RetailNet: A deep learning approach for people counting and hot spots detection in retail stores

Valério Nogueira Jr., Hugo Oliveira, José Augusto Silva, Thales Vieira and Krerley Oliveira
Federal University of Alagoas (UFAL), Maceió, AL, Brazil
{vnjr, htmo, jass2, thales}@ic.ufal.br, krerley@im.ufal.br

***Abstract*—Customer behavior analysis is an essential issue for retailers, allowing for optimized store performance, enhanced customer experience, reduced operational costs, and consequently higher profitability. Nevertheless, not much attention has been given to computer vision approaches to automatically extract relevant information from images that could be of great value to retailers. In this paper, we present a low-cost deep learning approach to estimate the number of people in retail stores in real-time and to detect and visualize hot spots. For this purpose, only an inexpensive RGB camera, such as a surveillance camera, is required. To solve the people counting problem, we employ a supervised learning approach based on a Convolutional Neural Network (CNN) regression model. We also present a four channel image representation named RGBP image, composed of the conventional RGB image and an extra binary image P representing whether there is a visible person in each pixel of the image. To extract the latter information, we developed a foreground/background detection method that considers the peculiarities of people behavior in retail stores. The P image is also exploited to detect the hot spots of the store, which can later be visually analyzed. Several experiments were conducted to validate, evaluate and compare our approach using a dataset comprised of videos that were collected from a surveillance camera placed in a real shoe retail store. Results revealed that our approach is sufficiently robust to be used in real world situations and outperforms straightforward CNN approaches.**

I. Introduction

The retail sector comprises a major fraction of the world's developed economies. It is well-known that understanding customer attitudes and behavior is crucial to maximize profit and increase the competitiveness of retail stores [1]. Besides, the impact of effective sales staff scheduling is of critical importance to the profitable operations, since the cost of labor is generally one of the largest expenses of a retailer [2]. Consequently, managing these aspects efficiently has been the focus of research for decades.

In particular, customer's flow analysis [3] and the detection of hot spots, *i.e.* physical areas of a store where people flow is higher [4], are essential in this regard. Nevertheless, it was usually impractical to acquire and analyze an appropriate number of data to accomplish such tasks.

In the last years, advances in hardware and machine learning algorithms based on deep learning allowed researchers to develop outstanding solutions to many Computer Vision tasks, such as object detection and localization [5], outperforming previous state-of-the-art machine learning techniques [6]. However, while many solutions for general classification and localization problems in images using deep models are available in the literature, not much attention has been given to more specific problems, such as accurate people counting in uncrowded environments [7].

In this paper, we present RetailNet: a low-cost deep learning approach to estimate the number of people in retail stores in real-time and to detect hot spots. For this purpose, we require only an inexpensive RGB camera, such as a surveillance camera. In this challenging situation, the resolution of the acquired RGB images can be as low as 1 megapixel. Also, severe occlusion is expected due to furniture and other people inside the store.

It is also worth mentioning that there is still a large gap between the theoretical research and real-world applications to reliably detect humans in various poses and in high interaction scenes, where the body parts of different humans are spatially nearby [8]. State-of-the-art deep neural networks, such as YOLO [5], are not appropriate due to three main reasons: 1) the expected low-resolution images acquired from low-cost surveillance cameras; 2) the expected recognition rate of less than 90% for each object, which is still low when many objects (people) are simultaneously inside the store; and 3) the high variability of customer poses, including extreme poses such as being seated, trying on shoes, and occluded by a chair.

We propose to solve the people counting problem through a supervised learning approach based on a Convolutional Neural Network (CNN) regression model. The CNN is expected not to specifically recognize people, but to learn the density of people in each part of the spatial domain of four channel images named RGBP (red, green, blue, people) images. The RGBP images are composed of: the original RGB channels to provide color information; and an extra binary channel to represent the presence of people, which is computed by detecting the foreground. With this aim, we propose a real-time foreground detection method based on concept drift [9]. Such an algorithm is adapted to the peculiarities of the problem taking into consideration, for example, that the background is not fully characterized by static pixels over time, since salespeople, for example, may stay motionless for long periods of time. We also present an approach to detect the hot spots of retail stores, which exploits the proposed foreground detection method. It is worth emphasizing that our people counting approach relies on single static images only.

To validate our method, we collected several videos from a surveillance camera placed in a real shoe retail store. We performed experiments to tune the hyper-parameters of the

2377-5416/19/$31.00 ©2019 IEEE
DOI 10.1109/SIBGRAPI.2019.00029

proposed CNN regression model and compare with other image representations: RGB images; and RGB images with blacked background pixels. We also present a case study on hot spot detection of the watched retail store and discuss the limitations of our approach.

In summary, the main contribution of this paper is a deep learning approach for people counting, featuring the following components:

- A foreground detection method to recognize people in RGB videos;
- An input image format named RGBP images to simultaneously provide color and foreground information;
- A CNN regression model that learns how to count people from the aforementioned RGBP images, achieving better accuracy to estimate the number of people in images when compared to other CNN regression models trained from conventional RGB or P images (Section VII-B).

II. Related Work

Crowd density estimation and accurate people counting are closely related problems, although existing methods for the former are generally not appropriate for the latter. According to Loy *et al.* [10], methods for crowd counting can be categorized into three groups: counting by detection, counting by clustering and counting by regression.

The counting by detection approach is based on the recognition of each existing individual in the image. Such an approach has been used to detect pedestrians using Histograms of Oriented Gradients (HOG) [11] and shapelets [12] for example. However, this approach is unreliable in crowded scenes where occlusion and scene clutter are frequent. Alternatively, part-based detection methods are more robust in such situations, detecting body parts like heads and shoulders [13]. Still, specific body parts may be hard to detect in low-resolution surveillance cameras, especially when the parts are far from the sensor, and still suffer from occlusion. In addition, accurate people counting relies on the successful detection of each person in the image, which also makes deep learning approaches for object detection [5] unfeasible here, since their accuracy is not sufficient yet.

Clustering-based methods assume that coherent feature trajectories can be grouped together to represent moving entities. Following this approach, Bayesian clustering was employed to group local features [14]. Similarly, a parallelized version of the Kanade-Lucas-Tomasi (KLT) tracker was proposed to extract and cluster a set of feature trajectories [15]. A limitation of such methods is the reliance on spatiotemporal coherence, being inappropriate to count people in single static images. Also, as in the counting by detection approach, failure to recognize a single person would impact the resulting people count.

Regression methods are not prone to the latter problem since they work by globally estimating the crowd density [10]. It is the most extensively used approach for crowd counting, including studies based on simple linear regression models [16], Gaussian processes [17], support vector regression [18] and, more recently, deep learning methods [19]–[21], which are carefully described in a recent survey [22].

However, as discussed in [22], the above-mentioned methods are focused on roughly estimating the number of people in highly dense crowds, for purposes such as outdoor crowd analysis in sporting events, political rallies and public demonstrations. Instead, we aim at accurately estimating the people count in small indoor environments, and particularly in retail stores. It is worth emphasizing that indoor scenes have their own peculiarities, making the counting task more challenging due to: a mix of stationary and moving people, which complicates the foreground detection; and severe occlusion from other people, furniture and walls.

To the best of our knowledge, this is the first attempt to employ deep learning to solve the accurate single image people count problem in indoor environments.

III. Overview

Our approach addresses two relevant computer vision problems for customer behavior analysis: accurate real-time people counting and hot spot detection. As illustrated in Fig. 1, we consider as input low-resolution RGB images (400×225 pixels), which can be acquired from cheap surveillance cameras. Such an image is first quantized and then given as input to a foreground detection method that was developed to properly deal with the peculiarities of people behavior in retail stores.

To perform people count, RGB image data and the estimated foreground information are merged in an image format that we call RGBP images, which are employed in two distinct phases of a supervised learning task: training and real-time prediction. In the training phase, images are first annotated by a trainer according to the number of people, aided by a tool that was developed to allow semi-automatic image annotation. Next, this supervised dataset is used to train a CNN based regression model that is then expected to accurately estimate the people count. In the prediction phase, RGB images acquired from the camera are processed by the foreground detection algorithm. The resulting RGBP image is given as input to the trained CNN, which finally estimates the people count in real-time.

Simultaneously, the detected foregrounds representing people in the image (P channel of the RGBP images) are added to a cumulative image. The accumulated data is then color-coded using a color map, and the resulting heat map may be exploited to visually recognize where people flow is higher (hot spots) from the camera viewpoint.

IV. Foreground detection and RGBP images

Foreground detection or background subtraction is a well-studied problem, commonly used as a preprocessing step in many vision-based tasks. The most straightforward approach is to first acquire a background image of the scene, *i.e.* an image without people in it, and then subtract the new images from the background. Although acquiring such a background image is practical in our case, it is not expected that it remains static, since products and furniture are frequently being moved or reorganized. In addition, shadows and illumination

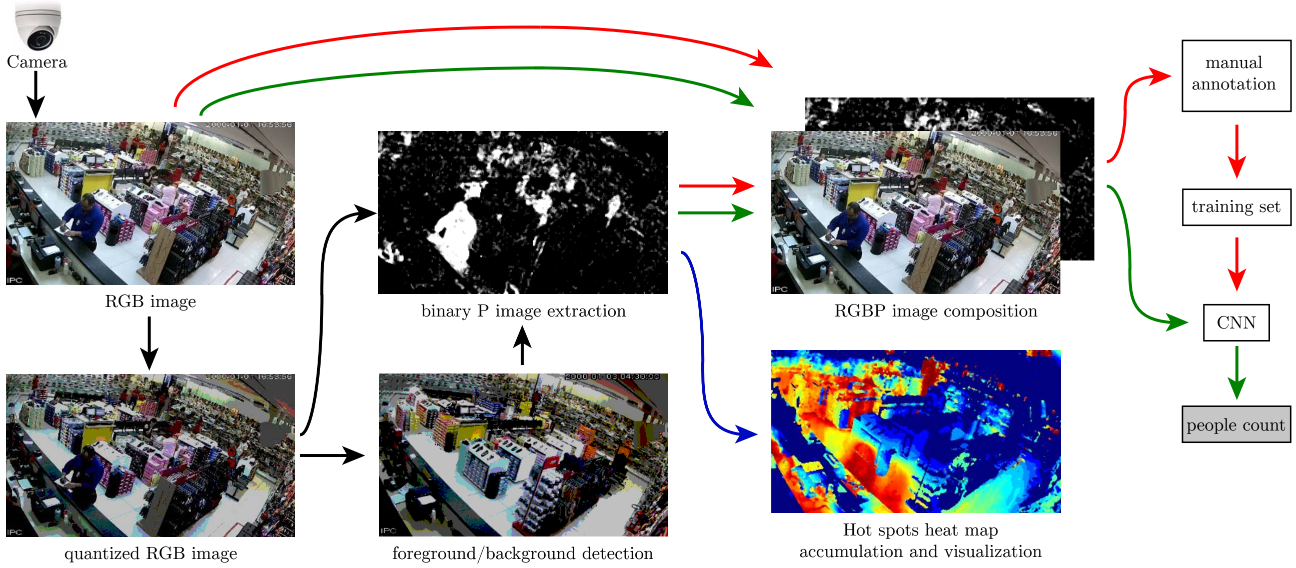

Camera
RGB image
quantized RGB image
binary P image extraction
foreground/background detection
RGBP image composition
Hot spots heat map accumulation and visualization
manual annotation
training set
CNN
people count

Fig. 1. Overview of our approach. An RGB image acquired from a surveillance camera is first quantized. Next, the quantized version is used to initialize or update the foreground/background, and to extract a binary P image representing people. The P image is merged with the original RGB image to compose an RGBP image, which is used to train a CNN model. Then, the trained regression model is expected to accurately estimate the people count. Simultaneously, the P image is also used to update a cumulative image that encodes the hot spots areas and can be visually analyzed as a heat map. The black arrows represent common steps to all phases. The red, green and blue arrows represent training, prediction and hotspots detection steps, respectively.

changes may impact the segmentation quality. Alternatively, another common approach is to analyze motion features to detect static and moving objects (background and foreground, respectively). Unfortunately, in the context of a retail store, it is expected that many people remain static, particularly salespeople and customers browsing or trying on products such as shoes, for example. We refer the reader to [23] for a comprehensive presentation of other existing methods.

To properly handle these issues, we propose a hybrid approach that initializes the background from a few consecutive images, or from an initial background image if practicable, and then continuously updates it.

A. Preprocessing

The first steps to detect the foreground of an acquired RGB image are: downsampling its spatial domain to 400 × 225 pixels, which is the fixed input of the neural network; and then generating an uniformly quantized version $\tilde{\mathcal{I}}$ of the downsampled image $\mathcal{I}$, with λ color levels, to minimize the effect of illumination changes and shadows. We achieved excellent results by empirically setting $\lambda = 64$ (4 levels per channel). The spatial resolution of the downsampled images was empirically chosen by asking a few individuals to count people in challenging images, at different resolutions. The selected resolution was the lowest one in which individuals could still correctly count people. It is also worth mentioning that $\tilde{\mathcal{I}}$ is used only for foreground detection, while $\mathcal{I}$ is used for the CNN training and prediction (jointly with the P channel, described in Section IV-D) since it holds more information.

B. Background initialization

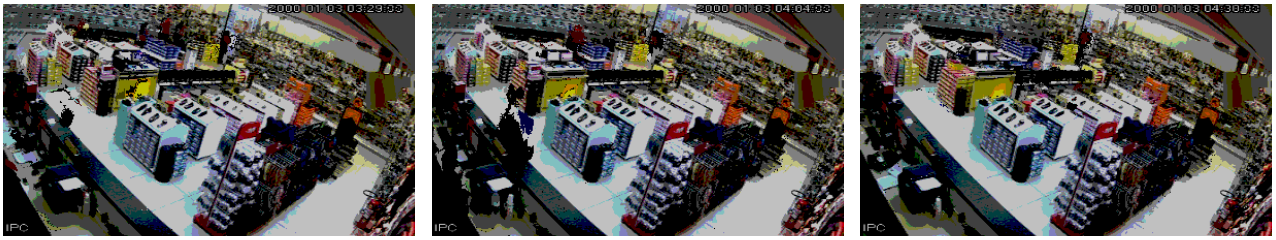
The background may be initialized using one of two approaches: considering a single background image, which may be collected when the store is empty for example; or by accumulating data from a few images in various histograms, one for each pixel, as we describe next.

Let $\mathcal{H}_{ij}$ be the histogram for pixel (i, j), where i ranges from 1 to 225, and j from 1 to 400. The number of bins of each histogram is set to be the number of levels λ of the quantized images $\tilde{\mathcal{I}}$. To compute the histograms, we use a circular image buffer $\mathcal{C}$ of maximum size η. Each time a new image $\mathcal{I}$ is acquired, we first compute a quantized image $\tilde{\mathcal{I}}$, which is then added to $\mathcal{C}$. If $\mathcal{C}$ is full, $\tilde{\mathcal{I}}$ replaces the oldest image $\tilde{\mathcal{I}}_o$ in the buffer. However, before performing this operation, the histograms are firstly updated: for each $i = 1 \ldots 225$ and $j = 1 \ldots 400$, $\mathcal{H}_{ij}[\tilde{\mathcal{I}}_o(i, j)]$ is decreased by one unit, and $\mathcal{H}_{ij}[\tilde{\mathcal{I}}(i, j)]$ is increased by one unit, where $\mathcal{H}_{ij}[\cdot]$ denotes the frequency of bin $\cdot$ of the histogram $\mathcal{H}_{ij}$. Note that $\tilde{\mathcal{I}}(i, j)$ and $\tilde{\mathcal{I}}_o(i, j)$ are quantized RGB values, which must be coherently mapped to unique corresponding bins of the histograms.

To initialize the background, we wait until the buffer $\mathcal{C}$ becomes full (holding η images), and use the histograms data to compute the mode for each pixel. Thus, the initial background image is given by

$$\mathcal{B}(i, j) = \text{mode}(\mathcal{H}_{ij}), \quad i = 1 \ldots 225,\ j = 1 \ldots 400\,. \quad (1)$$

In our experiments, we generated reliable backgrounds by sampling one frame per second from the camera, to fill a buffer of size $\eta = 100$, which implied that the initial background would be available after the first 100 seconds. However,



(a) Incorrect background estimation with many salespeople near the entrance (top center).

(b) The salespeople gradually begin to disappear with more buffered images.

(c) almost correct background with ideal buffer size.

Fig. 2. Analyzing the effects of buffer size in background initialization: with few images on the buffer, some people is still visible (left and middle). The background detection improves with more accumulated images, until reaching a high quality background (right).

these parameters depend on each situation, and particularly on salespeople and customers behavior in the store.

Fig. 2 shows an example of a background initialization with a different number of buffered images (η). In particular, in Fig. 2c there is still a salesman at the entrance. We deduced that this happened because there is always a salesman at the entrance waiting for customers. However, in our experiments, the model succeeded in learning how to handle such an issue by learning from training data that include these challenging examples. In such challenging situations, we believe that the model considered a specific combination of RGB data and the absence of foreground pixels in a specific part of the image.

C. Background updates

As already discussed above, it is not reliable to consider a constant background in time, due to dynamic changes of objects in the scene. Nevertheless, it is not straightforward to update regions of the background using recently acquired images. In particular, salespeople and customers may perform static behaviors for fairly long periods. Consequently, a potential confusion with a moving object in the scene must be considered. Thus, we adopt a more cautious condition to update the background, inspired by the concept drift approach: we try to detect relevant background changes over time, avoiding considering changes caused by illumination and people.

After the background initialization, the circular buffer keeps being fed at the same sampling rate, and the histograms keep being updated. Each time a new image updates the histograms, we independently analyze each histogram. Instead of considering the mode of each $\mathcal{H}_{ij}$ to update the corresponding pixel value $\mathcal{B}(i,j)$, as in Equation (1), we act in a more conservative way according to:

$$\mathcal{B}^t(i,j) = \begin{cases} \text{mode}(\mathcal{H}^t_{ij}), & \text{if } \max(\mathcal{H}^t_{ij}) \geq \tau \cdot \eta \\ \mathcal{B}^{t-1}(i,j), & \text{otherwise,} \end{cases} \tag{2}$$

where $\mathcal{B}^{t-1}$ and $\mathcal{B}^t$ are the previous and the updated background, respectively; $\mathcal{H}^t_{ij}$ is the updated histogram of pixel (i,j); and τ is a threshold set to 0.8 in our experiments, which means that $\mathcal{B}(i,j)$ is updated only if at least 80% of the frequencies of histogram $\mathcal{H}_{ij}$ are concentrated in a single bin. Consequently, erroneous background changes due to static people, for example, are lessened.

D. Composite RGBP images

The continuously updated background is used to detect people in RGB images acquired from the camera in real-time, or from the training set. Given a new image $\mathcal{I}$, we first generate $\tilde{\mathcal{I}}$ using the quantization procedure described in Section IV-A, and then compute the absolute difference from the current background $\mathcal{B}$:

$$\tilde{\mathcal{I}}_d = |\tilde{\mathcal{I}} - \mathcal{B}|\,, \tag{3}$$

The resulting absolute difference image $\mathcal{I}_d$ is then processed to highlight relevant differences, which we expect to be people. With this aim, $\tilde{\mathcal{I}}_d$ is first converted to grayscale, and then it is binarized by thresholding as:

$$P(i,j) = \begin{cases} 1, & \text{if gray}(\tilde{\mathcal{I}}_d(i,j)) > \beta \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where β is a binarization threshold set to 0.1, and gray$(\cdot)$ is a luminance-preserving grayscale conversion function.

Finally, the resulting binary image P is merged with the original RGB image $\mathcal{I}$, to compose a four channel image that we call RGBP image. Such an image combines both the original color information (RGB) and higher level people information (P) represented by pixels with value 1. In the following section, we employ this image representation to train a CNN to predict the people count.

V. Convolutional Neural Network based regression for people counting

We adopt a supervised learning approach to estimate the people count in images by training a CNN based regression model. To train such a model, a large training set comprised of annotated RGB images (acquired from a surveillance camera) is required. Then, we expect the trained CNN to be capable of accurately estimating people count in real-time.

A. CNN training and real-time prediction

At the training phase, a large number of RGBP images showing different numbers of people were used to train a CNN, generating a highly non-linear regression model. To compose the training set, RGB images are first acquired from the surveillance camera and then preprocessed by performing the foreground detection and generating their RGBP representation, as described in Section IV.

We emphasize that the accuracy of the model relies on a diversified training set, comprised of examples of each number of people that is expected to be found in the store. In our experiments, we noticed that the CNN was not capable of extrapolating its prediction to detect people counts higher than the most crowded training images. However, this partial limitation was mitigated by the physical limitations of the store, *i.e.* the maximum possible number of people inside it.

Besides, it is also necessary to annotate the large training set of images by visually counting the number of people shown in each image. As this is an extremely laborious task, we developed a simple semi-automatic visual tool to facilitate the process. First, we consider that the training set is composed of images collected from continuous video. When playing a video, we expect the number of people to smoothly increase or decrease. Thus, the developed tool first statically shows the first frame of the video, and wait until the user annotates it by typing the correct number of people. Then, the video starts playing and the user is required to just press one of two buttons: increase by one unit the number of people, when a new person arrives in the store; or decrease by one unit, when a person leaves the store. By using this tool, the training time is the same as the videos duration.

Finally, in the classification stage, RGB images acquired from the camera are similarly preprocessed, and the resulting RGBP image is given as input to the trained CNN, which predicts, in real-time, the number of people in the image.

B. Architecture

The class of Convolutional Neural Networks for image classification or regression includes models that receive as input any kind of data structured as n-dimensional images; perform convolutions on such data in a few convolutional layers, possibly using max-pooling and dropout; flatten the resulting feature maps into an uni-dimensional array before feeding a few dense layers; and finally reach an output layer with an activation function. Since we are dealing with a regression problem, we employ a linear activation function in the last layer.

In Section VII-A, we report results of experiments conducted to find the best performing architecture among several hyper-parameters values combinations. All investigated networks receive as input an RGBP image with fixed dimensions of 400 by 225 pixels. The input data is filtered by 3 or 4 convolutional layers (C), where each one may apply 8 or 16 filters (F) of size 3×3 or 5×5 (K). Each convolutional layer is always followed by a 2×2 max pooling layer to downsample the feature maps. Then, the last feature maps are flattened into an unidimensional array that is given as input to 1 or 2 dense layers (L), with 16 or 32 hidden units (U) each. In all convolutional and dense layers, we use the rectified linear (*ReLU*) as activation function. Finally, the output layer is composed of a single neuron with linear activation. In practice, we round the CNN prediction to the nearest integer, since the people count is expected to be an integer. The basic structure of the experimented CNNs is shown in Fig. 3.

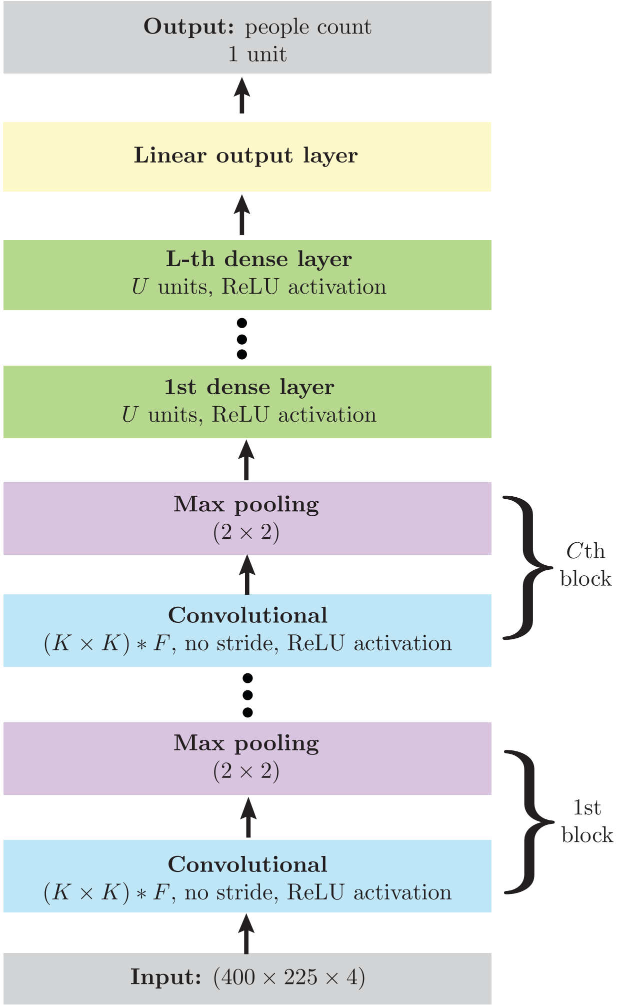


Fig. 3. CNN regression model architecture: the RGBD image is given as input to convolutional blocks composed of 2D convolutional layers (each one applying K filters of size $F \times F$, and 2×2 max pooling layers). Then, flattened features are given to dense layers that precede a linear output layer with a single neuron. The output of the network is the predicted people count, represented as a real number.

VI. Heat Map Generation for Hot Spot Detection

Hot spots are characterized as high-traffic areas within retail stores. We exploit the binary images P, described in Equation (4), in order to detect such places. Since P represents people (foreground), it is feasible to use data from P over the time to identify hot spots, by evaluating the frequency in which each area of the store is occupied by a person. This can be accomplished by evaluating the cumulative image $\mathcal{M}^T$ given by

$$\mathcal{M}^T = \frac{1}{T}\sum_{t=1}^{T} P^t \,, \tag{5}$$

where P^t is the binary image P computed at time t, and T is the last time step analyzed. Thus, regions of $\mathcal{M}^T$ with higher pixel values represent the estimated hot spots.

To visually analyze data in $\mathcal{M}^T$, we first perform the usual histogram equalization procedure on $\mathcal{M}^T$, and then color-code it using a conventional color map, as illustrated in Fig. 7, in which case the Jet color map was applied.

VII. Experiments

To carry out our experiments, we collected several RGB videos from a 1-megapixel surveillance camera placed in a real shoe retail store, which is publicly available[1]. To maximize the diversity of people flow, the videos were collected at 7 different time periods, including distinct times of the day and days of the week. The training set was initially comprised of 153 minutes of video. Next, semi-automatic annotation was performed using the tool described in Section V-A. Due to the high similarity of consecutive frames, we then discarded four out of every five consecutive images. The final training set was comprised of 37,678 annotated RGB images, including images of the store empty, and images with 1 to 30 people, as depicted in Fig. 4.

A. CNN validation and hyper-parameters optimization

The proposed class of CNNs for regression represents a large number of neural networks that are defined by a few hyper-parameters values. Instead of empirically setting values as many previous works, we optimize hyper-parameters by searching over a sample of the search space, as already described in Section V-B. In other words, we performed a grid search, experimenting with various combinations of hyper-parameters values.

Each combination is evaluated through a cross-validation procedure by randomly splitting the dataset examples into training and test sets, considering 75% of the examples for training and 25% for testing. Here, we carefully split training and test sets to avoid similar images from short periods of time to be part of the same subset. With this aim, training and test sets were comprised of images sampled from distinct video recordings. The validation accuracy was evaluated after an early stopping trigger to avoid overfitting. As we are dealing with a regression problem, whose target values are integers, we consider the following error measure to evaluate each combination of hyper-parameters values:

$$\mathcal{E} = \frac{1}{n}\sum_{i=1}^{n}\frac{|t_i - \text{round}(y_i)|}{t_i}\,, \tag{6}$$

where t_i is the ground truth (annotated) people count and y_i is the people count estimated by the CNN, of the i-th image; and round$(\cdot)$ is the function that rounds a number to the nearest integer. When $t_i = 0$, we set the denominator to 1.

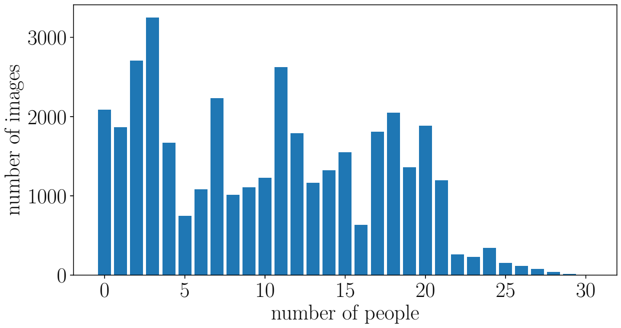


Fig. 4. Number of training images per number of people: images with up to 30 people were collected. Note that fewer examples displaying more than 21 people were acquired, since this was not a common situation.

TABLE I
Configuration and accuracy of the top 8 CNN configurations sorted by $\mathcal{E}$. The best result is shown in bold.

C	F	L	U	K	$\mathcal{E}$ (in %)	MAE	weights
3	8	2	16	5	**10.78%**	1.238	145,641
3	16	1	16	3	10.99%	1.236	324,753
3	16	2	16	3	11.35%	1.285	325,025
3	16	2	32	5	11.58%	1.237	580,817
4	8	2	32	3	11.60%	**1.232**	73,825
3	16	1	16	5	11.76%	1.263	297,105
3	8	1	32	3	11.90%	1.264	321,017
3	16	2	16	5	11.96%	1.286	297,377

We adopted the Adam optimization algorithm [24] of the Keras library [25], with an initial learning rate of 0.001. As $\mathcal{E}$ is not differentiable, we employ the mean absolute error (MAE) as a loss function to train the network. The top 8 best performing configurations according to $\mathcal{E}$ are shown in Table I, achieving $\mathcal{E} = 10.78\%$ for the best configuration, and similar results for the others. Besides, the MAE was roughly 1.2 people for all top configurations, which we consider to be an unimportant error for customer's flow analysis. Such results validate and confirm that our approach allows real-time accurate monitoring of people flow in retail stores.

Additionally, we better analyze the performance of the best CNN: we calculate the number of images that resulted in each absolute error given by

$$\mathcal{A}(t, y) = |t - \text{round}(y)|\,,$$

where t is the ground truth people count and y is the predicted people count. Note that $\mathcal{A}$ is always a non-negative integer.

As depicted in Fig. 5, the majority of images achieved a maximum absolute error of 1 person. In particular, the people count of 41.8% of the test set images was correctly predicted by the best CNN, and less than 8% of the tested images resulted in an absolute error above 2 people. Considering that an error of up to 2 people is negligible for long term people flow analysis, we conclude that our regression model can indeed be employed in real-world retail stores.

To corroborate this conclusion, we evaluated a video composed of 250 consecutive seconds showing a situation where the store was crowded, with an average of 18 people. As

[1] http://www.ic.ufal.br/professor/thales/retailnet/

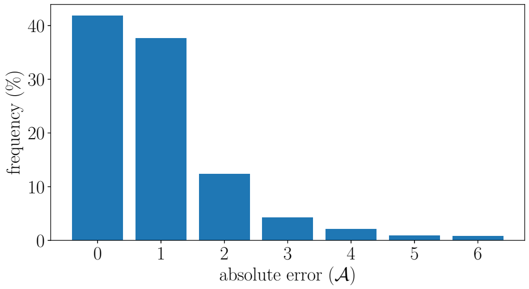


Fig. 5. Percentage of test images per absolute error: most images achieved a correct people count (41.8%) or an inaccurate people count of only one person (37.6%).

depicted in Fig. 6, the CNN achieved high accuracy: almost all predictions attained less than 10% error. By exploiting the temporal coherence of the problem, we believe that even better results could be achieved.

B. Comparison with alternative image representations

In the previous section, we simultaneously validate our approach and compared various CNN configurations. The other relevant component of our approach, which is the foreground detection and its use in the RGBP representation, is compared here with other commonly used image representations.

Similarly to the previous section experiments, we set up here cross-validation experiments that considered three different image representations as input to the best performing CNN configuration (Table I). The following representations were evaluated: *RGB-only* images; *P-only* images; and *RGB-blacked* images, which are the original RGB images with blacked background pixels, as given by P. In addition to comparing the results of these different image representations, some relevant questions can also be answered: 1) is the CNN recognizing people in RGB images? 2) is foreground detection a relevant step to improve people count accuracy? 3) Is the CNN capable of learning to count people from the P image? 4) is the background color information relevant?

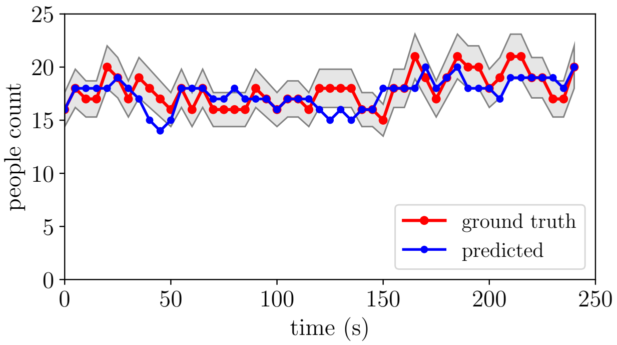


Fig. 6. Ground truth and predicted people count over 250 seconds of a challenging situation with many people: most predicted people count is inside the gray polygon, which represents a 10% relative error tolerance region. Images were evaluated by the CNN every 5 seconds.

TABLE II
COMPARISON OF OUR APPROACH WITH OTHER IMAGE REPRESENTATIONS. THE BEST RESULTS ARE SHOWN IN BOLD.

	image representation			
error	RGB-only	RGB-blacked	P-only	RGBP (ours)
$\mathcal{E}$	37.68%	17.45%	14.80%	**10.78%**
MAE	1.831	1.735	1.676	**1.232**

Such questions are answered by analyzing Table II, which reveals that the RGBP image outperforms the other image representations. The worst results were achieved by learning from RGB-only data, which indicates that raw color data is not the most appropriate kind of information to recognize people. Following, the RGB-blacked representation achieved significantly better results, which is strong evidence that foreground detection is indeed a relevant step. The third question is positively answered by observing the better results of the P-only image. Finally, the best results achieved by our RGBP images imply that: combining both color and foreground information is relevant to improve accuracy and outperforms all other approaches; and background color information is relevant, answering the fourth question (otherwise, the results for *RGB-blacked* would be comparable).

C. Case study on hot spots visualization

In this section, we present a case study conducted to validate our approach for hot spots generation and visualization. Using the same camera setup of the previous experiments, we applied the proposed algorithms to accumulate data of one hour of recording. The widespread Jet color map was chosen to generate images at different moments in time, as shown in Fig. 7, where the red and blue regions of the image indicate the hot and cold spots of the store, respectively.

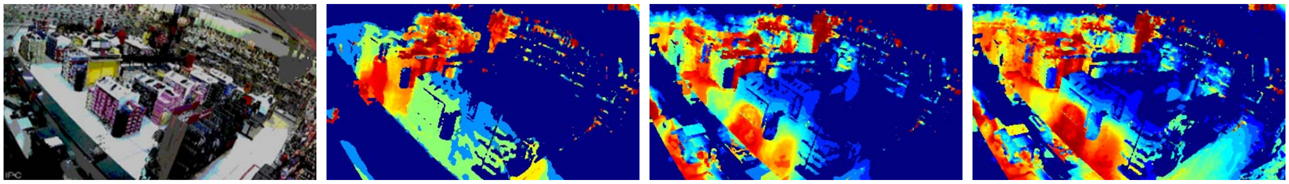
One can see that heat maps gradually change over time. By visually analyzing the last heat map (Fig. 7d), it is possible to extract some interesting information about people flow:

1) the peak of people flow is the entrance of the store, as expected;
2) due to many people remaining at the counter for a long time, that place was considered a hot spot. This information may be exploited by a retailer to position products in that area;
3) there is a hot spot around the chairs used to try shoes;
4) there is not much movement in the central area of the store, suggesting a repositioning of that furniture.
5) the corridors at the right of the image are also not much visited by customers.

It is worth emphasizing that a longer-term analysis would result in more reliable conclusions. Nevertheless, the presented case study was sufficient to validate our approach.

D. Limitations

The main limitations of our approach are related to the CNN training phase. According to the results presented in Section VII-B, it is not expected that a trained model can be



(a) store (b) 5 minutes of recording (c) 30 minutes of recording (d) 60 minutes of recording

Fig. 7. Hot spot detection example: several color-coded visualizations of the equalized cumulative images, where red and blue regions represent hot and cold spots, respectively. In (a), a picture of the store for reference.

applied in different places or viewpoints of the same place. Put differently, training is customized to each place and viewpoint. In particular, the surveillance camera must stand still to avoid viewpoint changes. Fortunately, the proposed annotation tool greatly increases the efficiency of each customized training. The second limitation is related to the high non-linearity of the trained models, which do not support extrapolation. Thus, a well-balanced training set, composed of an appropriate number of examples from each expected number of people, tends to result in better accuracy.

VIII. Conclusion

We presented a low-cost deep learning approach to estimate the number of people in retail stores and detect hot spots in real-time, using only an inexpensive surveillance camera. Several experiments were conducted to validate, evaluate and compare our approach. Results revealed that our approach is sufficiently robust to be used in real world situations, outperforming more conventional approaches.

As future work, we intend to investigate adaptations to detect and exclude salespeople from the people count, in such a way that retailers could better analyze customer-only flow. We also plan to experiment with deep networks that consider the temporal coherence of the problem, since people count gradually changes over time. Another future research topic would be incorporating, into the network architecture, layers capable of automatically detecting the foreground in an integrated manner.

Acknowledgment

The authors would like to thank the Alagoas Research Foundation – FAPEAL (Grant #60030 000421/2017), PIBITI/UFAL and PRMB Comércio e Distribuidora de Calçados LTDA.

References

[1] D. I. Hawkins and D. L. Mothersbaugh, *Consumer behavior: Building marketing strategy*. McGraw-Hill Education, 2015.

[2] S. Lam, M. Vandenbosch, and M. Pearce, "Retail sales force scheduling based on store traffic forecasting," *Journal of Retailing*, vol. 74, no. 1, pp. 61–88, 1998.

[3] Y.-k. Wu, H.-C. Wang, L.-C. Chang, and S.-C. Chou, "Customer's flow analysis in physical retail store," *Procedia Manufacturing*, vol. 3, pp. 3506–3513, 2015.

[4] A. J. Newman and G. R. Foxall, "In-store customer behaviour in the fashion sector: some emerging methodological and theoretical directions," *International Journal of Retail & Distribution Management*, vol. 31, no. 11, pp. 591–600, 2003.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, June 2016.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[7] A. Schofield, P. Mehta, and T. J. Stonham, "A system for counting people in video images using neural networks to identify the background scene," *Pattern Recognition*, vol. 29, no. 8, pp. 1421–1428, 1996.

[8] D. T. Nguyen, W. Li, and P. O. Ogunbona, "Human detection from images and videos: A survey," *Pattern Recognition*, vol. 51, pp. 148–175, 2016.

[9] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine learning*, vol. 23, no. 1, pp. 69–101, 1996.

[10] C. C. Loy, K. Chen, S. Gong, and T. Xiang, "Crowd counting and profiling: Methodology and evaluation," in *Modeling, simulation and visual analysis of crowds*. Springer, 2013, pp. 347–382.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE Computer Society, 2005, pp. 886–893.

[12] P. Sabzmeydani and G. Mori, "Detecting pedestrians by learning shapelet features," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[14] G. J. Brostow and R. Cipolla, "Unsupervised bayesian detection of independent motion in crowds," in *CVPR*, vol. 1. IEEE, 2006, pp. 594–601.

[15] V. Rabaud and S. Belongie, "Counting crowded moving objects," in *CVPR*, vol. 1. IEEE, 2006, pp. 705–711.

[16] A. C. Davies, Jia Hong Yin, and S. A. Velastin, "Crowd monitoring using image processing," *Electronics Communication Engineering Journal*, vol. 7, no. 1, pp. 37–47, Feb 1995.

[17] A. B. Chan and N. Vasconcelos, "Counting people with low-level features and bayesian regression," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2160–2177, April 2012.

[18] D. Conte, P. Foggia, G. Percannella, and M. Vento, "A method based on the indirect approach for counting people in crowded scenes," in *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Aug 2010, pp. 111–118.

[19] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *CVPR*, June 2015.

[20] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *CVPR*, 2016.

[21] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 2016, pp. 640–644.

[22] V. A. Sindagi and V. M. Patel, "A survey of recent advances in cnn-based single image crowd counting and density estimation," *Pattern Recognition Letters*, vol. 107, pp. 3–16, 2018.

[23] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer science review*, vol. 11, pp. 31–66, 2014.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] F. Chollet *et al.*, "Keras," https://keras.io, 2015.