Real-time single-shot brand logo recognition

Leonardo Bombonato, Guillermo Camara-Chavez and Pedro Silva
Computer Science Department
Federal University of Ouro Preto (UFOP)
Ouro Preto, Minas Gerais, Brazil,
Email: leonardobombonato@gmail.com

***Abstract*—The amount of data produced every day on the internet increases every day and with the increasing popularity of the social networks the number of published photos are huge, and those pictures contain several implicit or explicit brand logos. Detecting this logos in natural images can provide information about how widespread is a brand, discover unwanted copyright distribution, analyze marketing campaigns, etc. In this paper, we propose a real-time brand logo recognition system that outperforms all other state-of-the-art in two different datasets. Our approach is based on the Single Shot MultiBox Detector (SSD), we explore this tool in a different domain and also experiment the impact of training with pretrained weights and the impact of warp transformations in the input images. We conducted our experiments in two datasets, the FlickrLogos-32 (FL32) and the Logos-32Plus (L32plus), which is an extension of the training set of the FL32. On the FL32, we outperform the state-of-the-art by 2.5% the F-score and by 7.4% the recall. For the L32plus, we surpass the state-of-the-art by 1.2% the F-score and by 3.8% the recall.**

I. Introduction

Brand logos are graphic entities that represent organizations, goods, *etc.*; they are mainly designed for decorative and identification purposes. A specific logo can has several different representations, and some logos can be very similar in some aspects. Logo classification in natural scenes is a challenging problem since it often appears in various angles and sizes, making harder the keypoint extraction process, especially due to significant variations in texture, poor illumination, and high intra-class variations (see Figure 1). The automatic classification of logos gives to the marketing industry a powerful tool to evaluate the impact of brands. Marketing campaigns and medias can benefit with this tool, detecting unauthorized distributions of copyright materials.

Several techniques and approaches were proposed in the last decades for object classification, such as Bag of Visual Words(BoVW), Deep Convolutional Neural Networks (DCNN), feature matching with RANSAC, *etc*. The most successful approaches in logo classification were based on the BoVW model. Most recently, DCNN approach was proposed, being the Region-Based Convolution Neural Network (RCNN) a successful extension [1], this network introduces selective search to find candidates. This method has shown great results comparing to others proposed in previous research works. Despite the good results achieved by RCNN, it is hard to train and test due to the characteristics of selective search that generates several potential bounding boxes categorized by a classifier. After classification, a post-processing step refines the bounding box generation, eliminating duplicate detection, and re-scoring the boxes.

In this paper, we propose an approach for logo detection based on a deep learning model. Our results outperforms state-of-the-art approach results, achieving a higher accuracy on datasets FlickrLogos-32 and Logos-32Plus. Our proposals use transfer learning to improve the logo image representations, being not only more accurate but also faster in logo detection, processing 19 images per second using a Nvidia Titan X card.

II. Related Works

Different approaches for logo recognition have been proposed through the last years. Before a few years ago only shallow classifiers were offered to solve this issue, but with the increasing popularity of deep learning frameworks and of course because of its success in image recognition and detection, many research works explored this approach. The problem of detecting and classifying brand logos extends in two directions: specific logos, like in paper documents or vehicles logos, and generic brand logos. This article focuses on detection generic brand logos.

The first successful approach in logo recognition was based on contours and shapes with a clear background. Francesconi *et al.* [2] proposed an adaptive model using a recursive neural network, the authors used the area and the perimeter of the logo as features.

After 2007, with the popularization of SIFT [3], [4], many applications started to use it due to its robustness to rotation and scale transformations, and partial occlusions of interest objects. Many approaches for logo recognition based their proposals on SIFT descriptors [5]–[12].

RANSAC also became a popular learning module for object recognition since its use in Lowe *et al.* research work [3], [4]. They used RANSAC for descriptor comparison and find inline keypoints, thus locating the object. In logo recognition, some researchers explore this method and achieved significant results, e.g. [11], [12].

The *Flickrlogos-32* dataset became popular, and several types of research in classification, detection, and image retrieval evaluated their performance using this dataset. This dataset has the advantage of being balanced, the same number of images for each class, and it is a challenging dataset with a high variation of scale, intra-class variance, occlusion, rotation, illumination, etc. Many approaches evaluated their proposals on *Flickrlogos-32* [7], [9], [11], [13]–[16]

Fig. 1. This figure exemplifies the challenges of classifying logos in natural scenes, such as high intra-class variation, warping, occlusion, rotation, translation and scales.

Later works in logo detection started to use DCNN due to its great results in the object detection field. [13]–[15], [17]. The first research to introduce DCNNs in this field was [14] which explored the benefits of synthetically generated data for the task of brand logo detection. They used the R-CNN approach, extracting 2000 bounding boxes for each image, feeding it to a DCNN which return a fixed-length of features that are then classified by a set of linear SVMs. Their experiments show that when a little training data is available, synthetic data can improve the results of a deep learning approach.

Iandola *et al.* [13] evaluated three different problems: Logo Classification, Logo Detection without Localization and Logo Detection with Localization. For the logo classification task, they used the GoogLeNet with some variations, such as a Global Pooling layer before the fully connected, a softmax output layer after each inception module and a Full-inception approach where the first layer is also an inception layer. For the detection task, they performed 2 variations with the Fast R-CNN, combining it with the AlexNet and with the VGG-16, the VGG-16 achieved better results.

Oliveira *et al* [15] explore this field of research using the Fast R-CNN [1], they experiment two different DCNN architectures: Caffenet and VGG-M-1024. Also, they vary the learning rate, the selective search for generating the bounding boxes and jittering with shear and color.

Bianco *et al.* [17] proposed a system composed by a Selective Search combined with a tiny Network Architecture. Bianco *et al.* also proposed a new dataset called Logos-32Plus which is an extension of the training set of the FlickrLogos-32, to overcome the problem of low training instances for deep learning approaches.

III. Deep Convolutional Neural Networks

An Artificial Neural Network is a classification machine learning model where the layers are composed of interconnected neurons with learned weights. These weights are learned by a training process. Convolutional Neural Network (CNN) is a type of feed-forward artificial neural network and a variation of a multilayer perceptron. A neural network with three or more hidden layers is called deep network.

1) Transfer Learning: In a CNN, each layer learns to "understand" specific features from the image. The first layers usually learn generic features like edges and gradients, the more we keep forwarding in the layers, the more specific the features the layer detects. In order to "understand" these features, it is necessary to train the network, adjusting the net weights according to a predefined loss function. If the network weights initiate with random values, it requires much more images and training iterations compared to using pretrained weights. The use of net weights trained with other dataset is called "fine-tuning", and it demonstrates to be extremely advantageous compared to training a network from scratch [18]. This technique is useful when the number of training images per class is scarce (e.g. 40 images for this problem), which makes it hard for the CNN to learn. Furthermore, transfer learning also speeds up the training convergence [19].

2) Data Augmentation: Training a DCNN requires lots of data, especially very large/deep networks. When the dataset does not provide enough training images, we can add more images using data augmentation process. This process consists of creating new synthetic images that simulate different view angles, distortions, occlusions, lighting changes, etc. This technique usually increases the robustness of the network resulting in better results.

A. Single Shot MultiBox Detector

Single Shot MultiBox Detector (SSD) [20], [21] is an approach based on a feed-forward CNN, this network produces a collection of fixed size of bounding boxes and scores for the presence of object class instances. Finally, a non-maximum suppression step produces the final detections.

The SDD makes predictions based on feature maps taken at different stages; then it divides each one into a pre-established set of bounding boxes with different aspect ratios and scales. The bounding boxes adjust itself to better match the target object. The network generates scores using a regression technique to estimate the presence of each object category in each bounding box. The SSD increases its robustness to scale variations by concatenating feature maps from different resolutions into a final feature map. This network generates

scores for each object category in each bounding box and produces adjustments to the bounding box that better match the object shape. The non-maximum suppression process is used to reduce overlapping detection. Figure 2 shows how the feature maps are divided and the shapes of the default boxes.

Figure 3 shows the topology of the SSD framework, more specifically the SSD 300. The network receives an input image, then a base network extracts the features and at last the extra layers score predefined detection.

1) SSD variants: The SSD approach uses a base network to extract features from images and use them in detection layers. The extra layers in the SSD are responsible for detecting the object. There are some differences between SSD 300/500 and SSD 512. The SSD 512 is an upgrade of SDD 500, the improvements are presented as follows:

1) The pooling layer (*pool6*) between fully connected layers (*fc6* and *fc7*) was removed;
2) The authors added convolutional layers as extra layers;
3) A new color distortion data augmentation, used for improving the quality of the image, is also added;
4) The network populates the dataset by getting smaller training examples from expanded images;
5) Better proposed bounding boxes by extrapolating the image's boundary.

IV. Our approaches and contributions

Logos detection can be considered a subproblem of object detection since they usually are objects with a planar surface. Our approaches are based on the SSD framework since it performs very well in object detection and is also fast. We explored the performance of SDD model on logo images domain. We deeply analyze the impact of using pretrained weights with the technique called transfer learning and also the impact of balancing the dataset. We compare different implementations of the SSD and we also explore the impact of warping image transformations to meet the shape requirements of the SSD input layer.

A. Transfer learning methodology

To use the transfer learning technique was necessary to redesign the DCNN. This re-design remaps the last layer, adapting the class labels between two different datasets. Therefore, all convolution and pooling layers are kept the same, and the last fully-connected layers (responsible for classification) are reorganized for the new dataset. For logos detection, the fine-tuning was made over a pretrained network, trained for 160.000 iterations on PASCAL VOC2007+VOC2012+COCO datasets [21].

B. Our approaches

We explored 5 different approaches on the FlickrLogos-32(FL32) and 2 on the Logos-32Plus, Table I shows all different setups. The networks were trained for 100.000 (FL32) and 200.000 (L32plus) iterations using the Nesterov Optimizer [22] with a fixed learning rate of 0.001. The SSD 300 and SSD 500 were only explored using pretrained weights because they were easily surpassed by the SSD 512. The approach SSD 500 AR was an attempt to reduce the warp transformation of the input image since in the training and testing phase, the SSD needs to fit the input image into a square resolution. Since the dataset L32plus is imbalanced and knowing that CNN is very sensible to class-imbalanced instances [23], we explore the benefits of balancing the dataset. The dataset was balanced replicating the images randomly until all classes have the same number of logos.

TABLE I
Our proposal approaches

Acronym	Training Details	Extra	Dataset
SSD 300	Pretrained		FL32
SSD 500	Pretrained		FL32
SSD 500 AR	Pretrained	Preserving aspect ratio	FL32
SSD 512 FS	From Scratch		FL32
SSD 512 PT	Pretrained		FL32
SSD 512 PT U	Pretrained	Imbalanced dataset	L32plus
SSD 512 PT B	Pretrained	Balanced dataset	L32plus

V. Experiments

We evaluate and analyze our approaches on FlickrLogos-32 [7] and Logos-32Plus [17] datasets. Our experiments ran on the Caffe deep learning framework [24] and using a 2× Nvidia Tesla K80. First, we describe the dataset then compare the performance of our approaches. Finally, we compare our results to state-of-the-art methods in logo recognition.

A. DataSets

The first logo dataset was the BelgaLogos proposed in 2009 [25]. This dataset have the focus on image retrieval and is extremely imbalanced, some classes have only one training instance and one testing instance, not suitable for training a network. In 2011, two datasets were proposed the FlickLogos-32 [7] and the FlickrLogos-27 [26]. FlickrLogos-27 contains 27 classes with images extracted from Yahoo Flickr and has only 40 images per class, being 30 images for training and 10 for testing and a "distractor set" that contains 4207 logo images/classes, that depict, in most cases, clean logos. FlickrLogos-32 (FL32), on the other hand, has three times more testing instances, more training instances and also 6000 no-logos images. Six years after the FL32 was created, Logos-32Plus (L32plus), an extension of the training set is proposed in [17]. On average, the L32Plus has 10 times more training instances than FL32.

1) FlickrLogos-32: A challenging dataset, where the most promising approaches in logo recognition experimented their proposals on it. This dataset was proposed by Romberg [7], many approaches evaluated their performances on this dataset [8], [9], [11], [13]–[15], [27]. Romberg also defined an experimental protocol, splitting the dataset into training, validation and testing sets. In all approaches, we strictly follow this protocol. The FlickrLogos-32 have that name due to the fact that the images were collected from the Yahoo Flickr [1] and

[1] https://www.flickr.com/

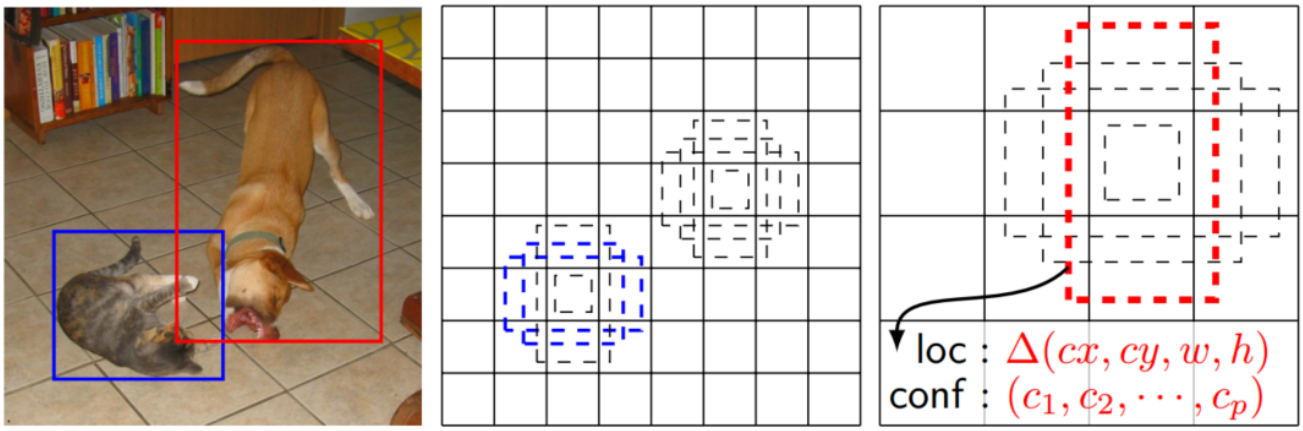

Fig. 2. (a) The final detection produced by the SSD. (b) A feature map with 8×8 grid. (c) A feature map with 4×4 grid and the output of each box, the location and scores for each class. Image extracted from [20].

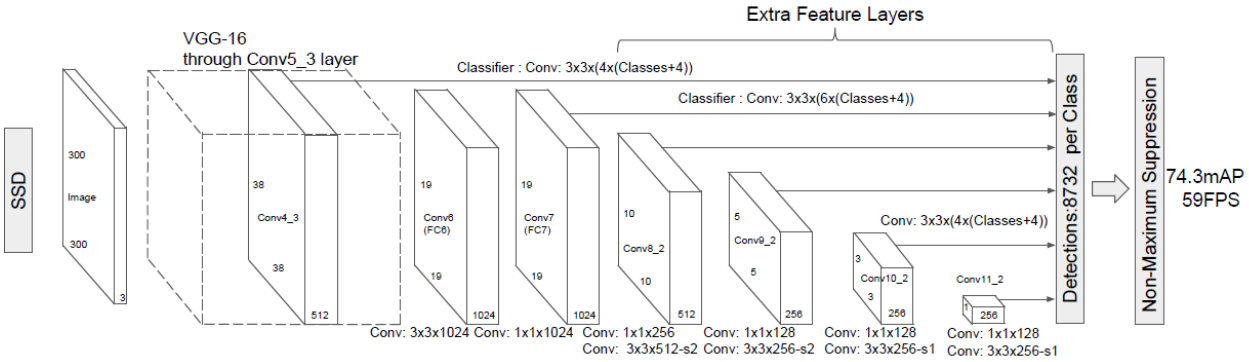

Fig. 3. Topology of the SSD 300. Base network plus extra feature layers plus non-maximum suppression.

TABLE II
EVALUATION PROTOCOL TABLE. EXTRACTED FROM [7].

Subset	Description	Images	Sum
$P1$	Hand-picked images, single logo, clean background	10 per class	320
$P2$	Images showing at least a single logo under various views	30 per class	3960
	Non-logo images	3000	
$P3$	Images showing at least a single logo under various views	30 per class	3960
	Non-logo images	3000	
Total			8240

TABLE III
COMPARISON BETWEEN FL32 AND L32PLUS. EXTRACTED FROM [17].

	FlickrLogos-32	Logos-32plus
Total images	8240	7830
Images containing logo instances	2240	7830
Train + Validation annotations	1803	12302
Average annotations for class	40	400
Total annotations	3405	12302

also has 32 different brand logos: Adidas, Aldi, Apple, Becks, BMW, Carlsberg, Chimay, Coca-Cola, Corona, DHL, Esso Erdinger, Fedex, Ferrari, Ford, Fosters, Google, Guinness, Heineken, HP, Milka, Nvidia, Paulaner, Pepsi, Ritter Sport, Shell, Singha, Starbucks, Stella Artois, Texaco, Tsingtao and UPS. Table II shows the distribution between, train, validation and test sets. We have used $P1 + P2$ (except no-logos) for training and $P3$ for testing.

2) Logos-32Plus: An extension of the training set of the FL32, proposed by [17]. This dataset was created to overcome the low number of training instances of the FL32 since CNN works better with more training images [23]. Table III shows the differences between the two datasets.

B. Results for FlickrLogos-32

We explored 5 different approaches on the FL32, we calculated the F-score for each approach varying the threshold from 0 to 1 with a step of 0.01. The chart with the F-scores can be seen in the Figure 4. As we can see, the SSD 512 PT outperforms all other approaches and its peak is at the threshold 90 with 93.5% F-score.

The Figure 5 shows the metrics for our best approach, the SSD pretrained. Analyzing the figure we can see that the approaches SSD 300, SSD 500 and SSD 500 AR achieved poor results if compared to the SSD 512. We see that in all cases using pretrained weights resulted in better performance.

TABLE IV
COMPARISON OF OUR BEST APPROACHES AGAINST OTHERS STATE-OF-THE-ART ON FL32

Method	Method	Year	Dataset	Precision	Recall	F1
Romberg et. al [7]	HCF	2011	FL32	0.981	0.610	0.752
Revaud et. al [8]	HCF	2012	FL32	0.980	0.726	0.841
Romberg et. al [9]	HCF	2013	FL32	0.999	0.832	0.908
Li et. al [27]	HCF	2014	FL32	**1.000**	0.800	0.890
Bianco et. al [28]	DL	2015	FL32	0.909	0.845	0.876
Eggert et. al [14]	DL	2015	FL32	0.996	0.786	0.879
Oliveira et. al [15]	DL	2016	FL32	-	-	0.890
Bianco et. al [17]	DL	2017	FL32	0.976	0.676	0.799
SSD 512 PT	DL	2017	FL32	0.954	**0.919**	**0.933**

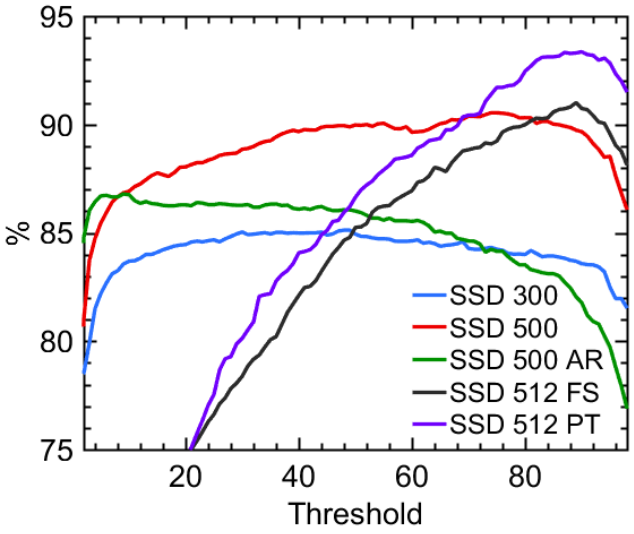

Fig. 4. Comparison of the F-score of our approaches on the FL32, varying from 0.01 to 0.99.

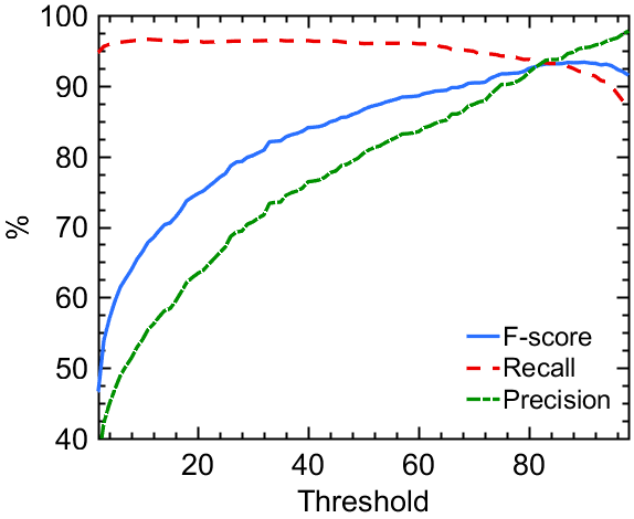

Fig. 5. The figure shows the F-score, Precision and Recall of the best approach.

Analyzing only the best result, SSD 512 PT, we see that we achieve our best F-score with a threshold of 0.9.

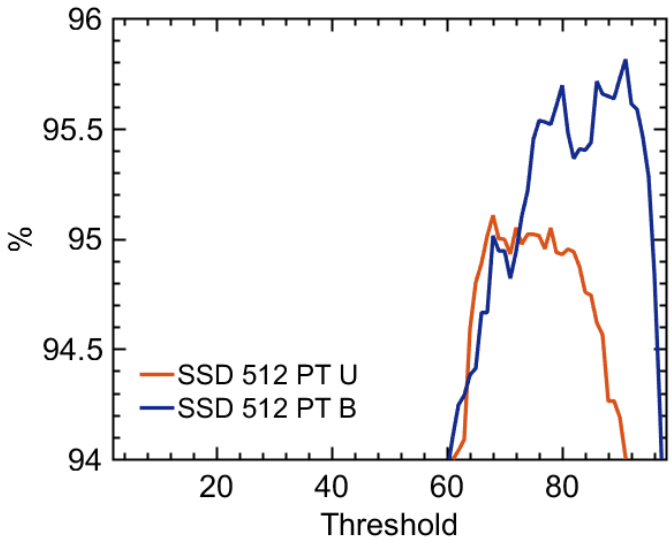

Fig. 6. Comparison of the F-score of our approaches on the L32Plus, varying from 0.01 to 0.99.

The comparison among other researches and our best result (the SSD 512 with pretrained weights) can be seen in the Table IV. Analyzing the results for FL32, we can see that our method outperforms by 2.5% the F-score and by 7.4% the recall of the state-of-the-art. The high recall achieved is due to the fact that the SSD uses some of its extra layers to estimate the object location and also it can well generalize the object. The approach proposed by Li et. al [27] achieved such high precision due to the process of feature matching that eliminates false positive matches.

C. Results for Logos-32Plus

We experimented on this dataset with the SSD 512 PT, our best result. We explore the effect of balancing the dataset since the L32Plus is imbalanced. In order to balance the dataset, we replicate the images randomly until all classes have the same number of logos. In Figure 6, we can see that balancing the dataset result in a better F-score.

The precision, recall and F-score of our best approach for L32Plus can be seen in Figure 7.

The Logos-32Plus is a relatively new dataset and does not have other research works beyond the one from its authors.

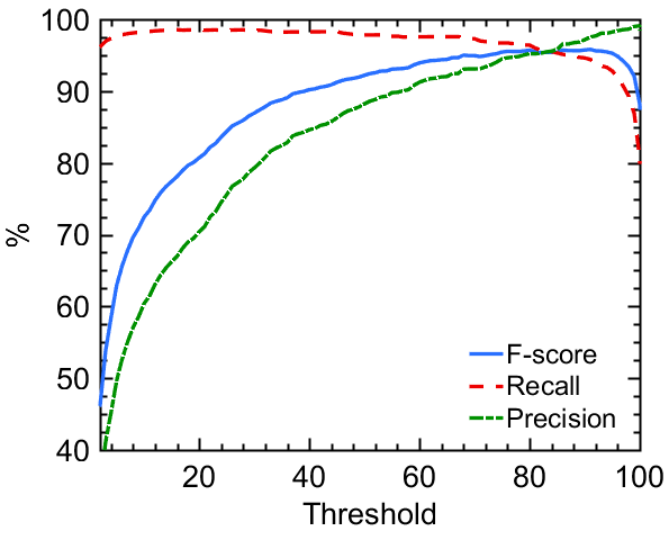

Fig. 7. F-score, Precision and Recall of the best approach for L32Plus dataset.

Analyzing the results, we see that we outperform by 1.2% the F-score and by 3.8% the recall. We compared our results against the "TC-VII" without the "Ground Truth to the obj. prop.", because we do not use the Ground Truth detection in the testing phase.

D. Error analysis

Analyzing out the best model with the threshold of 0.9, the model failed in 75 images of 3960 tested. In 57 images the approach did not find the logo discarding the image as no-logo, in 15 images the approach "found" a logo in the background (false positive) and in only 3 images the approach confused the logo. We have found some inconsistencies in the dataset was pointed out also by [15]. Analyzing this inconsistency, we found that one of our 3 images that confused the logo was, in fact, a true positive as shown in the Figure 8.

Figure 9 show some examples of false positive found by our approach. Analyzing, we see that the mistakes made have some logical explanation. In the left image, the apple logo resemble the Pepsi logo, as for the image in the middle the BMW logo is circular just like the Pepsi logo, and in the last one, the apple logo resembles the apple in the figure.

VI. Conclusion

In this work, we investigated the use of DCNN, transfer learning and data augmentation on logo recognition system. The combination among them has shown that DCNN is very suitable for this task, even with relatively small train set it provides greater recall and f-score. A relevant contribution of this paper is the use of data augmentation combined with transfer learning to surpass the lower data issue and allow to use deeper networks. These techniques improve the performance of DCNN in this scenario. The results of our approach reinforce the robustness of DCNN approach, which surpasses the F1-score literature results.

Fig. 8. Example of inconsistency in the FL32 dataset, this image belongs to the Tsingtao class, but our algorithm detected a logo of Guinness (in yellow) which is a "false" false positive. The image was cropped for better visualization.

VII. Acknowledgment

The authors are thankful to the Brazilian funding agencies CNPq, CAPES and FAPEMIG and to the Federal University of Ouro Preto (UFOP) for supporting this work.

References

[1] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[2] E. Francesconi, P. Frasconi, M. Gori, S. Marinai, J. Sheng, G. Soda, and A. Sperduti, "Logo recognition by recursive neural networks," in *Graphics Recognition Algorithms and Systems*. Springer, 1997, pp. 104–117.

[3] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. IEEE, 1999, pp. 1150–1157.

[4] ——, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] A. D. Bagdanov, L. Ballan, M. Bertini, and A. Del Bimbo, "Trademark matching and retrieval in sports video databases," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*. ACM, 2007, pp. 79–86.

[6] J. Kleban, X. Xie, and W.-Y. Ma, "Spatial pyramid mining for logo detection in natural scenes," in *Multimedia and Expo, 2008 IEEE International Conference on*. IEEE, 2008, pp. 1077–1080.

[7] S. Romberg, L. G. Pueyo, R. Lienhart, and R. Van Zwol, "Scalable logo recognition in real-world images," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*. ACM, 2011, p. 25.

[8] J. Revaud, M. Douze, and C. Schmid, "Correlation-based burstiness for logo retrieval," in *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012, pp. 965–968.

[9] S. Romberg and R. Lienhart, "Bundle min-hashing for logo recognition," in *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*. ACM, 2013, pp. 113–120.

[10] J. Krapac, F. Perronnin, T. Furon, and H. Jégou, "Instance classification with prototype selection," in *Proceedings of International Conference on Multimedia Retrieval*. ACM, 2014, p. 431.

[11] R. Boia and C. Florea, "Homographic class template for logo localization and recognition," in *Pattern Recognition and Image Analysis*. Springer, 2015, pp. 487–495.

TABLE V
COMPARISON OF OUR BEST APPROACHES AGAINST OTHERS STATE-OF-THE-ART ON L32PLUS

Method	Method	Year	Dataset	Precision	Recall	F1
Bianco et. al [17]	DL	2017	L32plus	0.989	0.906	0.946
SSD 512 PT U	DL	2017	L32plus	0.944	0.960	0.951
SSD 512 PT B	DL	2017	L32plus	0.975	**0.944**	**0.958**

Fig. 9. Example of false positive detections. In the left image the correct if Apple and the predicted was Pepsi. The image in the middle is BMW and was classified as pepsi. The right image is a no-logo but was classified as Apple.

[12] F. Yang and M. Bansal, "Feature fusion by similarity regression for logo retrieval," in *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE, 2015, pp. 959–959.

[13] F. N. Iandola, A. Shen, P. Gao, and K. Keutzer, "Deeplogo: Hitting logo recognition with the deep neural network hammer," *arXiv preprint arXiv:1510.02131*, 2015.

[14] C. Eggert, A. Winschel, and R. Lienhart, "On the benefit of synthetic data for company logo detection," in *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*. ACM, 2015, pp. 1283–1286.

[15] G. Oliveira, X. Frazão, A. Pimentel, and B. Ribeiro, "Automatic graphic logo detection via fast region-based convolutional networks," *arXiv preprint arXiv:1604.06083*, 2016.

[16] R. Boia, C. Florea, L. Florea, and R. Dogaru, "Logo localization and recognition in natural images using homographic class graphs," *Machine Vision and Applications*, vol. 27, no. 2, pp. 287–301, 2016.

[17] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini, "Deep learning for logo recognition," *arXiv preprint arXiv:1701.02620*, 2017.

[18] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[19] S. J. Pan and Q. Yang, "A survey on transfer learning," in *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10. IEEE, 2010, pp. 1345–1359.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "Ssd: Single shot multibox detector," *arXiv preprint arXiv:1512.02325*, 2015.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*. Springer, 2016, pp. 21–37.

[22] Y. Nesterov, "A method of solving a convex programming problem with convergence rate o (1/k2)," in *Soviet Mathematics Doklady*, vol. 2, 1983, pp. 372–376.

[23] D. Masko and P. Hensman, "The impact of imbalanced training data for convolutional neural networks," 2015.

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[25] A. Joly and O. Buisson, "Logo retrieval with a contrario visual query expansion," in *Proceedings of the 17th ACM international conference on Multimedia*. ACM, 2009, pp. 581–584.

[26] Y. Kalantidis, L. Pueyo, M. Trevisiol, R. van Zwol, and Y. Avrithis, "Scalable triangulation-based logo recognition," in *in Proceedings of ACM International Conference on Multimedia Retrieval (ICMR 2011)*, Trento, Italy, April 2011.

[27] K.-W. Li, S.-Y. Chen, S. Su, D.-J. Duh, H. Zhang, and S. Li, "Logo detection with extendibility and discrimination," *Multimedia tools and applications*, vol. 72, no. 2, pp. 1285–1310, 2014.

[28] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini, "Logo recognition using cnn features," in *International Conference on Image Analysis and Processing*. Springer, 2015, pp. 438–448.