Multi-label Logo Classification Using Convolutional Neural Networks

Antonio-Javier Gallego(✉), Antonio Pertusa, and Marisa Bernabeu

University Institute for Computing Research (IUII), University of Alicante, San Vicente del Raspeig, Spain
jgallego@dlsi.ua.es

Abstract. The classification of logos is a particular case within computer vision since they have their own characteristics. Logos can contain only text, iconic images or a combination of both, and they usually include figurative symbols designed by experts that vary substantially besides they may share the same semantics. This work presents a method for multi-label classification and retrieval of logo images. For this, Convolutional Neural Networks (CNN) are trained to classify logos from the European Union TradeMark (EUTM) dataset according to their colors, shapes, sectors and figurative designs. An auto-encoder is also trained to learn representations of the input images. Once trained, the neural codes from the last convolutional layers in the CNN and the central layer of the auto-encoder can be used to perform similarity search through kNN, allowing us to obtain the most similar logos based on their color, shape, sector, figurative elements, overall features, or a weighted combination of them provided by the user. To the best of our knowledge, this is the first multi-label classification method for logos, and the only one that allows retrieving a ranking of images with these criteria provided by the user.

Keywords: Logo image retrieval · Multi-Label Classification · Convolutional Neural Networks

1 Introduction

The detection and recognition of logos is an important task at the industrial level due to the need to find unauthorized usage of company logos, and also for trademark registration, where it needs to be verified that there are no similar existing logos within the same sector. However, there is not much research regarding this topic [5,9]. The reason may be because some years ago most logo datasets were not publicly available [3], but nowadays both the European Union Intellectual Property Office (EUIPO) and the United States Patent and Trademark Office (UPSTO) share their images and metadata.

Most previous computer vision works on logos are focused on Trademark Image Retrieval (TIR), i.e. similarity search to obtain the most similar logos to a query image. This is a very relevant task due to the volume of trademark

A. Morales et al. (Eds.): IbPRIA 2019, LNCS 11867, pp. 485–497, 2019.
https://doi.org/10.1007/978-3-030-31332-6_42

registration applications and the size of the databases containing existing trademarks, as it is almost impossible for humans to make all the comparisons visually [8]. Traditional methods addressed TIR by extracting hand-crafted features and then matching them with the prototypes of a dataset by using kNN to obtain a ranking of the most similar ones. Features used to represent logos include color histograms [3], texture descriptors, shape [10], a combination of them [4,6], or local descriptors such as SIFT [1]. In most cases, the feature dimensionality is reduced with a clustering method such as Bag of Words [5]. Finally, the processed characteristics are usually matched with those from a ground-truth dataset by using distance metrics or more complex methods such as template matching [9].

Recent TIR works such as [1] make use of deep neural networks. A combination of CNN pre-trained with ImageNet is employed in [8], fine-tuning their weights with two logo similarity datasets. Once the models are trained, authors extracted neural codes (NC) from different layers and compared them with the logo prototypes using cosine distance. For training these networks, one of the datasets used (DBc) contained a classification of figurative elements based on UPSTO with classes such as human beings, plants or foodstuffs in a multi-class setup.

Excepting the work in [8], which to the best of our knowledge is the only one using figurative classes, most TIR methods use datasets with logos that are only labeled by brand, assuming that images from a same brand would be similar. The fact that one brand may have different logo versions over time with changes in background, color, texture or shape (e.g. Disney has changed its logo more than 30 times [5]) is used to build labeled logo datasets. Examples of these are the METU dataset [12] which contains 32 classes (brands) in the query set with 10 images each, or Logos in the Wild [13], in which images are labeled with 871 brands. However, it must be noted that sometimes there are strong changes in the designs from the same brand, making them very different in appearance, therefore relying on visual similarity of logos from the same brand is not always reliable.

In addition to the trademark they belong to, logos can also be classified using different criteria, such as color, shape, sector or semantics. In this case, samples usually have more than one simultaneous label (for example, they may contain several colors or figurative elements), making this a Multi-Label Classification (MLC) task. MLC is different than traditional multi-class classification, as in the latter class labels are treated as independent target variables (relying on their mutually exclusive assumption), which is clearly suboptimal for MLC as the dependencies among classes cannot be leveraged.

In addition to MLC, there exist another task related to supervised learning from multi-label data [11]: Label Ranking (LR). The main difference is that MLC is concerned with learning a model that outputs a bipartition of the set of labels into relevant and irrelevant with respect to a query instance, while LR is concerned with learning a model that outputs a ranking of class labels according to their relevance to a query instance.

To the best of our knowledge, multi-label classification of logos remains unexplored. In this work we perform MLC and LR on logos using a large image dataset labeled with figurative elements, colors, shapes and sectors. Individual classifiers are trained for each of these tasks, together with an auto-encoder for the reconstruction of the input images. Then, the trained networks can be used for classification, but also for similarity search using kNN on the neural codes (NC) extracted from the last convolutional layers of the CNN or from the middle layer of the auto-encoder. This allows us to perform logo retrieval by mixing different criteria according to the user.

2 Dataset

For training the models we use the European Union TradeMark (EUTM) dataset, which gives owners an exclusive right in the 28 Member States from European Union. Original data was downloaded from the European Union Intellectual Property Office (EUIPO) website[1], selecting a subset of 11,000 logos corresponding to the year 2018 for experiments.

The EUTM dataset uses the Vienna Classification, a hierarchical system which proceeds from the general to the specific, dividing all elements into 29 main categories as can be seen in Table 1, that are subdivided into more specific elements (subcategories). The characteristics used in this work are the following:

- **Figurative.** In the scope of this work, we call figurative designs to those codes between 1 a 24, as they are related to the particular objects that can be found in the image logo. The figurative subcategories were not used since they are too specific (for example, 10.1.2. Cut tobacco, 10.1.3 Cigars, 10.1.5 Cigarettes, or 10.1.7 Tobacco in any other form) and the number of classes could be very large. The codes from 25 to 29 are related to the background, shape, text and color and, therefore, we did not include them for figurative classification. Codes 26 (shape) and 29 (color) were used separately as can be seen below.
- **Colors.** The 13 color codes used in this work are shown in Table 2 (left). Vienna classification also includes codes related to the number of colors (e.g., 29.01.12 means that there are two predominant colors), although they were not used.
- **Shapes.** Table 2 (right) shows the different shapes used for classification, including circles, triangles, lines, etc.
- **Sectors.** Trademarks have also the associated goods and/or services to be covered by the mark. For this, EUIPO has adopted the Nice Classification[2] that divides goods and services into 45 subcategories[3], not shown here due to space limitations, from two main sectors: goods (from codes 1 to 34), and

[1] https://euipo.europa.eu/ohimportal/en/open-data.
[2] https://euipo.europa.eu/ohimportal/en/nice-classification.
[3] https://www.wipo.int/classifications/nice/nclpub/en/fr/20180101/classheadings/explanatory_notes=show&lang=en&menulang=en.

Table 1. Vienna labels [14]. In the scope of this work, figurative elements are those with codes from 1 to 24. Codes from 25 onwards are related to shape, text and color, and were not used for figurative classification.

Code	Description
1	Celestial Bodies, Natural Phenomena, Geographical Maps
2	Human Beings
3	Animals
4	Supernatural, Fabulous, Fantastic or Unidentifiable Beings
5	Plants
6	Landscapes
7	Constructions, Structures for Advertisements, Gates or Barriers
8	Foodstuffs
9	Textiles, Clothing, Sewing Accessories, Headwear, Footwear
10	Tobacco, Smokers' Requisites, Matches, Travel Goods, Fans, Toilet Articles
11	Household Utensils
12	Furniture, Sanitary Installations
13	Lighting, Wireless Valves, Heating, Cooking or Refrigerating Equipment, Washing Machines, Drying Equipment
14	Ironmongery, Tools, Ladders
15	Machinery, Motors, Engines
16	Telecommunications, Sound Recording or Reproduction, Computers, Photography, Cinematography, Optics
17	Horological Instruments, Jewelry, Weights and Measures
18	Transport, Equipment for Animals
19	Containers and Packing, Representations of Miscellaneous Products
20	Writing, Drawing or Painting Materials, Office Requisites, Stationery and Booksellers' Goods
21	Games, Toys, Sporting Articles, Roundabouts
22	Musical Instruments and their Accessories, Music Accessories, Bells, Pictures, Sculptures
23	Arms, Ammunition, Armour
24	Heraldry, Coins, Emblems, Symbols
25	Ornamental motifs, Surfaces or backgrounds with ornaments
26	Geometrical figures and Solids
27	Forms of writing, Numerals
28	Inscriptions in various characters
29	Colors

services (from 35 to 45). The goods sector includes chemicals, medicines, metals, materials, machines, tools, vehicles, instruments, etc., while services include advertising, insurance, telecommunications, transport and education, among others. In this work we use both sectors and sub-sectors.

Table 2. Vienna codes used in this work for colors (code 29 in Table 1), and shapes (code 26 in Table 1).

Code	Color
29.01.01	Red
29.01.02	Yellow
29.01.03	Green
29.01.04	Blue
29.01.05	Violet
29.01.06	White
29.01.07	Brown
29.01.08	Black
29.01.95	Silver
29.01.96	Gray
29.01.97	Gold
29.01.98	Orange
29.01.99	Pink

Code	Shape
26.1	Circles, ellipses
26.2	Segments or sectors of circles or ellipses
26.3	Triangles, lines forming an angle
26.4	Quadrilaterals
26.5	Other polygons
26.7	Different geometrical figures, juxtaposed, joined or intersecting
26.11	Lines, bands
26.13	Other geometrical figures, indefinable designs
26.15	Geometrical solids

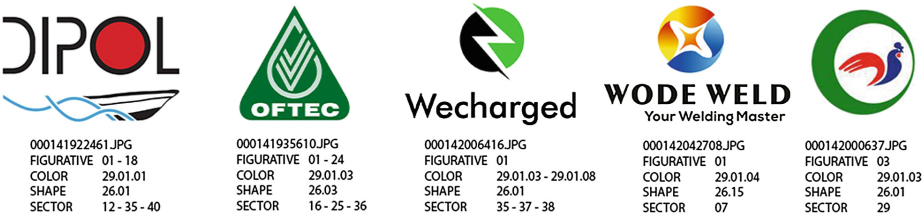


Fig. 1. Some trademark examples in the EUTM dataset.

2.1 Preprocessing

The EUTM metadata is stored in XML format. The XML files have fields with the categories and subcategories of the trademark, and the corresponding logo image path. We extracted these fields from each XML file and separated them in figurative (codes from 1 to 24), color (codes 29.xx.xx), shape (codes 26.xx), and sector (codes for goods and/or services). Only those trademarks that have all these fields were selected. Then, all the images related to the processed data were downloaded. Notice that a trademark can have two or more codes associated to each field, and some information is not always labeled, specially in the color and shape fields where not all elements are described, as can be seen in Fig. 1.

Regarding the color category in Fig. 1, we can see in the first example image that only red color was annotated, although it also has black and blue. In the third logo, green and black were labeled, whereas in the forth and fifth ones, which have three colors, only one of them was detailed. Meanwhile, only circles were annotated as shapes in the first, third and fifth images, although they also contain other shapes. The forth, which is also circular, was classified as geometrical solid.

The logo images were also preprocessed. First they were crop to eliminate the borders containing background. In this dataset, all the logos have a white background but the size of its border is variable, and in some cases too large. For this reason, images were crop by eliminating white borders so that all the logos

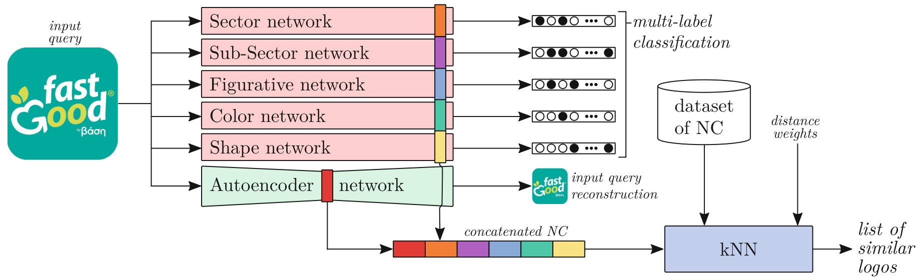


Fig. 2. Scheme of the proposed method.

occupied all the available space of the image. Once this process was completed, images were scaled to a size of 256×256 pixels, and their values were normalized into the range $[0, 1]$ to feed the networks.

3 Method

Figure 2 shows the scheme of the proposed approach. The CNNs are trained on each characteristic using an MLC setup with a sigmoid activation instead of a softmax in the last layer, as shown in Table 3. This activation function models the probability of each class as a Bernoulli distribution, where each class is independent of the others unlike it happens with softmax. Therefore, the output is a multi-label classification for each of the considered characteristics.

In addition, an auto-encoder was also trained to reconstruct the input image in order to get the NC from the middle layer. These codes are a low-dimensionality representation of the main characteristics of the image, which allows to perform similarity search with kNN. Once all networks are trained, NC are extracted from the CNNs and the auto-encoder on the input image. The size of the NC extracted from the CNNs is 128, and for the auto-encoder is 256. The NC are combined into a single vector of characteristics, which is the one used to perform similarity search. An ℓ_2-norm is applied to normalize the concatenated feature vector before searching for the most similar prototypes, since this technique usually improves the results [2].

We use a weighted distance to search for the nearest neighbors, allowing the user to adjust the search criteria modifying the weight assigned to each characteristic. We use the following equation to calculate the distance between two vectors of characteristics A and B:

$$d(A, B) = \frac{\sum\limits_{c \in \mathcal{C}} w^c d(A^c, B^c)}{\sum\limits_{c \in \mathcal{C}} w^c} \tag{1}$$

where $\mathcal{C}$ is the set of all the possible characteristics to be classified, A^c and B^c represent the subset of features corresponding to the characteristic c, w^c is the weight assigned to that characteristic, and $\forall c \in \mathcal{C} : w^c \in [0, 1]$.

Table 3. CNN and Auto-Encoder (Encoder+Decoder) architectures. Notation: Conv(f, $w \times h$) stands for a layer with f convolutional operators of size $w \times h$; ConvT(f, $w \times h$) stands for a layer with f transposed convolutional operators of size $w \times h$; MaxPool($w \times h$) stands for the Max-Pooling operator with a $w \times h$ kernel; Drop(d) refers to Dropout with ratio d; FC(n) is a Fully-Connected layer with n neurons. The NC were extracted from the layers marked in bold.

CNN configuration					
Conv(32,11×11) BatchNorm Act ReLU MaxPool(2×2) Drop(0.1)	Conv(32,9×9) BatchNorm Act ReLU MaxPool(2×2) Drop(0.2)	Conv(64,7×7) BatchNorm Act ReLU MaxPool(2×2) Drop(0.3)	Conv(64,5×5) BatchNorm Act ReLU MaxPool(2×2) Drop(0.4)	Conv(128,3×3 BatchNorm Act ReLU MaxPool(2×2) Drop(0.5)	Flatten() **FC(128, ReLU)** BatchNorm Drop(0.5) FC(n, Sigmoid)
Encoder configuration					
Conv(128,3×3) Stride(2×2) BatchNorm Act ReLU Drop(0.25)	Conv(64,3×3) Stride(2×2) BatchNorm Act ReLU Drop(0.25)	Conv(64,3×3) Stride(2×2) BatchNorm Act ReLU Drop(0.25)	Conv(64,3×3) Stride(2×2) BatchNorm Act ReLU Drop(0.25)	**Conv(1, 3×3)**	
Decoder configuration					
CnvT(64,3×3) Stride(2×2) BatchNorm Act ReLU Drop(0.25)	CnvT(64,3×3) Stride(2×2) BatchNorm Act ReLU Drop(0.25)	CnvT(64,3×3) Stride(2×2) BatchNorm Act ReLU Drop(0.25)	CnvT(128,3×3) Stride(2×2) BatchNorm Act ReLU Drop(0.25)	Conv(3,3×3) Act Sigmoid	

3.1 Training Process

The training of the networks is carried out by means of standard backpropagation using Stochastic Gradient Descent (SGD) and considering the adaptive learning rate method proposed in [15]. In the backpropagation algorithm, *binary crossentropy* was used as the loss function between the CNN output and the expected result. The training lasted a maximum of 200 epochs with *early stopping* when the loss did not decrease during 15 epochs. The mini-batch size was set to 32 samples.

In addition, data augmentation was performed on the training images by adding random rotations, horizontal and vertical flips, scale and shear transformations.

4 Evaluation

This section presents the metrics used for evaluation and the obtained results. For evaluation we have used the dataset described in Sect. 2. From the 11,000 logo images, 80% were selected for training and the remaining samples for test.

4.1 Metrics

In multi-label learning each sample can have any number of ground truth labels associated. In this work we use the following multi-label metrics [11]:

Table 4. Results obtained with the CNNs for the different tasks.

Network	CE	LRL
Color	2.4397	0.0399
Shape	2.4751	0.1326
Sector	1.6619	0.2345
Sub-sector	16.2427	0.2120
Figurative	2.0170	0.0616

Coverage Error (CE). This is an MLC metric which computes the average number of labels that have to be included in the final prediction such that all true labels are predicted. This is useful to know how many top-scored-labels must be predicted in average without missing any true one. The best value of this metrics is thus the average number of true labels. Formally, being N the total number of samples and L the number of labels, given a binary indicator matrix of the ground truth labels $y \in \{0,1\}^{N \times L}$ and the score associated with each label $\hat{f} \in \mathbb{R}^{N \times L}$, the coverage is defined as:

$$\mathrm{CE}(y, \hat{f}) = \frac{1}{N} \sum_{i=0}^{N-1} \max_{j:y_{ij}=1} \mathrm{rank}_{ij} \tag{2}$$

where $\mathrm{rank}_{ij} = \left| \left\{ k : \hat{f}_{ik} \geq \hat{f}_{ij} \right\} \right|$.

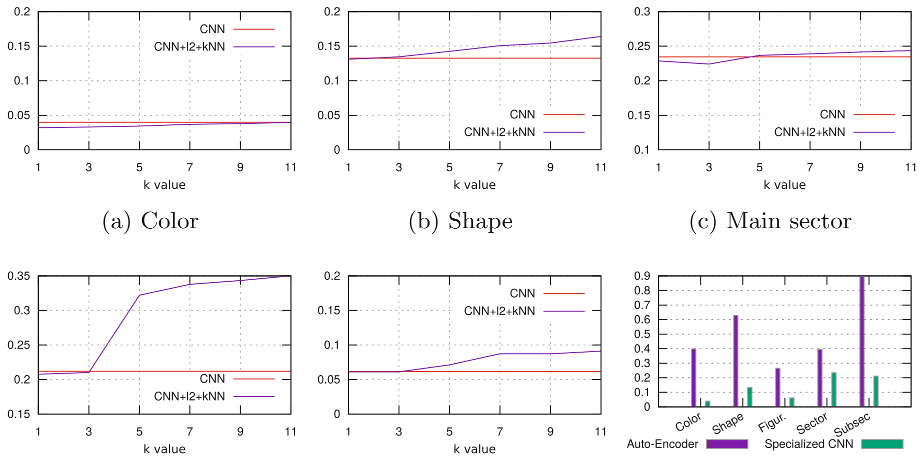
Label Ranking Loss (LRL). This LR metric computes the ranking which averages over the samples the number of label pairs that are incorrectly ordered (true labels with a lower score than false labels), weighted by the inverse of the number of ordered pairs of false and true labels. The lowest achievable label ranking loss is zero.

$$\mathrm{LRL}(y, \hat{f}) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{1}{\|y_i\|_0 (L - \|y_i\|_0)} \left| \left\{ (k,l) : \hat{f}_{ik} \leq \hat{f}_{il}, y_{ik} = 1, y_{il} = 0 \right\} \right| \tag{3}$$

where $|\cdot|$ calculates the cardinality (number of elements) of the set, and $\|\cdot\|_0$ is the ℓ_0-norm which computes the number of nonzero elements in a vector.

4.2 Results

The results of the different CNNs can be seen in Table 4. The best results using the LRL metric are obtained by the color classifier, followed by the main sector (note that it only has two labels), the figurative designs, and the shape, with fairly close results. The worst results are obtained for the sub-sector, possibly because it is the characteristic with the largest number of classes (45). In addition, it must be borne in mind that, while color and shape are objective characteristics, the sector, sub-sector and figurative design may be subjective, and in some cases the same design can be used for different sectors or sub-sectors.



(a) Color (b) Shape (c) Main sector

(d) Sub Sector (e) Figurative (f) Auto-Encoder

Fig. 3. LRL results for each characteristic using kNN on the NC (lower values are better). In the case of the auto-encoder, results are evaluated regarding all the other possible characteristics.

With respect to the results obtained by the auto-encoder for the validation set, we obtain a binary crossentropy loss of 0.3992 and a Mean Squared Error (MSE) of 0.0687 for reconstruction. In this case, it is not possible to calculate the other metrics since the auto-encoder is trained in an unsupervised way.

Regarding similarity search with kNN, Fig. 3 shows the results obtained (using the LRL metric) for each individual characteristic. In this experiment, the value of k was evaluated in the range $k \in [1 - 11]$ to assess its impact. As can be seen in Figs. 3a–e, the best results are obtained with low k values, slightly improving the CNN result when using 1 or 3 neighbors. Although the results are similar to those from the CNN, using kNN on the NC allows us to perform similarity search. Finally, Fig. 3f shows the results obtained using the NC extracted from the auto-encoder to classify the different characteristics considered (color, shape, etc.). In this way we can analyze what the characteristics learned by the auto-encoder represent. As can be seen, the auto-encoder focuses slightly on each of the characteristics, obtaining the best results when classifying the figurative design, followed by the main sector, color, and shape, and obtaining the worst results for the sub-sector.

Figure 4 shows an example query and the most similar results using the NC of the color classifier. As seen, colors of the retrieved results are correctly matched, even when they are multiple, independently of other characteristics such as the shape. Figure 5 shows that shapes are also correctly detected. Figurative designs (see Fig. 6) are more complicated to retrieve. The first example shows a plant, and most retrieved results are correct, although there are also unrelated logos retrieved. The second example contains an animal, but the first and last retrieved results do not.

Fig. 4. Color retrieval example with kNN. The first logo is the query. (Color figure online)

Fig. 5. Shape retrieval example with kNN. The first logo is the query.

Fig. 6. Figurative design retrieval example with kNN. The first logo is the query.

Fig. 7. Sector retrieval example with kNN. The first logo is the query.

Sector is very difficult to retrieve properly, moreover when there are only two classes that do not strongly depend on visual information: goods and services. The first row in Fig. 7 shows an example of a goods logo that was correctly

Fig. 8. Sub-sector retrieval example with kNN. The first logo is the query.

Fig. 9. Auto-encoder retrieval example with kNN. The first logo is the query.

retrieved, and the second is related to services. In the case of sub-sector (Fig. 8), the first query corresponds to a cosmetics brand. As can be seen, nearest logos are also related, and also the most visually similar are found. The second example corresponds to a brand from the construction sub-sector, and most retrieved elements are correct. The third row shows that even when logos have similar features but layout changes (BelXPo brand), they are retrieved properly.

As can be seen in Fig. 9, the auto-encoder mainly focuses on spatial distributions (the logo layout), in some cases taking also into account the colors.

Figure 10 shows the results when using the weighted distance. In this example, the characteristics of color and shape are used, comparing the result obtained by assigning 70% of the weight to the color and 30% to the shape, and vice versa. As can be seen, by increasing the weight of color, retrieved images have a similar color and slightly similar shapes (unlike the results shown in Fig. 4, in which the shape varies greatly). By inverting the weights and giving more importance to the shape, Fig. 10 shows images with more similar shapes but slightly different colors. Analyzing the results of the last row (brand "Acute angle") we can see that by giving more weight to the color only appears a completely triangular shape, since there are no more triangular logos labeled with the same color. If we compare these results with those obtained in the first row of Fig. 5 for the same mark, we can see how we also obtain only triangular shapes but with a different arrangement, in which when weighed the distance giving a certain weight to the color appears as a second result an inverted triangular shape, but with the same color.

Fig. 10. Results obtained using the weighted distance with color and shape. In the first column the applied weights are shown, and the first image is the query. (Color figure online)

5 Conclusions

In this work, several multi-label CNNs were trained for the classification of logo images according to their color, figurative designs, shape, and sector. The NC extracted from internal layers were used for similarity search using kNN with optional user-driven weights for each characteristic. In addition to the NC, an auto-encoder was also trained to extract a compact representation of the logo which contains its overall appearance, and that can also be used for similarity search. To the best of our knowledge, there are no previous works addressing multi-label logo classification, with these amount of characteristics that can be used to search similar images, and allowing weighting the characteristics.

Results show that the method is reliable for retrieving logos that are similar in color or shape. Worse results are obtained using sector, sub-sector and figurative elements, as these features are much more general. However, as can be seen in the example figures, most retrieved logos are visually similar.

The design of a logo is subjective and, in addition, it is not always made by a professional designer. Moreover, there are design criteria that depend on certain factors such as the sector, since logos of the same sector often share similar features (for instance, car brand logos usually contain metallic shapes and colors). An advantage of using deep neural networks is that these models learn the most frequent designs for each sector and this can serve, in addition to looking for similar logos, as an indication of whether the logo is appropriate or not for a particular activity.

As future works, it is planned to further explore the NC, for example from color and shape, using t-Distributed Stochastic Neighbor Embedding (t-SNE [7]). This would enable us to cluster samples of similar logos visually. Also, we plan to explore more specific categories for figurative elements such as "cheese" (instead of the main categories such as "food") to obtain similar logos more accurately, although this would require to train the classifiers with many more images. In addition, an evaluation of the logo retrieval results could also be performed by expert designers.

Acknowledgments. This work is supported by the Spanish Ministry HISPAMUS project with code TIN2017-86576-R, partially funded by the EU.

References

1. Chiam, J.H.: Brand logo classification. Technical report, Stanford University (2015)
2. Gallego, A.J., Pertusa, A., Calvo-Zaragoza, J.: Improving convolutional neural networks' accuracy in noisy environments using k-nearest neighbors. Appl. Sci. **8**(11), 2086 (2018)
3. Ghosh, S., Parekh, R.: Automated color logo recognition system based on shape and color features. Int. J. Comput. Appl. **118**(12), 13–20 (2015)
4. Guru, D.S., Vinay Kumar, N.: Interval valued feature selection for classification of logo images. In: Abraham, A., Muhuri, P.K., Muda, A.K., Gandhi, N. (eds.) ISDA 2017. AISC, vol. 736, pp. 154–165. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76348-4_16
5. Iandola, F.N., Shen, A., Gao, P., Keutzer, K.: DeepLogo: hitting logo recognition with the deep neural network hammer. CoRR abs/1510.02131 (2015)
6. Kumar, N.V., Pratheek, Kantha, V.V., Govindaraju, K., Guru, D.: Features fusion for classification of logos. In: Internetional Conference on Computational Modelling and Security (CMS), vol. 85, pp. 370–379 (2016)
7. van der Maaten, L., Hinton, G.: Visualizing high-dimensional data using t-SNE. J. Mach. Learn. Res. **9**(Nov), 2579–2605 (2008)
8. Perez, C.A., et al.: Trademark image retrieval using a combination of deep convolutional neural networks. In: International Joint Conference on Neural Networks (IJCNN), pp. 1–7, July 2018
9. Pornpanomchai, C., Boonsripornchai, P., Puttong, P., Rattananirundorn, C.: Logo recognition system. In: ICSEC 2015, pp. 1–6 (2015)
10. Qi, H., Li, K., Shen, Y., Qu, W.: An effective solution for trademark image retrieval by combining shape description and feature matching. Pattern Recogn. **43**(6), 2017–2027 (2010)
11. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Data Mining and Knowledge Discovery Handbook, pp. 667–685 (2010)
12. Tursun, O., Aker, C., Kalkan, S.: A large-scale dataset and benchmark for similar trademark retrieval. CoRR abs/1701.05766 (2017)
13. Tüzkö, A., Herrmann, C., Manger, D., Beyerer, J.: Open set logo detection and retrieval. In: International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP) (2018)
14. World Intellectual Property Organization: International Classification of the Figurative Elements of Marks: (Vienna Classification). WIPO Publication, World Intellectual Property Organization (2002)
15. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. CoRR abs/1212.5701 (2012)