Computers in Industry 144 (2023) 103786

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.sciencedirect.com/journal/computers-in-industry

Deep learning based hierarchical classifier for weapon stock aesthetic quality control assessment

Víctor Manuel Vargas [a,*], Pedro Antonio Gutiérrez [a], Riccardo Rosati [b], Luca Romeo [c], Emanuele Frontoni [c], César Hervás-Martínez [a]

[a] Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain
[b] Department of Information Engineering, Marche Polytechnic University, Ancona, Italy
[c] University of Macerata, Macerata, Italy

ARTICLE INFO

Keywords:
Hierarchical classification
Ordinal classification
Deep learning
Aesthetic quality control
Convolutional neural networks

ABSTRACT

In the last years, multiple quality control tasks consist in classifying some items based on their aesthetic characteristics (aesthetic quality control, AQC), where usually the aspect of the material is not measurable and is based on expert observation. Given the increasing amount of images in this domain, deep learning (DL) models can be used to extract and classify the most discriminative patterns. Frequently, when trying to evaluate the quality of a manufactured product, the categories are naturally ordered, resulting in an ordinal classification problem. However, the ordinal categories assigned by an expert can be arranged in different levels that somehow model a hierarchy of the AQC task. In this work, we propose a DL approach to improve the classification performance in problems where categories are naturally ordered and follow a hierarchical structure. The proposed approach is evaluated on a real-world dataset that defines an AQC task and compared with other state-of-the-art DL methods. The experimental results show that our hierarchical approach outperforms the state-of-the-art ones.

1. Introduction

In the real world, many tasks involve the classification of a given manufactured item depending on its quality. In some cases, this quality is referred to the absence of defects that could impact the capabilities of the product. While these quality control tasks are crucial for some business-to-business industries where any defect can represent a prominent quality problem, there are other scenarios like business-to-consumer industries, where the aesthetic quality is more important, like in the automotive or weapons industry. In this context, the finished product must guarantee high performance not only at the mechanical level but also at the aesthetic one according to the expectations of the customer, with the aim to make a manufactured item with excellent perceived quality (Stylidis et al., 2020). The aesthetic quality control (AQC) (Ouzounis et al., 2021) tasks can be defined as a subset of general quality control (QC) (Wagersten et al., 2011) problems where only the aesthetic quality is evaluated. This task is usually done by an expert technician that classifies each of the items one by one merely using its expert knowledge and focusing on qualitative and subjective analyses. However, using machine learning techniques, a decision support system can be constructed to predict the label of each item. In some cases, the data available to classify these elements come in the shape of images. Deep learning techniques are more suitable for this kind of data, given that, using Convolutional Neural Networks (CNN) (Hansen et al., 2018; Akshayarathna et al., 2021; Zhang et al., 2020), features can be automatically extracted from images without prior knowledge of each particular challenge related to the problem. CNN is a deep neural network model which uses convolution operations to extract higher-level features from the pixels of the input images. Also, they have mechanisms (pooling layers) to reduce the dimensionality of the input data to avoid overfitting and improve the convergence of the model. In recent works, some industrial (Villalba-Diez et al., 2019; Villalba-Díez et al., 2020; Rosati et al., 2021; Pazzaglia et al., 2021) and Internet of Things (Elsisi et al., 2021) problems have been tackled using these methods, including inspection of laser welding defects on batteries (Yang et al., 2020), damage prediction for steel beams (Onchis and Gillich, 2021), industrial object visual detection (Ge et al., 2020), predictive maintenance for motors (Kiangala and Wang, 2020), and damage-type identification in structures (Agrawal and Chakraborty, 2022), among others. In Chiarello et al. (2021), the authors defined different challenges that can be solved combining Engineering Design and Computer Science techniques.

* Corresponding author at: Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain.
E-mail address: vvargas@uco.es (V.M. Vargas).

https://doi.org/10.1016/j.compind.2022.103786
Received 29 November 2021; Received in revised form 4 July 2022; Accepted 19 September 2022
Available online 5 October 2022

Also, when trying to evaluate the quality of a manufactured product (AQC task), the categories usually show a natural order (i.e. the first category represents the worst quality while the last class indicates the maximum quality). Thus, the problem should be considered an ordinal classification problem instead of a standard nominal classification one. Ordinal classification problems (Gutierrez et al., 2016; Vargas et al., 2022; Cao et al., 2020) are those problems where the variable that we aim to predict is selected from a group of categories that follow a natural order which is inherent to the real problem that is being solved. In some aspects, ordinal classification, which is also named ordinal regression, is similar to standard regression problems. However, in regression, the objective variable is continuous, while in ordinal classification it is discrete and finite so that it must belong to one of the categories which are defined for each problem. This kind of problems are present in numerous research areas. For example, there are multiple recent works related to biomedicine that used an ordinal approach to solve the proposed problem (Durán-Rosal et al., 2021; Albuquerque et al., 2021).

In these terms, a classification problem can be defined as the problem of predicting the real class y from input data $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^K$. The real class y belongs to a set of categories $\mathcal{Y} = \{C_1, C_2, C_3, C_4, \dots, C_Q\}$, where Q is the total number of labels of the problem. An ordinal classification problem can be defined as a special case of a classification problem where the labels follow a natural order. Thus, these labels satisfy the expression: $C_1 \prec C_2 \prec C_3 \prec C_4 \prec \dots \prec C_Q$. The precedence ($\prec$) operator indicates that categories follow a specific order but, in contrast to regression, the distance between them is not quantifiable. Therefore, the distance between two different pairs of adjacent classes does not have to be the same. The aforementioned inherent order can be expressed as an integer number using the function $\mathcal{O}(\cdot)$, so that $\mathcal{O}(C_q) = q$ and $q = 1, \dots, Q$.

This kind of problem can be tackled as a standard classification problem, discarding the subsequent order between categories. However, the ordinal information can be used to improve the classification performance by reducing the error in distant classes, which are the most important errors in these problems. Thus, misclassifying a pattern in an adjacent class in a problem with 10 classes is way less important than classifying it in the furthest category.

The AQC we tackle in our work, as well as the main challenge we aim to solve, is originated from a specific company's demand. A well-known weapon manufacturer has collected for some time a decent number of images of the wooden stocks that they equip in their shotguns or rifles. For each of the images, they took two high-quality photographs from each side of the stock. All the images were made using the same lighting conditions and camera so that they are uniform. An expert technician labelled each stock according to the aesthetic quality of the wood which was employed to make it. Therefore, they built a dataset that can be used to construct and train a deep learning model that tries to predict the quality of a given stock from an input image. As will be described in Section 3, the existing ordinal labels are structured hierarchically, having macro classes and micro labels, where every macro class contain three sub-classes or micro classes.

Hence, the aim of this work is to propose a hierarchical approach that simplifies, generalises and automatises the AQC task by using multiple ordinal CNN models to predict hierarchically the final label in two steps: one for the macro label and one for the micro. A similar approach was introduced in a previous work (Sánchez-Monedero et al., 2018), where, in a medical application, an initial prediction was done to obtain a positive or negative result, and, a posterior classification determined different grades when the first result was positive. However, in this work, instead of using a binary classifier for the first step, an ordinal classifier with four classes is used. For the second step, three different ordinal classifiers are employed to obtain the micro label. A combination of the labels obtained in both steps results in the final label. Concerning the challenges introduced in Chiarello et al. (2021), in this work, we try to tackle Challenge 13, Redesign ad-hoc Data Science methods, and Challenge 20, Automatic data labelling. Our motivation is justified by the facts (i) to redesign and automatise standard QC task procedure in order to properly perform in a specific context and (ii) to mitigate the inter-subject variability of the overall AQC annotation procedure while providing classification that could be more suitable for supporting the expert human operator.

The rest of the manuscript is structured as follows: in Section 2 some previously proposed ordinal methods are described, in Section 3 the aforementioned dataset is characterised, in Section 4 the proposed method is described, in Section 5 the model and the design of the experiments are illustrated, in Section 6 the results of all the experiments are shown and compared, including an statistical analysis, and, finally, in Section 7 the conclusions of this work are summarised.

2. Related works

The AQC tasks are usually based on classifying images and thus several state-of-the-art works employed standard DL algorithms (Kao et al., 2017). The general aim is to automatise the overall QC analysis that is strictly human dependent, subjective and not directly measurable. However, the AQC poses additional real challenges: the annotation procedure exhibit a different level of complexity and scales. Approaching this problem with a nominal DL classification method (which does not exploit class order) may lead to a lower generalisation performance, especially between widely distant classes, which represents a significant error from the industrial perspective. Moreover, a single model for learning the entire annotation may not be able to capture the different scales and magnitudes of the ordinal QC classes which usually discloses a hierarchical structure.

In this section, we describe three methods that were proposed in previous works to improve the classification performance for ordinal problems, as well as two hierarchical methods based on Error Correcting Output Codes (ECOC).

The cumulative link models (CLM) are threshold models described in Agresti (2010), which can be used to model the posterior probabilities of a given ordinal problem. These kinds of models are based on the concept of a latent variable (or a linear projection) and a set of thresholds that separate different categories. In Vargas et al. (2019, 2020), the authors proposed to combine the CLM with a deep neural network model. Thus, the aforementioned projection was obtained from a convolutional neural network model and used to determine the final class of each sample.

Also, in de la Torre et al. (2018) the authors proposed to use a different loss function for the optimisation process to exploit the order information of the problem. In these terms, they defined the Quadratic Weighted Kappa (QWK) loss function which is based on the Kappa index and applied it to train models with a standard softmax output. Said loss function uses a penalisation matrix which applies a higher error to the prediction when the class obtained is far from the ground truth label. This loss function was also used by the authors of Vargas et al. (2020), and they both checked that the performance for ordinal problems improved when using this loss function instead of the standard categorical cross-entropy loss.

Then, in Zhang et al. (2021), Vargas et al. (2022) the authors proposed a method to represent the labels in a soft manner instead of using the standard one-hot encoding. To do that, they also used a custom loss function. In this case, they modified the standard cross-entropy loss adding a regularisation term. The standard cross-entropy loss is often defined as:

$$L(\mathbf{x}) = \sum_{q=1}^{Q} h(q)[-\log p(y = C_q|\mathbf{x})], \tag{1}$$

where $h(q) = \delta_{y,q}$, y is the ground truth class and $\delta_{y,q}$ is the Dirac delta, which equals to 1 for $q = y$, and 0 otherwise. Thus, the regularisation can be applied to the $h(q)$ term, turning it into a soft distribution instead of being 0 or 1. This soft term is denoted as $h'(q)$ and defined

as $h'(q) = (1-\eta)\delta_{y,q} + \eta f(x)$, where $f(x)$ is the density or probability function of a given distribution. Therefore, the regularisation term can be sampled from a uniform distribution $\left(\frac{1}{Q}\right)$, in the most simple case, or from other distributions like Poisson, binomial or exponential (Liu et al., 2020). In Vargas et al. (2022), the authors used a beta distribution to obtain these soft labels and they proved that using this kind of distribution improved the performance over other state-of-the-art approaches. Given that the beta distribution has two parameters that must be fixed (a and b), they also proposed a method to analytically determine them based on the number of classes of the problem.

Even though the described methods were proved to improve the performance of ordinal classification problems, none of them included the possibility of having a hierarchical structure implicit in the categories of the problem. When dealing with complex problems that are hierarchically labelled, using a method that exploits these hierarchical structures can lead to better classification performance and lower cost errors. In these terms, the ECOC approach (Dietterich and Bakiri, 1994) was proposed as a method to decompose a multi-class problem into multiple binary problems that can be solved independently using different models. Then, the predictions of each model are combined like in any ensemble. This method has been used in several previous works (Bora et al., 2020) in combination with CNN models. Although this approach is not originally hierarchical, a hierarchical approach can be easily derived from it, given that the codes generated for each of the labels can include the hierarchical dependencies of those classes. To do that, the codes can be simply composed of Q_i bits for each of the hierarchical levels, where Q_i is the number of classes in level i. The bits corresponding to the correct label on each of the levels will be active while the others will remain zero. In this way, the hierarchical structure is encoded in the generated codes. However, this approach does not take into account the ordinal information, given that each of the bits is going to be one or zero without taking into account the rest of the bits. Therefore, to address this problem, in Barbero-Gómez et al. (2022), the authors proposed a method to generate the ECOC codes in an ordinal way, resulting in better classification performance for ordinal problems. However, in this case, the hierarchical structure of the labels is not represented by the codes. Also, it is worth noting that the ECOC approach usually will spend more time for the training process, given that they decompose the original multi-class problem in multiple binary problems, and each of them is trained using all the training samples.

Taking into account the characteristics of the approaches described, in this work, we propose to combine the described ordinal methods with a new hierarchical classification approach that aims to predict the correct label for an ordinal problem in two separate steps. In these terms, our method is further described in Section 4.

3. Problem formulation

The weapons manufacturing industry is an important industry that can benefit from machine learning techniques for AQC. Concretely, there is a well-known manufacturer who makes different types of weapons, including shotguns and rifles. Most of these weapons equip a wooden stock that can be classified depending on the aesthetic quality of the wood that was used to make it. Since the manufacturing of wooden parts is made by external suppliers, the company carries out an AQC for defining whether the items comply with the quality requirements. Therefore, different quality categories follow an order that is determined by the AQC problem. Apart from following an ordinal relation, these categories are grouped in four macro classes: 1, 2, 3, and 4. The macro classes can be easily classified by an expert. However, each of these macro classes contains several micro labels (-, ·, +) which are harder to classify. Combining both types of labels (i.e. different scale), the proposed problem contains 10 ordinal categories: 1, 2^-, 2, 2^+, 3^-, 3, 3^+, 4^-, 4, 4^+ (note that the first class has not been divided into micro labels because the company usually produces model series with higher quality classes). Fig. 1 shows the hierarchical structure of the classes in a more detailed manner.

In the last years, the aforementioned weapons manufacturer has been collecting different images of the stocks they have been producing. These images have been taken from both sides of the stock and using the same camera and lighting conditions. In this way, the dataset includes high-quality images that were labelled according to the categories described before. Currently, the dataset comprises a total of 2120 1000 × 500 colour images belonging to 1060 different stocks. Table 1 shows the distribution of all the samples across the different categories described.

To use these images to train a CNN model, they have been resized to 470 × 270 and the background has been replaced with black plain colour. Fig. 2 shows some images from the dataset including their class label.

In the next section, the proposed method to train a classifier following this hierarchical structure and the strategy to obtain the final label for each sample are described.

4. Proposed method

In this section, the proposed hierarchical methodology is shown. The description of the hierarchical method is divided into two parts: (1) the ensemble architecture and the method to combine the predictions obtained from the individuals models, and (2) the architecture of the deep network that is used for each individual model that the hierarchical structure contains.

4.1. Ensemble architecture

In this work, we propose a new hierarchical method to build a classifier for the hierarchical ordinal problem described in Section 3. The main idea behind this method is to do a multi-step classification, where, in the first step, we try to distinguish the macro classes (1, 2, 3 or 4, in this case) and, in the second step, we aim to classify the micro classes (-, · or +). Completely independent models are used for each step: for the first step, a single model is used to derive the macro class. In the second step, one separate model is used to determine the micro class for each of the aforementioned macro classes. Therefore, for this problem, four different models are used to obtain the final prediction. The number of models can vary for other problems where the number of classes or their hierarchy is different.

The model used to predict the macro class is defined as $f_M(\mathbf{x}) \rightarrow y_M$, where y_M is the predicted macro label. On the other hand, the models used to predict the micro classes are denoted as $f_{m_i}(\mathbf{x}) \rightarrow y_{m_i}$, where ith classifier is associated with macro class i, and y_{m_i} is the micro label predicted by the classifier associated with ith macro class. The predictor y_M is trained using all the samples in the training set, but these samples are labelled using only their macro class. In the same way, the classifiers y_{m_i} are trained using only the samples that belong to the ith macro class and they are labelled using only the micro labels. Therefore, the complete hierarchical model can be defined as a ensemble model which is composed of 4 independent classifiers, whose decision function can be defined as:

$$f(\mathbf{x}) = \begin{cases} f_M(\mathbf{x}), & \text{if } \mathcal{O}(f_M(\mathbf{x})) = 1, \\ f_M(\mathbf{x}) \cup f_{m_i}(\mathbf{x}), & \text{if } \mathcal{O}(f_M(\mathbf{x})) > 1, \end{cases} \tag{2}$$

where $i = \mathcal{O}(f_M(\mathbf{x}))$, and $\mathcal{O}(\cdot)$ represents the order of any given ordinal class. Therefore, the final labels predicted by the hierarchical model can be denoted as:

$$y = \begin{cases} y_M, & \text{if } \mathcal{O}(y_M) = 1, \\ y_M \cup y_{m_i}, & \text{if } \mathcal{O}(y_M) > 1, \end{cases} \tag{3}$$

where $i = \mathcal{O}(y_M)$.

Taking into account the problem tackled in this work, $y_M \in \{1,2,3,4\}$, and $y_{m_i} \in \{-,\cdot,+\}$, where $i \in \{2,3,4\}$.

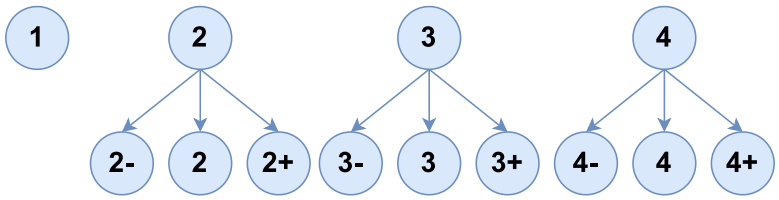

Fig. 1. Hierarchical classes structure.

Table 1
Stocks dataset classes distribution.

Label	1	2^-	2	2^+	3^-	3	3^+	4^-	4	4^+
Images	165	148	212	177	179	306	344	208	275	106

Fig. 2. Cropped dataset images from class 1 (top-left), 2 (top-right), 2^+ (bottom-left) and 3^+ (bottom-right).

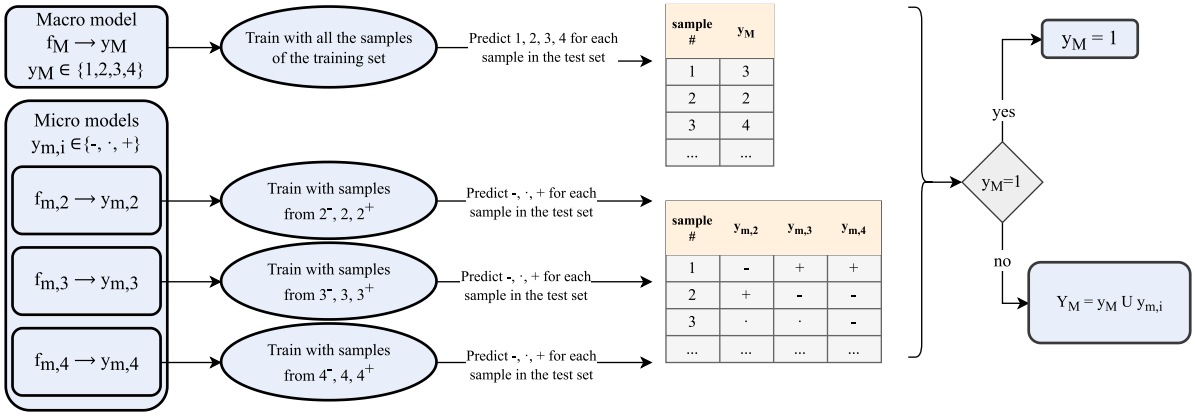

Fig. 3. Hierarchical models structure.

Fig. 3 illustrates the hierarchical scheme that has been described. On the left side, the models considered for this approach are defined, followed by the training procedure and the way that the final predictions are obtained.

4.2. Deep network architectures

In order to obtain more robust results regarding the proposed hierarchical methods, three different well-known architectures have been used: VGG-16 (Simonyan and Zisserman, 2014), ResNet-101 (He et al., 2016) and DenseNet-121 (Huang et al., 2017). In this way, we can prove that the proposed methodology is not architecture dependent, and it can work with any CNN model.

Also, the convolutional parts of the models use the pre-trained ImageNet weights instead of adjusting them from a random initialisation. Therefore, only the top part of the model needs to be adjusted. This optimisation approach, named transfer learning, is commonly used to train deep models as it speeds up the training process and improves convergence. For the top part of the model, a 50% dropout is performed before two dense layers with 4096 units. In the output layer, two different functions have been employed:

- Softmax function, which is the standard output function for classification tasks.
- Cumulative link models (described in Section 2), that enhance ordinal classification performance. In this case, two different link functions have been used: logit and probit.

The VGG-16, ResNet-101 and DenseNet-121 contain 266M (251M trainable), 67M (25M trainable) and 493M (486M trainable) parameters, respectively. There are some fixed parameters due to the transfer learning approach. It is worth noting that the ResNet-101 has less parameters compared to the other alternatives. However, the residual neural network architectures have demonstrated having an outstanding generalisation capability with a reduced number of parameters.

5. Experiments

In this Section, the experiments that have been conducted are detailed, including a description of the data partitions used along with the optimisation and evaluation processes followed.

5.1. Experimental design

The models described in Section 4.2 are evaluated following a holdout scheme, where 80% of the whole set is used to adjust the model while the remaining 20% forms the test set. From the training set, another 15% of the samples are taken for the validation set, which is used to stop the training process when the model performance stops improving. All the experiments are repeated 30 times using different seeds to create the data partitions and initialise the model parameters. In this way, we obtain robust results from the point of view of a statistical analysis.

When using the hierarchical approach, different loss functions can be employed for the model which predicts the macro class and the models that predict the micro class. The experiments are performed using different loss functions to guide the optimisation algorithm:

- Categorical cross-entropy (CCE): the standard CCE is commonly used for nominal classification problems where classes do not follow any order.
- QWK Loss: the quadratic weighted kappa loss function that was described in Section 2.
- Beta regularised categorical cross-entropy (CCE-Beta): the unimodal regularised CCE that was described in Section 2.

In this way, we test different loss functions combinations. Also, the output function can vary from one model to the others, leading to using an ordinal output like the CLM in the first step and the standard softmax in the second.

Regarding the proposed hierarchical methods, different methodologies are used for the first classifier, where 4 classes are considered: C1 = $\{1\}$, C2 = $\{2^-, 2, 2^+\}$, C3 = $\{3^-, 3, 3^+\}$ and C4 = $\{4^-, 4, 4^+\}$, and the second classifier, which considers only three different classes $(-, \cdot, +)$. These methodologies are listed below and summarised in Table 2:

1. Hierarchical baseline. Softmax in the output layer and the standard CCE for both the first classifier and the next three classifiers.
2. Hierarchical CLM with logit link in the output layer and Beta regularised cross-entropy as loss function for the first and the subsequent classifiers.
3. Hierarchical CLM with probit link in the output layer and Beta regularised cross-entropy as loss function for the macro and the micro classifiers.
4. Hierarchical CLM with logit link in the output layer and Beta regularised cross-entropy loss for the first model and softmax function with the standard CCE loss for the micro models of the second stage.
5. Hierarchical CLM with probit link in the output layer and Beta regularised cross-entropy loss for the first model and softmax function with the standard CCE loss for the micro models of the second stage.
6. Hierarchical CLM with logit link in the output layer for the first and the next three models and QWK loss function for all of them.
7. Hierarchical CLM with probit link in the output layer and QWK loss function for the first and the second stage.

Models described in items 4 and 5 use an ordinal output function and an ordinal loss function for the first models, which tries to distinguish between 4 classes, and a nominal approach, for the next three models. Even though the problem that is solved in the second step is ordinal too, the number of classes is too small to benefit from the advantages of using an ordinal approach. Therefore, using a nominal approach for these models has been considered as a good alternative and is going to be compared with the rest of the experiments that have been proposed.

For comparison purposes, the non-hierarchical alternatives previously proposed in the literature (see Section 2) are also run. In these cases, the number of classes considered is 10. These alternatives are listed below and summarised in Table 3:

8. Baseline. Softmax function in the output layer and the standard CCE as loss function.
9. CLM with logit link (Vargas et al., 2020) in the output layer and Beta regularised CCE (Vargas et al., 2022) as loss function.
10. CLM with probit link (Vargas et al., 2020) in the output layer and Beta regularised CCE (Vargas et al., 2022) as loss function.
11. CLM with logit link (Vargas et al., 2020) in the output layer and QWK loss function (de la Torre et al., 2018).
12. CLM with probit link (Vargas et al., 2020) in the output layer and QWK loss function (de la Torre et al., 2018).
13. ECOC with codes that represent the hierarchical structure of the classes (Dietterich and Bakiri, 1994). In this case, each code is composed of 7 bits, where the first 4 bits are related to the macro class and the last 3 bits represent the micro class (e.g. for class 2^+, 0100 001).
14. ECOC with codes that contain the ordinal information of the labels (Barbero-Gómez et al., 2022). They are composed of $Q-1$ bits and each bit q is set to 1 when the class that the code is associated to is higher than q (e.g. for class 2^+, which is the $4th$ class, $q = 4$, 111 000 000).

The optimisation algorithm used for all the experiments is the Adam algorithm with a learning rate of 0.01. Taking into account the size of the dataset, the mini-batch size is fixed to 16. The model is trained for a maximum of 50 epochs. However, the early stopping strategy stops the training process when the validation loss stops improving. This strategy uses a patience value of 15, which determines the number of epochs without validation loss improvements before stopping the training process.

6. Results

In this section, the results of the experiments described in Section 5 are shown. To better analyse the performance, we have considered several evaluation metrics over the 10-class problem:

- Quadratic weighted kappa (QWK) (de la Torre et al., 2018), defined in Section 2.
- Minimum sensitivity (MS) (Caballero et al., 2010; Cruz-Ramírez et al., 2014). It represents the lowest percentage of patterns correctly predicted as belonging to each class, concerning the total number of examples in that class. The MS is useful to guarantee that all the classes are decently classified and can be calculated as:

$$\mathrm{MS} = \min\left\{ S_q = \frac{n_{qq}}{n_{q\bullet}}; q = 1, \ldots, Q \right\}, \tag{4}$$

Table 2
Hierarchical experiments types. Number of classes refers to the classes considered for each task.

	Macro loss	Macro output	# classes	Micros loss	Micros output	# classes
1	CCE	Softmax	4	CCE	Softmax	3, 3, 3
2	CCE Beta	CLM Logit	4	CCE Beta	CLM Logit	3, 3, 3
3	CCE Beta	CLM Probit	4	CCE Beta	CLM Probit	3, 3, 3
4	CCE Beta	CLM Logit	4	CCE	Softmax	3, 3, 3
5	CCE Beta	CLM Probit	4	CCE	Softmax	3, 3, 3
6	QWK	CLM Logit	4	QWK	CLM Logit	3, 3, 3
7	QWK	CLM Probit	4	QWK	CLM Probit	3, 3, 3

Table 3
Different sets of non-hierarchical experiments. The ECOC alternatives (13 and 14) consist of 7 and 9 binary tasks respectively, given that the multi-class problem is decomposed in multiple binary tasks.

	Loss	Output	# classes
8	CCE	Softmax	10
9	CCE Beta	CLM Logit	10
10	CCE Beta	CLM Probit	10
11	QWK_L	CLM Logit	10
12	QWK_L	CLM Probit	10
13	CCE	Softmax	2 (7 tasks)
14	CCE	Softmax	2 (9 tasks)

where n_{qq} and $n_{q\bullet}$ are, respectively, the number of samples correctly classified in category q, and the total number of patterns predicted as class q.

- Mean absolute error (MAE) (Cruz-Ramírez et al., 2014). It is the average absolute deviation of the predicted class from the true class (number of categories in the ordinal scale). It can be defined as:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\mathcal{O}(y_i) - \mathcal{O}(\hat{y}_i)|, \tag{5}$$

where $\mathcal{O}(y_i)$ and $\mathcal{O}(\hat{y}_i)$ are the order of the true and predicted labels for the ith sample.

- Correctly Classified Rate (CCR) or Accuracy. It is the standard metric for classification and determines the ratio of test samples that have been correctly classified.

QWK, MS and CCR should be maximised, while MAE should be minimised.

Then, taking into account the aforementioned metrics, the results of the experiments with the three different architectures are shown in Tables 4–6. The experiments that are marked as hierarchical were run using the proposed hierarchical approach, while the non-hierarchical methods were run for comparison purposes.

From a solely descriptive point of view, the results show that the hierarchical method achieved the best results for CCR, MAE and MS for all the architectures. However, the non-hierarchical model that uses the Beta regularised cross-entropy with the CLM logit resulted in better performance for the QWK metric. For the VGG-16 models, the hierarchical alternative which uses the Beta regularised cross-entropy for the macro model combined with the CLM with logit link, and the standard cross-entropy loss with the standard softmax output for the micro models obtained the best results. In the case of the ResNet-101 architecture, the best results regarding MS and MAE were produced by the same alternative that achieved the best results in VGG-16. However, the best value for CCR was obtained when using the standard cross-entropy loss and the softmax for both, the macro and the micro models. Finally, when using the DenseNet-121 architecture, the best results are achieved with the standard cross-entropy and the softmax function for both steps. Also, in Fig. 4, a boxplot is represented for each of the metrics considered for the architecture that obtained the best results (i.e. ResNet-101). For a more in depth comparison, the boxplots corresponding to the other model architectures have been added in Appendix B.

As we mentioned in Section 2, the computational cost of the proposed hierarchical approach should be lower than the cost associated with the ECOC approaches. To confirm that fact, the mean time required to complete both experiments was compared for each of the architectures. In these terms, for the VGG-16 architecture, the hierarchical approach took 20 s while the ECOC needed 45 seconds per epoch. For the ResNet-101 model, the first took 27 s and the second 91 s. For the DenseNet-121 model, they took 23 s and 51 s respectively.

To sum up, the following conclusions can be obtained from the results tables:

- Our hierarchical approaches obtained the best results for MS, MAE and CCR considering all the model architectures.
- Our hierarchical model that uses the CCE Beta + CLM Logit for the macro task and CCE + Softmax for the micro classifiers obtained the best performance in most of the cases. The second best alternative is the hierarchical model which uses CCE + Softmax for both steps.
- The computational cost of the proposed hierarchical approach is lower than the cost associated with the ECOC approaches.

Even though the hierarchical method obtained the best performance for most of the metrics in all the model architectures, in the next section, a statistical analysis is performed to check whether the differences obtained are significant.

6.1. Statistical analysis

In this section, a statistical analysis has been performed to determine which of the tested alternatives are significantly better than the others. Also, we aim to check whether the proposed approach obtains better results than the baseline approach and previously proposed methods. To do that, we considered all the model architectures and each of the metrics was analysed separately.

First, for each of the four analysed metrics, a Kolmogorov–Smirnov (Massey Jr., 1951) test was performed to check whether the 30 test values obtained, for each method and architecture, from the different seeds are normally distributed. The test confirmed that the values are normally distributed (p-value < 0.001) for all the metrics and methodologies and architectures, except for the MS metric when using the QWK loss. Therefore, an Analysis of variance II (ANOVA II) (Miller Jr., 1997) test, where the factors considered are the methodology applied and the model architecture used, was performed for each of the metrics. It is worth noting that the statistical tests where performed using 90 points for each method (30 for each architecture).

First, the CCR metric was considered. $CCR_{ij}, (i = 1, \dots, 12; j = 1, 2, 3)$ denotes all the methodologies considered. The observations fit the following equation:

$$CCR_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \epsilon_{ijk}, \quad k = 1, \dots, 30, \tag{6}$$

where μ is the fixed effect that is common to all the populations, α_i is the effect associated with the ith level of the first factor, β_j is the effect associated with the jth level of the second factor, γ_{ij} is the interaction between the ith level of the first factor and the jth level of the second factor, and the term ϵ_{ijk} is the influence of the random effects in the final result. The results of this test are shown in Table 7.

Table 4
Mean results for the test set and 30 executions using the VGG-16 architecture. The Hier. column indicates whether the method uses the proposed hierarchical methodology or not.

	Hier.	Macro loss	Macro output	Micros loss	Micros output	QWK↑	MS↑	MAE↓	CCR↑
1	Yes	CCE	Softmax	CCE	Softmax	0.8921	0.2311	0.7435	0.5129
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit	0.9088	0.1778	0.7058	0.4908
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit	0.9099	0.1782	0.7012	0.4937
4	Yes	CCE Beta	CLM Logit	CCE	Softmax	0.9055	**0.2400**	**0.6947**	**0.5238**
5	Yes	CCE Beta	CLM Probit	CCE	Softmax	0.9033	0.1893	0.7057	0.5230
6	Yes	QWK	CLM Logit	QWK	CLM Logit	0.9056	0.1349	0.7097	0.5031
7	Yes	QWK	CLM Probit	QWK	CLM Probit	0.9062	0.1334	0.7152	0.4948
8	No	CCE	Softmax	–	–	0.8713	0.1643	0.8253	0.4839
9	No	CCE Beta	CLM Logit	–	–	**0.9192**	0.2152	0.7121	0.4478
10	No	CCE Beta	CLM Probit	–	–	0.9161	0.2110	0.7299	0.4389
11	No	QWK	CLM Logit	–	–	0.9106	0.0000	0.7370	0.4544
12	No	QWK	CLM Probit	–	–	0.9123	0.0000	0.7358	0.4517
13	No	CCE	Softmax	-	-	0.8735	0.1552	0.8625	0.4666
14	No	CCE	Softmax	-	-	0.8852	0.0457	0.8420	0.4313

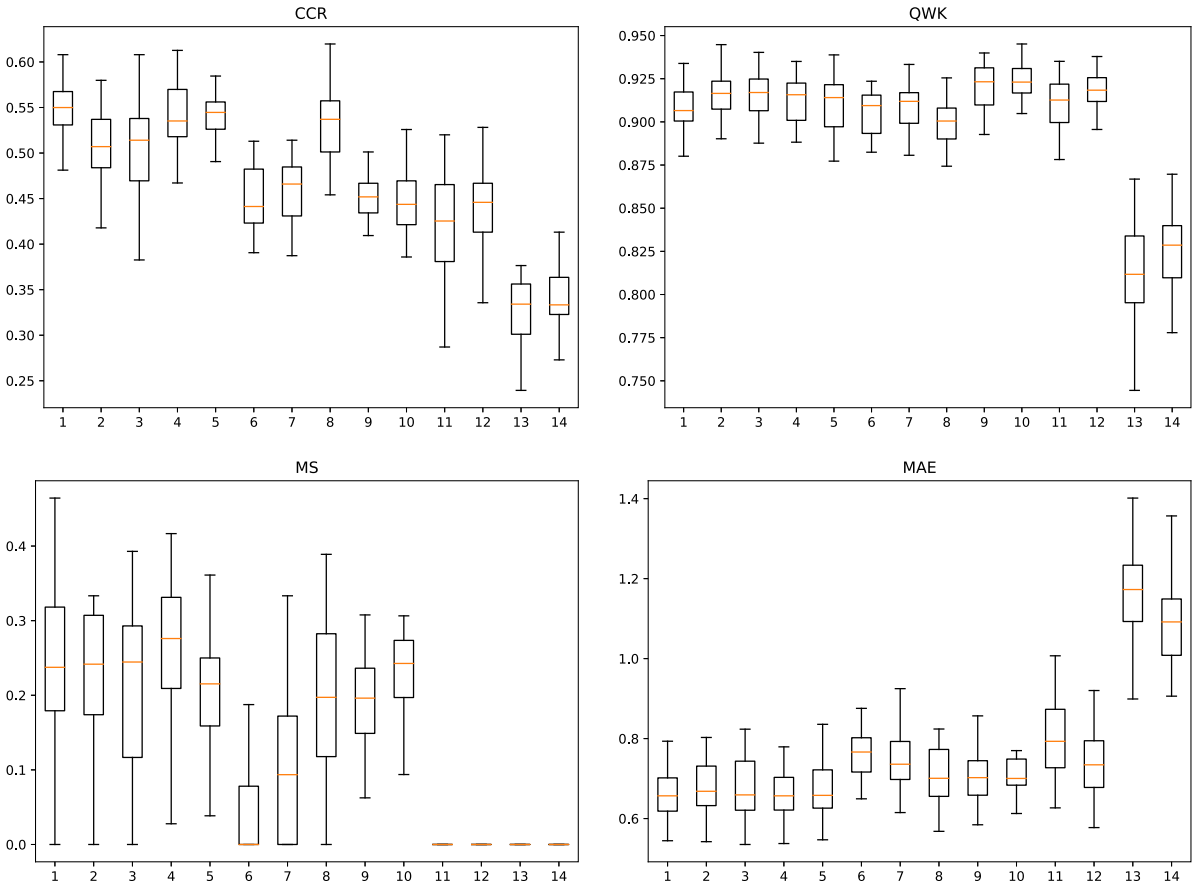

Fig. 4. Boxplots for all the test metrics using the ResNet-101 architecture. Methods are identified with the numbers defined in Table 5.

When the p-value represented in the ANOVA table is smaller than 0.01, the factor effect is statistically significant at a level of confidence of 99%. The results obtained from this test reported that the methodology and the architectures used significantly influence the test accuracy value obtained. Also, there is an interaction between both factors that also influences the final result.

Given that there are significant differences in mean CCR depending on the methodology considered, a post-hoc HSD Tukey's (Tukey, 1949) test was performed to compare the mean CCR values in the test set between all the methodologies. The results of this test are summarised in Table 8. It groups the methodologies into four different subsets according to their performance such that the elements within a subset are not significantly different between them, while the differences between members of different groups are significant. The first subset contains the worst methodologies while the last subset groups the best ones.

The results in Table 8 depict that the ordinal methodologies tend to reduce the accuracy even though they improve the performance regarding ordinal metrics. However, using the proposed hierarchical approach, the accuracy metric is statistically improved, obtaining higher values than the standard nominal approach. In this case, the best methodology was the hierarchical one that used the standard categorical cross-entropy loss and the softmax output for both, the macro and the micro models. However, there are no significant differences with the hierarchical methods which use the Beta regularised CCE and the CLM with logit or probit link for the macro model, and the standard CCE + softmax for the micro models. The fact that using a nominal

Table 5
Mean results for the test set and 30 executions using the ResNet-101 architecture. The Hier. column shows whether the method uses the proposed hierarchical methodology or not.

	Hier.	Macro loss	Macro output	Micros loss	Micros output	QWK↑	MS↑	MAE↓	CCR↑
1	Yes	CCE	Softmax	CCE	Softmax	0.9080	0.2310	0.6647	**0.5479**
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit	0.9165	0.2221	0.6737	0.5075
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit	0.9155	0.2036	0.6863	0.5031
4	Yes	CCE Beta	CLM Logit	CCE	Softmax	0.9129	**0.2469**	**0.6612**	0.5408
5	Yes	CCE Beta	CLM Probit	CCE	Softmax	0.9108	0.2126	0.6765	0.5352
6	Yes	QWK	CLM Logit	QWK	CLM Logit	0.9065	0.0429	0.7599	0.4500
7	Yes	QWK	CLM Probit	QWK	CLM Probit	0.9059	0.0953	0.7586	0.4517
8	No	CCE	Softmax	–	–	0.9002	0.1981	0.7050	0.5318
9	No	CCE Beta	CLM Logit	–	–	0.9215	0.1910	0.7162	0.4427
10	No	CCE Beta	CLM Probit	–	–	**0.9239**	0.2314	0.7075	0.4435
11	No	QWK	CLM Logit	–	–	0.9091	0.0000	0.8078	0.4166
12	No	QWK	CLM Probit	–	–	0.9169	0.0000	0.7461	0.4357
13	No	CCE	Softmax	-	-	0.8120	0.0010	1.1705	0.3306
14	No	CCE	Softmax	-	-	0.8228	0.0000	1.0966	0.3395

Table 6
Mean results for the test set and 30 executions using the DenseNet-121 architecture. The Hier. column indicates whether the method uses the proposed hierarchical methodology or not.

	Hier.	Macro loss	Macro output	Micros loss	Micros output	QWK↑	MS↑	MAE↓	CCR↑
1	Yes	CCE	Softmax	CCE	Softmax	0.8880	**0.1988**	**0.7261**	**0.5276**
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit	0.8924	0.1508	0.7869	0.4560
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit	0.8912	0.1506	0.7910	0.4631
4	Yes	CCE Beta	CLM Logit	CCE	Softmax	0.8888	0.1551	0.7621	0.5031
5	Yes	CCE Beta	CLM Probit	CCE	Softmax	0.8827	0.1703	0.7825	0.4997
6	Yes	QWK	CLM Logit	QWK	CLM Logit	0.8823	0.0906	0.8380	0.4344
7	Yes	QWK	CLM Probit	QWK	CLM Probit	0.8859	0.0933	0.8234	0.4435
8	No	CCE	Softmax	–	–	0.7696	0.0299	1.1420	0.4081
9	No	CCE Beta	CLM Logit	–	–	0.8867	0.1548	0.9041	0.3663
10	No	CCE Beta	CLM Probit	–	–	**0.8992**	0.1797	0.8314	0.4049
11	No	QWK	CLM Logit	–	–	0.8710	0.0000	1.0047	0.3467
12	No	QWK	CLM Probit	–	–	0.8790	0.0000	0.9606	0.3608
13	No	CCE	Softmax	-	-	0.8776	0.1773	0.8390	0.4707
14	No	CCE	Softmax	-	-	0.8921	0.1148	0.7996	0.4651

Table 7
Results of the ANOVA II test for the CCR metric. SS stands for Sum of Squares, DF refers to the Degrees of Freedom, MSq are the Mean Squares, and F is the F-ratio.

Source	SS	DF	MSq	F	*p*-value
Corrected model	3.843	41	0.094	42.924	< 0.001
Intercept	267.244	1	267.244	122396.130	< 0.001
Method	2.252	13	0.173	79.338	< 0.001
Model	0.346	2	0.173	79.255	< 0.001
Method * Model	1.245	26	0.048	21.923	< 0.001
Error	2.659	1218	0.002		
Total	273.746	1260			
Corrected total	6.502	1259			

Table 8
Results of the post-hoc HSD Tukey's test for the CCR metric.

	Hier.	Macro loss	Macro out	Micro loss	Micro out	Subsets			
						1	2	3	4
11	No	QWK	CLM Logit	–	–	0.4059			
14	No	CCE	Softmax	–	–	0.4120			
12	No	QWK	CLM Probit	–	–	0.4161			
9	No	CCE Beta	CLM Logit	–	–	0.4189			
13	No	CCE	Softmax	–	–	0.4226			
10	No	CCE Beta	CLM Probit	–	–	0.4289			
11	Yes	QWK	CLM Logit	QWK	CLM Logit		0.4625		
12	Yes	QWK	CLM Probit	QWK	CLM Probit		0.4633	0.4633	
8	No	CCE	Softmax	–	–		0.4746	0.4746	
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit		0.4848	0.4848	
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit			0.4866	
5	Yes	CCE Beta	CLM Probit	CCE	Softmax				0.5193
4	Yes	CCE Beta	CLM Logit	CCE	Softmax				0.5226
1	Yes	CCE	Softmax	CCE	Softmax				0.5295
p-values						0.060	0.080	0.052	0.975

Table 9
Results of the post-hoc HSD Tukey's test for the QWK metric.

	Hier.	Macro loss	Macro out	Micro loss	Micro out	Subsets			
						1	2	3	4
8	No	CCE	Softmax	–	–	0.8470			
13	No	CCE	Softmax	–	–	0.8544	0.8544		
14	No	CCE	Softmax	–	–		0.8667		
11	No	QWK	CLM Logit	–	–			0.8952	
1	Yes	CCE	Softmax	CCE	Softmax			0.8960	
6	Yes	QWK	CLM Logit	QWK	CLM Logit			0.8982	
5	Yes	CCE Beta	CLM Probit	CCE	Softmax			0.8989	0.8989
7	Yes	QWK	CLM Probit	QWK	CLM Probit			0.8993	0.8993
12	No	QWK	CLM Probit	–	–			0.9007	0.9007
4	Yes	CCE Beta	CLM Logit	CCE	Softmax			0.9024	0.9024
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit			0.9056	0.9056
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit			0.9059	0.9059
9	No	CCE Beta	CLM Logit	–	–			0.9091	0.9091
10	No	CCE Beta	CLM Probit	–	–				0.9130
p-values						0.916	0.198	0.076	0.065

Table 10
Results of the post-hoc HSD Tukey's test for the MS metric.

	Hier.	Macro loss	Macro out	Micro loss	Micro out	Subsets						
						1	2	3	4	5	6	7
11	No	QWK	CLM Logit	–	–	0.000						
12	No	QWK	CLM Probit	–	–	0.000						
14	No	CCE	Softmax	–	–		0.080					
6	Yes	QWK	CLM Logit	QWK	CLM Logit		0.090	0.090				
7	Yes	QWK	CLM Probit	QWK	CLM Probit		0.107	0.107				
8	No	CCE	Softmax	–	–			0.1308	0.1308			
13	No	CCE	Sofmtax	–	–				0.166	0.166		
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit					0.178	0.178	
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit					0.184	0.184	0.184
9	No	CCE Beta	CLM Logit	–	–					0.187	0.187	0.187
10	No	CCE Beta	CLM Probit	–	–					0.207	0.207	0.207
5	Yes	CCE Beta	CLM Probit	CCE	Softmax					0.208	0.208	0.208
4	Yes	CCE Beta	CLM Logit	CCE	Softmax						0.214	0.214
1	Yes	CCE	Softmax	CCE	Softmax							0.220
p-values						1.000	0.667	0.064	0.220	0.063	0.179	0.173

approach for the second phase achieves better results is due to the lower number of labels in the micro-tasks.

The same statistical analysis has been performed for the QWK metric. The ANOVA II test also reported significant differences for the different factors (p-value < 0.001) and a significant interaction between them. Therefore, the results of the post-hoc HSD Tukey's test for the different methods considered are shown in Table 9.

In this case, the results show that the best methodology is the non-hierarchical one that uses the Beta regularised cross-entropy and the CLM with logit link in the output. However, there are no significant differences with the other methods in the same group. The worst results were obtained by the standard nominal approach and the ECOC methods. The ordinal and hierarchical methodologies highly improved the performance concerning the standard nominal approach.

Following the same methodology, the MS metric was analysed. Again, the ANOVA II test performed over the MS test results reported that there are significant differences between the methodologies and between the different architectures. Moreover, there is a significant interaction between these two factors. Therefore, a posthoc HSD Tukey's Test taking into account the methodologies was performed and the results are shown in Table 10.

In this case, the same hierarchical approach that performed the best for the CCR metric also obtained the best MS results.

By analysing the composition of each of the seven subsets, some conclusions can be obtained:

1. The non-hierarchical methodologies that use the QWK loss function fail to classify at least one of the classes, obtaining always a value of 0 for the minimum sensitivity. Therefore, they should be discarded even though they perform well regarding the other metrics.
2. The hierarchical approach solves the problem related to the QWK loss: all the classes are represented in the final predictions.
3. Five out of seven methods grouped in the best three subsets are hierarchical, including the three methods that obtained the best average performance.

Finally, for the MAE metric, the same analysis was performed. The ANOVA II test reported significant differences between the methods considered and the architectures tested (p-value < 0.001). Also, there is a significant interaction between factors (p-value < 0.001) Then, a post-hoc HSD Tukey's test was performed to determine which methods achieve better performance. The results of this test are given in Table 11.

The results in this table show that the best mean result was obtained by the hierarchical methodology which employs the Beta regularised cross-entropy loss with the CLM Logit in the first model and the CCE with softmax for the others. However, there are no significant differences with the other methods in group 1. Also, the table shows that most of the methods in group 1 are hierarchical methods. The last four groups only contain non-hierarchical methods and the ECOC approaches. Therefore, the overall results with hierarchical methods are better than the ones obtained by the corresponding non-hierarchical ones.

After comparing the different methods using the post-hoc tests, the three model architectures were compared too. The post-hoc HSD Tukey's test shown that the VGG-16 and the ResNet-101 are not significantly different and they both are significantly better than the DenseNet-121 concerning the QWK and MS metrics. For the CCR

Table 11
Results of the post-hoc HSD Tukey's test for the MAE metric.

	Hier.	Macro loss	Macro out	Micro loss	Micro out	Subsets							
						1	2	3	4	5	6	7	8
4	Yes	CCE Beta	CLM Logit	CCE	Softmax	0.706							
1	Yes	CCE	Softmax	CCE	Softmax	0.711	0.711						
5	Yes	CCE Beta	CLM Probit	CCE	Softmax	0.722	0.722	0.722					
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit	0.722	0.722	0.722					
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit	0.726	0.726	0.726					
10	No	CCE Beta	CLM Probit	–	–	0.756	0.756	0.756	0.756				
7	Yes	QWK	CLM Probit	QWK	CLM Probit	0.766	0.766	0.766	0.766				
6	Yes	QWK	CLM Logit	QWK	CLM Logit		0.769	0.769	0.769				
9	No	CCE Beta	CLM Logit	–	–			0.778	0.778				
12	No	QWK	CLM Probit	–	–				0.814	0.814			
11	No	QWK	CLM Logit	–	–					0.850	0.850		
8	No	CCE	Softmax	–	–						0.891	0.891	
14	No	CCE	Softmax	–	–							0.913	0.913
13	No	CCE	Softmax	–	–								0.957
p-values						0.063	0.086	0.113	0.084	0.791	0.590	0.995	0.442

Table A.12
Results of the post-hoc HSD Tukey's test for the model and the CCR metric.

Model	Subsets		
	1	2	3
DenseNet-121	0.4393		
ResNet-101		0.4626	
VGG-16			0.4797
p-values	1.000	1.000	1.000

Table A.13
Results of the post-hoc HSD Tukey's test for the model and the QWK metric.

Model	Subsets	
	1	2
DenseNet-121	0.8776	
VGG-16		0.8987
ResNet-101		0.9006
p-values	1.000	0.624

Table A.14
Results of the post-hoc HSD Tukey's test for the model and the MS metric.

Model	Subsets	
	1	2
DenseNet-121	0.1190	
VGG-16		0.1519
ResNet-101		0.1562
p-values	1.000	0.734

Table A.15
Results of the post-hoc HSD Tukey's test for the model and the MAE metric.

Model	Subsets		
	1	2	3
VGG-16	0.7443		
ResNet-101		0.7736	
DenseNet-121			0.8565
p-values	1.000	1.000	1.000

metric, the ResNet-101 is significantly better than the other two architectures, which also are significantly different between them. Finally, regarding the MAE, the VGG-16 architecture achieved the best mean results. The complete statistical comparison of the architectures can be found in Appendix A.

Therefore, taking into account all the metrics, some general conclusions can be derived:

1. The proposed hierarchical methodologies perform significantly better concerning the CCR, MS and MAE metrics, while the non-hierarchical ones achieve better results for the QWK. However, there are no significant differences concerning QWK with most of the other methodologies.
2. Using an ordinal loss function and output function for the macro classifier and the standard nominal approach for the micro classifier usually obtained the best results.
3. The ResNet-101 obtained the best results followed by the VGG-16 architecture. Nevertheless, the proposed methodology obtains a significant performance enhancement for all architectures tested. Therefore, the proposed method can be generalised to different types of architectures.

7. Conclusions and future work

In this work, we have proposed an ordinal hierarchical method that is aimed to solve tasks where the labels follow are structured in different levels and naturally ordered. Then, we have applied this method to solve an AQC task related with the manufacturing industry. In particular, the problem relies on classifying wooden stocks based on the aesthetic quality of the wood used to manufacture them. Differently from other metric quality control task that are usually solved with nominal CNN approach, the considered AQC task is characterised by different scale of labels (macro and micro). Taking into account the complexity and the challenges that originated from the considered task, our main contribution lies in the proposal of a hierarchical approach that is specifically tailored for learning ordinal classes in two separate phases. In the first one, a single model was used to predict the macro label of each pattern. In the second one, one model was used for each macro label to predict the corresponding micro class. Moreover, it was combined with an ordinal loss regularisation and an output layer based on the CLM to encourage an ordinal classification. Different alternatives were tested with three different model architectures and the experimental results showed that the hierarchical methods obtained the best results for most of the metrics and architectures. In general terms, the hierarchical approaches obtained better results than other state-of-the-art non-hierarchical approaches. The main benefit of the described approach is that it improves the performance of this kind of tasks at the same time that it simplifies the problem by dividing the classification task into multiple models. From a practical perspective, this ensures a greater flexibility of the DSS to also provide individual predictions for each macro class. This fact may support the human operator to classify different level of grades, and eventually focus the attention only on the classification of specific level (i.e. macro classes). Moreover the proposed approach ensures a greater accuracy for classifying samples in the overall 10 classes, but maintaining, at

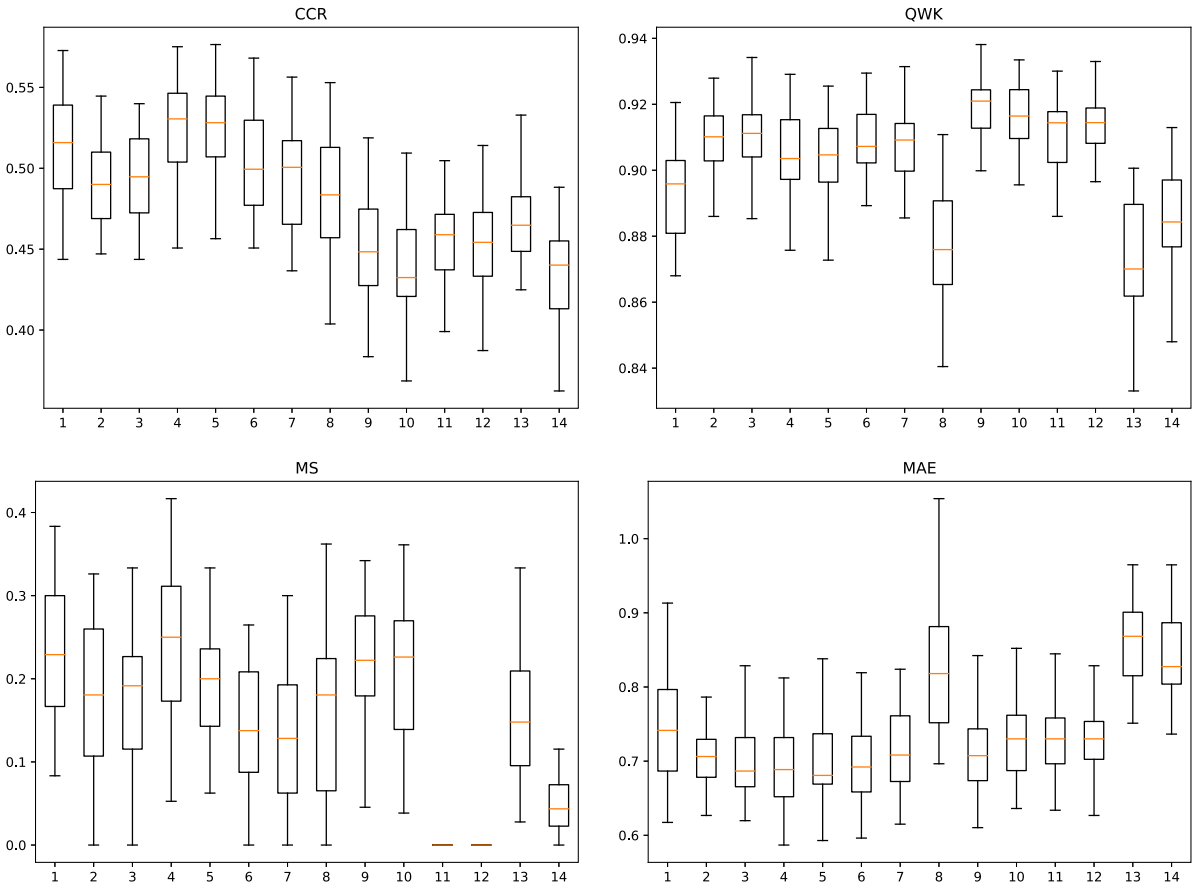

Fig. B.5. Boxplots for all the test metrics using the VGG-16 architecture. Methods are identified with the numbers defined in Table 4.

the same time, misclassification errors close to the correct class, which is a relevant aspect for the industrial production domain.

Given that the proposed approach improved the classification performance compared to the non-hierarchical methodologies, in future works, the same methodology could be applied to other types of hierarchical problems. Its applications include but are not limited to any other type of QC problem, which includes AQC but also other problems related to the overall quality of a product from the engineering point of view. The only limitation of the proposed approach is that the labels must follow a natural order and must also be decomposed hierarchically. However, there are some problems where the hierarchical structure can be inferred from the characteristics of the problem. Also, the proposed methodology can be extended to other areas different from the industry. For example, predicting the age of people from photographs of their faces can be a hierarchical ordinal problem if the age ranges are divided into sub-groups (as in the case of clinical risk based on age and disease (Romeo and Frontoni, 2022)).

CRediT authorship contribution statement

Víctor Manuel Vargas: Methodology, Software, Writing – original draft, Investigation, Visualization. **Pedro Antonio Gutiérrez:** Conceptualization, Methodology, Validation, Writing – review & editing. **Riccardo Rosati:** Methodology, Validation, Data curation, Writing – review & editing. **Luca Romeo:** Methodology, Data curation, Resources, Writing – review & editing. **Emanuele Frontoni:** Formal analysis, Supervision, Writing – review & editing, Project administration, Funding acquisition. **César Hervás-Martínez:** Formal analysis, Supervision, Writing – review & editing, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been partially subsidised by "Agencia Española de Investigación (España)" (grant reference: PID2020-115454GB-C22 / AEI / 10.13039 / 501100011033), by "Consejería de Salud y Familia (Junta de Andalucía)" (grant reference: PS-2020-780) and by "Consejería de Transformación Económica, Industria, Conocimiento y Universidades (Junta de Andalucía) y Programa Operativo FEDER 2014-2020" (grant references: UCO-1261651 and PY20_00074). This work has been also supported within the research agreement between Università Politecnica delle Marche and Benelli Armi Spa for the "4USER Project" (User and Product Development: from Virtual Experience to Model Regeneration) funded on the POR MARCHE FESR 2014-2020-ASSE 1-OS 1-ACTION 1.1-INT. 1.1.1. Promotion of industrial research and experimental development in the areas of smart specialisation -LINEA 2 -Bando 2019, approved with DDPF 293 of 22/11/2019. Víctor Manuel Vargas's research has been subsidised by the FPU Predoctoral Program of the Spanish Ministry of Science, Innovation and Universities (MCIU), grant reference FPU18/00358.

Appendix A. Model architectures statistical comparison

In addition, to complete the statistical analysis performed in Section 6.1, the three model architectures considered in our work are compared using statistical tests. During the general statistical analysis,

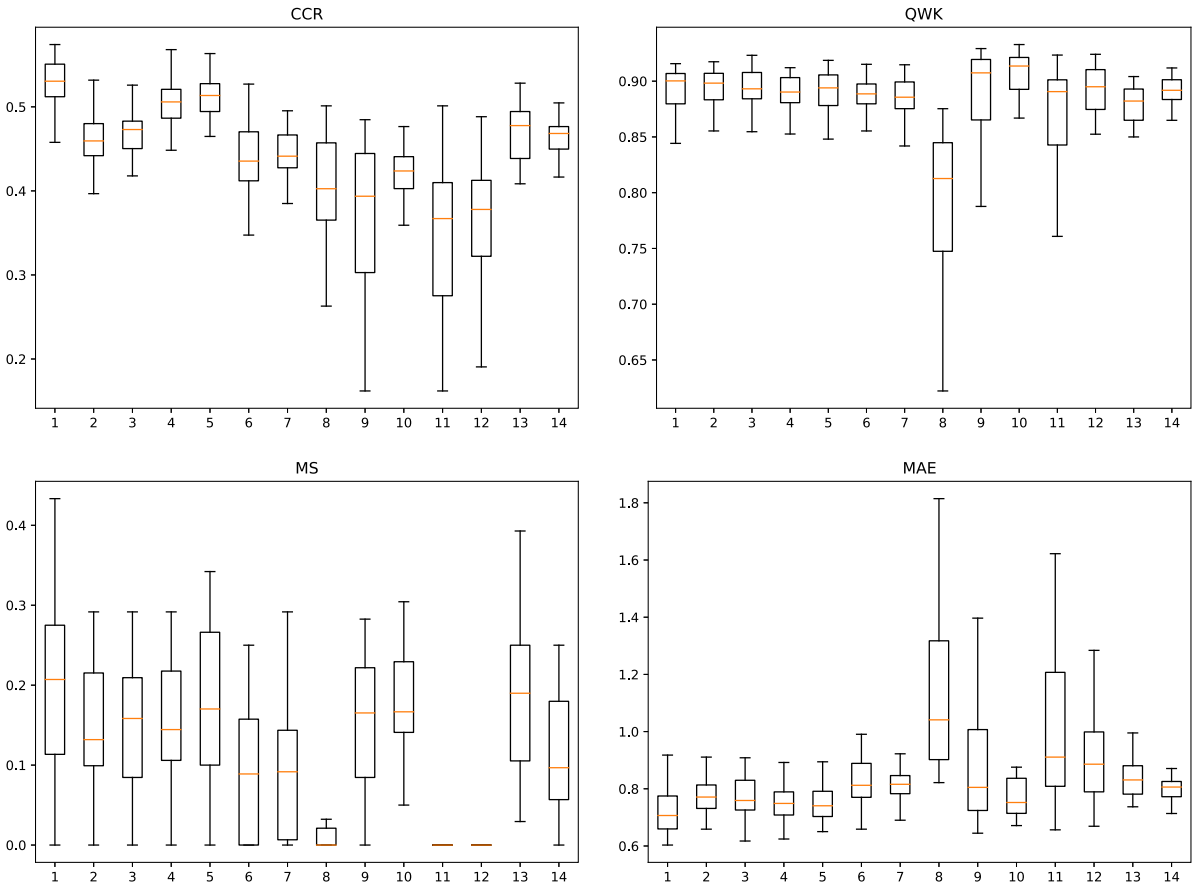

Fig. B.6. Boxplots for all the test metrics using the DenseNet-121 architecture. Methods are identified with the numbers defined in Table 6.

the ANOVA II tests reported significant differences between the architectures for all the metrics. Therefore, a post-hoc HSD Tukey's test is performed for each of the metrics.

First of all, the statistical test was performed for the accuracy metric. The results of the post-hoc test are shown in Table A.12. The two best architectures for the CCR metric are the VGG-16 and the ResNet-101. They are significantly better than the DenseNet-121 model.

For the QWK metric, the same analysis is performed. The results are shown in Table A.13, and, this time, they show that all the architectures are significantly different. The DenseNet-121 model is, again, the worst, but, in this case, the VGG-16 and the ResNet-101 show significant differences and the residual network is better.

Then, the minimum sensitivity metric values are analysed. The results of the post-hoc test are shown in Table A.14. In this case, the conclusions are the same that were obtained for the accuracy metric. The ResNet-101 and the VGG-16 architectures obtained the best results and are significantly better than the DenseNet-121.

Finally, the MAE metric is analysed. The results are shown in Table A.15. Again, the results are similar: the ResNet-101 and the VGG-16 models obtained the best results. The differences between the results obtained using these two architectures and the results obtained using the DenseNet-121 model are significant.

Therefore, from these tests, we can conclude that the best model architecture is ResNet-101, given that it is significantly better than the other alternatives regarding the QWK metric, and it is as good as the VGG-16 model considering the other metrics.

Appendix B. Boxplots of VGG-16 and DenseNet-121

In this appendix, the boxplots corresponding to the results of the VGG-16 and DenseNet-121 architectures are shown. Fig. B.5 shows the boxplots for each metric for the VGG-16 architecture, while Fig. B.6 shows the boxplots for the DenseNet-121 architecture.

References

Agrawal, A.K., Chakraborty, G., 2022. Semi-supervised implementation of SVM-based error-correcting output code for damage-type identification in structures. Struct. Control Health Monit. e2967. http://dx.doi.org/10.1002/stc.2967.

Agresti, A., 2010. Analysis of Ordinal Categorical Data. Vol. 656, J. Wiley & Sons.

Akshayarathna, A., Divya Darshini, K., Dhalia Sweetlin, J., 2021. A convolutional neural network model to predict air and water hazards. In: Machine Vision and Augmented Intelligence—Theory and Applications. Springer, pp. 413–432. http://dx.doi.org/10.1007/978-981-16-5078-9_35.

Albuquerque, T., Cruz, R., Cardoso, J.S., 2021. Ordinal losses for classification of cervical cancer risk. PeerJ Comput. Sci. 7, 1–21. http://dx.doi.org/10.7717/peerj-cs.457.

Barbero-Gómez, J., Gutiérrez, P.A., Hervás-Martínez, C., 2022. Error-correcting output codes in the framework of deep ordinal classification. Neural Process. Lett. 1–32. http://dx.doi.org/10.1007/s11063-022-10824-7.

Bora, M.B., Daimary, D., Amitab, K., Kandar, D., 2020. Handwritten character recognition from images using CNN-ECOC. Procedia Comput. Sci. 167, 2403–2409. http://dx.doi.org/10.1016/j.procs.2020.03.293.

Caballero, J.C.F., Martínez, F.J., Hervás, C., Gutiérrez, P.A., 2010. Sensitivity versus accuracy in multiclass problems using memetic pareto evolutionary neural networks. IEEE Trans. Neural Netw. 21 (5), 750–770. http://dx.doi.org/10.1109/TNN.2010.2041468.

Cao, W., Mirjalili, V., Raschka, S., 2020. Rank consistent ordinal regression for neural networks with application to age estimation. Pattern Recognit. Lett. 140, 325–331. http://dx.doi.org/10.1016/j.patrec.2020.11.008.

Chiarello, F., Belingheri, P., Fantoni, G., 2021. Data science for engineering design: State of the art and future directions. Comput. Ind. 129, 103447. http://dx.doi.org/10.1016/j.compind.2021.103447.

Cruz-Ramírez, M., Hervás-Martínez, C., Sánchez-Monedero, J., Gutiérrez, P.A., 2014. Metrics to guide a multi-objective evolutionary algorithm for ordinal classification. Neurocomputing 135, 21–31. http://dx.doi.org/10.1016/j.neucom.2013.05.058.

de la Torre, J., Puig, D., Valls, A., 2018. Weighted kappa loss function for multi-class classification of ordinal data in deep learning. Pattern Recognit. Lett. 105, 144–154. http://dx.doi.org/10.1016/j.patrec.2017.05.018.

Dietterich, T.G., Bakiri, G., 1994. Solving multiclass learning problems via error-correcting output codes. J. Artificial Intelligence Res. 2, 263–286. http://dx.doi.org/10.1613/jair.105.

Durán-Rosal, A.M., Camacho-Cañamón, J., Gutiérrez, P.A., Guiote Moreno, M.V., Rodríguez-Cáceres, E., Vallejo Casas, J.A., Hervás-Martínez, C., 2021. Ordinal classification of the affectation level of 3D-images in parkinson diseases. Sci. Rep. 11 (1), 1–13. http://dx.doi.org/10.1038/s41598-021-86538-y.

Elsisi, M., Tran, M.-Q., Mahmoud, K., Lehtonen, M., Darwish, M.M., 2021. Deep learning-based industry 4.0 and internet of things towards effective energy management for smart buildings. Sensors 21 (4), 1038. http://dx.doi.org/10.3390/s21041038.

Ge, C., Wang, J., Wang, J., Qi, Q., Sun, H., Liao, J., 2020. Towards automatic visual inspection: A weakly supervised learning method for industrial applicable object detection. Comput. Ind. 121, 1–11. http://dx.doi.org/10.1016/j.compind.2020.103232.

Gutierrez, P.A., Perez-Ortiz, M., Sanchez-Monedero, J., Fernandez-Navarro, F., Hervas-Martinez, C., 2016. Ordinal regression methods: survey and experimental study. IEEE Trans. Knowl. Data Eng. 28 (1), 127–146. http://dx.doi.org/10.1109/TKDE.2015.2457911.

Hansen, M.F., Smith, M.L., Smith, L.N., Salter, M.G., Baxter, E.M., Farish, M., Grieve, B., 2018. Towards on-farm pig face recognition using convolutional neural networks. Comput. Ind. 98, 145–152. http://dx.doi.org/10.1016/j.compind.2018.02.016.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition. pp. 770–778.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4700–4708.

Kao, Y., He, R., Huang, K., 2017. Deep aesthetic quality assessment with semantic information. IEEE Trans. Image Process. 26 (3), 1482–1495. http://dx.doi.org/10.1109/TIP.2017.2651399.

Kiangala, K.S., Wang, Z., 2020. An effective predictive maintenance framework for conveyor motors using dual time-series imaging and convolutional neural network in an industry 4.0 environment. IEEE Access 8, 121033–121049. http://dx.doi.org/10.1109/ACCESS.2020.3006788.

Liu, X., Fan, F., Kong, L., Diao, Z., Xie, W., Lu, J., You, J., 2020. Unimodal regularized neuron stick-breaking for ordinal classification. Neurocomputing 388 (7), 34–44. http://dx.doi.org/10.1016/j.neucom.2020.01.025.

Massey Jr., F.J., 1951. The Kolmogorov-Smirnov test for goodness of fit. J. Amer. Statist. Assoc. 46 (253), 68–78. http://dx.doi.org/10.1080/01621459.1951.10500769.

Miller Jr., R.G., 1997. Beyond ANOVA: Basics of Applied Statistics. Chapman and Hall/CRC.

Onchis, D.M., Gillich, G.-R., 2021. Stable and explainable deep learning damage prediction for prismatic cantilever steel beam. Comput. Ind. 125, 1–8.

Ouzounis, A., Sidiropoulos, G.K., Papakostas, G., Sarafis, I., Stamkos, A., Solakis, G., 2021. Interpretable deep learning for marble tiles sorting. In: Proceedings of the 2nd International Conference on Deep Learning Theory and Applications. pp. 101–108. http://dx.doi.org/10.5220/0010517001010108.

Pazzaglia, G., Martini, M., Rosati, R., Romeo, L., Frontoni, E., 2021. A deep learning-based approach for automatic leather classification in industry 4.0. In: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV. Springer, pp. 662–674.

Romeo, L., Frontoni, E., 2022. A unified hierarchical XGBoost model for classifying priorities for COVID-19 vaccination campaign. Pattern Recognit. 121, 108197. http://dx.doi.org/10.1016/j.patcog.2021.108197.

Rosati, R., Romeo, L., Cecchini, G., Tonetto, F., Perugini, L., Ruggeri, L., Viti, P., Frontoni, E., 2021. Bias from the wild industry 4.0: Are we really classifying the quality or shotgun series? In: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV. Springer, pp. 637–649. http://dx.doi.org/10.1007/978-3-030-68799-1_46.

Sánchez-Monedero, J., Pérez-Ortiz, M., Saez, A., Gutiérrez, P.A., Hervás-Martínez, C., 2018. Partial order label decomposition approaches for melanoma diagnosis. Appl. Soft Comput. 64, 341–355. http://dx.doi.org/10.1016/j.asoc.2017.11.042.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Stylidis, K., Wickman, C., Söderberg, R., 2020. Perceived quality of products: a framework and attributes ranking method. J. Eng. Des. 31 (1), 37–67. http://dx.doi.org/10.1080/09544828.2019.1669769.

Tukey, J.W., 1949. Comparing individual means in the analysis of variance. Biometrics 5 (2), 99–114. http://dx.doi.org/10.2307/3001913.

Vargas, V.M., Gutiérrez, P.A., Hervás, C., 2019. Deep ordinal classification based on the proportional odds model. In: Proceedings of the International Work-Conference on the Interplay Between Natural and Artificial Computation. Springer, pp. 441–451. http://dx.doi.org/10.1007/978-3-030-19651-6_43.

Vargas, V.M., Gutiérrez, P.A., Hervás-Martínez, C., 2020. Cumulative link models for deep ordinal classification. Neurocomputing 401, 48–58. http://dx.doi.org/10.1016/j.neucom.2020.03.034.

Vargas, V.M., Gutiérrez, P.A., Hervás-Martínez, C., 2022. Unimodal regularisation based on beta distribution for deep ordinal regression. Pattern Recognit. 122, 1–10. http://dx.doi.org/10.1016/j.patcog.2021.108310.

Villalba-Díez, J., Molina, M., Ordieres-Meré, J., Sun, S., Schmidt, D., Wellbrock, W., 2020. Geometric deep lean learning: Deep learning in industry 4.0 cyber–physical complex networks. Sensors 20 (3), 1–16. http://dx.doi.org/10.3390/s20030763.

Villalba-Diez, J., Schmidt, D., Gevers, R., Ordieres-Meré, J., Buchwitz, M., Wellbrock, W., 2019. Deep learning for industrial computer vision quality control in the printing industry 4.0. Sensors 19 (18), 3987. http://dx.doi.org/10.3390/s19183987.

Wagersten, O., Forslund, K., Wickman, C., Söderberg, R., 2011. A framework for non-nominal visualization and perceived quality evaluation. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Vol. 54792, pp. 739–748. http://dx.doi.org/10.1115/DETC2011-48270.

Yang, Y., Yang, R., Pan, L., Ma, J., Zhu, Y., Diao, T., Zhang, L., 2020. A lightweight deep learning algorithm for inspection of laser welding defects on safety vent of power battery. Comput. Ind. 123, 1–8. http://dx.doi.org/10.1016/j.compind.2020.103306.

Zhang, Q., Fu, F., Tian, R., 2020. A deep learning and image-based model for air quality estimation. Sci. Total Environ. 724, 138178. http://dx.doi.org/10.1016/j.scitotenv.2020.138178.

Zhang, C.-B., Jiang, P.-T., Hou, Q., Wei, Y., Han, Q., Li, Z., Cheng, M.-M., 2021. Delving deep into label smoothing. IEEE Trans. Image Process. 30, 5984–5996. http://dx.doi.org/10.1109/TIP.2021.3089942.