Understanding the Energy Saving Potential of Smart Scale Selection in the Viola and Jones Facial Detection Algorithm

Noel Pérez, Sergio Faria and Miguel Coimbra
Instituto de Telecomunicações de Portugal, FCUP, Rua do Campo Alegre 1021/1055, 4169-007 Porto, Portugal

Keywords: Face Detection, Battery Consumption, Viola and Jones Detector.

Abstract: In this paper we study the energy saving potential of smart scale selection methods when using the Viola and Jones face detector running on smartphone devices. Our motivation is that cloud and edge-cloud multi-user environments may provide enough contextual information to create this type of scale selection algorithms. Given their non-trivial design, we must first inspect its actual benefits, before committing important research resources to actually produce relevant smart scale selection methods. Our experimental methodology in this paper assumes the optimum scenario of a perfect selection of scales for each image (drawn from ground truth annotation, using well-known public datasets), comparing it with the typical multi-scale geometrical progression approach of the Viola Jones algorithm, measuring both classification precision and recall, as well as algorithmic execution time and battery consumption on Android smartphone devices. Results show that if we manage to approximate this perfect scale selection, we obtain very significant energy savings, motivating a strong research investment on this topic.

1 INTRODUCTION

The problem of face detection has been considered as one of the most important topics in computer vision during the past 20 years. It has been widely studied by the scientifics due to its application on face recognition (Taigman et al., 2014), face tracking (Kalal et al., 2010), facial shape analysis (Chen et al., 2014) as well as for facial behavioral analysis (Fu et al., 2010).

At present, facial image processing constitutes a powerful resource for researchers involved in the detection of potential psychiatric disorders (e.g. melancholic depression) (Hyett et al., 2016), the recognition of surgically altered faces (Bhatt et al., 2013) and emotion detection in educational environments (Saneiro et al., 2014). These potential scenarios could be benefited with the use of facial detection algorithms located on mobile device clouds (Li et al., 2015). Two benefits of using this type of cloud are related to (1) more storage and processing capacity and (2) minimization of the mobile's energy consumption (e.g. since the prior knowledge of the image details are shared among all nodes, the detector performance could be optimized). However, those benefits imply the use of satisfactory cloud offloading strategies to mitigate the issue of limited resources of mobile devices (Barbera et al., 2013).

Although the majority of mobile energy optimization research in the software layer have been focused in cloud offloading (Khan, 2015) (Kwon and Tilevich, 2013), the algorithm complexity of implemented apps deserve a special attention as well. The complexity of a face detection app depends on the strategy to follow (the goal) and the available resources in the hosting platform. Usually, these apps sacrifice accuracy computations in order to avoid high-level battery consumption or overheating (Oneto et al., 2015). Thus, the implementation of face detection apps with low battery consumption constitutes a challenging task.

After the seminal work of Viola and Jones (VJ) (Viola and Jones, 2001), many face detectors have been proposed using different sets of features. In (Viola et al., 2003) and (Li et al., 2002), they used a variation of Haar-like features for multi-view face detection by capturing non-symmetrical details and diagonal structures in the image. In (Zhang et al., 2007), the detector used a set of multi-block local binary patterns features for capturing large scale structures. The evaluated set of features was 1/20 smaller than Haar-like features for a basic resolution of 20x20 pixels. Moreover, in (Meynet et al., 2007), the use of anisotropic Gaussian filters and its first derivative enabled the weak classifier to capture contour singularities with a smooth low resolution and, to approx-

Perez N., Faria S. and Coimbra M.
Understanding the Energy Saving Potential of Smart Scale Selection in the Viola and Jones Facial Detection Algorithm.
DOI: 10.5220/0006247501220127
In *Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2017)*, pages 122-127
ISBN: 978-989-758-215-8

imate the edge transition in the orthogonal direction respectively. The introduction of speeded up robust features, logistic regression based weak classifier and the area under receiver operating curve based criterion (instead of detection rate and false positive rate) produced a faster convergence in the detector by using fewer cascade stages (cascade optimization) (Li et al., 2011).

Despite the rapid growth of face detection approaches, the VJ detector (Viola and Jones, 2001) arises as the most intuitive implementation for mobile devices because of its fast performance and low algorithm complexity. However, this detector explores several pre-computed scales depending on the image's resolution, given its absence of prior knowledge and context about the image to be processed. This excellent ability to adapt to any image comes with the cost of processing many more scales than the ones relevant for each specific image, leading to excessive battery consumption that could be mitigated via a smarter, context informed scale-selection. Rich information environments such as cloud or edge-cloud ones associated with specific events, motivate us to think that when a photo is actually taken, couldn't we exploit context information from nearby smartphones taking photos, that give us insights about what to expect from this new photo? If so, we need to research smart scale selection algorithms, based on information obtained from this cloud environment, which is a non-trivial task. Is it worth pursuing this goal? Will the battery savings be relevant enough to justify all this research investment?

In this paper we will address this research question.

The reminder of the work is ordered as follows: Section II describes the materials and methods of our experiments, followed by its results and their associated discussion on Section III. Finally, conclusions are drawn in Section IV.

2 MATERIALS AND METHODS

2.1 Datasets

The public WIDER FACE dataset is a large face detection benchmark dataset with 32,203 images and 393,703 face annotations. It presents a high degree of variability in scale, pose, occlusion, expression, appearance and illumination. In this dataset, all the faces are well-documented according to its bounding box (the detection ground truth). The dataset distribution is based on 61 event classes (Yang et al., 2016).

For our detection purposes, we selected two different datasets from the WIDER FACE set of datasets. Both of these are mostly formed with images containing frontal faces from different event classes. The first dataset named DS1 is formed by 55 images and contains a total of 3415 small faces representing the *large group, group team, group, meeting, press conference, cheering, award ceremony, demonstration protesters* and *family group* event classes (see Figure 1 a). Meanwhile the second dataset called DS2 is formed by 55 images and has a total of 173 medium-large faces belonging to the *waiter, couple, family group, large group* and *group* event classes (see Figure 1 b).

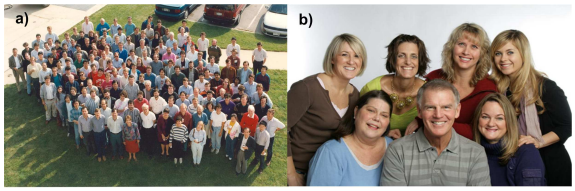

Figure 1: Image examples from DS1 and DS2 datasets, a) `Group_12_108` and b) `Family_Group_20_239`.

2.2 Viola and Jones Detector

Since 2001, the VJ face detector (Viola and Jones, 2001) has been considered the most popular algorithm in the area of face detection. It implements three main modules that enable it to perform as a real-time face detector: the integral image consisting of a fast and efficient computation of the sum of values in a rectangle subset of a grid; the classifier learning with adaboost (adaptive boosting), which finds highly accurate hypothesis by combining many "weak" hypothesis (each with moderate accuracy) and, the attentional cascade structure (viewed as a degenerate decision tree in which each node contains a set of weak classifiers), which rejects most of the negative examples (in early stages) while keeping almost all the positive examples (in late stages). This detector has palpable limitations such as: the cascade is constructed manually, thus the threshold (for early rejections) and the number of weak classifiers per node is also defined manually and, the training phase for a good face detector can take weeks or even more time (depending on hardware conditions) for fine-tuning. Nevertheless, the major concern of implementing this detector on mobile devices is the exhaustive scales exploration, which dramatically increases the execution time and battery consumption.

2.3 Smart Scale Selection Method

The smart scale selection (SSS) method could be considered as an important module in the framework of the VJ face detector due to the necessity of saving energy on mobile devices. The concept behind this method is a smart strategy able to determine the relevant scale for each specific image independently of the hosting platform.

The standard architecture of the VJ detector uses a typical multi-scale geometrical progression approach for performing the face detection task. Thus, the taken picture is processed by the detector until the last scale is reached, leading to an excessive battery consumption and execution time (see Figure 2 a). With the introduction of the SSS method is possible to use the VJ detector in a single device with only one scale (the relevant one) for accomplishing the face detection task and both the battery consumption and execution time could be reduced.

In a more complex scenario such as a cloud or edge-cloud where nearby smartphones are taking photos and sharing insights about them, the SSS method performs the best scale selection by taking advantages of the current metadata (images, indexes, GPS location, devices in the neighborhood, etc.) that are being shared on these environments. As it is shown in Figure 2 b) the SSS method receives the shared metadata in the edge-cloud and process them to determine the best performance scale (output), which will be returned and shared throughout the edge-cloud to all connected devices as well. Then, each device will perform the face detection task by using the VJ detector with only one scale (the one shared in the edge-cloud) instead of the whole pre-computed set of scales (as in the Figure 2 a).

Some advantages of using the SSS method are related to: (1) reducing the number of the detector iterations; (2) minimizing the execution time per image and, (3) maximizing the battery life of mobile devices. Although these advantages provide an efficient way for saving energy on mobile devices, there is a trend to lose detection precision and recall due to the own nature of the method (avoids the exhaustive scales exploration).

2.4 Experimental Methodology

This section outlines the experimental evaluation of the VJ detector with the SSS method through the consideration of an optimum scenario where the SSS method makes a perfect selection of scales for each image (the scales were selected from the ground truth annotation of the datasets). Then, we compared it

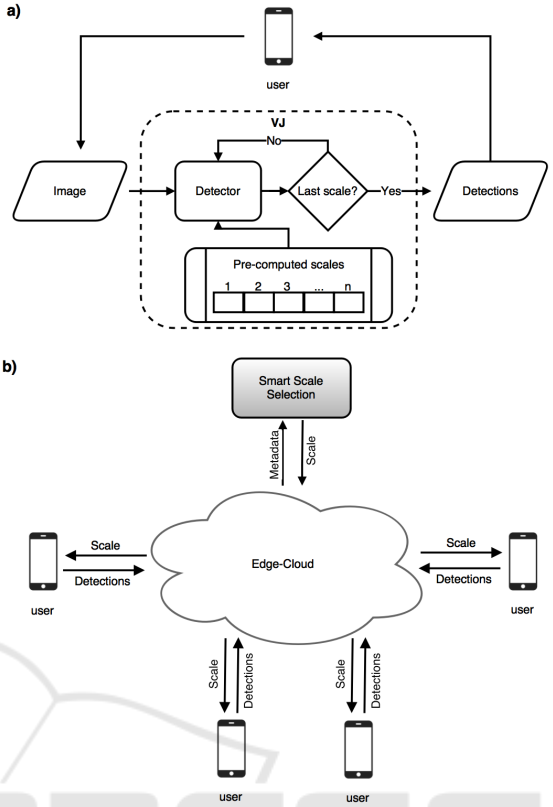

Figure 2: The VJ face detection framework a) in a single device, b) in the edge-cloud with the SSS method.

against the standard VJ detector following the overall procedure:

- Applying the standard VJ detector and the VJ detector with the SSS method to the DS1 and DS2 previously formed datasets.
- Validating the detection results and energy consumption of both detectors according to the selected ground truth and energy measurement protocol respectively.
- Establishing the comparative analysis of obtained results according the mean of precision (mP), recall (mR), execution time (mET) and battery consumption (mBC) metrics. All comparisons were based on the Wilcoxon statistical test (Hollander et al., 2013) with a confidence level of alpha=0.05 to assess the meaningfulness of differences between both detectors.

2.4.1 Ground Truth Selection

The experimental datasets present a high degree of variability in terms of detection scales, which could lead to unfair detection results because of the minimum scale used by the detectors (24x24 pixels).

Thus, we implemented a procedure to find the border line between the detector minimum scale and the ground truth set by considering two variables: the dimension and area of the bounding box.

The procedure started by (1) removing the bounding boxes with any dimension lesser than 10 pixels; (2) removing the bounding boxes with area lesser than 100 square pixels; (3) validating the detection results against the two new ground truth sets (outputs of the steps 1 and 2) and, (4) selecting the ground truth set that maximizes the detector's precision and recall. This procedure was repeated two more times by setting the dimension and area to 15, 20 pixels and 225, 400 square pixels respectively. The optimal ground truth set for our experiment was the one formed by bounding boxes with an area equal and greater than 225 square pixels.

2.4.2 Detector Configuration and Implementation

We used the tree-based 20x20 gentle adaboost frontal face detector kernel from the OpenCV project with two changes made in the implementation as follows: (1) the *base scale* and *increment* parameters were set to 1 for producing linear integer scales instead of floating point scales and (2) the displacement of the detector was changed from a multiplicative increment to additive one, for reducing the window's step during the displacement. The face detector app was developed in JAVA programming language and installed in a Samsung SM-T530 tablet with Android 5.0.2 and battery capacity of 2100 mAh (estimated by the device).

2.4.3 Energy Measurement Protocol

The Android Debug Bridge service (adb command line tool) (Developers, 2014) facilitated us a satisfactory way to recover two logs from the mobile device. The BatteryStats log, containing important stats regarding the battery consumption of every single process (for our purpose, we took only the information from the process that executed the detector) and the ProcessActivity log, containing stats about the execution time (in nanoseconds) of the detector, as well as the number of detections per image. Both logs were employed for further computation of the the execution time and battery consumption scores.

3 RESULTS AND DISCUSSION

According to the experimental methodology section, a total of 110 images containing 3588 faces were analyzed using the the VJ detector with the SSS method, which used the scale 1 and 8 for processing the small (DS1) and medium-large (DS2) datasets respectively. The straightforward statistical comparison based on four metrics over 100 runs exhibited interesting results for both datasets.

3.1 Performance on Small Faces Dataset

In this dataset, the VJ detector with the SSS method was statistically better in precision than the standard VJ detector ($p<0.01$), obtaining a value of mP=0.92 against mP=0.87 respectively. However, it was statistically lower in recall ($p<0.01$), reaching a value of mR=0.75 against mR=0.8 in the standard VJ detector.

Regarding the execution time, the VJ detector with the SSS method reached a mean value of mET=14.03 minutes versus the mET=21.71 minutes needed by the standard VJ detector for processing the whole dataset (see Figure 3). The difference value of 7.68 minutes between both results was statistically significant ($p<0.01$).

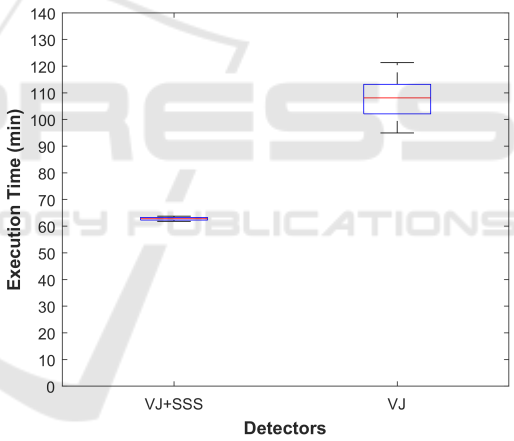

Figure 3: Detectors performance according to the mean of execution time on DS1 dataset.

Concerning the battery consumption, the VJ detector with the SSS method consumed a mean value of mBC=18.14 mAh respect to the mBC=22.20 mAh spent by the standard VJ detector (see Figure 4). These results did not provide statistical evidence of any difference between both detectors (p=0.01).

3.2 Performance on Medium-large Faces Dataset

The best precision in this dataset was reached by the VJ detector with the SSS method (with scale 8), obtaining a mP=1 score versus mP=0.24 reached by the standard VJ detector. The score difference between

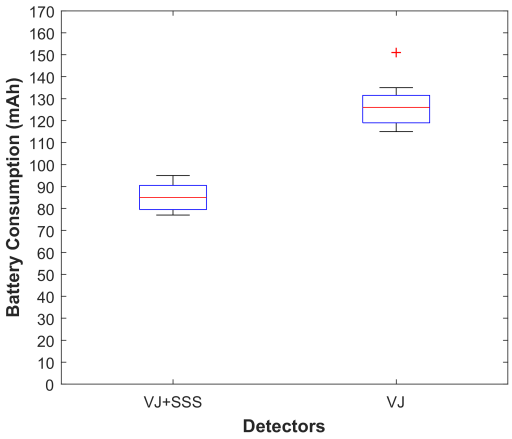

Figure 4: Detectors performance according to the mean of battery consumption on DS1 dataset.

both detectors was statistically significant ($p<0.01$). In term of recall there was also significant difference between both detectors performance. The VJ detector with the SSS method touched a mR=0.71 versus mR=0.86 stretched by the standard VJ detector.

Regarding the execution time, the faster detector was the VJ with the SSS method, reaching a mean value of mET=0.94 minutes against mET=32.43 obtained by the standard VJ detector (see Figure 5). The difference between both detectors performance was statistically significant ($p<0.01$).

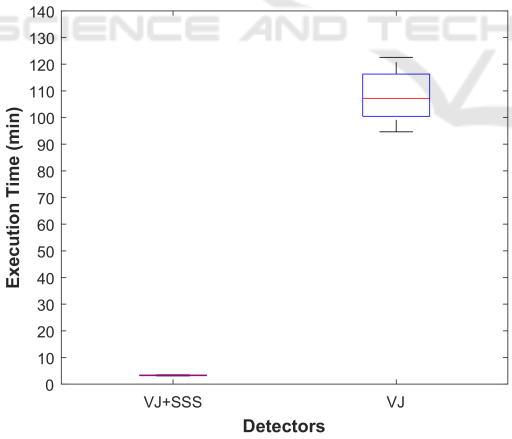

Figure 5: Detectors performance according to the mean of execution time on DS2 dataset.

Concerning the battery consumption, the VJ detector with the SSS method turned out in a mean value of mBC=7.22 mAh versus mBC=46.36 mAh consumed by the standard VJ detector (see Figure 6). The high difference between both results was considered statistically significant ($p<0.01$).

These results were partially expected since the VJ detector with the SSS method explored only one scale

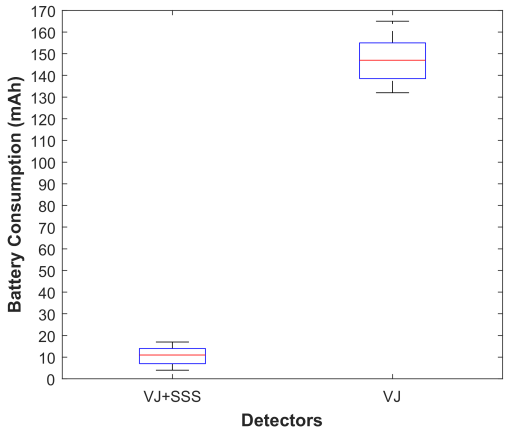

Figure 6: Detectors performance according to the mean of battery consumption on DS2 dataset.

per dataset (scale 1 and 8 for DS1 and DS2 datasets respectively). Thus, it is possible to reach higher precision scores because of the lower introduction of false positive detections. It is possible to decrease the execution time and battery consumption because of the fewer number of the detector iterations. However, it is not possible to reach higher recall scores due to the fewer number of true positive detections (based on one scale). It should be noted that the similar values of battery consumption obtained on DS1 dataset could be related to the particularity of this dataset and the detectors iterations. Since this dataset is formed by small faces, both detectors will interact more times because of the smaller scales and the amount of battery consumed will increase too.

Likewise in DS1 dataset, the obtained results in DS2 dataset were fully expected. With the exception of the recall metric, the VJ detector with the SSS method statistically surpassed the standard VJ detector. This situation could be related to the fact of this dataset being formed by medium-large faces, thus it is reasonable to expect better performance when using large scales. In this case, the VJ detector with the SSS method considered the scale 8 while the standard VJ detector had to explore several pre-computed scales before and after the optimal one. This exhaustive exploration process of the standard VJ detector led to an increment in the precision, execution time and battery consumption metrics as well.

4 CONCLUSIONS

The main contribution of this work is the study of the energy saving potential of a smart scale selection method when using the VJ face detector running on mobile devices. According to the four evaluated met-

rics, the VJ detector with the SSS method was statistically better than the standard VJ detector in precision, execution time and battery consumption and slighly lower in recall on both datasets. These results show that if we manage to approximate this perfect scale selection, we obtain very significant energy savings on limited resources devices. Thus it is worth to invest research time on this topic.

Future work will be aimed to design and develop smart scale selection methods.

ACKNOWLEDGEMENTS

The authors want to thank the "Instituto de Telecomunicações" for the financial support through the HYRAX project (REF: CMUP-ERI/FIA/0048/2013).

REFERENCES

Barbera, M. V., Kosta, S., Mei, A., and Stefa, J. (2013). To offload or not to offload? the bandwidth and energy costs of mobile cloud computing. In *INFOCOM, 2013 Proceedings IEEE*, pages 1285–1293. IEEE.

Bhatt, H. S., Bharadwaj, S., Singh, R., and Vatsa, M. (2013). Recognizing surgically altered face images using multiobjective evolutionary algorithm. *IEEE Transactions on Information Forensics and Security*, 8(1):89–100.

Chen, D., Ren, S., Wei, Y., Cao, X., and Sun, J. (2014). Joint cascade face detection and alignment. In *European Conference on Computer Vision*, pages 109–122. Springer.

Developers, A. (2014). Android debug bridge. https://developer.android.com/studio/command-line/adb.html.

Fu, Y., Guo, G., and Huang, T. S. (2010). Age synthesis and estimation via faces: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 32(11):1955–1976.

Hollander, M., Wolfe, D. A., and Chicken, E. (2013). *Nonparametric statistical methods*. John Wiley & Sons.

Hyett, M. P., Parker, G. B., and Dhall, A. (2016). The utility of facial analysis algorithms in detecting melancholia. In *Advances in Face Detection and Facial Image Analysis*, pages 359–375. Springer.

Kalal, Z., Mikolajczyk, K., and Matas, J. (2010). Face-tld: Tracking-learning-detection applied to faces. In *2010 IEEE International Conference on Image Processing*, pages 3789–3792. IEEE.

Khan, M. A. (2015). A survey of computation offloading strategies for performance improvement of applications running on mobile devices. *Journal of Network and Computer Applications*, 56:28–40.

Kwon, Y.-W. and Tilevich, E. (2013). Reducing the energy consumption of mobile applications behind the scenes. In *ICSM*, pages 170–179. Citeseer.

Li, J., Peng, Z., Xiao, B., and Hua, Y. (2015). Make smartphones last a day: Pre-processing based computer vision application offloading. In *Sensing, Communication, and Networking (SECON), 2015 12th Annual IEEE International Conference on*, pages 462–470. IEEE.

Li, J., Wang, T., and Zhang, Y. (2011). Face detection using surf cascade. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 2183–2190. IEEE.

Li, S. Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H., and Shum, H. (2002). Statistical learning of multi-view face detection. In *European Conference on Computer Vision*, pages 67–81. Springer.

Meynet, J., Popovici, V., and Thiran, J.-P. (2007). Face detection with boosted gaussian features. *Pattern Recognition*, 40(8):2283–2291.

Oneto, L., Ghio, A., Ridella, S., and Anguita, D. (2015). Learning resource-aware classifiers for mobile devices: from regularization to energy efficiency. *Neurocomputing*, 169:225–235.

Saneiro, M., Santos, O. C., Salmeron-Majadas, S., and Boticario, J. G. (2014). Towards emotion detection in educational scenarios from facial expressions and body movements through multimodal approaches. *The Scientific World Journal*, 2014.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708.

Viola, M., Jones, M. J., and Viola, P. (2003). Fast multi-view face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Citeseer.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–511–I–518. IEEE.

Yang, S., Luo, P., Loy, C. C., and Tang, X. (2016). Wider face: A face detection benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhang, L., Chu, R., Xiang, S., Liao, S., and Li, S. Z. (2007). Face detection based on multi-block LBP representation. In *International Conference on Biometrics*, pages 11–18. Springer.