ArenaPeds: Pedestrian Flow in a highly crowded stadium

Jan Erik van Woerden

30 October 2020

Contents

Chapter 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

Chapter 2

Background

In this chapter a selection of terms is explained which gives a basis to understand the rest of this thesis. This background is created with the assumption that the reader has a basic background in Machine Learning and (Convolutional) Neural Networks.

2.1 Region of Interest

Transform to Region of Interest

Crowd Counting in Machine Learning is a hot topic with a lot of new and recent papers. The goal of Crowd Counting is to count the amount of pedestrians present in a given image. This can be an individual image or a frame of a video sequence. The goal with Crowd Counting is only to give the amount of people in the image. The exact location of each pedestrian is irrelevant for this task. Crowd Counting can be done on a whole image or only given a part of the image. This region is then specified as the Region of Interest.

So the goal is to predict based on the given image the amount of pedestrians in an image. So can we directly predict the amount of pedestrians in the frame using a Machine Learning method? With a direct approach the amount of supervision on the weights is very low, so to train the model correctly, the amount of images required is very high. So all recent State-of-the-Art methods make use of an intermediate representation to give the model enough supervision to perform Crowd Counting with a low amount of training samples.

2.1.1 Object Recognition

A simple solution would be found in Object Recognition, as well a subfield of Machine Learning. This tries to locate objects given an image or video. By counting the amount of found objects in a frame we can predict the amount of pedestrians in a frame. For an area with a low count of pedestrians which are large enough, existing object recognition methods would be sufficient to recognize each pedestrian and give the correct count of the pedestrians in the given region.

So why not use general Object Recognition for Crowd Counting? The accuracy of Object Recognition will quickly degrade when pedestrians get smaller and the amount of pedestrians in the frame will increase. A lot of Object Recognition software has limitations with the total amount of objects it can recognize in a single image (Mostly around 50-100). Additionally they have a hard time when objects get occluded by each other, especially when they are small. Therefore most of the

This should be backedup by real numbers/papers. YOLO check limit and some

benchmarks used in Crowd Counting contain several hundred pedestrians to several thousands pedestrians per frame.

Show the average pedestrians per benchmark to backup numbers

2.1.2 Density Map

To the best of our knowledge all of the State-of-the-Art methods currently use a density map as extra supervision representation. A density map for Crowd Counting is a map which represents the density of pedestrians of each pixel. The density map is generated by taking the locations of each pedestrian (x_p and y_p in equation 2.1) and place those locations on the the density map.

Individual dots are very hard for a Neural Network to detect correctly and are proned to errors. To circumvent this a Gaussian shaped circle is crated around this location, still with with a sum of 1. The amount of pedestrians in the frame can be extracted from the density map by taking the sum over all the pixels of the density map (Equation 2.2, where $D_i(x, y)$ is the density for location x, y for trainings frame i).

$$D_i(x, y) = \frac{1}{2\pi\sigma_p^2} \sum_{p=1}^{P} e^{\frac{(x_p - x)^2 + (y_p - x)^2}{-2\sigma_p^2}} \tag{2.1}$$

$$C_i = \sum_{x=0,y=0}^{X,Y} D_i(x, y) \tag{2.2}$$

Several methods have been presented to optimize the generation of density maps. For most medium dense frames the difference in methods is minimal. Often in benchmarks with medium dense frames a fixed sigma is used ($\sigma_p = \sigma_i$ in equation 2.1). For highly dense frames the use of different methods can have a difference, especially when the difference in size between close pedestrians and pedestrians in the background is large.

Show image of crowd and of the density map

2.2 Flow Estimation

The research which is done on the Flow Estimation problem is widely used. Approaches on this topic can be used in a wide range of applications which makes it very interesting. Already in the early 1980's Horn and Schunck [Horn and Schunck, 1981] published the first paper which tried to predict flow. Since then lot's of different approaches have been published. Long conventional mathematical approaches have ruled the flow estimation field.

Maybe explain some extra information about this field and the history

In 2015 FlowNet was introduced. This supervised network can predict the flow map based on two consecutive frames. The flow map is a map which predict per pixel of the frame the amount of movement to another location. In equation 2.3, $F(x, y)^{(i)}$ shows the flow map as a difference between the location of the pixel in the current frame ($\begin{bmatrix} x \\ y \end{bmatrix}$) and the location of this pixel in the next frame ($L_i(x, y)$).

$$F_i(x, y) = L_i(x, y) - \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.3}$$

Creating a real world dataset that utilizes the power of pixel-wise flow estimation is very hard. There are no real world devices which could capture both video and create pixel perfect ground-truths to train the flow estimation models on. Most of the flow estimation benchmarks are therefore generated videos. Computer 3D-engines make it possible to generate pixel-perfect flow estimation based on the generated videos in the engine.

With the lack of real world usability several papers introduced methods to learn unsupervised the flow maps.

Read some more and explain some high level shit :)

2.3 Line of Interest

Line of Interest is very similar to Region of Interest. Where Region of Interest is the interest of the amount of people inside the ROI, the Line of Interest is the focus on the amount of pedestrians that cross the specified line during a certain timeframe.

Add an image with a line drawn inside the TUB dataset

With the Line of Interest problem the goal is to give the amount of pedestrians crossing of each side given a set of frames (a pre-captured video or video stream). The output of the prediction should give two numbers. In the rest of this thesis we refer to these sides as side left-to-right (side L) and right-to-left (side R). This with the assumption that a line is drawn vertically from top to bottom without losing generality.

Only a handful of papers are published about Line of Interest. In the earlier papers [Ma and Chan, , Cao et al.,], slicing was a widely used approach to estimate the Line of Interest. With slicing a small portion of the given frame around the LOI was taken. Over a set of consecutive frames each slice of the frame was taken and stitched together into a single image. On the images slow walking pedestrians appear rather wide and fast walking pedestrians shallow. By counting the amount of pedestrians present on the stitched image, the total amount of pedestrians crossing the line can be counted.

Recent papers discard this method, because it makes it hard to track pedestrian with different speeds and walking in different directions give artifacts which make it hard to track those pedestrians. In recent papers this method is discarded and actual frame by frame prediction is introduced. Using two consecutive frames the amount of pedestrians crossing the line is measured. These newer methods predict both location and direction of the pedestrian.

Maybe a bit more indepth

Based on these new papers, the problem of Line of Interest is divided into three separate problems. Locating the pedestrians (Region of Interest), estimate the direction (Flow Estimation) of the pedestrians and combining these two streams of information into the count for Line of Interest.

Chapter 3

Related Work

In this chapter we try to explain a couple of key papers on which the proposed methods are build.

3.1 Region of Interest

3.1.1 CSRNet

In 2018 CSRNet [Li et al., 2018] was introduced. This model had a massive improvement over earlier proposed models in the Region of Interest field. In the proposed model they proposed the use of dilated convolutions which use convolution filters over a much wider area. With this method the area a convolution filter stretches is much higher, without increasing the amount of processing time.

Explain in depth the contribution

3.2 Flow Estimation

Is there a paper which uses pyramid architecture

3.2.1 PWCNet

A popular Flow Estimation network is PWCNet [Sun et al.,]. It uses the original ideas of Flownet, but it improves Flownet in a lot of ways. Flownet traditionally is used to fully predict the flow with a neural network. PWCNet removes a couple of parts and replaces some parts of the network with conventional methods. This massively reduces the number of weights, which results in faster training and much quicker prediction. Additionally the network shows a higher accuracy on several benchmarks.

3.2.2 DDFlow

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.

3.3 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.

3.3.1 Two-stage Network

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.

3.3.2 Two-stage Local Crowd Estimation and Accumulation

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.

Chapter 4

Method

4.1 Accumulator

4.2 Network

For our system we split up the program into 3 parts. The Crowd Counting models, the Flow Estimation models and the Line of Interest. The methods which combine the Crowd Counting and the Flow Estimation in a single model are called Crowd Flow models (Useful as directory in the final code)

train.py # Train the models (All models should be trainable in this file, because they all share the same kind of dataloader)

test.py # Test the LOI methods using a trained network from train.py (Based on the datasets provided)

run.py # Run from a video stream live. (Would be super cool and very useful to actual demo this thing)

loi_models/ (Line of Interest models) - Pixelswise/Regionwise

fe_models/ (Flow Estimation models) - PWCNet cc_models/ (Crowd Counting models) - CSRNet cf_models/ (Crowd Flow models) - New Network

Chapter 5

Implementation

5.1 Trainings environment

5.2 Models

5.2.1 Baseline

5.2.2 Network

5.3 Experimental Setup

Measure per sequence the amount of people who have crossed the Line of Interest.

5.3.1 Sample generation

5.3.2 Metrics

The accuracy of the models is measured using two measurements, the Mean Average Error (5.1) and the Mean Squared Error (5.2).

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |G_l^{(i)} - P_l^{(i)}| + |G_r^{(i)} - P_r^{(i)}| \tag{5.1}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (G_l^{(i)} - P_l^{(i)})^2 + (G_r^{(i)} - P_r^{(i)})^2 \tag{5.2}$$

Where $G_l^{(i)}$ is the ground truth for sample i for side left-to-right. And $P_r^{(i)}$ is the predicted value for right-to-left.

In this thesis we make use of four different datasets. Three of them are already public datasets and one dataset is created using City of Amsterdam footage.

5.4 Datasets

In this thesis we make use of four different datasets. Three of them are already public datasets and one dataset is created using City of Amsterdam footage.

5.4.1 UCSD

The UCSD Pedestrian dataset [Chan and Vasconcelos, 2008] is a public dataset created in 2008. The dataset is in black and white and has only a resolution of 234x158. It has 6 scenes with each around 30 videos which each has around 10 seconds at 10fps of footage. Only a small amount of videos has precise labeled data for Crowd Counting. Most of the other video's have small parts of information about the pedestrians crossing.

The UCSD dataset is in previous papers used as default benchmark. Although other datasets do represent the capabilities of the presented methods much better, this dataset is used to give some comparison with older methods.

Explain more in detail which labeled data is present and add other data papers

5.4.2 CrowdFlow

The CrowdFlow dataset [Schroder et al., 2019] is a public dataset generated at the TU Berlin in 2018. The dataset contains 10 sequences generated from 5 different scenes. Each scene is captured once with a drone view camera and once with a fixed view camera. Each scene is a virtual urban environment and it is generated in the Unreal Engine, a 3D-engine. Each sequence is roughly 10 seconds long with 25 fps with a resolution of 1280x720. The generated sequences are dense in pedestrians and have up to 1451 pedestrians in a single frame. All the camera views are captured from a high surveillance style view.

5.4.3 Fudan-ShanghaiTech

The Fudan ShanghaiTech dataset [Fang et al., 2019] is a public dataset with 100 videos of 13 different scenes. Each video contains 6 seconds of footage at 25 fps and have a resolution of 1920x1080. The scenes have between 20-100 pedestrians per frame. In each frame the pedestrians in the frame are labeled with a bounding-box and a center-point of the bounding-box.

5.4.4 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.

Chapter 6

Results

6.1 UCSD

- Table with Region of Interest
- Table with Line of Interest

6.2 CrowdFlow

- Table with Region of Interest
- Table with Line of Interest

6.3 Fudan-ShanghaiTech

- Table with Region of Interest
- Table with Line of Interest

6.4 ArenaPeds

- Table with Region of Interest
- Table with Line of Interest

6.5 Flow estimation impact

Qualitative comparison

Chapter 7

Discussion

Bibliography

[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.

[Chan and Vasconcelos, 2008] Chan, A. B. and Vasconcelos, N. (2008). Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):909–926.

[Fang et al., 2019] Fang, Y., Zhan, B., Cai, W., Gao, S., and Hu, B. (2019). Locality-constrained spatial transformer network for video crowd counting. *Proceedings - IEEE International Conference on Multimedia and Expo*, 2019-July:814–819.

[Horn and Schunck, 1981] Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203.

[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.

[Schroder et al., 2019] Schroder, G., Senst, T., Bochinski, E., and Sikora, T. (2019). Optical Flow Dataset and Benchmark for Visual Crowd Analysis. *Proceedings of AVSS 2018 - 2018 15th IEEE International Conference on Advanced Video and Signal-Based Surveillance*.

[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.

[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.

Appendix A

Appendix