# A Mask-based Output Layer for Multi-level Hierarchical Classification

Tanya Boone-Sifuentes
Deakin University
Geelong, VIC, Australia
tboonesifuentes@deakin.edu.au

Mohamed Reda Bouadjenek
Deakin University
Geelong, VIC, Australia
reda.bouadjenek@deakin.edu.au

Imran Razzak
University of New South Wales
Sydney, NSW, Australia
imran.razzak@unsw.edu.au

Hakim Hacid
Technology Innovation Institute
Abu Dhabi, UAE
hakim.hacid@tii.ae

Asef Nazari
Deakin University
Geelong, VIC, Australia
asef.nazari@deakin.edu.au

## ABSTRACT

This paper proposes a novel mask-based output layer for multi-level hierarchical classification, addressing the limitations of existing methods which (i) often do not embed the taxonomy structure used, (ii) use a complex backbone neural network with $n$ disjoint output layers that do not constraint each other, (iii) consequently, may output predictions that are often inconsistent with the taxonomy in place, and (iv) have often a fixed value of $n$. Specifically, we propose a model agnostic output layer that embeds the taxonomy and that can be combined with any model. Our proposed output layer implements a top-down divide-and-conquer strategy through a masking mechanism to enforce that predictions comply with the embedded hierarchy structure. Focusing on image classification, we evaluate the performance of our method on three different datasets, including CIFAR-100, Caltech BIRDS-200-2011, and Stanford Cars, each with a three-level hierarchical structure. Experiments on these datasets show that our proposed mask-based output layer allows to improve several multi-level hierarchical classification models for various performance metrics.

## KEYWORDS

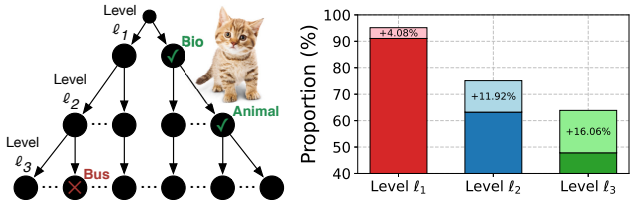
Hierarchical Classification, CNN, Deep Learning.

## 1 INTRODUCTION

Multi-level Hierarchical Classification (MLHC) is a specific classification task, which addresses the problem of classifying items into a multi-level hierarchy structure of classes [1–7]. MLHC has attracted a lot of attention over the past few years, mainly because many real-world applications and services now use a hierarchical structure to organize their data, e.g., online retailers such as Amazon, Wikipedia, DMOZ, etc.

To illustrate and assess the benefit of a MLHC, we refer to Figure 1, which shows (1a) an example of an image classified by a MLHC that has $n$ independent and disjoint output layers, and (1b) the proportion of correctly classified images for each level of our taxonomy of the CIFAR-100 dataset, as well as the proportion of images incorrectly classified but for which the other levels in the taxonomy were correctly identified. There are a few important observations here: (i) First, a MLHC allows to structure large amounts of information using a hierarchical taxonomy, which can be convenient as it allows to describe relations between classes by mean of the "*subclass-of*" notion. (ii) Second, from the example shown in Figure 1a, if we could tell to the last classification layer that the image is a "Bio organism" and an "Animal", we could help it to identify the correct class in $\ell_3$ – or at least being consistent by selecting a *subclass-of* "Animal". Finally, (iii) the results presented in Figure 1b show that 4.08% of images incorrectly classified by $\ell_1$ were correctly classified by $\ell_2$ or $\ell_3$, 11.92% of images incorrectly classified by $\ell_2$ were correctly classified by $\ell_1$ or $\ell_3$, and 16.06% of images incorrectly classified by $\ell_3$ were correctly classified by $\ell_1$ or $\ell_2$. This shows and motivates the potential benefit of a MLHC that embeds the taxonomy structure with a top-down or a bottom-up classification approach.

There have been several methods proposed for MLHC, and they can be categorized according to how the hierarchical structure is




**(a) Taxonomy example.** **(b) CIFAR-100.**

**Figure 1: (a) An image of a "Cat" classified by 3 independent classifiers as a "Bio organism", an "Animal", and incorrectly as a "Bus" for the CIFAR-100 dataset discussed in Section 3.1. The two correctly identified classes in $\ell_1$ and $\ell_2$ could have helped to identify the correct class for $\ell_3$. (b) Proportion of correctly classified images for each level of our taxonomy of the CIFAR-100 dataset, and the proportion of images incorrectly classified but for which the other levels in the taxonomy were correctly identified. This shows the potential benefit of a multi-level hierarchical classifier.**


*Conference'17, July 2017, Washington, DC, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

explored [4, 8, 9]. In particular, we distinguish: (i) the *flat classification approach* [10, 11], consisting of completely ignoring the class hierarchy, typically predicting only classes at the leaf nodes and considering that all its ancestor classes are also implicitly assigned to that instance; (ii) the *local classification approach* [12–14], where for each parent node in the class hierarchy, a multi-class classifier is trained to distinguish between its child nodes; and (iii) the *global classification approach* [9, 15–25], where a single classifier dealing with the entire class hierarchy structure is used. In this paper, we argue that *flat classification approaches* are inefficient as they do not take into account the taxonomy structure during the training stage, thus, resulting in very low *Hierarchical Evaluation Metric* values as shown in Section 3.2 – aberrant predictions are also obtained as for example the image in Figure 1a would be hierarchically classified as "Bus", "Automotive", and "Object". Moreover, we also argue that *local classification approaches* are not applicable, as they require $n$ networks to be trained and maintained, which can be tedious in practice. Therefore, in our work we opt and favor *global classification approaches* as they overcome the above constraints. However, we claim that existing *global classification approaches* here still suffer from several drawbacks as they: (i) do not "naturally" embed the taxonomy structure used, (ii) use a complex backbone neural network with $n$ disjoint output layers that do not constraint each other, (iii) may output predictions that are inconsistent with the taxonomy in place, and (vi) have often a fixed value of $n$, which means that they lack flexibility as they need substantial changes for a different value of $n$.

This paper addresses these deficiencies by proposing a novel mask-based output layer for MLHC. Specifically, we propose a model agnostic output layer that embeds the taxonomy and that can be combined with any model. Our proposed output layer implements a top-down divide-and-conquer strategy through a masking mechanism to enforce that predictions comply with the embedded hierarchy structure. Focusing on image classification, we evaluate the performance of our method on three different datasets including CIFAR-100 [26], Caltech BIRDS-210-2011 [27], and Stanford Cars [28], each with a three-level hierarchical structure. Experiments on these datasets show that our proposed mask-based output layer allows to improve several MLHC models.

## 2 METHODOLOGY

This section formally presents the hierarchical classification problem, and then introduces our mask-based output layer for MLHC.

### 2.1 Notation and the MLHC problem

**Classification:** Most classification problems in the literature involve flat classification, where each example is assigned to a class out of a finite set of flat classes. Formally, given a dataset $\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(m)}, y^{(m)})\}$ with $m$ instances, where each $\mathbf{x}^{(i)} \in \mathbb{X} \subseteq \mathbb{R}^n$ is an $n$-dimensional input feature vector of the instance $i$ and $y^{(i)} \in \mathcal{Y} = \{y_1, y_2, \cdots, y_k\}$ represents its class, a classification algorithm must learn a mapping function $f : \mathbb{X} \rightarrow \mathcal{Y}$, which assigns to each feature vector $\mathbf{x}^{(i)}$ its correct class $y^{(i)}$.

**Hierarchical classification:** In contrast to *flat classification* in which classes are considered unrelated, in a hierarchical classification problem classes are organized in a taxonomy. The taxonomy is often organized as a tree, where classes have a single parent each, or a directed acyclic graph (DAG), where classes can have multiple parents. Given a set of classes $\mathcal{Y}$, Wu et al. [29] defined a taxonomy as a pair $(\mathcal{Y}, \prec)$, where $\prec$ is the *"subclass-of"* relationship with the following properties [8, 29]: (i) asymmetry ($\forall y_i, y_j \in \mathcal{Y}, if\, y_i \prec y_j$ then $y_j \not\prec y_i$), (ii) anti-reflexivity ($\forall y_i \in \mathcal{Y}, y_i \not\prec y_i$), and (iii) transitivity ($\forall y_i, y_j, y_k \in \mathcal{Y}, y_i \prec y_j$ and $y_j \prec y_k$ implies $y_i \prec y_k$).

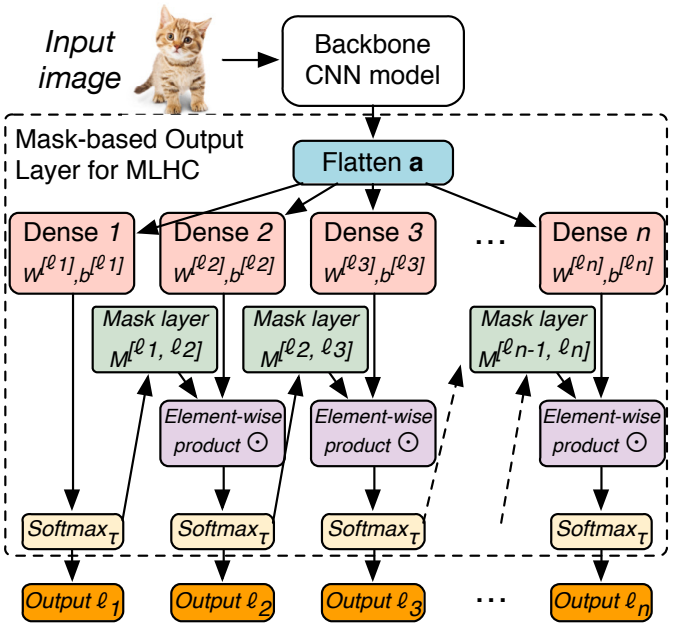



**Figure 2: Overview of the Mask-based Output Layer.**

In this paper, we consider only *tree* taxonomies, which are organized with a hierarchy structure of $n$ levels $\ell_i$, such that $\ell_i \subset \mathcal{Y}$, $\ell_1 \cup \ell_2 \cdots \cup \ell_n = \mathcal{Y}$, and $\forall y_j \in \ell_{i+1}, \exists y_k \in \ell_i$ s.t. $y_k \prec y_j$ (see Figure 1a for a three-level taxonomy) . Finally, we encode the relationship between two successive levels $\ell_i$ and $\ell_{i+1}$ in a taxonomy using an $|\ell_i| \times |\ell_{i+1}|$ matrix $M^{[\ell_i, \ell_{i+1}]}$, where the binary value $M^{[\ell_i, \ell_{i+1}]}_{y_k, y_j} \in \{0(y_k \not\prec y_j), 1(y_k \prec y_j)\}$, with $y_k \in \ell_i$ and $y_j \in \ell_{i+1}$.

**Problem definition:** The multi-level hierarchical classification problem we study in this paper is then defined as learning a mapping function $f : \mathbb{X} \rightarrow \mathcal{Y}$, which assigns to each feature vector $\mathbf{x}^{(i)}$ a prediction vector $\mathbf{y}^{(i)} = \{y^{[\ell_1]}, y^{[\ell_2]}, \cdots, y^{[\ell_n]}\}$ such that $y^{[\ell_i]} \in \ell_i$ is the class that $f$ assigns for each level $\ell_i$.

### 2.2 Proposed mask-based output layer

Figure 2 shows an overview of the architecture of our Mask-based Output Layer for MLHC. As mentioned previously, the output layer uses a masking mechanism to enforce that predictions comply with the hierarchy structure, thus, it embeds all matrices $M^{[\ell_i, \ell_{i+1}]}, i \in \{1, \cdots, n\}$ that encode the taxonomy. First, the layer computes an embedding for every level of the taxonomy as follows:

$$\mathbf{z}^{[\ell_i]} = W^{[\ell_i]} \times \mathbf{a} + b^{[\ell_i]} \tag{1}$$

where $\mathbf{a}$ is the embedding of the input, and $W^{[\ell_i]}, b^{[\ell_i]}$ are parameters learnt during training that are associated with every level $\ell_i$ of the taxonomy. Because the layer implements a top-down strategy,

**Table 1: Description of datasets.**

| Dataset | CIFAR-100 | Stanford Cars | CUB-200-2011 |
|---|---|---|---|
| **Statistics** | | | |
| Training set | 50,000 | 8,144 | 5,944 |
| Validation set | 5,000 | 4,020 | 3,000 |
| Test set | 5,000 | 4,021 | 2,071 |
| #classes | 100 | 196 | 200 |
| **Taxonomy** | | | |
| #classes $\ell_1$ | 2 | 13 | 39 |
| #classes $\ell_2$ | 20 | 113 | 123 |
| #classes $\ell_3$ | 100 | 196 | 200 |

the prediction at $\ell_1$ is obtained using a simple temperature softmax on its embedding as follows: $\hat{\mathbf{y}}^{[\ell_1]} = softmax_\tau(\mathbf{z}^{[\ell_1]})$, where $\tau$ is the temperature parameter that has to be tuned. Next, for all remaining levels, a mask is first computed to enforce that the prediction complies with the taxonomy (i.e., $\hat{\mathbf{y}}^{[\ell_{i+1}]} \prec \hat{\mathbf{y}}^{[\ell_i]}$) as follows:

$$\mathbf{m}^{[\ell_{i+1}]} = \hat{\mathbf{y}}^{[\ell_i]} \times M^{[\ell_i,\ell_{i+1}]} \tag{2}$$

The mask is then applied using a simple Hadamard product on the embedding to enforce that $\hat{\mathbf{y}}^{[\ell_{i+1}]} \prec \hat{\mathbf{y}}^{[\ell_i]}$:

$$\hat{\mathbf{y}}^{[\ell_{i+1}]} = softmax_\tau(\mathbf{z}^{[\ell_{i+1}]} \circ \mathbf{m}^{[\ell_{i+1}]}) \tag{3}$$

Finally, the model is trained by minimizing the following objective function:

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\left[\pi^{[\ell_i]} \times \mathcal{L}(\hat{y}^{(j)[\ell_i]}, y^{(j)[\ell_i]})\right] \tag{4}$$

where $\mathcal{L}(\bullet,\bullet)$ denotes the cross-entropy function and $\pi^{[\ell_i]}$ are hyperparameters that need to be tuned to calibrate the relative importance of the objectives. The loss function takes all levels' loss into account to make sure the structure prior can play a role of internal guide to the whole model and make it easier to flow the gradients back to the all layers.

## 3 EXPERIMENTAL EVALUATION

In this section, we fist describe the experimental setup we have used in our evaluation before discussing the obtained results.

### 3.1 Experimental setup

**Datasets:** Our experiments are performed on three different image datasets: CIFAR-100 [26], Stanford Cars [28], and Caltech-UCSD Birds-200-2011 (CUB-200-2011) [27]. Hyperparameters are tuned on validation sets obtained by splitting the test sets. The CIFAR-100 is a 2-level hierarchy dataset to which we have added a third level at the top: "Object" and "Bio Organism". For the Stanford Cars and CUB-200-2011 we have used the hierarchy structure provided by [21]. Detailed statistics of the datasets are provided in Table 1.

**Implementation details:** All models used in our experiments are based on the VGG19 [30] backbone neural network pretrained on ImageNet [31]. For CIFAR-100 we used an image size of 32x32 and for the other datasets an image size of 64x64. Finally, a batch size of 128 was used and Adam Optimizer [32] with a learning rate of 1e-4 with a decay factor on plateau of 0.1. For the loss function we used an equal weights for all $\pi^{[\ell_i]}$.

**Baseline models:** Our mask-based output layer for MLHC is combined with the following baseline models in our experiments:

(1) ***n*-nets:** *n* independent networks for each hierarchy level.
(2) ***n*-outs:** a single network with *n* output layers.
(3) **B-CNN:** Branch-CNN described in [22].
(4) **B-CNN_v2:** a variant of B-CNN, which takes the ReLu activation of every branch output and uses them as the input of the next Fully-Connected layer.
(5) **Bi-CNN:** Bilinear-CNN described in [33].
(6) **MLPH:** Multi-linear Pooling with Hierarchy described in [21].

In addition, we use a flat classification approach as a baseline for comparison. We recall that it consists of completely ignoring the class hierarchy, typically predicting only classes at the leaf nodes. It provides an indirect solution to the problem of hierarchical classification, because, when a leaf class is assigned to an example, one can consider that all its ancestor classes are also implicitly assigned to that instance.

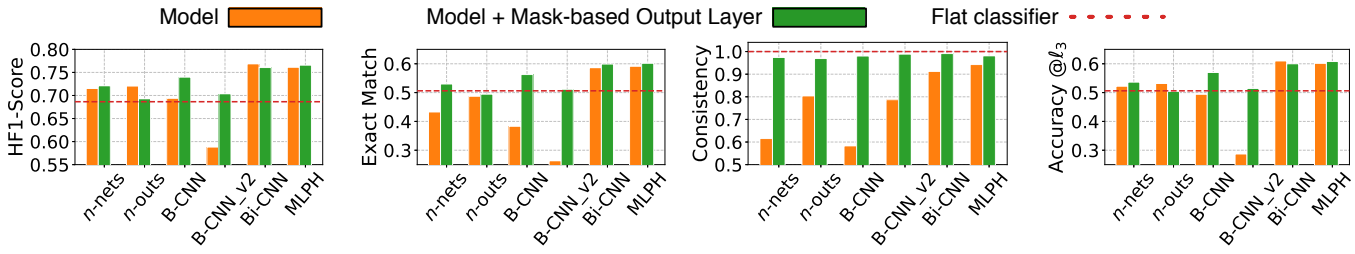
**Metrics:** Commonly used measures of Precision, Recall, F1-Score, and Accuracy are not appropriate for Hierarchical Classification, because they do not take into account the relations that exist between classes. Hence, we report our results using the following hierarchical metrics: (i) *Hierarchical F1-Score* [1], which is a variant of F1-Score that uses the hierarchy, (ii) *Exact Match*, which measures the percentage of predictions that match exactly the ground truth for all levels of the hierarchy, and (iii) *Consistency*, which estimates the proportion of test examples that are consistent with the hierarchy structure, regardless the ground truth. Finally, we also use (vi) Accuracy@$\ell_3$ to estimate the impact of our top-down output layer on the last level of the taxonomy.

### 3.2 Results

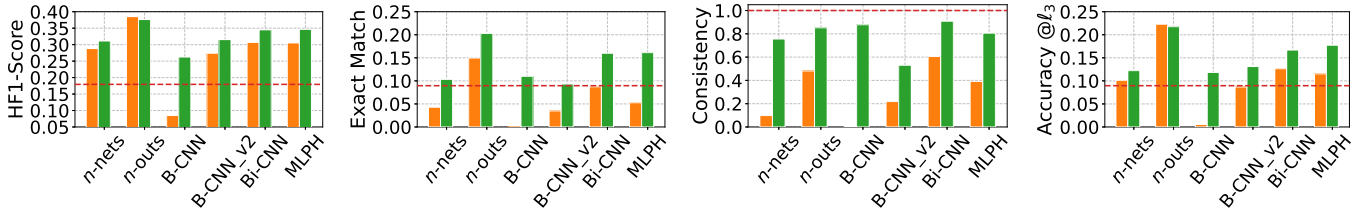
**Performance:** Figure 3 shows the effect of our mask-based output layer on each model, and the performance obtained by the flat classifier as a baseline. From the obtained results we make the following key observations:

(1) Our mask-based output layer allows to improve all models for all metrics.
(2) Almost all methods outperform the flat classifier baseline, which indicate that although it is an intuitive approach, it does not provide good performance, thus the need to develop specific MLHC methods.
(3) B-CNN is the model for which we notice the highest improvement for all metrics.
(4) CUB-200-2011 and Stanford Cars datasets are the datasets that improved the most compared to the flat classifier Baseline.
(5) Stanford Cars is the dataset that benefited the most from our mask-based output layer.
(6) Our mask-based output layer allows a huge improvement in terms of consistency, thus providing more reliable predictions.

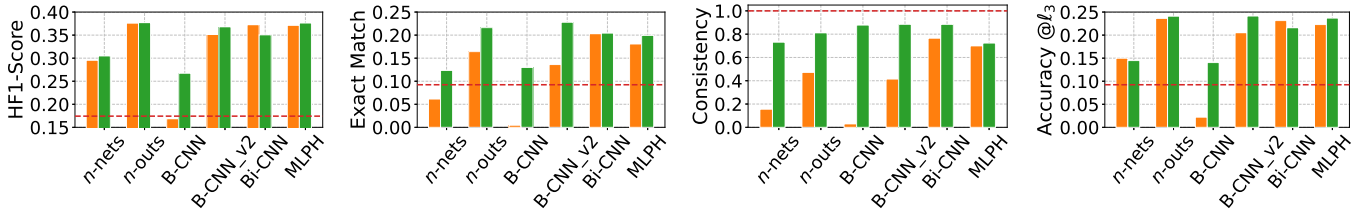
In conclusion, we observe that our mask-based output layer offers a good trade-off between performance metrics and consistency,




**(a) CIFAR-100.**




**(b) Stanford Cars.**




**(c) CUB-200-2011.**

**Figure 3: Performance comparison.**

hence, combining both validity and reliability for MLHC. Also, we observe that there is a clear discrepancy between the results obtained on CIFAR-100 dataset and on the CUB-200-2011 dataset, which we analyse in the next section.

**Task complexity analysis:** In this section we analyse the discrepancy observed between the results obtained on CIFAR-100 dataset and on the CUB-200-2011 dataset, which we explain by the complexity of the task. Hence, we compare *n*-nets with our masked-based output layer against the flat classifier baseline on the CUB-200-2011 dataset, while varying the complexity of task by varying the number of classes in $\ell_1$.

The obtained results are shown in Figure 4, from which we observe that for a hard task (> 20 classes for $\ell_1$), our method substantially outperforms the flat classifier, whereas for an easy task (< 10 classes for $\ell_1$), the the flat classifier substantially outperforms our method. Hence, we simply conclude that for complex tasks such as the Stanford Cars classification or the CUB-2010-2011 classification, our method allows a substantial improvement, whereas for a simple tasks such as the CIFAR-100 classification problem a flat classifier is enough to achieve high classification performance.

## 4 CONCLUSION AND FUTURE WORK

We introduced in this paper a new mask-based output layer for multi-level hierarchical classification, which embeds the taxonomy structure and that can be combined with any model. Our

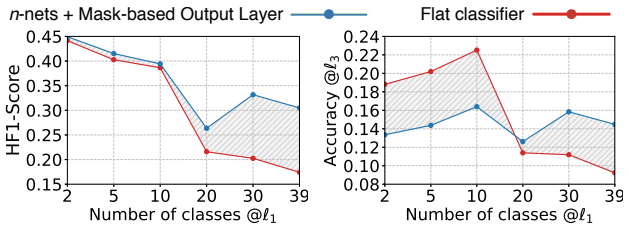



**Figure 4: Task complexity analysis.**

proposed output layer implements a top-down divide-and-conquer strategy through a masking mechanism to enforce predictions comply with the embedded hierarchy structure. Focusing on image classification, we presented a thorough experimental evaluation of the performance of our method on three different datasets, including CIFAR-100, Caltech BIRDS-200-2011, and Stanford Cars, each with a three-level hierarchical structure. Experiments on these datasets show that our proposed mask-based output layer allows to improve several multi-level hierarchical classification models for various performance metrics.

Future work includes improving our method with a new loss function specifically designed for hierarchy structures, combining a bottom-up approach, and exploring the attention mechanism for improving the mask mechanism.

# REFERENCES


[1] Aris Kosmopoulos, Ioannis Partalas, Eric Gaussier, Georgios Paliouras, and Ion Androutsopoulos. Evaluation measures for hierarchical classification: a unified view and novel approaches. Data Mining and Knowledge Discovery, 29(3):820–865, 2015.

[2] A. Kosmopoulos, E. Gaussier, G. Paliouras, and S. Aseervatham. The ecir 2010 large scale hierarchical classification workshop. SIGIR Forum, 44(1):23–32, aug 2010.

[3] Eduardo Costa, Ana Lorena, ACPLF Carvalho, and Alex Freitas. A review of performance evaluation measures for hierarchical classifiers. In Evaluation methods for machine learning II: Papers from the AAAI-2007 workshop, pages 1–6, 2007.

[4] Alex Freitas and André Carvalho. A tutorial on hierarchical classification with applications in bioinformatics. Research and trends in data mining technologies and applications, pages 175–208, 2007.

[5] Aixin Sun, Ee-Peng Lim, and Wee-Keong Ng. Performance measurement framework for hierarchical text classification. Journal of the American Society for Information Science and Technology, 54(11):1014–1028, 2003.

[6] Aixin Sun, Ee-Peng Lim, and Wee-Keong Ng. Hierarchical text classification methods and their specification. In Cooperative internet computing, pages 236–256. Springer, 2003.

[7] Huzefa Rangwala and Azad Naik. Large scale hierarchical classification: foundations, algorithms and applications. In The European conference on machine learning and principles and practice of knowledge discovery in databases, 2017.

[8] Carlos N Silla and Alex A Freitas. A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery, 22(1):31–72, 2011.

[9] Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In Proceedings 2001 IEEE International Conference on Data Mining, pages 521–528, 2001.

[10] Susan Dumais and Hao Chen. Hierarchical classification of web content. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '00, page 256–263, New York, NY, USA, 2000. Association for Computing Machinery.

[11] Jayme Garcia sArnal Barbedo and Amauri Lopes. Automatic genre classification of musical signals. EURASIP Journal on Advances in Signal Processing, 2007:1–12, 2006.

[12] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In Proceedings of the Fourteenth International Conference on Machine Learning, ICML '97, page 170–178, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[13] Andrew McCallum, Ronald Rosenfeld, Tom M. Mitchell, and Andrew Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98, page 359–367, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[14] Andrew D Secker, Matthew N Davies, Alex A Freitas, Jon Timmis, Miguel Mendao, and Darren R Flower. An experimental comparison of classification algorithms for hierarchical prediction of protein function. Expert Update (Magazine of the British Computer Society's Specialist Group on AI), 9(3):17–22, 2007.

[15] M. M. Zloof. Query by Example: Operations in Hierarchical Databases. Microfiche, 1975.

[16] W. Dickson. Feature grouping in a hierarchical probabilistic network. Image and Vision Computing, 9(1):51–57, 1991.

[17] S. Rizzi. Genetic operators for hierarchical graph clustering. Pattern Recognition Letters, 19:1293–1300, 1998.

[18] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In Comp. Vision and Pattern Recognition, pages II:524–531, 2005.

[19] L. Khan, M. Awad, and B. Thuraisingham. A new intrusion detection system using support vector machines and hierarchical clustering. The International Journal on Very Large Data Bases, 16(4):507–521, 2007.

[20] Y. Ioannou, D. P. Robertson, R. Cipolla, and A. Criminisi. Deep roots: Improving cnn efficiency with hierarchical filter groups. Computer Vision and Pattern Recognition, pages 5977–5986, 2016.

[21] Yuqi Huo, Yao Lu, Yulei Niu, Zhiwu Lu, and Ji-Rong Wen. Coarse-to-fine grained classification. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'19, page 1033–1036, New York, NY, USA, 2019. Association for Computing Machinery.

[22] Xinqi Zhu and Michael Bain. B-cnn: branch convolutional neural network for hierarchical classification. arXiv preprint arXiv:1709.09890, 2017.

[23] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. Hd-cnn: Hierarchical deep convolutional neural networks for large scale visual recognition. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), December 2015.

[24] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In Yoshua Bengio and Yann LeCun, editors, 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.

[25] Yanming Guo, Yu Liu, Erwin M Bakker, Yuanhao Guo, and Michael S Lew. Cnn-rnn: a large-scale hierarchical image classification framework. Multimedia tools and applications, 77(8):10251–10271, 2018.

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[27] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[28] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, Australia, 2013.

[29] Feihong Wu, Jun Zhang, and Vasant Honavar. Learning classifiers using hierarchically structured class taxonomies. In International symposium on abstraction, reformulation, and approximation, pages 313–320. Springer, 2005.

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

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

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

[33] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In Proceedings of the IEEE international conference on computer vision, pages 1449–1457, 2015.