A CBR System for Image-Based Webpage Classification: Case Representation with Convolutional Neural Networks

Daniel López-Sánchez, Juan M. Corchado, Angélica González Arrieta
University of Salamanca, BISITE Research Group,
Edificio I+D+i, 37008 Salamanca, Spain

Abstract

During the past decade, there was an exponential growth in the number of webpages available. Automatic webpage categorization systems can help to manage these immense amounts of content, making search tasks and recommendation easier. However, most webpages have a significant proportion of visual content that conventional, text-based web mining systems can not handle. In this paper, we present a novel hybrid CBR framework designed to perform image-based webpage categorization. Our system incorporates state-of-the-art deep learning techniques which help attain high accuracy rates. In addition, the system was designed with the goal of minimizing computational costs.

Introduction

The recent increase in web-page numbers can be explained, to some extent, by the fact that free content management systems have been popularized; systems such as Drupal, WordPress or Tumblr are accessible to the general public. This means that anyone can create a great variety of web content without possessing any knowledge of digital system management. The very nature of hosting providers promotes the proliferation of eminently multimedia content.

The democratization of the internet has brought about challenging problems. For example, it has become increasingly difficult for users to find precisely what they are looking for because they are searching among a massive and growing number of sites. For this reason, digital systems should be able to understand the content they manage, at least to a certain degree. Thus, they could assist users in a variety of ways. Unfortunately, conventional, text-based web mining systems cannot make use of the information provided in the form of visual content.

In this paper, we propose a novel hybrid case-based reasoning (CBR) framework and evaluate its performance on the task of image-based web page categorization. We propose using a deep convolutional neural network (DCNN) as a case-representation generator in the context of a case-based reasoning methodology. In the proposed framework, the DCNN is intended to produce a compact and abstract representation of the visual content present in the web-page being analysed.

The designed hybrid system benefits from the advantages of both deep-learning and CBR approaches: (1) The ability to learn over time, as new revised cases are added to the case-base. (2) The capacity to justify an emitted solution by supplying previous cases and their solutions. (3) The neural-based case representation model enables the system to deal with highly complex image classification tasks, significant intra-class variability and noisy samples. (4) Thanks to the use of the transfer-learning approach, the system is able to achieve high accuracies even if the initial case-base is small. Also, the computational cost of the training phase is drastically reduced.

The rest of this article is structured as follows: In the "Related Work" section we briefly review previous proposals from the literature and contextualize the proposed system. Our proposed framework is described in detail in the section "Proposed method". The section "Experimental results" presents the outcome of a number of experiments which were performed to evaluate the effectiveness of our framework and compare it to the alternative approaches. Finally, in section "Conclusions", we summarize the results obtained in the previous section and propose the future research lines.

Related Work

Although classical case-based reasoning systems have been widely applied to the task of image processing (Coulon and Steffens 1994) (Richter and Weber 2013), specially within the medical field (Nasiri, Zenkert, and Fathi 2015), their naive retrieval method (i.e., retrieval based on Euclidean distances over raw pixel intensities) is not the best option for complex vision tasks. For this reason, hand-crafted feature descriptors are often applied to create a meaningful case representation. Several hand-crafted image descriptors have proven to be effective for a variety of problems, particularly in the context of CBR and content-based image retrieval systems. Some of the most popular examples are the MPEG-7 image descriptor (Allampalli-Nagaraj and Bichindaritz 2009), the Local Binary Pattern Histogram (LBPH) descriptor (Vatamanu et al. 2015) and the bag-of-visual words descriptor (BoVW) (Welter et al. 2011). In addition, the use of Virtual Reality-based case representations of real world objects has also been explored (Watson and Oliveira 1998).

On the other hand, artificial vision has achieved a major breakthrough after the recent popularization of the deep learning paradigm. The success of the deep learning approach can be explained by the increasing complexity of the models and the availability of massive training datasets (Chen and Lin 2014). An example of this paradigm is the VGG-16 Deep Convolutional Neural Network (Simonyan and Zisserman 2014). This network was designed and trained by the Visual Geometry Group (hence the name of the network) at the University of Oxford. One of the main goals of its designers was to develop an efficient DCNN architecture. For this reason, each convolutional layer in the network uses a small 3×3 kernel size (thus reducing the number of parameters in the model). The network consists of sixteen successive layers (excluding maxpooling), arranged into five convolution-pooling blocks and a final block of three fully connected layers. The VGG-16 model was trained on the ImageNet database, which compiles approximately fourteen million images, distributed among 1000 categories. It achieved a top-5 error rate of 92.7% in the test set, and although VGG-16 did not achieve the first place in the Large Scale Visual Recognition Challenge 2014 [1], it remains competitive with more recent proposals in terms of effectiveness/efficiency balance.

The major limitations of the deep learning approach are its computational requirements both in training and test stages. The transfer learning methodology, where the key idea is to use the knowledge gained while solving one problem to a distinct but somehow related problem, has been successfully applied in the literature of deep learning (Sharif Razavian et al. 2014) to mitigate the problem of computational cost. Despite its success, the possible integration of the deep learning paradigm into the CBR methodology has not yet been studied in the available literature.

Proposed method

This section describes the proposed method in detail. We first introduce the framework from a global perspective. Then, the different algorithms applied at various stages are described and the way in which they are combined is motivated and explained.

As mentioned before, the goal of the proposed system is to classify web pages according to their multimedia content. Specifically, whenever a URL is presented to the framework, it must: (1) Obtain the HTML document pointed by that URL, download every picture present in the document and dispose of advertisement and navigation images; (2) Generate a case representation (i.e., a feature descriptor) such that images from the same class are grouped together; and (3) produce a prediction about the category of the webpage in its entirety (i.e., categorization), based on the individual predictions for the images. The overall architecture of the framework is shown in figure 1.

Content extraction and filtering

When a new, unclassified webpage is presented to the system, the first task it must accomplish consists of the extraction of all the relevant images present in the web page. Ad-

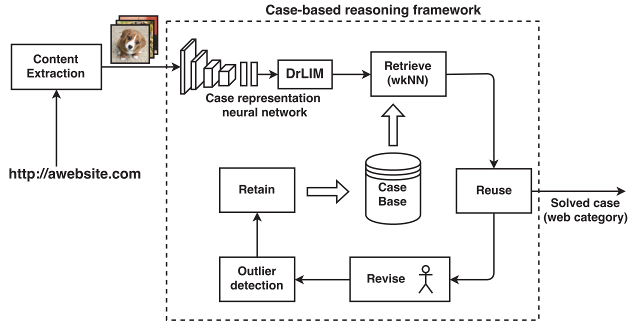

Figure 1: Overview of the proposed Hybrid CBR framework

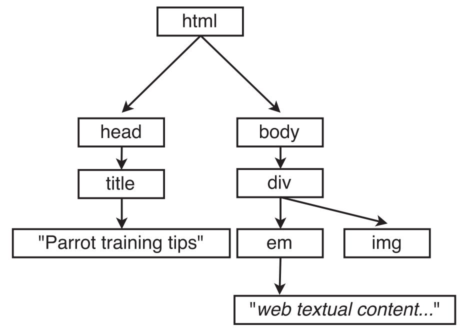

Figure 2: Sample web page hierarchy (DOM)

ditionally, it should dispose of uninformative images such as navigation icons, banners and advertisements. To that end, a web-scrapping approach was taken. The content extraction module uses the beautiful soup library (Richardson 2007) to download the HTML document associated to a given URL and build a hierarchical tree with its contents. In this representation, the leaves represent HTML elements like links, text, headers and images. Figure 2 shows the corresponding hierarchical tree of a very simple web page. Once this model of the document has been build, it is explored recursively to find all the image elements present in it. The *src*[2] fields of the extracted image elements of the document are used to download the images and store them for further processing. The next step in the processing pipeline is intended to discard uninformative images. Here, we decided to take a very simple approach; this is, we defined a boundary in the dimension of the images. This method was developed after the informal observation that images with very high or low proportion ratios tend to be uninformative (e.g. correspond to advertisement banners). Images with an aspect ratio outside $[0.5, 2]$ are not considered for further analysis.

Convolutional Neural Networks for Case representation

In this section, the proposed method for case representation is described in detail. The simplest approach one could take consist of using the raw pixel values of images as the case representation in the CBR system. During the retrieval

[1] webpage of the Large Scale Visual Recognition Challenge: http://image-net.org/challenges/LSVRC/

[2] The *src* attribute specifies the URL of the image element.

stage, the Euclidean distance between cases would be used to retrieve the most relevant cases from the case-base. Although simple, this naive method would not be very effective. The reason for this being that the Euclidean distance function over raw pixel intensities does not represent a meaningful measure of the similarity between the abstract elements present in two images. For example, this metric is extremely sensitive to illumination conditions, the arrangement of the different objects depicted in the images and other transformations that do not actually modify the contents of the image. For this reason, image-based CBR systems usually include a case representation stage that generates image descriptors. The image descriptors shall be chosen in such a way that Euclidean distances between them constitute a meaningful similarity measure for the problem being addressed (i.e., cases separated by a short Euclidean distance should admit similar solutions). However, building such a case representation module is not an easy task. There are two major approaches to solve this problem: (1) adopting a hand-crafted feature descriptor (which is the most common approach in the literature as explained before); and (2) training a feature extraction model.

The first option is very popular because it does not require a training stage. However, some feature descriptors might not be well suited for a given problem, and the choice of a particular descriptor must often be taken on a trial-and-error basis. Regarding the second option, there are also some complications. The success in the training process of a feature extraction model, especially those based on deep-learning methods, highly depends on the availability of large training datasets. Of course, such datasets are not always available, and collecting them is a very time-consuming task. Additionally, this training process would consume a significant amount of time (although modern parallelization techniques could help to palliate this problem). To avoid these problems, we propose applying the so called *transfer learning* approach. In this context, the transfer of knowledge consists of initializing some of the layers in a DCNN with the synaptic weights from another network, which was trained to perform a different task. The remaining layers (usually the deepest ones) are initialized following the standard procedure (i.e., according to some random distribution). These final layers must be trained to perform the new task, but this training phase is usually fairly efficient and only requires a few training samples.

In our proposal, the chosen model from which the synaptic weights were transferred is the above described VGG-16 Deep Convolutional Neural Network (Simonyan and Zisserman 2014). Once the network for knowledge transfer has been chosen, the next typical step would be to train a simple classifier on top of the transferred layers. Such a classifier could consist of a number of densely connected neuron layers. The number of neurons in the final layer would be equal to the number of classes in our categorization problem. However, our goal here is not to train a classifier, but to create a model for case representation. Our preliminary experiments pointed out that the features coming out of the last max-pooling layer of the VGG-16 network are not convenient for case representation; this is because samples from

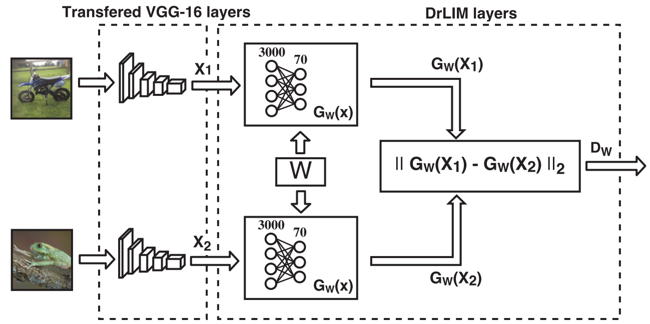

Figure 3: Siamese architecture for the training procedure of the DrLIM layers of the case-representation network.

different classes tend to overlap in that feature space. To refine that feature space and make sure that a distance-based retrieval method would work over our case representation, we decided to apply the DrLIM method (Hadsell, Chopra, and LeCun 2006).

The key idea of the DrLIM method is to train a model to minimize a contrastive loss function, which penalizes samples of different classes being placed near each other, and rewards samples from the same class being placed close together in the output-vector space. We removed the last three layers of the VGG-16 model and plugged in three new dense layers which we trained using DrLIM. To perform the training procedure with DrLIM, the network is duplicated to build a *siamese* architecture. Here, the replicated DrLIM layers $G_w(\cdot)$ share a set of weights W (see figure 3). Then, the training process occurs as follows: the whole network is fed with successive pairs of images (I_1, I_2). The images are forwarded through the transferred VGG-16 layers, yielding two $512 \times 7 \times 7$ activations X_1 and X_2 at the last max-pooling layer. These activations are flattened and processed by the siamese networks to produce $G_W(X_1)$ and $G_W(X_2)$, which are the case representations associated with each input image. A cost module is then in charge of computing the Euclidean distance between both representations. This value is used by the contrastive loss function, which is later computed taking into account whether I_1 and I_2 belong to the same class. For more details about the pair selection method and the contrastive loss see (Hadsell, Chopra, and LeCun 2006). When the training process is over, the set of weights W is fixed. The architecture of the final case representation network is described in table 1. Figure 4 shows a 2-dimensional MSD[3] embedding of the training images used in section 4 at the last transferred VGG-16 layer and after processing by the DrLIM-trained layers. As expected, the feature space at the output of the DrLIM layers seems more suited for distance-based classification.

[3]Multidimensional scaling (MDS) is a popular technique used to visualize the degree of similarity of individual cases of a dataset. For mode detail see (Borg and Groenen 2005).

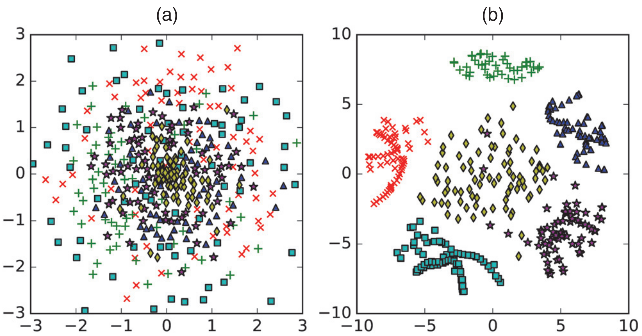

Figure 4: 2-dimensional MDS embedding of training samples from: (a) Maxpooling2D_5 and (b) Dense_2 (DrLIM). Food (×), motor (+), interior (△), animal (□), fashion (⋆), landscapes (◇).

Case retrieval

The previous section described the training process of the case representation network. After its training procedure has been completed, it is used to construct the case-base by processing a set of labeled images. The available n labeled images $\{I_1, I_2, \cdots, I_n\}$ are forwarded through the network, yielding n feature descriptors that along with the labels (i.e., stored solutions) form the case-base $B = \{(x_i, y_i), i = 1, 2, \cdots, n\}$. Once the case-base has been generated, the framework can be used to emit predictions about unseen web pages. The images present in the new web page are extracted by the content extractor module and individually processed by the case representation network. Then, the class associated to each descriptor is predicted and later used to emit a prediction concerning the whole webpage. Consider one individual image I from the webpage under analysis and its corresponding feature descriptor $x \in \mathbb{R}^{70}$. First, the k-nearest neighbors of x in the case-base B are found. After this, the un-normalized weight vector is calculated according to the following equation:

$$w = [\frac{1}{d(x, x_i)}, j = 1, 2, \cdots, k] \tag{1}$$

where $d(x, x_j)$ represents the desired distance metric (typically the squared Euclidean distance) between the case x and its j-th nearest neighbor. The probability of case x belonging to class c is estimated by using the standardized added weights as follows (Hechenbichler and Schliep 2004):

$$P(y = c \mid x, B) = \frac{\sum_{j=1}^{k} w_j \cdot I(y_j = c)}{\sum_{j=1}^{k} w_j} \tag{2}$$

where $I(\cdot)$ returns the value of one whenever it receives a true statement as its input. Otherwise, it returns the value of zero. A probability vector (containing the membership probability for each possible class) is computed for each image in the webpage being analyzed. The probability vector for the descriptor x of a given image I is defined as follows:

$$\vec{p} = [P(y = 0 \mid x, B), \cdots, P(y = c_n \mid x, B)] \tag{3}$$

Table 1: Case representation network. Dense layers use the *tanh* activation. Only the final two layers were trained, the first eighteen layers were transferred from the VGG-16 pretrained model.

Layer name	**Output shape**	**Param. #**
Convolution_1	(64, 224, 224)	1,792
Convolution_2	(64, 224, 224)	36,928
Maxpooling_1	(64, 112, 112)	0
Convolution_3	(128, 112, 112)	73,856
Convolution_4	(128, 112, 112)	147,584
Maxpooling_2	(128, 56, 56)	0
Convolution_5	(256, 56, 56)	295,168
Convolution_6	(256, 56, 56)	590,080
Convolution_7	(256, 56, 56)	590,080
Maxpooling_3	(256, 28, 28)	0
Convolution_8	(512, 28, 28)	1,180,160
Convolution_9	(512, 28, 28)	2,359,808
Convolution_10	(512, 28, 28)	2,359,808
Maxpooling_4	(512, 14, 14)	0
Convolution_11	(512, 14, 14)	2,359,808
Convolution_12	(512, 14, 14)	2,359,808
Convolution_13	(512, 14, 14)	2,359,808
Maxpooling_5	(512, 7, 7)	0
Dense_1 (DrLIM)	(3000)	75,267,000
Dense_2 (DrLIM)	(70)	210,070
Total	(70)	90,191,758

where each class has been assigned an integer label and c_n is the number of possible classes. Note that, conveniently:

$$\sum_{i=0}^{c_n} P(y = i \mid x, B) = 1 \tag{4}$$

Case reuse

After a probability vector has been obtained for each image present in the webpage being analysed, these probability vectors are combined to emit a prediction concerning the web in its entirety. Given the set of probability vectors for the m images extracted from a webpage $\{\vec{p}_0, \vec{p}_1, \cdots, \vec{p}_m\}$, its corresponding probability vector is computed as follows:

$$\vec{p}_{web} = \frac{1}{m} \sum_{i=0}^{m} \vec{p}_i \tag{5}$$

Note that the elements of $\vec{p}_{web}$ add up to exactly one (as a consequence of eq. 4). Therefore, the i-th element of $\vec{p}_{web}$ can be interpreted as the predicted probability of the webpage belonging to the class with label i. Accordingly, the most probable class label for the webpage can be computed as follows:

$$y_{web} = \arg\max(\vec{p}_{web}) \tag{6}$$

Case-base reduction

The proposed system has been specifically designed to work in scenarios where only a small number of training data samples is available. However, one of the major advantages of

CBR systems lies in their ability to learn over time, as new cases are available. Given the nature of the retrieve procedure, the query time of the system will increase lineally as the number of cases stored in the case-base increase. Specifically, finding the k nearest neighbors to a given case takes $\mathcal{O}(ndk)$ time, where n is the number of stored cases and d is the dimension of the case representation. This implies that, once a number of stored cases is reached, the query time might become inadmissible for a given application. To overcome this limitation, it is possible to periodically execute some case-base reduction method. The different methods mainly differ in the way they select the cases that will be deleted from the database. We decided to apply a recent technique, named the NearMiss-3 algorithm (Lemaître, Nogueira, and Aridas 2016). Experimental results concerning the performance loss produced by the application of the NearMiss-3 method are reported in the experimental results section.

Revision and retaining

In the context of the proposed webpage categorization framework, the revision stage should be performed by the final human user, who might indicate whether a given webpage has been misclassified or not. However, knowing the actual category of a web page is not enough, as the system described in the previous sections makes use of the individual image labels. Even if the user provides the true class for a given webpage, it does not mean that every single image in the web belongs to that category. Asking the user to label all the images individually is impractical and would drastically reduce the applicability of the designed system in real world applications. For this reason, it is necessary to implement a method to filter mislabeled images before they are stored into the case-base. Thanks to the DrLIM algorithm used to generate a proper case representation, misclassified samples will typically appear as isolated points surrounded by prototypes of different classes. The proposed system includes an outlier detection procedure that takes advantage of this property. The outlier detection method is applied before the retaining of a set of revised cases and solutions $R = \{(r_i, s_i), i = 1, 2, \cdots, r\}$ into the existing case-base $B = \{(x_i, y_i), i = 1, 2, \cdots, n\}$. Note that a revised case r_i consist of the descriptor extracted from a single image by the case representation network; the revised solution s_i is the class label assigned by the user to the webpage from which the image was originally extracted. The proposed retaining procedure (including outlier detection) is described by algorithm 1. Experimental results concerning the ability of the proposed system to learn over time are reported in the next section.

Algorithm 1 Retain procedure with outlier detection

Require: A case-base $B = \{(x_i, y_i), i = 1, 2, \cdots, n\}$
Ensure: A set of revised cases $R = \{(r_i, s_i), i = 1, \cdots, r\}$
  for $i := 0$ **to** r **step** 1 **do**
    trainSet = $R \setminus \{(r_i, s_i)\} \cup B$
    if k-NN(trainSet).predictClass(r_i) $\neq s_i$ **then**
      $R = R \setminus \{(r_i, s_i)\}$
    end if
  end for
  $B = B \cup R$

Experimental results

In this section, we first describe the dataset we collected to perform the experiments. After that, experimental results on the accuracy of the proposed system and its ability to learn from revised cases are presented. In addition, the performance of CBR systems using various image descriptors, namely MPEG-7 (Allampalli-Nagaraj and Bichindaritz 2009), LBPH (Vatamanu et al. 2015) and BoVW (Welter et al. 2011), is compared to the proposed framework. A simple voting procedure was used to combine the individual image predictions and emit a webpage-level prediction in the case of these descriptor-based CBR systems [4].

To the extent of our knowledge, no standard dataset for webpage categorization compiles the visual content of a number of webpages. Consequently, we decided to create a proper dataset composed of webpages from different categories and the visual content present on them at the moment of recollecting the dataset. A total of 280 webpages were collected from different websites, with an almost uniform distribution among six categories: ”food”, ”motor and vehicles”, ”interior design”, ”animals”, ”fashion” and ”nature/landscapes”. Using the content extraction module $3,223$ images were extracted from those webs. A train/test split was performed at random with 35 webpages for training and 210 for testing; the remaining 35 webs were retained to evaluate the learning capabilities of the designed CBR system as more revised cases are incorporated to the case-base. For the experiments in this paper, the DrLIM layers of the case representation network were trained using batch gradient descent. The learning rate was set to 0.0004 and 1000 iterations were executed. Table 2 compiles both the image-level and the webpage-level classification accuracies for various combinations of techniques described in this paper. It also reports the increase in the accuracies after a number of revised cases becomes available. Figure 5 shows the confusion matrices of the proposed framework (without the application of NearMiss3) before and after retaining additional cases.

Conclusions

Our experimental results prove that the proposed method outperforms other alternative case-representation techniques from the literature in the task of image-based webpage classification. This is because classical feature descriptors are not well suited for complex vision problems where only a small training dataset is available. On the other hand, the proposed framework overcomes this limitation by incorporating the knowledge gained by another model (i.e., transfer from the VGG-16 model). The experiments also reflect the ability of the proposed method to learn as new revised cases

[4] These systems were also evaluated with a weighted voting approach (Hechenbichler and Schliep 2004), but no significant accuracy increase was registered with respect to simple voting.

Case representation	Classifier	CB reduction	Acc.	Web acc.	Acc. *	Web acc. *
VGG16 & DrLIM	CBR (our proposal)	-	86.15%	96.19 %	87.08%	97.61 %
VGG16 & DrLIM	CBR (our proposal)	NearMiss3 (75%)	85.44%	95.71%	87.04%	97.14%
MPEG-7 (CS,EH)	CBR (kNN, k=5)	-	54.98%	80.00%	59.06%	81.42%
MPEG-7 (CS,EH)	CBR (kNN, k=5)	NearMiss3 (34%)	53.00%	76.66%	57.50%	77.61%
LBPH (P=8, R=1)	CBR (kNN, k=5)	-	44.34%	60.47%	46.36%	67.14%
LBPH (P=8, R=1)	CBR (kNN, k=5)	NearMiss3 (37%)	39.41%	52.38%	44.21%	63.33%
BoVW (SIFT & kMeans)	CBR (kNN, k=5)	-	40.73%	57.76%	42.93%	61.65%
BoVW (SIFT & kMeans)	CBR (kNN, k=5)	NearMiss3 (39%)	35.96%	49.02%	41.55%	56.79%

Table 2: Image-level and webpage-level classification accuracy for various combinations of techniques described in this paper. The symbol (*) indicates that this accuracy was attained after a number of additional revised cases were retained.

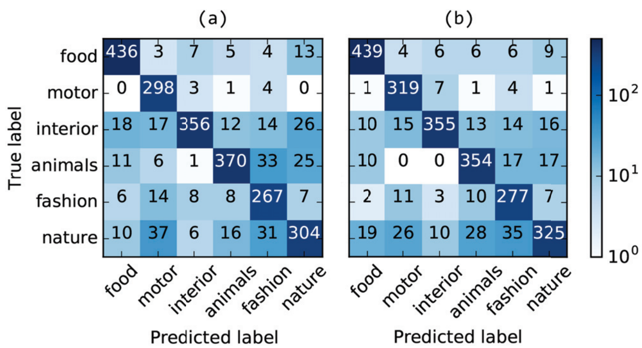

Figure 5: Confusion matrices for the proposed framework: (a) with the initial case-base, (b) after a number of revised cases were retained.

become available, and the possibility of reducing the size of the case-base with a low accuracy loss (using NearMiss3). As a future research line, we propose exploring the combination of the proposed system with classical, text-based web classification approaches.

Acknowledgements: The research of Daniel López-Sánchez has been financed by the FPU programme, reference number FPU15/02339.

References

Allampalli-Nagaraj, G., and Bichindaritz, I. 2009. Automatic semantic indexing of medical images using a web ontology language for case-based image retrieval. *Engineering Applications of Artificial Intelligence* 22(1):18–25.

Borg, I., and Groenen, P. J. 2005. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media.

Chen, X.-W., and Lin, X. 2014. Big data deep learning: challenges and perspectives. *IEEE Access* 2:514–525.

Coulon, C.-H., and Steffens, R. 1994. Comparing fragments by their images. *Similarity concepts and retrieval methods* 13:36–44.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, 1735–1742. IEEE.

Hechenbichler, K., and Schliep, K. 2004. Weighted k-nearest-neighbor techniques and ordinal classification.

Lemaître, G.; Nogueira, F.; and Aridas, C. K. 2016. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *CoRR* abs/1609.06570.

Nasiri, S.; Zenkert, J.; and Fathi, M. 2015. A medical case-based reasoning approach using image classification and text information for recommendation. In *International Work-Conference on Artificial Neural Networks*, 43–55. Springer.

Richardson, L. 2007. Beautiful soup documentation.

Richter, M. M., and Weber, R. 2013. *Case-based reasoning: a textbook*. Springer Science & Business Media.

Sharif Razavian, A.; Azizpour, H.; Sullivan, J.; and Carlsson, S. 2014. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 806–813.

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

Vatamanu, O. A.; FRANDEùa, M.; Lungeanu, D.; and MIHALAùa, G.-I. 2015. Content based image retrieval using local binary pattern operator and data mining techniques. *Digital Healthcare Empowering Europeans: Proceedings of MIE2015* 210:75.

Watson, I., and Oliveira, L. 1998. Virtual reality as an environment for cbr. In *European Workshop on Advances in Case-Based Reasoning*, 448–459. Springer.

Welter, P.; Deserno, T. M.; Fischer, B.; Günther, R. W.; and Spreckelsen, C. 2011. Towards case-based medical learning in radiological decision making using content-based image retrieval. *BMC medical informatics and decision making* 11(1):1.