ArenaPeds: Pedestrian Flow in a highly crowded stadium

Jan Erik van Woerden

October 2020

1 Introduction

In a world with increasing accessibility to information, it is important to summarize all this information into useful information for the intended users. In this case we have access to a large amount of surveillance camera's which are used during busy events. To monitor all these camera's it requires a lot of manual watching what is happening on those camera's. There has a lot of research been done on reducing the amount of manual watching and translating the information into more actionable and measurable tasks.

In this thesis we will focus on the Line of Interest problem (Convert this into a better term). The goal is to count the amount of pedestrians that cross a specified line in a certain time frame. In area's with low density of pedestrians a generic object tracking solution would suffice. In high density crowds the results of the solutions will degrade quickly. (Because most object trackers can handle up to 50 people, YOLOv4 paper and another solution)

To handle high density crowds specialized solutions are presented over the years. Those methods typically combine two components, crowd density prediction (Region of Interest) and flow estimation. (Already citing papers, same as below?). Traditionally the density prediction is done using keypoint extraction and feeding those keypoints in a regression model (Cite some papers). Flow estimation is typically done with a Lucas-Kanade method (papers who implement this method).

More recently the rise of Convolutional Networks made it possible to improve both components. Both crowd counting (Papers) and flow estimation (Papers) research fields have state of the arts which heavily rely on convolutional networks. Several papers combine these CNN focused methods to a Line of Interest solution (Papers, Zhao et al. (2016) and more) which give good and promising results.

One of the major drawbacks of Neural Networks is the huge amounts of that that is required for training these networks (Probably should be papers, but this is like a very trivial thing in the AI field, so should it?). For the crowd density estimation several public datasets are available and labeling new images can be done easily. Labeling data for the flow estimator is much more

difficult and time intensive (Should I explain this more in detail. No real paper, but can show somewhere my calculation of time). In earlier papers (Previous paragraph papers) this is done, but this is less feasible for implementation in actual applications.

Besides supervised learning of the flow estimation, more and more traction is gained for the unsupervised flow estimation research area. These methods provide a much more scalable solution for real world applications and come closer to supervised flow estimation performance. In this thesis we will focus on a solution which utilizes the unsupervised flow estimation. (See research question 1)

Besides the huge amount of data another problem with several convolutional neural networks is the running time. CNN's can take a long time to produce their predictions. To make methods applicable for real life applications quickly producing results with minimal performance degrading is crucial. This will be our second focus of this work. (See research question 2)

Lastly another major focus of the work is to make the proposed solution versatile and make it possible to easily use for new scenes. So we will focus on the minimization of extra required data when a the model is deployed on another scene. (See research question 3)

1.1 Research questions

In this work, we introduce a novel solution for the Line of Interest problem. The solution uses unsupervised flow estimation and supervised crowd density prediction to minimize the use of labeling.

To motivate our work we endeavor to answer the next research questions:

- What is the performance of the new model against fully supervised models?
- Is it possible to let the methods run real time, and if so what is the performance?
- How much new data is required to properly perform in new scenes?

1.2 Thesis outline

The rest of this thesis is divided into the next chapters:

- **Related work**, which explains more in depth related work and tries to give a good background to understand the proposed solution.
- **Method**, presents the method of the proposed solution. (Nothing fancy to tell about)
- **Implementation**, presents the hyperparameters, evaluation methods and the used datasets.

- **Results**, displays the results for the discussion.
- **Discussion**, discusses the research questions and try to answer it based on the results.
- **Conclusion**, wraps it up and summarizes what we can conclude

2 Related Work

In the following parts we try to explain several key subjects to understand the field of Line of Interest.

2.1 Region of Interest

One of the building blocks of all the information reduction is Crowd Counting/Region of Interest. In low density area's individual object detection, such as YOLO, is a good solution. In high density area's the occlusion by pedestrians makes it hard to detect individual pedestrians. Density Maps still provide the possibility to count the amount of people inside an area, but don't have to deal with the exact locations of the heads, which makes it more robust against errors in the prediction.

In contrast to object detection, it is not possible to identify the full body of the pedestrian in the frames. When using downwards facing camera's, such as high hanging surveillance camera's, the only clearly visible part of a pedestrian is the head. Therefore the labeling of the pedestrians is done by selecting a single point on the head.

An individual point is very hard for a Neural Network to find correctly and is prone to errors. Additionally the exact location of an individual pedestrian are not of high importance, only the total count given a certain region.

Density Maps for Crowd Counting are therefore a solution. Instead of finding individual points in the image, which represent a pedestrian, the points are spread out on the density map, by a Gaussian Distribution, which summed together add up to one.

This method results in a robust method to train an end-to-end model for ROI especially in high density images.

The density map is always a area with several Gaussians. The method to generate those Gaussian's and their size differ.

- All Gaussians same size - Gaussians differ in size depending on the density of the pedestrians in the neighbourhood - Gaussian differs in size based on the angle of the video. Hard to do when there is no access to height/angle.

Good paper to cite for benchmark and why we chose this method [Wang et al., 2020] [Li et al., 2018]

2.2 Flow Estimation

Since the early days of image detection different flow estimation methods are proposed. These methods are very general and can be applied to solve a range of different problems.

One of the most widely used methods is the Lucas-Kanade method (For more details [ul Rojas,]), which was invented in the 80s. It tries to predict the displacement of a pixel in two consecutive frames using equation (1).

$$I_x(x, y) \cdot u + I_y(x, y) \cdot v = -I_t(x, y) \tag{1}$$

With equation (1) the goal is to estimate (y, v) which is the direction of the pixel. $I_x(x, y)$ and $I_y(x, y)$ are the derivative of the intensity in both the x and y direction.$I_t(x, y)$ is the difference in intensity between the two consecutive frames $(b - a)$.

The Lucas-Kanade method is an easy and efficient method to estimate moving object, but it lacks on crucial points. It only incorporates gray scale and doesn't work very well with occluded pixels. (Linear movement? Not sure if NN's assume this as well. Probably not, but doesnt matter in our case)

2.2.1 FlowNet

One of the first good working models for Flow Estimation using a Neural Network based architecture is FlowNet [Fischer et al.,] and the architecture is still widely used. (FlowNetV2 [Ilg et al.,])

The first usable Flow Estimation using an end-to-end CNN was FlowNet, which is still widely used. The network tries per-pixel prediction which direction the flow is moving. This is done using a deep CNN and at the end a refinement to enlarge the remaining volume into an prediction with the same size as the original input frames. (Upconvolving using the output of several in between stages)

Till then a big problem was the trainability of large end-to-end CNN's and there massive hunger of data. Fischer et al. fixed this by generating a massive synthetic dataset called Flying Chairs. This dataset has more than 22000 image pairs which are build up from Flickr backgrounds and in the front chairs which have move using a random affine transformation.

Several issues with FlowNet were the complete reliability on Neural Networks. Some of the tasks inside the Flow Estimator can efficiently be solved with conventional methods. PWCNet ([Sun et al.,]) uses a combination of an CNN and some conventional methods. This results in a much smaller network, which results in faster training, additionally the smaller network can handle frames quicker.

One of the issues with CNN based flow estimators is the requirement for pixel level labeling per frame. This results in a huge amount of labeled data required for training. One of the solutions for these problems is the generation of synthetic data which, but these solutions do not scale well to the real world.

A solution for the huge amount of labeled data is unsupervised learning. Both DDFlow [Liu et al., a] and SelFlow [Liu et al., b] introduce methods to learn from unlabeled video.

- Video Counting explained: Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction

2.3 Line of Interest

In the earlier days of Line of Interest the method to estimate was by temporal slicing. In each frame a slice of the image is taken, by taking a slice around the LOI. These slices are stitched together, so a small sequence of frames result in a single image of the stitched slices, temporal sliced image. Based on the image the algorithms try to predict how many pedestrians passed the line in the short sequence.

Ma et al. [Ma and Chan,] first uses crowd segmentation on the sliced image, which results in several crowd segments in the image. Then the algorithm performs normalization on the pedestrians, because of the slicing the faster walking pedestrians appear smaller in the temporal slice image. After this the algorithm performs feature extraction using both global features and local features, which are extracted using local HOG (Histogram of Oriented Gradient) features and a bag-of-words model. This results in a single feature vector per crowd segment. Using a regression function with the features as input, the size of the crowd is predicted. The total pedestrian count of the sliced image is then used to calculate the exact count of people crossing the line between two individual frames. This is done by using several sliced images and solving the problem as a integer programming problem.

During the same time Cao et al. [Cao et al.,] first introduced CNN's in the field of LOI. The paper proposes a model which uses three CNN's which predicts the ins and outs of a individual temporal slice. The first CNN predicts the total amount of people in the temporal slice. The second CNN classifies from which direction people are crossing the line based on the optical flow of the temporal sliced image. The third one predict the ratio of pedestrians crossing during the slice, based on the optical flow. By using the outputs of each network, the algorithm is able to accurately predict the instant count. Additionally Cao et al. [Cao et al.,] shows that the use of CNN's improves the versatility of the models. The model performs better on almost all scenario's and is much more robust with changing weather scenario's and changing angles, than the method of Ma et al. [Ma and Chan,]

Zhao et al. [Zhao et al.,] introduced a new approach to process the images. Instead of using temporal sliced images. The method directly tries to predict the crowd count based on two consecutive frames. Additionally it merges the Crowd Counting and Crowd Flow models into a single CNN. The paper uses FlowNet as base for the model. Only the last layer predicts both the flow and the crowd density map, as described in Crowd Counting. Additionally the model doesn't try to predict precise direction of every pixel, because the labeled data provided for the model doesn't use pixel precise Flow Estimation. The flow estimator

only uses the dot-annotated location of the heads.

Zheng et al. [Zheng et al.,] provides in a era where CNN's have most of the records in hand, a method which scores SOTA on the benchmark for LOI. The model is extremely fast and only uses SVM's and linear regression to end state of the art. Problem with the model, it is very hard to tweak to very dynamic datasets such as the ArenaPeds dataset.

- Explain quickly why we can't just use YOLO (Low density) -

3 Method

3.1 Flow Estimation and Crowd Counting

3.2 Merger

3.3 Datasets

3.3.1 ArenaPeds

For this thesis we have access to a dataset of the City of Amsterdam. It has xxx amount of footage of environments with crowds ranging from 10 pedestrians in the frame to 1000 pedestrians a few hours later. Only a tiny proportion is labeled with where each pedestrian is. So there is no labeled data on the Crowd Direction, only for the Crowd Counting. This is the reason why we want to try to give unsupervised learning a shot.

4 Implementation

5 Discussion

References

[Cao et al.,] Cao, L., Zhang, X., Ren, W., and Huang, K. Large scale crowd analysis based on convolutional neural network. 48(10):3016–3024.

[Fischer et al.,] Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. FlowNet: Learning optical flow with convolutional networks.

[Ilg et al.,] Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. FlowNet 2.0: Evolution of optical flow estimation with deep networks.

[Li et al., 2018] Li, Y., Zhang, X., and Chen, D. (2018). Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1100.

[Liu et al., a] Liu, P., King, I., Lyu, M. R., and Xu, J. DDFlow: Learning optical flow with unlabeled data distillation.

[Liu et al., b] Liu, P., Lyu, M., King, I., and Xu, J. SelFlow: Self-supervised learning of optical flow.

[Ma and Chan,] Ma, Z. and Chan, A. B. Counting people crossing a line using integer programming and local features. 26(10):1955–1969.

[Sun et al.,] Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. PWC-net: CNNs for optical flow using pyramid, warping, and cost volume.

[Wang et al., 2020] Wang, Q., Gao, J., Lin, W., and Li, X. (2020). Nwpu-crowd: A large-scale benchmark for crowd counting. *arXiv preprint arXiv:2001.03360*.

[Zhao et al.,] Zhao, Z., Li, H., Zhao, R., and Wang, X. Crossing-line crowd counting with two-phase deep neural networks. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, volume 9912, pages 712–726. Springer International Publishing.

[Zheng et al.,] Zheng, H., Lin, Z., Cen, J., Wu, Z., and Zhao, Y. Cross-line pedestrian counting based on spatially-consistent two-stage local crowd density estimation and accumulation. 29(3):787–799.

[ul Rojas,] ul Rojas, P. D. R. *Lucas-Kanade in a Nutshell*.