

This is a postprint version of the published document at:

Sourtzinou, P., Velastin, S.A., Jara, M., Zegers, P. y Makris, D. (2016). People Counting in Videos by Fusing Temporal Cues from Spatial Context-Aware Convolutional Neural Networks. In *European Conference on Computer Vision 2016 Workshops*, Part II, LNCS 9914, pp. 655–667.

DOI: https://doi.org/10.1007/978-3-319-48881-3_46

People Counting in Videos by Fusing Temporal Cues from Spatial Context-Aware Convolutional Neural Networks

Panos Sourtzinis¹, Sergio A. Velastin^{2(✉)}, Miguel Jara³,
Pablo Zegers⁴, and Dimitrios Makris¹

¹ School of Computer Science and Mathematics,
Kingston University, Kingston, UK
psourt@gmail.com, d.makris@kingston.ac.uk

² Department of Computer Science,
Universidad Carlos III de Madrid, Getafe, Spain
sergio.velastin@ieee.org

³ Departamento de Informática,
Universidad de Santiago de Chile, Santiago, Chile
miguel.jara.rodriguez@gmail.com

⁴ Faculty of Engineering and Applied Sciences,
Universidad de los Andes, Santiago, Chile
pablozegers@gmail.com

Abstract. We present an efficient method for people counting in video sequences from fixed cameras by utilising the responses of spatially context-aware convolutional neural networks (CNN) in the temporal domain. For stationary cameras, the background information remains fairly static, while foreground characteristics, such as size and orientation may depend on their image location, thus the use of whole frames for training a CNN improves the differentiation between background and foreground pixels. Foreground density representing the presence of people in the environment can then be associated with people counts. Moreover the fusion, of the responses of count estimations, in the temporal domain, can further enhance the accuracy of the final count. Our methodology was tested using the publicly available Mall dataset and achieved a mean deviation error of 0.091.

Keywords: People counting · Convolutional neural networks · Video analysis

1 Introduction

Counting people can provide useful information for monitoring purposes in public areas, assist urban planners in designing more efficient environments, provide cues for situations that might endanger the safety of civilians, and also be used by shopping mall and retail store managers for evaluating their business practices. In principle, such knowledge can be obtained by analysing image and video footage from location specific cameras with the goal to measure the number of people in them. For this reason in this work we present an efficient method for counting people in images and video

sequences, from fixed cameras which incorporates the fusion of context aware cues from CNN in the temporal domain.

People counting is a very challenging problem, and although commercial solutions exist, these focus mainly in top-view cameras, where occlusions between people are minimal. An effective approach is to detect the heads of the pedestrians present in an image, since they are less prone to disappear in the image through occlusions, and then sum the head detections to measure the total count. Such an approach seems consistent to how humans would approach the problem, as implied by expressions such as ‘headcount’. Furthermore since our interest is in measuring the count of people using stationary cameras, where background is assumed fairly static, a local context-aware detector that is spatially tuned to distinguish foreground objects (e.g. heads) from the background scene is more promising than a general-purpose detector.

The main contribution of this work is the proposal of a convolutional neural network (CNN) that uses global image information, rather than cropped images, for people counting and the use of temporal coherence for enhancing the precision in the obtained results. Feeding the CNN with whole images allows modelling of the local context, i.e. the expected local appearance (e.g. size, orientation) of the foreground pedestrian heads and the spatial distribution of pixel luminance in the background. The output of the CNN for each frame is an intermediate density map and head counts are estimated using regression. Temporal coherence is exploited by refined regression of count estimations from multiple frames. In Sect. 2 a background study on the methods for people counting is presented, while in Sect. 3 the methodology of our approach is described. Finally in Sect. 4 the results and a critical discussion of our methodology are given followed by the concluding section in Sect. 5.

2 Previous Work

Counting methods can be mainly categorised into two groups. Counting by detection and counting by regression. In the former case, human shape models are used to localise people on the image plane, while the latter is based on the relationship between a distribution of low level features in the whole image and the number of people in it. Hybrid methods combine these two approaches, i.e. a person detector is used to create a footprint on a distribution describing the whole image, which then is used to infer the number of people in it. The use of CNNs for the task of people counting is by its nature such an approach. In the following sections we identify some methods, but as the literature on the topic is extensive, space limitations prevent us from giving a fuller review.

In counting by detection [16] the idea is to detect the presence of people in an image and then sum the detections to produce the final count. People detections is achieved by object detectors (whole or part-based), based on learned models that use features such as histogram of oriented gradients (HOG), poselets, edgelets and others which describe a shape model of a human body using pixel information. Traditionally, a location invariant object detector is applied using a sliding window technique followed by non-maximal suppression to localise the objects of interest.

In [16] a HOG detector is used to create a probability distribution over the image. To deal with occlusions, the HOG detector is trained to learn only the upper part of the human body. Next the optical flow between two consecutive frames is computed. Assuming that the upper human body exhibits a uniform motion in contrast with the motion generated from the limbs, a mask resembling the shape of upper human body, is scanned through the optical flow response and a probability distribution of uniform motion is computed. The probability distributions learned from the shape model and the uniform motion model are then combined and the fused probability distribution is searched, using Mean Shift Mode Estimation to localise head detections.

A pitfall of using counting by detection techniques is that they do not perform well in images with low resolution, since objects, in these, appear small and they do not generate enough information in order to be detected. Moreover, since most of these approaches use a sliding window to scan the whole image multiple times in different scales, they are computational heavy and thus slow.

In counting by regression [1, 11, 13], a mapping from some low level image characteristics, like edges or corners, to the number of objects is estimated using machine learning methods. Although this approach avoids the hard task of object detection, ambiguities may arise from the presence of objects of other classes that may also generate responses. Furthermore since some of these methods are location-invariant, the training phase requires large amount of data, in order to cover all the possible perspective nonlinearities of the image plane. On the other hand, annotating the ground truth data is simpler as it only involves manual counting.

In [13] the main idea is that integrals of density functions over pixel grids should match the object counts in an image. It is assumed that each pixel is characterised by a discretized feature vector and the training data are dot annotated (e.g. torso). Each annotated pixel is then characterised, using a randomised tree approach, by a feature descriptor combining the modalities of the actual image, the difference image and the foreground image. For each pixel, a linear transformation of its feature descriptor is learned, using a random forest to match the ground truth density function.

In [11] a mixture of Gaussians is initially applied to extract foreground information. Histograms of the area of the foreground blobs and edge orientation are then used as features to describe the image. Finally a feed forward back propagation neural network is used with the histograms of the normalised features as inputs, learning the number of pedestrians in the image.

In [1] a method for counting people using the Harris corner detector is presented. Motion vectors are used to differentiate between static and moving corners. Assuming that each person in the image generates the same amount of moving corners, the number of people in a frame is therefore computed based on the ratio of the moving corners detected over the average number of corners per person. As a consequence, this approach fails to recognise static people. Also the camera perspective effect is not taken into consideration which could invalidate the regression assumption.

A drawback of all regression approaches is that they cannot discriminate well between intra class variations (i.e. differences in human sizes, humans carrying objects, humans with bicycles etc.) and since they lack learning object shape models, they are unable to differentiate between interclass (e.g. animals) differences. Thus their application is mostly location specific.

Hybrid methods [7, 10, 17, 20] aim to combine the benefits of both approaches by fusing their techniques. For instance, in [17] a density image is computed where each pixel value defines the confidence output of the person detector used in [13]. This value is then discretized and represented by a binary feature vector. SIFT features are extracted from the image in order to compute another binary feature vector. The concatenation of the two binary feature vectors is then used to describe each pixel, and by minimizing the regularised MESA distance, the weight of each discretized feature is learned. The density of each pixel is thus calculated by multiplying its feature descriptor with the learned weight vector, and the count of people in the image is then estimated by the integral of the density of the image.

Another example of a hybrid approach is presented in [10] that copes with crowded situations. A Gaussian mixture model is initially applied on a grayscale video sequence to obtain the foreground information. After perspective correction this is further processed using a closing operation. Counting then becomes a problem of finding a relationship between the number of foreground pixels and the number of humans present in the image, a relationship which is learned using a neural network.

Finally two hybrid approaches [7, 20] are the only ones, as far as we know, that use CNN purely for people counting. Both attempt to exploit the CNN characteristic of the spatial invariance in the detection of patterns, and thus the networks described are trained as human detectors by using spatial crops from whole images for training. In [7] a CNN learns to estimate the density of people in an image by using cropped images from the full resolution training dataset. The trained network is then applied to the whole image information to produce a density map of human presence and moreover its parameters are transferred to two similar networks that are applied on different resolutions of the global image. The response from the three networks is then averaged to produce a final density map. To count the number of people in the density image, each point of the density estimated is fed to a linear regression node. The weights then of the node are learned independently for the density estimation. In [20] cropped images are also used for training, however the learning of the density and the total count is not sequential, but takes place in parallel. Both the density map and the linear regression node are connected to the same CNN and learning takes place by altering the cost function between the one used for the density estimation and the one used for count estimation.

3 Method

Deep learning machines have addressed many problems that were deemed as unsolvable in a surprisingly easy way. However, most of the research has focused on the use of static architectures ignoring relevant dynamics aspects of some of the problems. This is especially true in video analytics, where analysis is mainly frame-based, and traditionally the information obtained from each of them has been integrated using some heuristic-based algorithm. This has been recognized by many in the field and many recent publications extended and complemented the convolutional neural network (CNN) architecture into the time domain achieving good results (e.g. [4, 8, 9, 19]). Our work explores how to use time cues in an efficient manner, therefore we avoided

recurrent neural networks or other time domain architectures. Specifically, three replicated CNNs are used to process consecutive time frames and their combined output is fed into a final layer to produce the final estimate. Our approach, following the methodology proposed in [7], first generates a density map to indicate the presence of humans in the image followed by learning the regression relationship between the distribution of activations and the actual count number. The proposed architecture is shown in Fig. 1 and it uses the outcomes of three different instants in time learning the relationship between them to produce the desired results. This is a generalization of averaging the three results. The pipelines are identical in their parameters settings and only one is needed to be trained to reproduce the others. For more information on convolutional layers and their structure the reader is referred to [15].

The performance of a supervised neural network is dependent mainly on three factors: (a) the input data, (b) the network’s architecture and its parameters and (c) the ground truth data. Appropriate representation of the input can lead to better and faster learning of the network [15]. In our case the input layer of a single pipeline is an RGB image of size 240×320 pixels. Every frame, is pre-processed by initially calculating the mean in all training images and subtracting it from all the pixels, before entering the network. Then data is centred around zero in all dimensions and scaling in values between -1 and 1 is performed applying Eq. 1 on each pixel:

$$p_{x,y,c,s} = 2 \cdot \frac{p_{x,y,c,m} - \min(f_i)}{\max(f_i) - \min(f_i)} - 1 \quad (1)$$

where $p_{x,y,c,m}$ is the pixel value of frame f_i at location x,y of channel c , after the mean subtraction and $p_{x,y,c,s}$ is the pixel value after the scaling which we will refer as $p_{x,y,c}$. The data is zero centred to facilitate learning of the network and specifically for the gradient descent algorithm to avoid zigzagging while minimising the cost of the network.

3.1 Density Estimation

The density learning pipeline (Fig. 1) is comprised of four convolutional layers followed by a fully connected one. For the convolutional part of the density estimation pipeline, C_1 has 15 features of size 316×236 , C_2 has 10 features of size 154×114 , C_3 has 20 features of size 73×53 and finally C_4 has 10 features of size 33×23 . The detection kernel of all convolutional layers is 5×5 with a stride of 1 and the feature activations, except from those of C_4 , are max pooled with a kernel of shape 2×2 and stride of 2; thus halving each dimensionality of a feature before feeding it as an input to a subsequent convolutional layer.

In contrast to [7], where all activations in a feature share the same bias, in our case each feature activation is characterised from its own bias. Since the input is the whole image, the network is allowed to further tune the importance of a feature to a spatial location. Following the notation of Eq. 1 the activation function applied for a neuron belonging to a feature f in the proposed CNN is the hyperbolic tangent given by

$$a_{f,j,k} = \frac{1 - e^{-2 \cdot z}}{1 + e^{-2 \cdot z}} \quad (2)$$

$$\text{where } z = b_{f,j,k} + \sum_{f=1}^4 \sum_{l=0}^4 \sum_{m=0}^4 v_{f,l,m} \cdot a_{f-1,j+l,k+m} \quad (3)$$

where the leftmost summation sums over all the features present in the previous layer (as mentioned before, in the case where the previous layer is the input image, each channel of the image represents one feature). The last layer of the density estimation pipeline is a fully connected one (F_1 in Eq. 1) and has as many neurons as there are present in one feature of the previous layer (i.e. C_4). Each neuron in F_1 is connected to all the neurons present in C_4 and thus the weight vector v_i of each neuron i has 7590 ($33 \times 23 \times 10$) dimensions. The activation function used for each neuron of this layer is the sigmoid thus Eqs. (4) and (5) apply.

$$a_i = \frac{1}{1 + e^z} \quad (4)$$

$$z = b_i + \sum_{f=1}^{10} \sum_{l=1}^{33} \sum_{m=1}^{23} v_{i,r} \cdot a_{f,l,m}, r = 759 \cdot (f - 1) + 23 \cdot (l - 1) + m \quad (5)$$

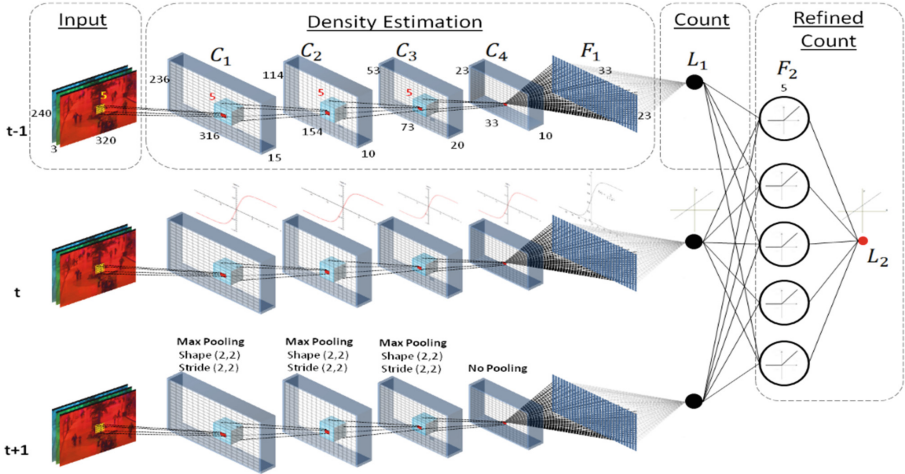


Fig. 1. The proposed architecture for pedestrian counting. In the left we can see the temporal data input in a form of consecutive in time RGB frames, while for the density estimation a pipeline with 4 convolutional layers followed by a full connected sigmoid layer having the task to produce the density images. For the count of a single pipeline a linear regression unit combines the 759 inputs to produce a final result. Finally by combining the results from the counts of 3 pipelines in full connected rectifier layer we feed a node to perform linear regression and produce the final result.

F_1 is the last layer in our density estimation pipeline and the 33×23 responses a_i of the layer are compared against the equivalent y_i of a ground truth density of same dimensionality to measure the error that will be back propagated for the learning. The cost function we use for the comparison is the Kullback–Leibler divergence shown in Eq. 6, and the error produced is the mean cost across all the examples seen.

$$KL(y_i||a_i) = y_i \cdot \log \frac{y_i}{a_i} + (1 - y_i) \cdot \log \frac{1 - y_i}{1 - a_i} \quad (6)$$

3.2 Counting

The final layer of each pipeline is dedicated to estimate the relationship between this density and the actual count of people. So, a single linear neuron (L_1 in Fig. 1) is fully connected with the sigmoid neurons of F_1 . Learning is performed by linear regression using the mean square error across a number of examples as cost function. Thus if a_i denotes the i th activation from layer F_1 and v_i the entry in the weight vector of L_1 associated with a_i , and b_c the bias and a_c activation value of L_1 then,

$$a_c = b_c + \sum_i v_i \cdot a_i \quad (7)$$

and the cost for a single example, when y is the ground truth count, is given by $(a_c - y)^2$.

3.3 Refined Counting

The accuracy of people counting, is further improved by fusing measurements from networks operating on subsequent frames along the temporal dimension. Hence, three pipelines operating on frames with timestamps $t - 1$, t and $t + 1$ are fully connected to a vector of five rectified linear units. Each rectified neuron has as activation function similar to Eq. 7 with the only difference that negative values, produced by the summation of the weighted input with the bias, produce a zero output. Finally, all five outputs from the rectified linear units are connected to the linear neuron L_2 for the refined count. The only difference in the linear regression performed in this neuron compared to the one in L_1 is the cost function, since for this we use the absolute difference of the estimated count against the ground truth.

4 Results

The network described earlier was implemented using Python and the pylearn2 and theano machine learning libraries [2, 3]. For our experiments we used the publicly available Mall crowd counting dataset [5, 6, 14], of which a couple of illustrative frames are shown in Fig. 2(a)-(b).



Fig. 2. By measuring the size of people in different time frames (a), (b), the perspective map denoting the relative scale of pixels in the real word dimension.

The dataset consists of 2000 time consecutive frames recorded by a fixed camera in an indoor shopping mall in 640×480 resolution with a frame rate around 2 Hz. Over 60,000 pedestrians are annotated, with a point indicating their head location. It is a challenging dataset with constant movement, where pedestrians wander freely, alone or in groups, forming a cluttered environment with many occlusions. Moreover, reflections occur in both the shop windows and the floor, the lighting conditions change, and the viewing angle of the camera causes pedestrians to vary in scale.

We also implemented the only two other methods, that to our knowledge, [7, 20] perform people counting using CNNs. As in our case, the three main pipelines of the architecture in [7] are identical in their configuration and in their parameter settings. However they apply each pipeline at different scales of the images in order to infuse scale invariance in their network. To train a single pipeline they use cropped images thus aiming to get a location invariant person detector, which is then applied to the whole image for density estimation. Since the scale of the input images in the pipelines is different, they use one bias per feature in contrast to our approach where every node in a feature is associated with a single bias. Each pipeline estimates a human density and their average merge layer merges the three different density estimations into one followed by a linear regression node for the count estimation.

The second method we implemented is the one presented in [20]. In this approach, similarly to [7], cropped images are used for training, however the learned network is applied on the whole image in a sliding window fashion, where each detection window generates a local density. The density estimate for the whole image is calculated by creating a mosaic from the local ones. Instead of learning a density and then performing a linear regression to estimate the count, the training of the density and the counting takes place in an alternate way. The layers are alternated until neither cost is further improved. To learn the density estimation, the ground truth consists of a density image created from the responses of a Gaussian distribution, centred at the head of a person, and a bivariate normal distribution, placed at the body of the person. The combined distributions describing a person are then normalised to add up to one. Then, counting is just a summation of the entries in the ground truth density image.

The head regions were represented by squares centred at the annotation points and size consistent to the perspective map of the scene. Pixels that belong to head regions have a value of 1, while all other pixels have a value of 0. Since the density estimation

resolution in our pipeline is 33×23 , the generated binary images of 640×480 were scaled down and each image was normalised to have values in the range between zero and one. For [7], the ground truth was based on cropped images of size 320×240 from the original 640×480 binary images created in the previous step, scaled to a resolution of 33×23 and normalised with values between 0 and 1. For [20] the ground truth density images were generated using a Gaussian kernel summing to one, centred at each annotation point and with a standard deviation based on the values of the perspective map of the dataset. Crops of size 72×72 from the 640×480 density images were then extracted and scaled down to size 18×18 . Figure 3 shows some examples.

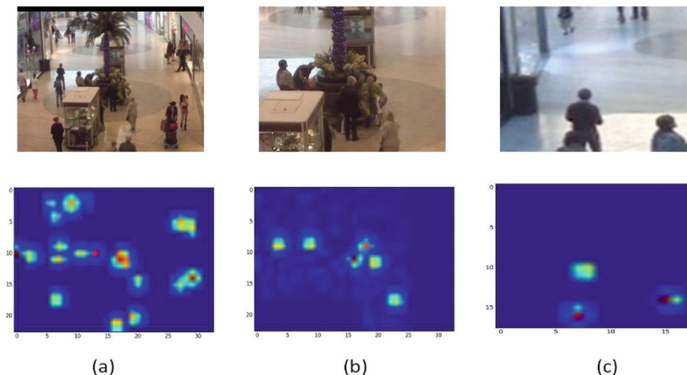


Fig. 3. Examples of training input images (upper row) generated from the same frame, to the three different networks and their associated ground truth (lower row) for density estimation.(a) resample whole frame 320×240 used by our approach, (b) cropped images 320×240 used by [7], (c) cropped images 72×72 used by [20]

From the 2000 frames of the dataset, 1000 were used for training 250 for validation and 750 for testing. For [7] we used 5 cropped images (size 320×240) per training whole image (640×480), while for [20] we extracted 50 cropped images. The input image resolution we used to test out methodology is 320×240 .

Training a CNN requires fine tuning of various parameters. However some of the training parameters were kept constant through all the experiments. The dropout rate was fixed to 0.5 for all layers. This means that during training each node has 50 % chance to be activated, and its parameters to get updated, which assists for regularisation and thus avoiding overfitting the network parameters to the training dataset. Another parameter we kept constant was pooling, by always using the same pooling kernel with same stride. Also all weights were initialised using a uniform distribution and with range $(-0.05, 0.05)$. Other parameters however, such as the learning rate, the use of momentum, the maximum norm of the weight vectors were selected separately for each experiment by testing their impact on the learning behaviour of a network on small subset of the training dataset. The algorithm used for the training was stochastic gradient descent with mini batches. Thus the update of the network parameters

occurred regularly and not at the end of each epoch (i.e. estimating the cost after seeing all training data once).

Figure 4 shows density estimation results from the different methods. Our approach manages to describe the distribution of the pedestrians quite well. In contrast, the responses from [7] are not descriptive, since it appears that although there is a change in the density estimation from frame to frame it follows a general pattern, and it seems that the network failed to learn the people’s density. Also the density results derived by [20], although more descriptive regarding the presence and the position of the pedestrians in the space than [7], still generates many false positives.

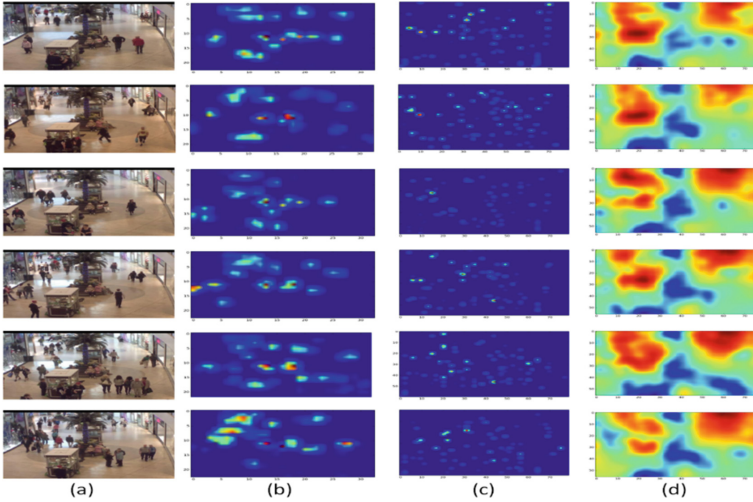


Fig. 4. Density estimation results. (a) Input frame, (b) The response from our approach, (c) The response from [20], (d) The response from [7]

Let’s consider the number of parameters in the configuration of each network. The total number of free parameters for learning in the network of [6] is 14,930. While for the one in [20] the number of parameters that are available for learning is 21,373,532. In our proposed network the number of parameters is 5,871,954. Finally the difference between our approach and the other two is that we use for training input whole images while they use cropped images.

Based on the information provided above, our assumption is that the method of [7], not only has too few parameters to offer a reliable solution to the problem, but also because it lacks any fully connected layer, no information is exchanged between the nodes that can result to a combination of features detected. On the other hand the approach in [20], has a plethora of parameters to adjust and to solve the problem of detecting people in an image, and furthermore they exchange node information by using fully connected layers. However by using cropped images as input it does not provide any spatial localised information that would facilitate learning the presence of the background in the whole image.

Our proposed network, with almost a quarter of parameters compared to [20], assumes whole input images, combines the information from the various nodes of the detectors and therefore it can learn localised background/foreground information. In other words, if our task was to find a fly on a wall, the approach in [7] scans the wall to find the fly with a lens that makes things to appear very blurry and the presence of the fly is diffused on the wall, while the one in [20] scans the wall with a lens that can see every little detail, thus some irrelevant complex patterns of the wall may confuse it. In contrast to the other two methods, our approach instead of scanning the wall, it just subtracts it and observes the difference.

After the density is estimated, the next step is to perform the counting. The mean deviation error (MDE) ε of the counting step,

$$\varepsilon = \frac{1}{N} \cdot \sum_N \frac{|y - \bar{y}|}{y} \quad (8)$$

where y is the ground truth, \bar{y} is the estimated count and N the number of images in the test dataset, is shown on Table 1.

As expected from the resulting density images, the error of the two competitive methods is relatively high. For [7] the linear regression is unable to learn the proper relationship between density and the count number. Even the approach in [20], which estimates the count by summing up all the responses from the density map, the counting error is significant. The counting error can further be reduced, by combining temporal information to remove noise from the measurements. Specifically, the combination of three pipelines with input frames at $t - 1$, t and $t + 1$ in order to estimate the count of frame t generates a mean relative error of 0.091. Table 2 presents results obtained by combining information from varying number of frames (one pipeline per frame is used) using the MDE and the Mean Squared Error (MSE) and the Mean Absolute Error (MAE). Considering the frame rate of the MALL dataset (2fps), optimal coherence is achieved in a temporal window of 1 s, i.e. using 3 consecutive frames. For videos with higher frame rate, optimal performance would be achieved by using more frames.

Table 3 compares our method with other non-CNN approaches for people counting performed in the MALL dataset. Our approach seems to perform similarly with other people counting methods.

Table 1. Mean Deviation Error for Counting.

<u>Approach</u>	<u>MDE</u>
Ours	0.094
Method presented in [20]	0.770
Method presented in [7]	0.230

Table 2. Comparison of varying number of pipelines, where one pipeline per frame is used.

Number of pipelines	MAE	MSE	MDE
1	3.15	16.9	0.093
3	3.00	15.7	0.091
5	3.77	23.8	0.109
7	5.91	46.6	0.200

Table 3. Comparison with other non-CNN methods in the Mall dataset

Method	MAE	MSE	MDE
CHEN_1 [6]	3.59	19.0	0.110
CHEN_2 [5]	3.43	17.7	0.105
LOY [14]	–	17.8	–
ZHANG [21]	2.69	12.1	0.082
KUMAGAI [12]	2.89	13.4	0.091
PHAM [18]	2.50	10.0	0.080
OURS	3.00	15.7	0.091

5 Conclusion

In this work a methodology using CNN was presented for people counting. We have demonstrated that using the whole image information as training input instead of using cropped images, performs better as the network is able to learn how to distinguish between the foreground and the background. Furthermore by fusing the count estimate in the temporal domain, count estimations are further improved. To the best to our knowledge, our method is the first to propose the application of a CNN on the whole image for the task of people counting and furthermore to use temporal information for the same task. Possible future lines of research may include to minimise the information theoretical measure instead of the Euclidean error in order to take into account the probabilistic nature of the problem. Moreover network architectures that utilise recurrent nodes can be used to take advantage of their application in the temporal domain, but also the use of other CNN architectures which incorporate temporal features, such as optical flow, can be investigated.

References

1. Albiol, A., Silla, M.J., Albiol, A., Mossi, J.E.M.: Video analysis using corner motion statistics. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, pp. 31–38 (2009)
2. Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., Bengio, Y.: Theano: new features and speed improvements. arXiv preprint arXiv:1211.5590 (2012)

3. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: a CPU and GPU math expression compiler. In: Proceedings of the Python for scientific computing conference (SciPy), vol. 4, p. 3 (2010)
4. Byeon, W., Breuel, T. M., Raue, F., Liwicki, M.: Scene labeling with lstm recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3547–3555 (2015)
5. Chen, K., Gong, S., Xiang, T., Change Loy, C.: Cumulative attribute space for age and crowd density estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2467–2474 (2013)
6. Chen, K., Loy, C.C., Gong, S., Xiang, T.: Feature Mining for Localised Crowd Counting. In: BMVC, vol. 1, no. 2, p. 3 (2012)
7. Conti, F., Pullini, A., Benini, L.: Brain-inspired classroom occupancy monitoring on a low-power mobile platform. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2014, pp. 624–629. IEEE (2014)
8. Gkioxari, G., Malik, J.: Finding action tubes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 759–768 (2015)
9. Hosang, J., Omran, M., Benenson, R., Schiele, B.: Taking a deeper look at pedestrians. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4073–4082 (2015)
10. Hou, Y.-L., Pang, G.K.: People counting and human detection in a challenging situation. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **41**(1), 24–33 (2011)
11. Kong, D., Gray, D., Tao, H.: A viewpoint invariant approach for crowd counting. In: 18th International Conference on Pattern Recognition, ICPR 2006, vol. 3, pp. 1187–1190. IEEE (2006)
12. Kumagai, S., Hotta, K.: LAC between Cells of HOG Feature for Crowd Counting. In: Advances in Visual Computing, pp. 688–697. Springer International Publishing, Switzerland (2014)
13. Lempitsky, V., Zisserman, A.: Learning to count objects in images (2010)
14. Change Loy, C., Gong, S., Xiang, T.: From semi-supervised to transfer counting of crowds. In: IEEE International Conference on Computer Vision (ICCV), pp. 2256–2263. IEEE (2013)
15. Nielsen, M.: Neural Networks and Deep Learning. Determination Press 1 (2014)
16. Patzold, M., Evangelio, R.H., Sikora, T.: Counting people in crowded environments by fusion of shape and motion information. In: Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 157–164. IEEE (2010)
17. Perko, R., Schnabel, T., Fritz, G., Almer, A., Paletta, L.: Counting people from above: Airborne video based crowd analysis. arXiv preprint arXiv:1304.6213 (2013)
18. Pham, V.Q., Kozakaya, T., Yamaguchi, O., Okada, R.: COUNT Forest: CO-voting uncertain number of targets using random forest for crowd density estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3253–3261 (2015)
19. Vu, T.H., Osokin, A., Laptev, I.: Context-aware CNNs for person head detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2893–2901 (2015)
20. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: Proceeding CVPR (2015)
21. Zhang, Z., Wang, M., Geng, X.: Crowd counting in public video surveillance by label distribution learning. Neurocomputing **166**, 151–163 (2015)