Noname manuscript No.
(will be inserted by the editor)

Spam Image Detection in Social Networks with Deep Convolutional Neural Networks

Leonel Cruz · Mauro Gomez · Jonatan Poveda · Mouna Makni · Beatriz Otero · Ruben Tous

Received: date / Accepted: date

Abstract In this paper, we report a work consisting in using deep convolutional neural networks (CNNs) for detecting spam images in social networks. The final goal is to facilitate searching and discovering user-generated content (UGC) with potential value for digital marketing tasks. The resulting models are able to detect the ten most common types of spam in Instagram. The CNNs were trained in a distributed manner with Apache Spark over MareNostrum, a petascale supercomputer.

Keywords Deep Learning · Spam Detection · Instagram · Spark

1 Introduction

Nowadays, there is a growing interest in exploiting the photos that users share on social networks such as Instagram or Twitter [5], a part of the so-called user-generated content (UGC). On the one hand, users' photos can be analyzed to obtain knowledge about users behavior and opinions in general, or with respect to a certain products or brands. On the other hand, some users' photos can be of value themselves, as original and authentic content that can be used, upon users' permission, in different communication channels.

However, discovering valuable images on social media streams is challenging. The amount of images to process is huge and, while they help, user defined tags are scarce and noisy. The most part of current solutions rely on costly manual curation tasks over random samples. This way many contents are not even processed, and many valuable photos go unnoticed.

Leonel Cruz, Mauro Gomez and Jonatan Poveda
Adsmurai. Barcelona, Spain
E-mail: leonel.cruz@example.com

Mouna Makni, Beatriz Otero and Ruben Tous
Universitat Politècnica de Catalunya (UPC). Barcelona, Spain

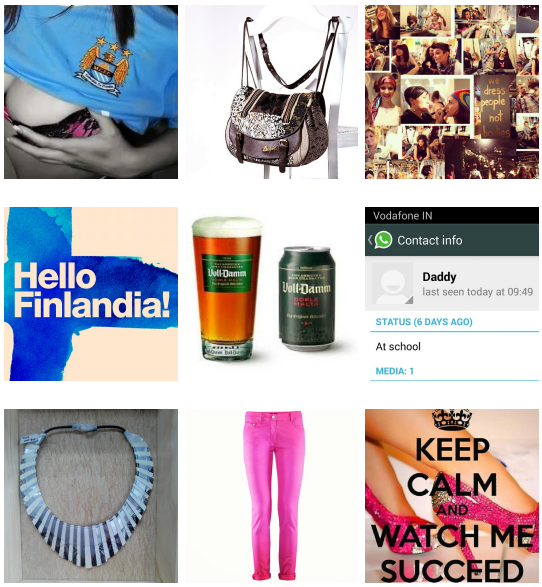

Fig. 1: Example images posted by Instagram users and tagged with Desigual's promotional hashtags (e.g. #lavidaeschula)

This situation is aggravated by the huge amount of *spam images*. In the context of UGC, we consider spam any image that, in the most usage scenarios, has no value neither as a knowledge carrier nor as a exploitable content. This includes images posted with commercial purposes, but also inappropriate images or very common images with low value such as photos of pets (Figure 4 shows some examples). Of course, this may be relative in some cases (e.g. for a pet food company).

We propose using deep convolutional neural networks (CNNs) to automatically detect the most common types of spam images, facilitating their early filtering. We do not merge the different spam categories in order to enable a fine-grained filtering, as, as mentioned before, the definition of spam is relative to the usage scenario.

TODO (LEONEL): A comment of the type of CNN, size of datasets.

We train the CNNs in a distributed manner, with the help of the Apache Spark cluster-computing framework, over MareNostrum, a petascale supercomputer. This implies several challenges, not only in terms of deployment but specially regarding scalability and proper configuration. Simply running on hundreds of cores may yield poor benefits or even degraded performance due to overheads. We deal with this issues and we have identifed solutions to potential inefficiencies of such a data-centric workload running over a compute-centric infrastructure.

TODO (RUBEN): SUMMARY OF CONTRIBUTIONS

2 Related Work

The work presented in this paper is related to recent works attempting to facilitate the classification and search of images in social networks such as Instagram and Twitter. Some works, such as [2], [4] or [1], also apply image recognition techniques to enrich the metadata originally present in the images in order to facilitate their processing. All latest works rely on CNNs as an underlying technique. In our case, the applied image recognition techniques, while also relying in CNNs, are tuned for spam detection. TODO: Explain particularities of the approach (distributed, spark, etc.)

TODO (?): Something about the distributed part, including this ref: [3]

3 Spam recognition CNN architecture

TODO (LEONEL): Layers, etc.

4 Dataset

TODO (?)

5 Distributed training setup

We trained the models in a distribued manner on MareNostrum, the Spanish Tier-0 supercomputer. It is an IBM System X iDataplex based on Intel Sandy Bridge EP processors at 2.6 GHz (two 8-core Intel Xeon processors E5-2670 per machine), 2 GB/core (32 GB/node). Currently the supercomputer consists of 48,896 Intel Sandy Bridge cores in 3,056 JS21 nodes, and 84 Xeon Phi 5110P in 42 nodes (not used in this work), with more than 104.6 TB of main memory and 2 PB of GPFS (General Parallel File System) disk storage. All compute nodes are interconnected through an Infiniband FDR10 network, with a non-blocking fat tree network topology. MareNostrum has a peak performance of 1.1 Petaflops.

While CNNs are trained more efficiently over GPUs, the availability of such huge computation power made us design a strategy for training our models over an HPC setup. The key element was Apache Spark [**?**], a distributed computing framework that is already available at MareNostrum and provides a very convenient abstraction layer. Spark is an implementation of the so-called Resilient Distributed Dataset (RDD) abstraction, which hides the details of distribution and fault-tolerance for large collections of items. Spark is designed to avoid the file system as much as possible, retaining most data resident in distributed memory across phases in the same job. Such memory-resident feature stands to benefit many applications, such as machine learning or clustering, that require extensive reuse of results across multiple iterations. Memory usage is the key aspect of Spark and the main reason that it outperforms Hadoop

tag	#positives	#total images	BoW	CNN	CNN-TL
selfie	295	9,254	72%	87%	93%
group_selfie	98	8,982	76%	88%	95%
spam	319	9,298	69%	78%	91%
burguer	474	9,319	81%	89%	95%
nails	434	9,300	83%	92%	97%
sushi	571	9,491	86%	93%	96%

Table 1: Training setup and results of some of the 100 new models that we have trained for image semantics recognition.

for many applications. In order to be able to efficiently run a Spark cluster on the MareNostrum supercomputer (which runs an IBM LSF Platform workload manager), we employ the Spark4MN framework described in [3].

As we were distributing computation over Spark, based on Java, we chose the DL4J framework TODO CITATION to be able to train our models in a distributed manner over this setup. DL4J is... TODO BREVE COMENTARIO

6 Results

We have submitted and tested several hundreds of jobs to MareNostrum, but we describe only the results that are of significance. Our runs include an extensive set of configurations; for brevity, when those parameters were shown to be either irrelevant or to have negligible effect, we use default values. Each experimental configuration was repeated at least 5 times. Unless otherwise stated, we report median values in seconds.

TODO (Leonel)...

6.1 Classification accuracy

6.2 Performance and Scalability

The main goal of these experiments is to evaluate the speed-up, scale-up, and size-up properties of the proposed distributed training strategy. To this end, we use datasets up to hundreds of GBs TODO of raw data. The size of RDDs is reported to be 2-5 times larger than that; in our experiments 400GBs of data in the sort-by-key application correspond to an RDD of 1TB. The cluster sizes range from 8 cores up to 1024 (i.e., 64 machines)...TODO

6.2.1 speed-up

In the first set of experiments, we keep the input dataset constant and we increase the size of nodes/cores running the Spark application; whenever we refer to nodes, we mean MareNostrum machines that run the executors, while

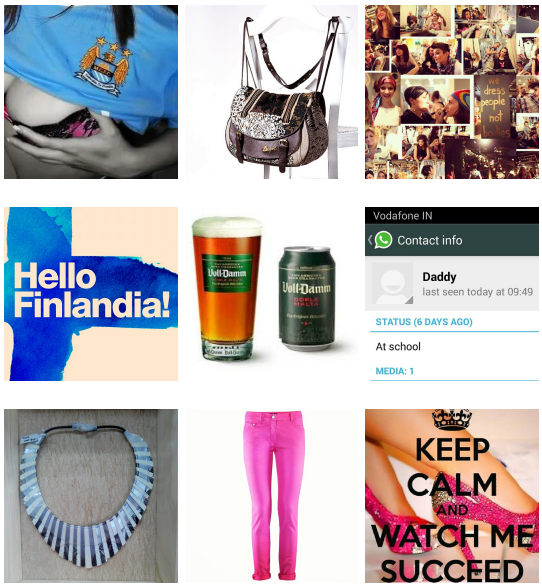

Fig. 2: Example true positives)

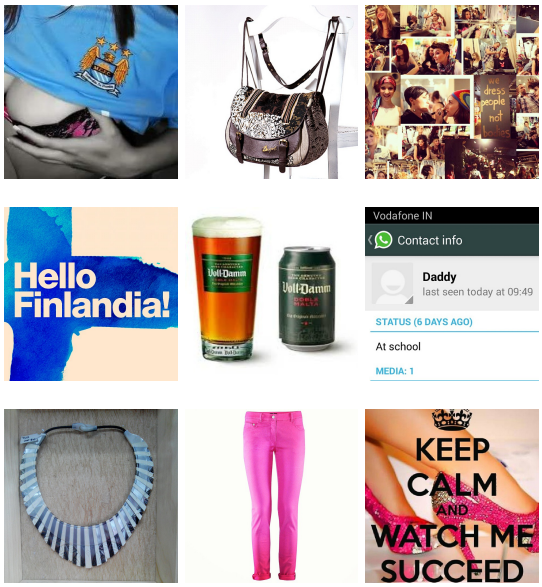

Fig. 3: Example false positives)

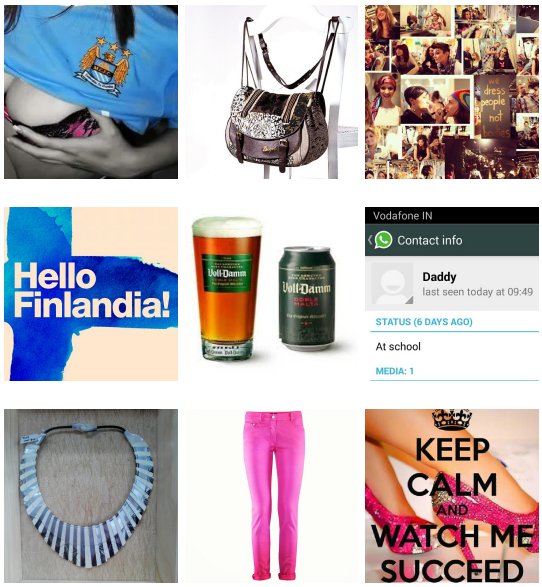

Fig. 4: Example false negatives)

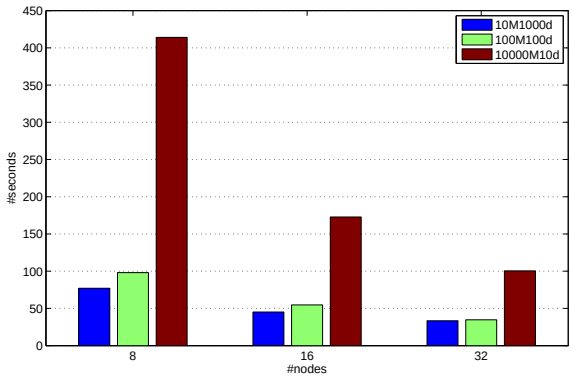

Fig. 5: Times for training....

the driver always runs on a separate machine; each machine is equipped with 16 cores and 32 GB of RAM. The results from 128 (8 nodes) up to 512 cores (32 nodes) are shown in Figure 7, where we can see that for large datasets in terms of number of records, the training can scale well. In the figure, we present the performance for the most efficient configurations; we discuss these configurations in detail later. TODO

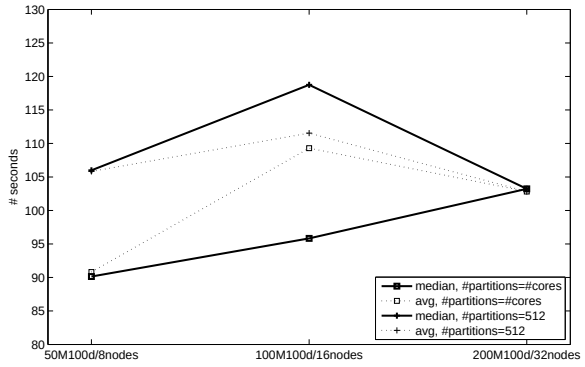

Fig. 6: Times for training....

6.2.2 scale-up

We process the same datasets, and we now modify both the number of records and the number of machines, i.e., the infrastructure scales-out. .... The results are shown in Figure **??**(top). In this figure, we show both the average and the median values. Ideally, all the plots should be horizontal; our system behaves closely to that. ....

6.2.3 size-up

We perform a third set of experiments, to assess the capability of sizing-up. We keep the number of nodes constant (either 16 or 32), and we gradually increase the dataset from 100GBs to 200GBs (raw data sizes). As shown in Figure **??**(bottom), Spark4MN exhibits a behavior where the curves are (almost) linear. ....

7 Conclusions

The research work presented in this paper analyzes the usage of distributed deep convolutional neural networks (CNNs) for .....

TODO (?)

Acknowledgements This work is partially supported by the Spanish Ministry of Economy and Competitivity under contract TIN2015-65316-P and by the SGR programme (2014-SGR-1051) of the Catalan Government.

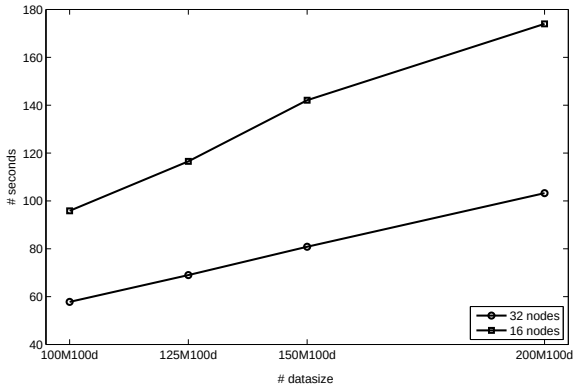

Fig. 7: Times for training....

References

1. Denton, E., Weston, J., Paluri, M., Bourdev, L., Fergus, R.: User conditional hashtag prediction for images. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, pp. 1731–1740. ACM, New York, NY, USA (2015). DOI 10.1145/2783258.2788576. URL http://doi.acm.org/10.1145/2783258.2788576
2. Park, M., Li, H., Kim, J.: HARRISON: A benchmark on hashtag recommendation for real-world images in social networks. CoRR **abs/1605.05054** (2016)
3. Tous, R., Gounaris, A., Tripiana, C., Torres, J., Girona, S., Ayguad, E., Labarta, J., Becerra, Y., Carrera, D.N., Valero, M.: Spark deployment and performance evaluation on the marenostrum supercomputer. In: Big Data, pp. 299–306. IEEE (2015)
4. Tous, R., Torres, J., Ayguade, E.: Multimedia big data computing for in-depth event analysis. In: BigMM, pp. 144–147. IEEE (2015)
5. Tous, R., Wust, O., Gomez, M., Poveda, J., Elena, M., Torres, J., Makni, M., Ayguade, E.: User-generated content curation with deep convolutional neural networks. IEEE (2016)