A Spatio-Temporal Attentive Network for Video-Based Crowd Counting

1st Marco Avvenuti
Dept. of Information Engineering
University of Pisa
Pisa, Italy
marco.avvenuti@unipi.it

2nd Marco Bongiovanni
Dept. of Information Engineering
University of Pisa
Pisa, Italy
m.bongiovanni2@studenti.unipi.it

3rd Luca Ciampi
Inst. of Information Science and Tech.
National Research Council (ISTI-CNR)
Pisa, Italy
luca.ciampi@isti.cnr.it

4th Fabrizio Falchi
Inst. of Information Science and Tech.
National Research Council (ISTI-CNR)
Pisa, Italy
fabrizio.falchi@isti.cnr.it

5th Claudio Gennaro
Inst. of Information Science and Tech.
National Research Council (ISTI-CNR)
Pisa, Italy
claudio.gennaro@isti.cnr.it

6th Nicola Messina
Inst. of Information Science and Tech.
National Research Council (ISTI-CNR)
Pisa, Italy
nicola.messina@isti.cnr.it

Abstract—**Automatic people counting from images has recently drawn attention for urban monitoring in modern Smart Cities due to the ubiquity of surveillance camera networks. Current computer vision techniques rely on deep learning-based algorithms that estimate pedestrian densities in still, individual images. Only a bunch of works take advantage of temporal consistency in video sequences. In this work, we propose a spatio-temporal attentive neural network to estimate the number of pedestrians from surveillance videos. By taking advantage of the temporal correlation between consecutive frames, we lowered state-of-the-art count error by 5% and localization error by 7.5% on the widely-used FDST benchmark.**

Index Terms—**Crowd Counting, Deep Learning, Visual Counting, Smart Cities**

I. INTRODUCTION

Computer Vision obtained a tremendous boost in the last few years thanks to the astonishing advances in Machine Learning. In particular, Deep Learning allowed the research community to define new state-of-the-arts in many Computer Vision tasks, such as object detection [1], or image retrieval [2], to name a few. With the increasing interest in Smart Cities and the grown availability of surveillance cameras that have become pervasive, there is a unanimous effort to employ these novel technologies for urban monitoring and surveillance. Like no other sensing mechanism, networks of city cameras can observe and simultaneously provide visual data to AI systems to extract relevant information from this deluge of data. In this context, many smart applications, ranging from lot occupancy detection [3] to pedestrian detection [4]–[6] and re-identification [7], have been proposed and are nowadays widely employed worldwide. In this work, we treat the *counting* task, which consists of providing the number of instances

of a specific class present in the scene — for example, the number of vehicles that are transiting a particular road. Specifically, we focus on crowd counting to automatically estimate the number of people present in images gathered from city surveillance cameras. This application is crucial in many scenarios, like monitoring and eventually limiting people aggregations during the recent COVID-19 pandemic.

Recent literature extensively faced the crowd counting task by estimating and integrating density maps of still, individual images. Nevertheless, relatively few works take advantage of the temporal consistency of *video* streams. However, using temporal constraints across consecutive frames could be an essential key point for enhancing the counting performance.

Driven by these concerns, we propose an enhanced extension to the video-counting framework introduced by Liu et al. [8], [9]. These works presented an interesting semi-supervised learning method that infers people density maps starting from the estimation of people flows among adjacent keyframes. At inference time, the flow is integrated to obtain the actual people count. Although the learning framework is solid, their proposed network comprises a simple fully-convolutional encoder-decoder pipeline, which estimates the flow from a pair of consecutive images. In this work, inspired by the recent attentive mechanisms proposed to process visual data, such as Vision Transformer [10], we enhance the architecture in [8], [9] with self-attentive connections. Specifically, we introduce an attentive-based temporal fusion layer to improve the flow predictor, taking advantage of the temporal correlation between consecutive frames. Through an experimental evaluation, we show that our proposed method can boost the performance compared with the original framework on the widely-used FDST dataset [11], one of the largest and most diverse collections of temporally correlated frames, suitable for video-based counting. We assess not only the counting performance considering the counting errors occurring at inference time (i.e., the difference between the predicted and

the actual person numbers) but also the ability to correctly localize the counted persons. Indeed, count errors do not take into account *where* the pedestrians have been detected in the images and, consequently, counting models might achieve low values of errors while providing wrong predictions (e.g., a high number of false positives and false negatives).

To sum up, we propose the following contributions:

- We propose an extension to a recent semi-supervised video-based counting framework [8], [9], employing an attentive-based temporal fusion layer that takes advantage of the temporal correlation between consecutive images and improves the people flow estimation.
- We demonstrate through detailed experiments that the proposed variation can reach state-of-the-art results on the FDST dataset, lowering the count error by 5%.
- We conduct a performance evaluation also considering the ability to correctly localize the counted persons, lowering the localization error by 7.5% compared to the original counting framework.

The code and the trained models will be publicly available at **https://tinyurl.com/yb42ce38**.

II. RELATED WORK

A. Image-based Counting

Image-based counting aims at estimating the number of object instances, like people [12], [13], cells [14], [15], or vehicles [16], [17], in *still* images or video frames [18]. Current solutions are formulated as supervised deep learning-based problems belonging to one of two main categories: counting by *detection* and counting by *regression*. Detection-based approaches, such as in [19] and [20], require prior detection of the single instances of objects. On the other hand, regression-based techniques like [21] and [22] try to establish a direct mapping between the image features and the number of objects in the scene, either directly or via the estimation of a density map (i.e., a continuous-valued function). Regression techniques show superior performance in crowded and highly-occluded scenarios [18].

B. Video-based Counting

Crowd counting approaches are mainly based on single-image inputs, even when a video sequence is available, leading to the impossibility of exploiting the temporal interdependence between consecutive frames in the sequence. Nevertheless, in the literature, there are a handful of works that rely on the estimation of density maps, but, on the other hand, try to exploit also temporal information to improve counting accuracy. For instance, [23] introduced one of the first video-based counting approaches, based on an LSTM-based method called ConvLSTM, i.e., a fully connected LSTM extended with convolutional layers in both the input-to-state and state-to-state connections. In [24], the authors exploited a Locality-constrained Spatial Transformer (LST) module to model the spatial-temporal correlation between estimated neighboring density maps. Another remarkable work is [25], where the authors introduced a Temporal Aware Network (TAN), which
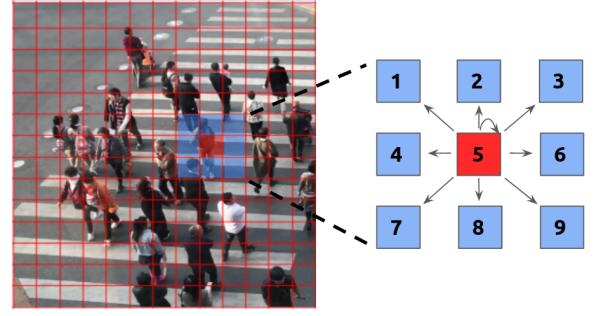

Fig. 1. Visualization of a neighborhood $N(j)$ of an image patch j (shown in red). This is used to impose people conservation constraints, as formulated by Equations 2 and 3.

can compute the number of people present at frame X_t by exploiting frames from X_{t-k} to X_{t+k}. More recently, Liu et al. [8], [9] proposed an alternative approach where density maps are not directly regressed from images but are inferred from people flows across image locations between consecutive frames. However, the proposed network in charge of estimating flows from pair of images lies in a simple fully-convolutional encoder-decoder pipeline. In this work, we extend and enhance this framework by exploiting self-attentive connections, introducing an attentive-based temporal fusion to enhance the flow predictor.

C. Attentive Models

Attention mechanisms have been largely used in the last two years and had a significant impact on tasks that involve both vision and language, such as VQA [26], image captioning [27], [28] or image-text matching [2], [29]. Recently, the Transformer-like attention mechanism [30] obtained the best results in processing images and videos. In particular, the authors in [10] introduced the Vision Transformer, demonstrating the power of the self-attentive mechanism in the image classification task. Similarly, the DETR architecture [31] used the full Transformer architecture for tackling the object detection task, obtaining remarkable results with respect to state-of-the-art fully-convolutional approaches. This work is inspired by the recent advances in attentive video processing, where most methods use spatio-temporal attention to understand frame patches from multiple timesteps [32], [33].

III. METHOD

The proposed architecture tries to reconstruct the people flows, enhancing the weak-supervised learning framework proposed in [8], [9]. In Section III-A we briefly review their original framework; in Section III-B we explain our attentive-based temporal fusion used to improve the flow predictor.

A. The People-Flow Approach

The approach presented in [8], [9] is a video-based counting scheme that does not directly estimate people densities from images but infers them from the so-called *people flow* between two consecutive frames. People flows are vector fields that associate pedestrian movement vectors to every point in the

frame space. People flows are zeros at a certain point in space if there are no moving pedestrians.

Once the people flow $\boldsymbol{f}^{t-1,t}$ between two consecutive frames $\boldsymbol{I}^{t-1}$ and $\boldsymbol{I}^t$ has been predicted, the j-th spatial location of the density map d_j^t for the frame $\boldsymbol{I}^t$ can be reconstructed by summing all the flow contributions entering j from neighboring locations of the previous frame:

$$d_j^t = \sum_{i \in N(j)} f_{i,j}^{t-1,t} \tag{1}$$

where the neighbouring locations $N(j)$ of the j-th path are shown in Figure 1. The final people count at time t can then be found by summing up all the pixel values of the obtained density map: $\sum_j d_j^t$. However, regressing the flows is not a straightforward operation, as many video-counting datasets do not embed any explicit people flow ground-truth that can be used to supervise the network.

For this reason, the work in [8], [9] proposed a weak-supervised learning approach that estimates the flows using only the ground-truth density maps $\bar{\boldsymbol{d}}^{t-1}$ and $\bar{\boldsymbol{d}}^t$ at consecutive timesteps $(t-1, t)$. In particular, the flows are constructed by only imposing strong people conservation constraints, i.e., people cannot appear or disappear between consecutive frames if not in the frame edges. In particular, the constraints can be expressed as follows:

$$\sum_{i \in N(j)} f_{i,j}^{t-1,t} = \sum_{k \in N(j)} f_{j,k}^{t,t+1} \tag{2}$$

$$f_{i,j}^{t-1,t} = f_{j,i}^{t,t-1} \tag{3}$$

where Eq. 2 imposes people conservation in the neighborhood of a location j (Figure 1) across consecutive frame intervals, and Eq. 3 enforces the spatio-temporal symmetry of the flows, i.e. the people should move in the opposite direction when the time flows backwards. More details can be found in [8].

With this weakly-supervised learning framework, the only missing piece is the function that regresses the flows. In particular, this function is a deep neural network $\mathcal{R}(\boldsymbol{I}^{t-1}, \boldsymbol{I}^t, \theta)$ that outputs the flow $\boldsymbol{f}^{t-1,t}$ given two consecutive images in input. Its parameters θ are optimized during the training process by enforcing the constraints in Eq. 2 and 3. In the next paragraph, we propose to use a spatio-temporal attentive network as $\mathcal{R}$.

B. The Attentive Flow Regressor

The proposed network resembles a convolutional encoder-decoder architecture, which takes two consecutive RGB images in input and produces the predicted flows. Specifically, the encoder $\mathcal{E}$ processes the input images $\boldsymbol{I}^{t-1}$ and $\boldsymbol{I}^t$ to obtain the corresponding internal feature maps $\boldsymbol{w}^{t-1}, \boldsymbol{w}^t \in \mathbb{R}^{W \times H \times C}$, where (W, H, C) are the width, the height and the number of channels, respectively. Formally:

$$\boldsymbol{w}^{t-1} = \mathcal{E}(\boldsymbol{I}^{t-1}) \tag{4}$$

$$\boldsymbol{w}^t = \mathcal{E}(\boldsymbol{I}^t) \tag{5}$$

These feature maps are then composed together using an aggregator function $\tilde{\boldsymbol{w}} = \mathcal{A}(\boldsymbol{w}^{t-1}, \boldsymbol{w}^t)$, which outputs a feature map $\tilde{\boldsymbol{w}} \in \mathbb{R}^{W \times H \times C'}$, with the same spatial resolution but possibly a different number C' of channels. In the end, the final flows are obtained by applying the decoder to the aggregated feature maps: $\boldsymbol{f}^t = \mathcal{D}(\tilde{\boldsymbol{w}}) \in \mathbb{R}^{W \times H \times 10}$. Notice that the output flow is 10-dimensional, as there are ten possible directions in which a person can move inside the frame (nine locations in the neighborhood including the starting cell, as depicted in Figure 1, plus one cell representing *the rest of the world* at the edges of the image).

In the original formulation in [8], the aggregator function is a straightforward concatenation along the channels dimension: $\mathcal{A} = [\cdot, \cdot]^C$, which therefore outputs $\tilde{\boldsymbol{w}} \in \mathbb{R}^{W \times H \times 2C}$. In this work, we propose an attentive spatio-temporal aggregation module that can produce more space-time aware feature maps $\tilde{\boldsymbol{w}}$, which, in turn, provide better flows.

The idea is to employ the feature maps coming from one of the two input frames to condition the features of the other frame. We denote the feature maps from the two frames as $\boldsymbol{w}_{\text{input}}$ and $\boldsymbol{w}_{\text{target}}$, where *input* refers to the visual information that conditions the *target* one in the attention mechanism. In particular, we used an approach similar to the convolutional self-attention by [34]. Specifically, we initially derive other three feature maps from the two input maps produced by the encoder, namely $\boldsymbol{v} = g_v(\boldsymbol{w}_{\text{target}}) \in \mathcal{R}^{W \times H \times d_v}$ and $\boldsymbol{q} = g_q(\boldsymbol{w}_{\text{target}}) \in \mathcal{R}^{W \times H \times d_k}$ from $\boldsymbol{w}_{\text{target}}$, and $\boldsymbol{k} = g_k(\boldsymbol{w}_{\text{input}}) \in \mathcal{R}^{W \times H \times d_k}$ from $\boldsymbol{w}_{\text{input}}$. These $\boldsymbol{v}, \boldsymbol{k}, \boldsymbol{q}$ are the *values*, the *keys*, and the *queries* respectively, used to drive the Transformer-like attention mechanism. They are produced from the input feature maps using three different convolutional layers g_v, g_k, g_k having a 3×3 kernel and padding=1 to leave the spatial resolution untouched.

The output, as in the Transformer [30] attention mechanism, is computed through the scaled dot-product attention:

$$\boldsymbol{o} = \text{softmax}(\frac{\boldsymbol{q}\boldsymbol{k}^\top}{\sqrt{d_k}})\boldsymbol{v} \tag{6}$$

$$= \text{softmax}(\frac{g_q(\boldsymbol{w}_{\text{target}})g_k(\boldsymbol{w}_{\text{input}})^\top}{\sqrt{d_k}})g_v(\boldsymbol{w}_{\text{target}}) \tag{7}$$

Note that, differently from the original Transformer-like attention, we empirically found that computing values from the target sequence — instead of the input sequence — led to better results. Still, the core idea aims at incorporating the context from one frame into the feature map of the other frame.

To account for the original feature maps after the attentive processing, we concatenate a transformation of the original feature maps $\boldsymbol{w}_{\text{input}}$ to the output $\boldsymbol{o}$ along the channel dimension:

$$\bar{\boldsymbol{o}} = [g_{\text{skip}}(\boldsymbol{w}_{\text{input}}), \boldsymbol{o}]^C \tag{8}$$

where $g_{\text{skip}}(\cdot)$ is a convolution operation that does not change the spatial resolution, and outputs $C - d_v$ channels so that $\bar{\boldsymbol{o}}$ has again C channels. We denote the above-described spatio-temporal attentive fusion module as $\bar{\boldsymbol{o}} = \mathcal{ST}(\boldsymbol{w}_{\text{input}}, \boldsymbol{w}_{\text{target}})$ (see Figure 3). Now, the final goal is to condition the past frame $\boldsymbol{I}^{t-1}$ given the present frame $\boldsymbol{I}^t$, and vice-versa. For
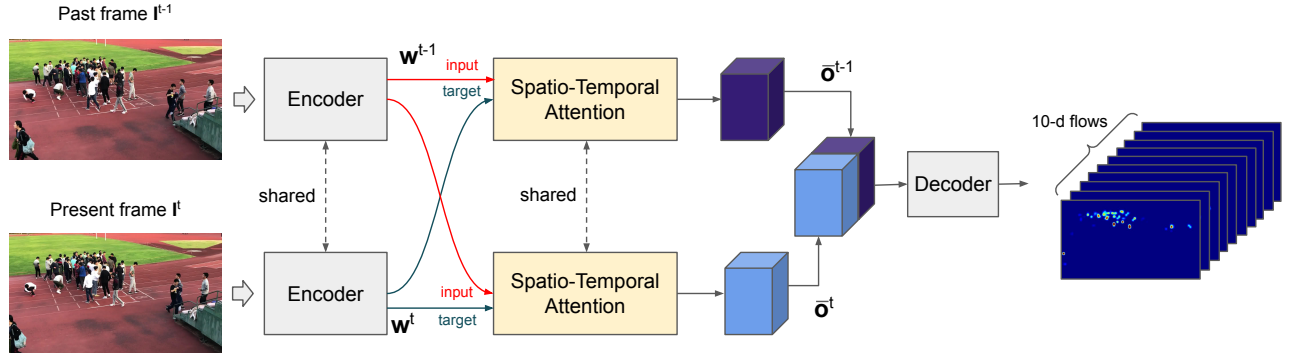
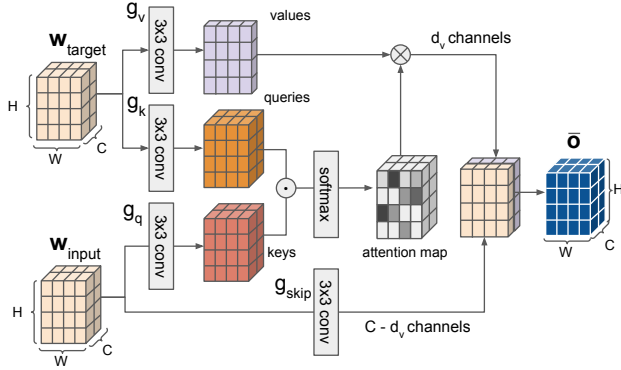Fig. 2. The overall spatio-temporal attentive regressor architecture.

Fig. 3. Inner architecture of the spatio-temporal fusion module.

this reason, we compute the two symmetric attentional feature maps, by simply swapping the two frames in input:

$$\bar{\boldsymbol{o}}^t = \mathcal{ST}(\boldsymbol{w}^{t-1}, \boldsymbol{w}^t) \tag{9}$$

$$\bar{\boldsymbol{o}}^{t-1} = \mathcal{ST}(\boldsymbol{w}^t, \boldsymbol{w}^{t-1}) \tag{10}$$

At this point, using plain concatenation along the channel direction, we merge the obtained attentive feature maps obtaining the final spatio-temporal aware feature map: $\bar{\boldsymbol{w}} = [\bar{\boldsymbol{o}}^{t-1}, \bar{\boldsymbol{o}}^t]^C$. Finally, as in the original encoder-decoder convolutional architecture, the 10-dimensional output flow $\boldsymbol{f}^t$ is obtained by applying the decoder to this last feature map: $\boldsymbol{f}^t = \mathcal{D}(\tilde{\boldsymbol{w}})$. The overall architecture is shown in Figure 2.

IV. EXPERIMENTAL EVALUATION

This section describes the experiments performed to validate our approach and discusses the obtained results. As benchmark, we exploit the widely-used FDST dataset [11]. It consists of 100 videos gathered from 13 different scenarios. A total of 150,000 frames have been extracted, thus representing one of the largest and most diverse collections of real-world images suitable for this task. Annotations are expressed using dots localizing peoples' heads, as usual for the counting task, for a total of 394,081 labeled pedestrians. We follow the same setting as in [11], considering 60 videos (9,000 frames) as the training set and the remaining 40 videos (6,000 frames) as the test set.

In the first stage of the evaluation, we compare the counting errors obtained using our proposed solution against the framework introduced in [8], [9], and other recent state-of-the-art counting solutions present in the literature. On the other hand, in the second part of our experiments, we also assess the ability to correctly localize the counted pedestrians.

A. Comparison with the State-of-the-art

Here, we compare our solution in terms of counting with other state-of-the-art methodologies. Following standard counting benchmarks, we used the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) to measure the counting performance. Specifically, they are defined as:

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^{N} \left| c_{\text{gt}}^n - c_{\text{pred}}^n \right|, \tag{11}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (c_{gt}^n - c_{pred}^n)^2}. \tag{12}$$

where N is the number of test images, c_{gt}^n is the actual count (i.e., the ground truth), and c_{pred}^n is the predicted count of the n-th image. It is worth noting that, as a result of the squaring of each difference, the RMSE effectively penalizes large errors more heavily than small ones, and so it is more useful when outliers are particularly undesirable.

We report our quantitative results obtained on the FDST dataset in Table I. We called our solution *People Flow Temporal Fusion (TF)*, to distinguish it from the original framework *People Flow Plain Concatenation (PC)*. We divided the training set into train and validation splits, considering the 80% and the 20% of the available data, respectively. To mitigate the overfitting problem, we considered training and validation images belonging to different video sequences (12 sequences for validation and the remaining 48 for the training). For better capturing movement between consecutive images, we sampled the previous and the consecutive frames by setting an offset of 5 frames. More, we performed data augmentation randomly, applying common transformations such as horizontal flipping, cropping, and normalization. We repeated the experiments using our solution three times, reporting the mean. Finally,

TABLE I
Comparison with SOTA on FDST dataset. We obtained SOTA
results in terms of MAE and we lowered the error by 5%
compared to the original framework.

Model	Temporal	MAE ↓	RMSE ↓
ConvLSTM [23]	✓	4.48	5.82
WithoutLST [24]		3.87	5.16
MCNN [13]		3.77	4.88
LST [24]	✓	3.35	4.45
CAN [22]		2.44	2.96
People Flow PC [8], [9]	✓	2.17	**2.62**
People Flow TF (Our)	✓	**2.07**	2.69

TABLE II
Localization Analysis. We lowered the *GAME* by about 7.5%
compared to the original framework on the FDST dataset.

Model	MAE ↓	GAME ↓
People Flow PC*	2.17	16.00
People Flow TF (Our)	**2.07**	**14.81**

* Retrained in this work.

sub-dividing the image in 4^L non-overlapping regions and summing the MAE computed in each of these sub-regions:

$$GAME(L) = \frac{1}{N} \sum_{n=1}^{N} (\sum_{l=1}^{4^L} |c_{gt}^l - c_{pred}^l|), \qquad (13)$$

where N is the total number of test images, c_{pred}^l is the estimated count in a region l of the n-th image, and c_{gt}^l is the ground truth for the same region in the same image. The higher L, the more restrictive the GAME metric will be.

In this work, we considered a grid with a fixed dimension of 80×45 (width and height, respectively), corresponding to the spatial resolution of the feature maps produced by the encoder $\mathcal{E}$ fed with input images of 640×360 pixels. Table II shows our quantitative results obtained on the FDST dataset exploiting our solution *People Flow Temporal Fusion (TF)* compared with the original framework *People Flow Plain Concatenation (PC)*. We repeated the experiments three times, reporting the mean. Specifically, concerning the original framework, we performed three different experiments considering the model publicly provided [1] by the authors of [8], [9], and two models obtained re-training from scratch the original network using different seeds (as illustrated in the Table, the mean concerning the MAE is fully comparable with the one reported in [8], [9]). As can be seen, our approach outperforms the original methodology, lowering the GAME by about 7.5%, thus suggesting that our solution provides better performance not only in terms of count errors but that it is also capable of better localizing the found pedestrians. In Figure 4 we report some qualitative outputs from our approach.
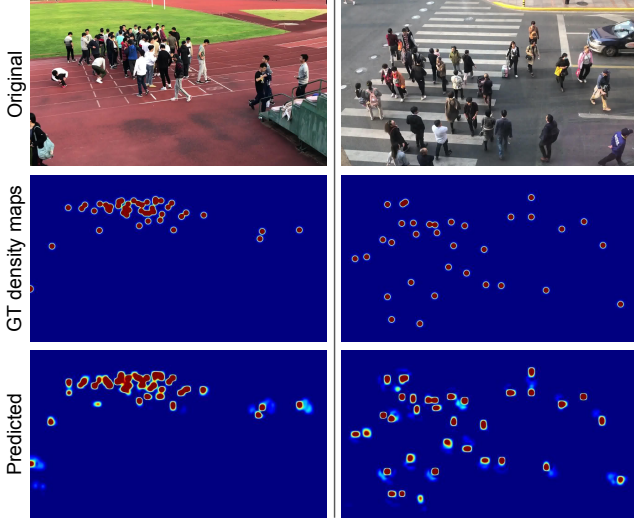

Fig. 4. Some qualitative examples showing the predicted density maps.

images are resized to a dimension of 640×360 pixels (width and height, respectively). As can be seen, our approach outperforms the competing methods. In particular, we lowered the MAE by about 5% compared to the baseline framework, with a comparable RMSE value.

B. Localization Analysis

Although the MAE is a fair metric for establishing a comparative in terms of count error, it can often lead to masking erroneous estimations. A potential pitfall of the counting approaches is that the models may miss hard-to-detect instances. To compensate for these missed detections and estimate the correct count, they may falsely mark background sub-regions having similar regional image properties as possible object instances instead. The reason is that the MAE does not take into account *where* the estimations have been done in the images. In this section, we conduct experiments to assess the ability of our solution to localize the counted pedestrians correctly, comparing the obtained results against the framework introduced by [8], [9]. Specifically, we consider the *Grid Average Mean absolute Error (GAME)* [35], a hybrid metric that simultaneously considers the object count and the estimated locations of the persons. It is computed by

V. Conclusions and Future Work

This paper proposed an attentive neural network to estimate the number of pedestrians in videos from surveillance cameras. In particular, we extended a promising method that estimates people flows to obtain people densities, which are then integrated to get the final count. To create a more suitable spatial and temporal context for predicting the flows, we proposed a spatio-temporal attentive network, which can contextualize the features maps from the present with those from the past frame vice-versa. In our experiments on the largely-used FDST dataset, we demonstrated the effectiveness of our architecture. We obtained a considerable improvement in counting performance compared to other state-of-the-art approaches in video counting. Furthermore, through the GAME

[1] **https://github.com/weizheliu/People-Flows**

metric, we demonstrated that our method achieves a more precise localization of the pedestrians inside the frame than the network without the spatio-temporal attentive module.

In the future, we plan to extend the experimentation to other datasets, also employing virtual data to train the network, to avoid manual annotation of large dot-annotated pedestrian videos. Furthermore, we plan to use domain adaptation techniques to fill the well-known *domain-gap* existing between the different monitored scenarios.

REFERENCES

[1] M. Traquair, E. Kara, B. Kantarci, and S. Khan, "Deep learning for the detection of tabular information from electronic component datasheets," in *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, jun 2019.

[2] N. Messina, G. Amato, A. Esuli, F. Falchi, C. Gennaro, and S. Marchand-Maillet, "Fine-grained visual textual alignment for cross-modal retrieval using transformer encoders," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 4, pp. 1–23, nov 2021.

[3] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, "Car parking occupancy detection using smart camera networks and deep learning," in *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, jun 2016.

[4] G. Amato, L. Ciampi, F. Falchi, C. Gennaro, and N. Messina, "Learning pedestrian detection from virtual worlds," in *Image Analysis and Processing - ICIAP 2019 - 20th International Conference, Trento, Italy, September 9-13, 2019, Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 11751. Springer, 2019, pp. 302–312.

[5] L. Ciampi, N. Messina, F. Falchi, C. Gennaro, and G. Amato, "Virtual to real adaptation of pedestrian detectors," *Sensors*, vol. 20, no. 18, p. 5250, sep 2020.

[6] M. D. Benedetto, F. Carrara, L. Ciampi, F. Falchi, C. Gennaro, and G. Amato, "An embedded toolset for human activity monitoring in critical environments," *Expert Systems with Applications*, vol. 199, p. 117125, aug 2022.

[7] Y. Ma, T. Bai, W. Zhang, S. Li, J. Hu, and M. Lu, "Multi-scale relation network for person re-identification," in *2021 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, sep 2021.

[8] W. Liu, M. Salzmann, and P. Fua, "Estimating people flows to better count them in crowded scenes," in *Computer Vision – ECCV 2020*. Springer International Publishing, 2020, pp. 723–740.

[9] ——, "Counting people by estimating people flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[11] Y. Fang, B. Zhan, W. Cai, S. Gao, and B. Hu, "Locality-constrained spatial transformer network for video crowd counting," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, jul 2019.

[12] L. Liu, Z. Qiu, G. Li, S. Liu, W. Ouyang, and L. Lin, "Crowd counting with deep structured scale integration network," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, oct 2019.

[13] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016.

[14] L. Ciampi, F. Carrara, G. Amato, and C. Gennaro, "Counting or localizing? evaluating cell counting and detection in microscopy images," in *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2022.

[15] J. P. Cohen, G. Boucher, C. A. Glastonbury, H. Z. Lo, and Y. Bengio, "Count-ception: Counting by fully convolutional redundant counting," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. IEEE, oct 2017.

[16] G. Amato, L. Ciampi, F. Falchi, and C. Gennaro, "Counting vehicles with deep learning in onboard UAV imagery," in *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, jun 2019.

[17] L. Ciampi, C. Gennaro, F. Carrara, F. Falchi, C. Vairo, and G. Amato, "Multi-camera vehicle counting using edge-ai," *CoRR*, vol. abs/2106.02842, 2021.

[18] V. S. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010*. Curran Associates, Inc., 2010, pp. 1324–1332.

[19] G. Amato, P. Bolettieri, D. Moroni, F. Carrara, L. Ciampi, G. Pieri, C. Gennaro, G. R. Leone, and C. Vairo, "A wireless smart camera network for parking monitoring," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, dec 2018.

[20] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Where are the blobs: Counting by localization with point supervision," in *Computer Vision – ECCV 2018*. Springer International Publishing, 2018, pp. 560–576.

[21] L. Ciampi, C. Santiago, J. Costeira, C. Gennaro, and G. Amato, "Domain adaptation for traffic density estimation," in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2021.

[22] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2019.

[23] F. Xiong, X. Shi, and D.-Y. Yeung, "Spatiotemporal modeling for crowd counting in videos," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2017.

[24] Y. Fang, B. Zhan, W. Cai, S. Gao, and B. Hu, "Locality-constrained spatial transformer network for video crowd counting," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, jul 2019.

[25] X. Wu, B. Xu, Y. Zheng, H. Ye, J. Yang, and L. He, "Fast video crowd counting with a temporal aware network," *Neurocomputing*, vol. 403, pp. 13–20, aug 2020.

[26] M. Malinowski, C. Doersch, A. Santoro, and P. Battaglia, "Learning visual question answering by bootstrapping hard attention," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–20.

[27] S. He, W. Liao, H. R. Tavakoli, M. Yang, B. Rosenhahn, and N. Pugeault, "Image captioning through image transformer," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[28] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, "Meshed-memory transformer for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 578–10 587.

[29] N. Messina, F. Falchi, A. Esuli, and G. Amato, "Transformer reasoning network for image-text matching and retrieval," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 5222–5229.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[31] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[32] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding," *arXiv preprint arXiv:2102.05095*, vol. 2, no. 3, p. 4, 2021.

[33] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6836–6846.

[34] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3286–3295.

[35] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascón, and D. Oñoro-Rubio, "Extremely overlapping vehicle counting," in *Pattern Recognition and Image Analysis*. Springer International Publishing, 2015, pp. 423–431.