

HOG-Dot: A Parallel Kernel-Based Gradient Extraction for Embedded Image Processing

Luca Maggiani, Cédric Bourrasset, Matteo Petracca, François Berry, Paolo Pagano, Claudio Salvadori

▶ **To cite this version:**

Luca Maggiani, Cédric Bourrasset, Matteo Petracca, François Berry, Paolo Pagano, et al.. HOG-Dot: A Parallel Kernel-Based Gradient Extraction for Embedded Image Processing. IEEE Signal Processing Letters, 2015, 10.1109/LSP.2015.2463092 . hal-01220417

HAL Id: hal-01220417
https://hal.science/hal-01220417

Submitted on 26 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HOG-Dot: a Parallel Kernel-Based Gradient Extraction for Embedded Image Processing

Luca Maggiani *‡, Cedric Bourrasset ‡, Matteo Petracca*, Francois Berry‡, Paolo Pagano†, Claudio Salvadori*
* TeCIP Institute, Scuola Superiore Sant'Anna, Pisa, Italy
† National Laboratory of Photonic Networks, CNIT, Pisa, Italy
‡ Institute Pascal, Universite Blaise Pascal, Clermont Ferrand, France

***Abstract*—In this paper we propose HOG-Dot, a method for the direct computation of the polar image gradients coordinates from the pixels values. The proposed algorithm, to be used as the first step of the Histogram of Oriented Gradient (HOG) pipeline, approximates the exact gradient with its projection onto a versor chosen among the projection plane set. Instead of non linear computations, the HOG-Dot method exploits linear operations while introducing a bounded approximation error with respect to other HOG approaches, thus resulting a more suitable solution for embedded devices. Concerning the state of the art, it also achieves improved accuracy with the mathematical spatial gradient formulation.**

I. Introduction

Embedded image processing devices, namely Smart Cameras (SCs), are nowadays considered as a powerful solution to extract complex information from the surrounding environment in which they are deployed. In designing SC one of the biggest efforts is the porting of complex and computational intensive computer vision pipelines to resource constrained embedded devices able to reduce deployment costs. In such a context, the Histogram of Oriented Gradient (HOG) algorithm [1] represents the state of the art for the pedestrian detections, allowing the classification of images windows represented by using histograms of local spatial derivatives through machine learning algorithms (i.e., Support Vector Machine [2]). The HOG pipeline is composed by a gradient extraction followed by a local spatial aggregation histogram. The resulting histogram is then normalised and compared to a model reference within the classifier. In particular, the spatial gradient extraction is based on intensive pixel-wise operations usually exploited with non-linear operators [3]. Thus its deployment into resource constrained embedded devices does not guarantee real-time processing performance. Within the HOG algorithm, spatial gradient extraction is usually computed with square root and arctangent operators from the 1-D spatial derivatives. In Eq. 1 the 1-D components are computed as:

$$G_x = \frac{\partial I}{\partial x} = I\big(x+1,y\big) - I\big(x-1,y\big) \qquad (1)$$
$$G_y = \frac{\partial I}{\partial y} = I\big(x,y+1\big) - I\big(x,y-1\big)$$

The gradient magnitude $\|\nabla I(x,y)\|$ and orientation angle θ are then computed for each pixel as follows:

$$\|\nabla I(x,y)\| = \sqrt{G_x^2 + G_y^2} \qquad (2)$$
$$\theta = \arctan\frac{G_y}{G_x}$$

The resulting gradient orientation θ is then sampled over a set of discrete samples, called bin, to create a dense grid of uniformly spaced histogram cells.

Since gradient extraction represents a common image processing technique, hardware and software implementations, especially for embedded devices, have been proposed in literature. In [4] several VLSI gradient extraction implementations have been reviewed. Such solutions deploy gradient magnitudes evaluations directly in a custom CMOS camera focal plane as pre-processing step. In [5] a Look-Up table (LUT) implementation has been proposed. It exploits a set of precomputed Look-Up Tables (LUTs) to approximate both the *square-root* and the *arc-tangent*. The results are used to index a 2D LUT and to address the preloaded magnitude and angle values. In such implementation the memory footprint grows accordingly to the size of the 2D LUT, and consequently to the gradient resolution. More recently, in [6] the polar conversion is computed implementing both the *arc-tangent* and the *square-root* functions with iterative hardware operations, thus reducing significantly the data throughput.

In this paper we present HOG-Dot, a kernel-based gradient extraction algorithm. It permits to compute directly the gradient polar components by deploying linear operations. Instead of evaluating an approximation of the Eq. 2 or deploying a custom circuitry as shown in [4], we rather propose a mathematical reformulation which offers a bounded approximation error with respect to the exact gradient. By modifying the number of bins, the resulting error is then controllable and deterministic.

The rest of this letter is organized as follows. First, in Sec. II the proposed spatial gradient expression is mathematically evaluated. In Sec. III our method is then compared with other state of the art gradient extraction formulations and finally Sec. IV concludes the paper.

II. Methodology

One of the main issue in developing the HOG feature extraction for embedded SC is the polar coordinates computations through non linear floating-point operations. Even

though floating-point arithmetic could be exploited even in resource constrained devices, as shown in [7] its implementation requires a non-negligible amount of time thus decreasing the output throughput. Thus, in this paper we propose a linear method capable of approximating the *square-root* and the *arc-tangent*. It is based on a set of linear operations that can be easily implemented within embedded device. Let us consider a generic vector v, its module can be seen as a dot product with its parallel versor:

$$\| v \| = v \cdot \hat{i}_\theta \tag{3}$$

where θ is the angle and then the direction of the versor parallel to v. By exploiting this property, we can state that the exact gradient is equal to:

$$\| \nabla I(x,y) \| = \nabla I(x,y) \cdot \hat{i}_\theta \tag{4}$$

where θ is the direction of the vector $\nabla I(x,y)$. Moreover, considering a generic versor $\hat{i}_k$, the following relation is valid:

$$\frac{\partial I}{\partial \hat{i}_k}(x,y) = \nabla I(x,y) \cdot \hat{i}_k \tag{5}$$

Since the HOG algorithm considers a discrete angle domain that consists of equidistant samples within the interval $[0,\pi)$, we consider a limited set of N directions, represented by the following set of versors (illustrated in Fig. 1):

$$B_{\hat{i}} = \{\ \hat{i_k}(\theta_k) \mid \theta_k = k\pi/N,\ k = 0,\dots,N-1\} \tag{6}$$

where each versor can be represented in cartesian coordinates as:

$$\hat{i_k} = \hat{x}\cos\theta_k + \hat{y}\sin\theta_k \tag{7}$$

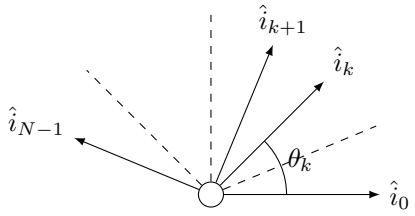


Fig. 1. The versors beam used to compute the dot product.

According to the above reported equations, the intensity of the gradient projection $\nabla I(x,y)$ onto each versor of the set $B_{\hat{i}}$ can be evaluated as follows:

$$\nabla I(x,y) \cdot \hat{i_k} = \frac{\partial I}{\partial \hat{i}_k}(x,y) = \cos\theta_k \frac{\partial I}{\partial x} + \sin\theta_k \frac{\partial I}{\partial y} \tag{8}$$

Since we consider the HOG case, the partial derivative functions are expressed in terms of pixel intensity and position as in Eq. 1.

Thus, by merging Eq. 8 and 1, the value of $\frac{\partial I}{\partial \hat{i}_k}$ is then derived as follows:

$$\frac{\partial I}{\partial \hat{i}_k}(x,y) = \cos\theta_k \Big[I(x+1,y) - I(x-1,y) \Big] + \sin\theta_k \Big[I(x,y+1) - I(x,y-1) \Big] \tag{9}$$

As in Eq. 9, the $\frac{\partial I}{\partial \hat{i}_k}$ value represents the magnitude of the gradient projection onto the versor $\hat{i_k}$. It can be computed by convolving an image with a kernel matrix filled with constant elements (θ_k are a-priori defined as well as the $\cos\theta_k$ and $\sin\theta_k$ values).

Among the k computed projections, the $argmax$ function is used to extract the greatest gradient projection as results of magnitude comparisons, as shown in Eq. 10. The extracted $\bar{k}$ projection also represents the closest gradient approximation provided by our method.

$$\bar{k} = argmax \left\{ \frac{\partial I}{\partial \hat{i}_k}(x,y) | k = 0,...,N-1 \right\} \tag{10}$$

as results of Eq. 10, the angle θ of the $\nabla I(x,y)$ vector belongs to the following interval:

$$\theta \in \left[\theta_{\bar{k}} - \frac{\pi}{2N} \quad , \quad \theta_{\bar{k}} + \frac{\pi}{2N} \right) \tag{11}$$

Thus, the HOG-Dot method approximates the exact gradient $\nabla I(x,y)$ as follows:

$$\nabla I_{Dot}(x,y) = \left(\frac{\partial I}{\partial \hat{i}_{\bar{k}}}(x,y) \quad , \quad \theta_{\bar{k}} \right) \tag{12}$$

where $\theta_{\bar{k}} = \bar{k}\pi/N$. According to the original HOG implementation [1] $\nabla I_{HOG}(x,y)$ is expressed as:

$$\nabla I_{HOG}(x,y) = (\| \nabla I(x,y) \|, \theta_{\bar{k}}) \tag{13}$$

Considering now Eq. 12 and Eq. 13, the vector obtained by our method is overlaying the HOG one but their magnitude differs. The *Dot* gradient is indeed representing the projection of the exact gradient over the $\hat{i}_{\bar{k}}$ versor (as in Eq. 14) while the HOG version has the same magnitude as the exact gradient. In Fig. 2 the geometric correspondences between both methods are shown.

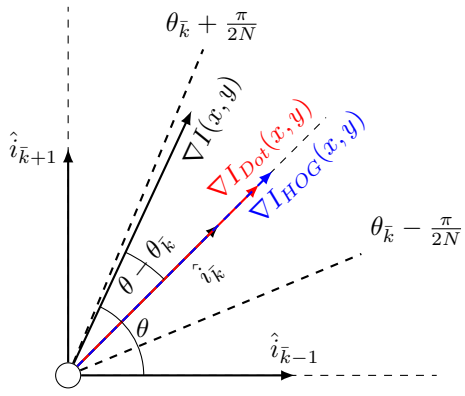


Fig. 2. The dot-product between $\nabla I(x,y)$ and the "closest" $\hat{i_k}$, compared with Dalal et al. technique.

As result of Eq. 4 and Eq. 12, the formal relation between $\nabla I_{HOG}(x,y)$ and $\nabla I_{Dot}(x,y)$ modules is then expressed in Eq. 14. In particular, the $\nabla I_{Dot}(x,y)$ represents a round down approximation of the $\nabla I_{HOG}(x,y)$ gradient.

$$\| \nabla I_{Dot}(x,y) \| = \| \nabla I(x,y) \| \cos(\theta - \theta_{\bar{k}}) \qquad (14)$$

In the following Section, the proposed approximation is quantitatively evaluated with respect to other HOG implementations.

III. Error evaluation

In this section the HOG-Dot gradient is first compared with the original HOG algorithm (Sec. III-A). In Sec. III-B an extended comparison is then shown with respect to the exact gradient formulation. As assessment metric we selected the *Error Vector Magnitude* (*EVM*) [8], an index commonly used in telecommunications to measure the difference between transmitted and received constellation points (distance between reference and real points). In mathematical terms EVM can be expressed as:

$$EVM \quad = \quad \sqrt{\frac{P_{error}}{P_{ref}}} \qquad (15)$$

where P_{error} is the power of the error vector and P_{ref} is the power of the reference measure. For our purposes the EVM can be rewritten as:

$$EVM = \sqrt{\frac{\| e \|^2}{\| \nabla I_{ref}(x,y) \|^2}} = \frac{\| e \|}{\| \nabla I_{ref}(x,y) \|} \qquad (16)$$

where e is $\| \nabla I_{ref}(x,y) - \nabla I_{real}(x,y) \|$, the difference vector between the reference and its approximated version. In the following of the section, the notation EVM_{A_B} is used to specify the reference (A) and the real (B) point considered in the assessment, e.g., EVM_{HOG_Dot} is evaluated between the HOG and the HOG-Dot approximations.

A. HOG-Dot versus HOG

As in Eq. 14 the difference vector between $\nabla I_{HOG}(x,y)$ and $\nabla I_{Dot}(x,y)$ is straightforward evaluated.

$$\begin{aligned} \| e_{HOG_Dot} \| &= \| \nabla I_{HOG}(x,y) - \nabla I_{Dot}(x,y) \| = \\ &= \| \nabla I(x,y) \| \left(1 - cos(\theta - \theta_{\bar{k}})\right) \end{aligned} \qquad (17)$$

Thus, the EVM_{HOG_Dot} is a deterministic function of $\theta - \theta_{\bar{k}}$, and, consequently of N ($\theta_{\bar{k}} = \bar{k}\pi/N$). The EVM_{HOG_Dot} is then described as follows:

$$EVM_{HOG_Dot} \quad = \quad 1 - cos(\theta - (\bar{k}\pi/N)) \qquad (18)$$

The EVM_{HOG_Dot} function is then bounded within the interval $\left[0, 1 - cos(\frac{\pi}{2N})\right)$. It reaches the lower bound value when $\theta = \theta_{\bar{k}}$ and the maximum value when $\theta = \frac{\pi}{2N}$ (the border of the interval). Moreover, since EVM_{HOG_Dot} is function of N, the maximum approximation error can be controlled by fixing an appropriate angle resolution as in Eq. 11. In Fig. 3 the EVM_{HOG_Dot} as function of $\theta - \theta_{\bar{k}}$ is depicted considering two different values of N, 8 and 16.

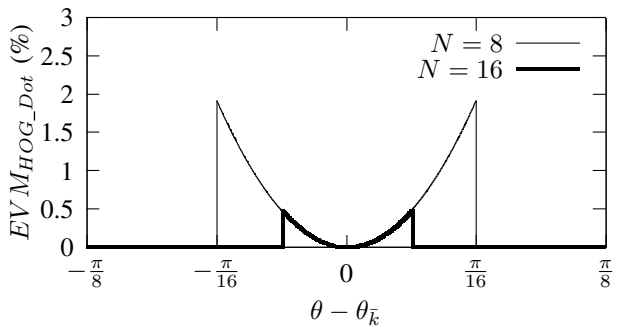


Fig. 3. The EVM_{HOG_Dot} evaluation as function of $\theta - \theta_{\bar{k}}$.

To complete the comparison, maximum and average errors, respectively annotated $max(EVM_{HOG_Dot})$ and $avg(EVM_{HOG_Dot})$, are reported as function of N in Fig. 4. By increasing the angle sampling parameter N, the difference between θ and $\theta_{\bar{k}}$ decreases, thus the maximum EVM decreases as well. Given all values on its period, the average EVM is evaluated as follows:

$$\begin{aligned} avg(EVM_{HOG_Dot}) &= \frac{2}{T} \int_0^{T/2} EVM_{HOG_Dot} \, \mathrm{d}\theta \\ &= 1 - \frac{2N}{\pi} sin(\frac{\pi}{2N}) \end{aligned} \qquad (19)$$

where T is equivalent to the EVM_{HOG_Dot} period $\frac{\pi}{N}$. The average value is clearly bounded by Eq. 18 and decreases when N increases. Although we exploit the gradient extraction with a linear kernel based method, our results are comparable with those obtained by the HOG technique. As shown in Fig. 4, by fixing N equal to 8 as Dalal et al. suggests, the resulting average EVM is close to 0.5%. Whether an improved accuracy is requested, the sampling factor N can be further increased to match the output specifications. The higher is N, the lower EVM_{HOG_Dot} results. As far as N increases, the ∇I projections are closer to the reference ∇I_{HOG} vector, which asymptotically will coincide.

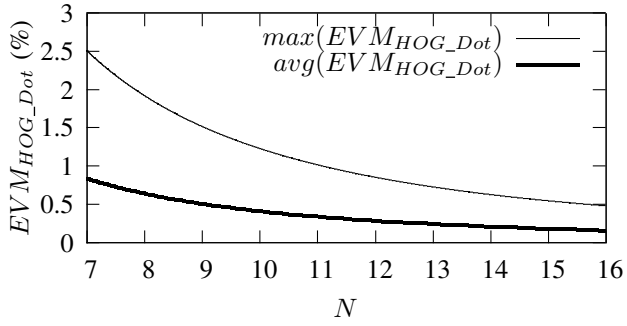


Fig. 4. Maximum and average EVM_{HOG_Dot} as function of N.

B. HOG methods versus ∇I

In Sec. III-A, the HOG-Dot technique has been compared with the original HOG gradient extraction. In this Section

HOG-Dot and other state of the art gradient extraction techniques are compared with the exact gradient formulation ∇I through the EVM metric. To better understand the main differences between the the HOG-Dot and HOG methods, ∇I_{Dot} and ∇I_{HOG} are reported in Fig. 5, in which the exact gradient ∇I is also shown.

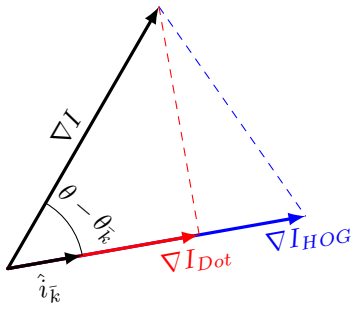


Fig. 5. ∇I, ∇I_{HOG} and ∇I_{Dot} comparisons.

In order to compare the considered methods with respect to the ∇I, in this occasion the EVM has been evaluated with respect to the exact ∇I. Thus, according to Eq. 12, ∇I_{Dot} represents the projection of ∇I along the versor $\hat{i}_{\bar{k}}$ and ∇I_{HOG} overlies the $\hat{i}_{\bar{k}}$ versor with the same ∇I module. The resulting EVM can be expressed as follows:

$$EVM_{\nabla I_Dot} = \|sin(\theta - \theta_{\bar{k}})\| \tag{20}$$

where $EVM_{\nabla I_Dot}$ is the magnitude of the difference vector between ∇I and ∇I_{Dot}. As shown in Fig. 5, the $EVM_{\nabla I_\nabla I_{HOG}}$ is then:

$$EVM_{\nabla I_HOG} = \sqrt{\left(1 - cos(\theta - \theta_{\bar{k}})\right)^2 + sin(\theta - \theta_{\bar{k}})^2} \tag{21}$$

Thus, by applying Eq. 20 to Eq. 21 the final $EVM_{\nabla I_HOG}$ is then obtained.

$$EVM_{\nabla I_HOG} = \sqrt{\left(1 - cos(\theta - \theta_{\bar{k}})\right)^2 + EVM_{\nabla I_Dot}{}^2} \tag{22}$$

According to Eq. 21, the original HOG gradient approximation proposed by Dalal et al. presents an $EVM_{\nabla I_HOG}$ higher than $EVM_{\nabla I_Dot}$. In Lee et al. [6] a gradient approximation has been implemented within an hardware circuitry. In this case the $EVM_{\nabla I_Lee}$ can be expressed as:

$$EVM_{\nabla I_Lee} = \frac{1}{\nabla I}\left(\nabla I - \frac{\|G_x\| + \|G_y\| + \sqrt{2}G_{max}}{1 + \sqrt{2}}\right) \tag{23}$$

where G_x and G_y are the gradient 1-D components as in Eq. 1 and G_{max} is the greater absolute value among them. Finally, the Bhardwaj and al. [3] implementation is considered. In this case, the gradient magnitude is approximated by directly adding the G_x and G_y components. The $EVM_{\nabla I_Bhardwaj}$ is then straightforward evaluated as follows:

$$EVM_{\nabla I_Bhardwaj} = \|cos\theta_k\| + \|sin\theta_k\| - 1 \tag{24}$$

where θ_k is the HOG binning angle as in Eq. 6.

The proposed algorithm is numerically evaluated with respect to the other methods. Simulations have been performed over different datasets (INRIA [9], CVC [10]) by calculating the spatial gradient with the above presented techniques. The maximum, the average and the variance of the EVM, respectively reported as $max(EVM)$, $avg(EVM)$ and $\sigma^2(EVM)$ in Table I, have been evaluated for each method. The comparison with the exact gradient shows a minimum EVM error of 19,5% mainly due to the HOG binning operation which introduces an angular error. With this respect, our proposed solution outperforms the other state of the art techniques. As a matter of comparison, the evaluation has been performed with N equal to 8 for all the considered methods.

TABLE I
SPATIAL GRADIENT ALGORITHMS COMPARISON

	$max(EVM)$	$avg(EVM)$	$\sigma^2(EVM)$
Lee et al. [11]	21.11 %	3.14	24.69
Bhardwaj et al. [3]	45.06 %	7.52	146.27
Dalal et al. [1]	19.59 %	2.60	19.38
Our	19.50 %	2.59	19.26

Nonetheless our method involves only linear computation, the resulting $EVM_{\nabla I_Dot}$ is smaller than others with respect to the ideal gradient. With respect to the method proposed by Lee et al., HOG-Dot achieves better results in terms of the maximum EVM and a reduced average error. With respect to the Bhardwaj et al. method, the HOG-Dot performs significantly better and reduces the maximum EVM by 25%. Finally, the comparison between the Dalal et al. and our technique essentially provides the same EVM performance, with the limited improvement as expected by Eq. 22. Although the proposed linear approximations, HOG-Dot shows better results with respect to the other state of the art techniques. By exploiting only linear computations, HOG-Dot is then suitable for constrained embedded devices, where non-linear, square root and arctan methods will be reducing the processing performance.

IV. Conclusions

In the paper the HOG-Dot technique has been proposed. The HOG-Dot is a kernel-based gradient extraction technique capable to compute directly the polar coordinates of image gradients by linear approximating the exact gradient. As comparison with the other state of the art algorithms, HOG-Dot shows a bounded and controllable approximation error. Even though our method introduces an error with respect to the exact mathematical formulation, it shows good accuracy results, outperforming the other state of the art HOG methods. Moreover, due to its internal parallel architecture, HOG-Dot is suitable for hardware oriented implementations. As future work the HOG-Dot will be implemented in a FPGA-based SC to evaluate the latency performance and the impact on the approximation error within the HOG pipeline.

ACKNOWLEDGMENT

This work has been sponsored by the French government research programme "Investissements d'avenir" through the IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01), by the European Union through the program Regional competitiveness and employment 2007-2013 (ERDF Auvergne region), and by the Auvergne region.

REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, vol. 1, pp. 886–893.

[2] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sept. 1995.

[3] Saket Bhardwaj and Ajay Mittal, "A survey on various edge detector techniques," *Procedia Technology*, vol. 4, no. 0, pp. 220 – 226, 2012.

[4] N. Katic, V. Popovic, R. Cojbasic, A. Schmid, and Y. Leblebici, "A relative imaging cmos image sensor for high dynamic range and high frame-rate machine vision imaging applications," *Sensors Journal, IEEE*, vol. 15, no. 7, pp. 4121–4129, July 2015.

[5] R. Kadota, H. Sugano, M. Hiromoto, H. Ochi, R. Miyamoto, and Y. Nakamura, "Hardware architecture for hog feature extraction," in *Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP '09. Fifth International Conference on*, 2009, pp. 1330–1333.

[6] S. Lee, K. Min, and T. Suh, "Accelerating histograms of oriented gradients descriptor extraction for pedestrian recognition," *Computers and Electrical Engineering*, pp. 1043 – 1048, 2013.

[7] C. Salvadori, M. Petracca, J. Martinez del Rincon, S. Velastin, and D. Makris, "An optimisation of gaussian mixture models for integer processing units," in *Journal of Real-Time Image Processing*. Springer, 2014.

[8] Michael D. McKinley, Kate A. Remley, Maciej Myslinski, J. Stevenson Kenney, Dominique Schreurs, and Bart Nauwelaers, "Evm calculation for broadband modulated signals," in *Proceedings of the 64th ARFTG Conference, December 2004*, 2008, pp. 45–52.

[9] INRIA, "Person dataset," http://pascal.inrialpes.fr/data/human/.

[10] CVC, "CVC-01 Pedestrian Dataset," http://www.cvc.uab.es/adas/site/?q=node/7.

[11] Seonyoung Lee, Haengseon Son, Jong Chan Choi, and Kyoungwon Min, "Hog feature extractor circuit for real-time human and vehicle detection," in *TENCON 2012 - 2012 IEEE Region 10 Conference*, Nov 2012, pp. 1–5.