuracy on coarse categories recognition allowing our model to get some useful information even when the main model makes misclassifications (Table 3).

6 Conclusion

In literature, multi-label classification is an important field in machine learning and it is strongly related to many real-world applications for example, in biomedical images annotation, document categorization and whatever problem which the instances inside the classes are not disjoint but they keep a hierarchical structure. In this paper, we have conducted widely empirical studies on different datasets to prove by experimental results the effectiveness and robustness of our proposed model, that can be applied as an extension of any Convolutional Neural Network configured for classification tasks.

Acknowledgements The authors kindly appreciate the NVIDIA gift of a Titan Xp GPU for this research.

Funding Open Access funding provided by Università degli Studi dell'Insubria.

References

1. Wehrmann J, Cerri R, Barros R (2018) Hierarchical multi-label classification networks. In: International Conference on Machine Learning, pp 5225–5234
2. Cesa-Bianchi N, Gentile C, Zaniboni L (2006) Incremental algorithms for hierarchical classification. J Mach Learn Res 7:31–54
3. Dimitrovski I, Kocev D, Loskovska S, Džeroski S (2011) Hierarchical annotation of medical images. Pattern Recogn 44(10-11):2436–2449
4. Valentini G (2009) True path rule hierarchical ensembles. In: International Workshop on Multiple Classifier Systems. Springer, pp 232–241
5. Yan X, Li L, Xie C, Xiao J, Gu L (2019) Zhejiang university at imageclef 2019 visual question answering in the medical domain. Working Notes of CLEF
6. Chen H, Miao S, Xu D, Hager GD, Harrison AP (2018) Deep hierarchical multi-label classification of chest x-ray images
7. Abacha AB, Hasan SA, Datla VV, Liu J, Demner-Fushman D, Müller H (2019) Vqa-med: Overview of the medical visual question answering task at imageclef 2019. In: CLEF2019 Working Notes. CEUR Workshop Proceedings, pp 09–12
8. Silla CN, Freitas AA (2011) A survey of hierarchical classification across different application domains. Data Min Knowl Disc 22(1-2):31–72
9. Hobbs JR (1990) Granularity. In: Readings in qualitative reasoning about physical systems. Elsevier, pp 542–545
10. McCalla G, Greer J, Barrie B, Pospisil P (1992) Granularity hierarchies. Comput Math Appl 23(2-5):363–375
11. Su H et al (2015) Multilabel classification through structured output learning-methods and applications
12. Costa EP, Lorena AC, Carvalho AndreCPLF, Freitas AA, Holden N (2007) Comparing several approaches for hierarchical classification of proteins with decision trees. In: Brazilian Symposium on Bioinformatics. Springer, pp 126–137
13. Xu D, Shi Y, Tsang IW, Ong Y-S, Gong C, Shen X (2019) A survey on multi-output learning. arXiv:1901.00248
14. Cerri R, Barros RC, de Carvalho AndreCPLF (2011) Hierarchical multi-label classification for protein function prediction: A local approach based on neural networks. In: 2011 11th International Conference on Intelligent Systems Design and Applications. IEEE, pp 337–343
15. Chen H-Y, Tsai L-H, Chang S-C, Pan J-Y, Chen Y-T, Wei W, Juan D-C (2019) Learning with hierarchical complement objective. arXiv:1911.07257
16. La Grassa R (2020) Hdl. https://gitlab.com/artelabsuper/hdlv3
17. Wen Y, Zhang K, Li Z, Qiao Y (2016) A discriminative feature learning approach for deep face recognition. In: European conference on computer vision. Springer, pp 499–515
18. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet Large Scale Visual Recognition Challenge. Int J Comput Vis (IJCV) 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y
19. Miller GA (1998) Wordnet: An electronic lexical database. MIT press
20. Krizhevsky A, Hinton G et al (2009) Learning multiple layers of features from tiny images
21. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
22. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: European conference on computer vision. Springer, pp 630–645
23. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv:1412.6980

Riccardo La Grassa received the Bachelor's degree (2015) and Master's degree (2018) in Computer Science from University of Palermo with published thesis. He is currently a Ph.D candidate at University of Insubria in Computer Science and Computational Mathematics.

His research activity involves Deep learning especially in Mathematical Optimization of the objective functions.

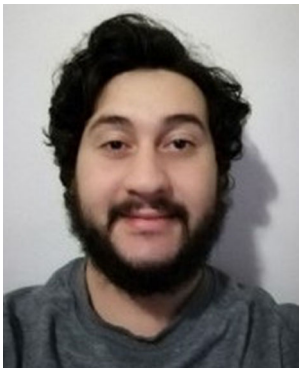
He published in several journals and conferences in the above areas.

Ignazio Gallo is Assistant Professor at the University of Insubria, Varese, Italy. He conducts research in Computer Vision, Deep Learning, Natural Language Processing, Image Processing, Pattern Recognition, and Neural Computing.

He is also interested in the stereoscopic reconstruction of 3D images, in feature selection and classification methods applied to remote sensing data.

His recent interest is in the fields of multimodal applications for classification and retrieval. He leads the Applied Recognition Technology Laboratory (ArTe-Lab) promoting the development and transfer of new technologies between the University and private companies.

He is author or co-author of more than 110 research publications.



Nicola Landro is a PhD student at the University of Insubria, Varese, Italy. He conducts research in Deep Learning, Natural Language Processing, Image Processing and Neural Computing.

He is also interested in Optimizers, Generative Adversarial Networks, Autoencoder, Multimodal, audio and 3D application of neural networks and deep learning.

His recent interest is in the fields of Optimizers and neural network for classification and retrieval. He joins the Applied Recognition Technology Laboratory (ArTe-Lab).