TÉCNICO LISBOA

Extended Abstract

City Cam: Similarity based classification of vehicle count time series

Tomás Matos Fernandes da Cunha Cordovil

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. João Paulo Salgado Arriscado Costeira
Carlos Jorge Andrade Mariz Santiago

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. João Paulo Salgado Arriscado Costeira
Member of the Committee: Prof. Pedro Manuel Quintas Aguiar

November 2021

Chapter 1

Introduction

In this section the motivation and context for this thesis are explained. Additionally the data capture process is also introduced.

With the present and assumed future rapid growth of inhabitants of urban areas, the number of road vehicles in these areas also witnesses a substantial increase. A direct result of this is a rise in the frequency of occurrences of traffic congestion situations. Since mobility in urban scenes is becoming more and more relevant because of its impact in the economy and the environment, being able to better characterize and predict traffic can be a valuable tool to improve circulation. Due to their low price and maintenance costs,video cameras are a widespread tool for traffic monitoring by being mounted in roads, city streets, inter-sections and parking lots [1]. They are able to capture data at all times giving continuous information on traffic states. With such a volume of data being produced, it can be expected that some kind of pattern will emerge with its analysis.

This thesis's objective is to propose a process to analyze this data in order to identify and find similarities between traffic patterns consisting of vehicle counts over segments of time at one specific or multiple geographic locations and classify the segments based on this notion of similarity. The main contribution of this work is the application of a Sparse Subspace Clustering algorithm to obtain the similarity metric and classification of the time segments.

We process still image data to obtain both vehicle count and traffic density estimations which in turn are used to generate traffic descriptors. The descriptors are then converted to graphs and a clustering algorithm is applied to the graph in order to extract similarities between groups of images.

The specific use case of this thesis revolves around Tallinn, the capital city of Estonia. We make use of Tallinn's camera network as a data source. The images are obtained through the use of a web crawler connected to the city's public website.

Chapter 2

Methodology

In this section we aim to present the main steps used to obtain and analyse the data mentioned in the previous section.

The data used in this thesis consists of images obtained from traffic surveillance cameras in the city of Tallinn with a web crawler connected to the city's public website ([2]). The script is fired every five minutes and during this interval. Each time the script is fired, a random number between 1 and 299 is generated and assigned to each camera. It represents how many seconds to wait before downloading one image of the corresponding camera. Images are downloaded and identified with both the id of the camera which obtained it and the timestamp generated upon the image capture. Data obtention was active at two distinct periods. The first one spans from 9 of March to 6 of April(except 11, 12 and 13 of March) 2020 and comprises 734284 images The second one spans from 2 3of August to 17 of September 2020 and includes the remaining 713293 images. In both cases, images were acquired between 8 A.M and 10 P.M or between 8:30 A.M and 8 P.M, depending on the selected period. Of the total 176 available cameras, we selected 11 to be the basis for this process.

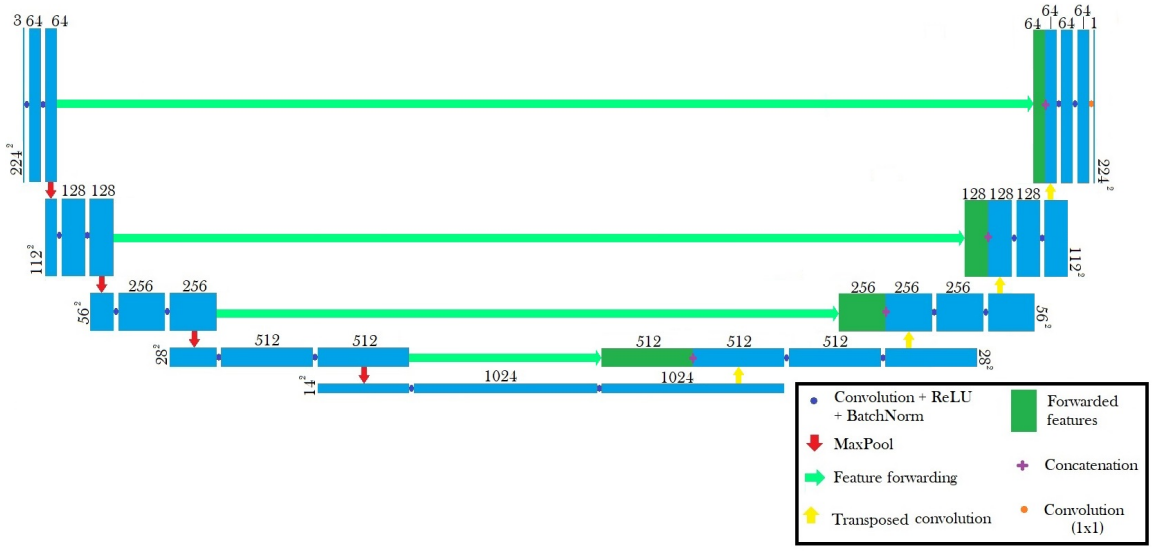

Figure 2.1: U-Net architecture

These images contain traffic density information which we want to extract in order to process and analyse it. We can break down the process into four main components: density estimation, vehicle count regression, sparse subspace representation and clustering. The end-to-end process receives sequence of images and outputs a classification relating time periods of the images based on a similarity metric for one or multiple cameras.

The number of vehicles in each image, or vehicle counts, are extracted from images using a neural network. The neural network follows the U-NET architecture as shown in Figure 2.1 ([3], [4]).

This is a convolutional neural network which is composed of an encoder followed by a decoder. The encoder contains five convolutional blocks and the decoder contains four. The output of the first convolutional block of the encoder shall be the input of the second block of the encoder but will also be used as part of the input of the forth block of the decoder. These connections between the encoder and the decoder are called skip connections and should allow the retention of low-level information in the inference. The input of the network itself is an RGB image of size 224x224 and the output is a grayscale image of the same size called density map. In the output image/map we expect to have the image's vehicle density. An example can be seen in Figure 2.2 The density concept used here is a way of quantifying vehicles in an image. In this case, the sum of the pixel values of an output density map should be equal to the number of vehicles in the corresponding input image ([5]). This principle is also applicable to sections of the image.

Figure 2.2: Example of annotated image and corresponding ground truth density map

The results of inference over a collection of n images is thus a same sized collection of density maps. By doing a sum over all pixels of each map, n vehicle counts are obtained. Using the timestamp generated at the moment of acquisition of each image together with the vehicle counts, time-series of vehicle counts over time are obtained for each camera. Because the interval at which samples were obtained was not constant, an interpolation step is performed to force all cameras to have the same number of samples and make all samples be equally spaced. For each camera, this interpolated time-series is broken down into smaller time-series, each comprising one day's worth of image acquisition. As a result, for a camera which contains k datapoints per day and was active for n days, a matrix of size $n \times k$ is obtained. This matrix, designated as daily vehicle count descriptor, contains in each column

all vehicle counts obtained for a specific day, and in each row, all datapoints obtained at a specific time for every acquired day. A direct analysis of the descriptors can be performed to gather some intuitive information regarding the distribution of traffic density throughout a day and across multiple days.

To reduce the dimensionality of the data, an alternate representation of the descriptor is generated through the use of an optimization program ([6]). This alternate representation consists in expressing a sequence of vehicle counts as a linear combination of a few other sequences of vehicle counts. Thus, every sequence can be reconstructed as a sum of the other sequences multiplied by their constant factor and adding a reconstruction error. The collection of the factors used in this reconstruction form the so-called sparse subspace representation of the data. The subspace representation C is a square matrix of size $n \times n$ where n is the number of columns of the used daily vehicle count descriptor and each cell $c_{i,j}$ contains the constant associated with column j in the reconstruction of column i. To avoid the obsolete solution of reconstructing a column with itself, the condition $c_{i,i} = 0$ is imposed. Since a sparse representation is desired, the sum of cell values is minimized with a $l1$ norm. With this alternative representation of the data (sparse subspace representation) a similarity graph W is obtained by adding the absolute value of C to the absolute value of C^T. In the similarity graph W, the cell $w_{i,j}$ contains the weight or similarity of the edge connecting the nodes or days i and j. Figure 2.3 shows a daily count descriptor, its sparse representation matrix, the reconstruction error associated with that representation and the affinity matrix that resulted from it.

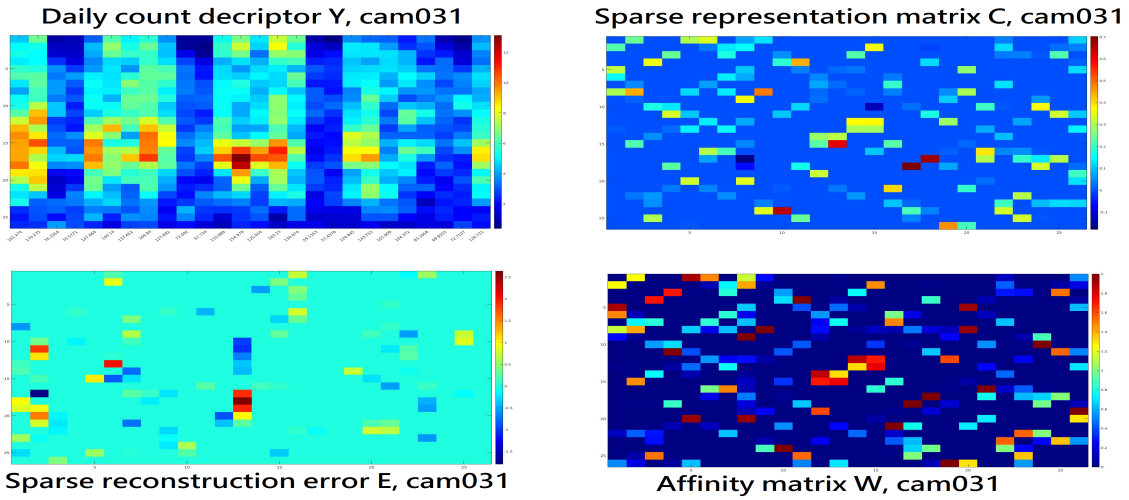

Figure 2.3: Top left: Daily count descriptor, Y, of Cam031. Top right: Sparse subspace representation C, using $\lambda_e = \lambda_z = 1 \times 10^{-3}$. Bottom left: Reconstruction error E. Bottom right: affinity matrix W (each with its own scale)

With the data now converted to a graph, it is possible to apply graph clustering methods which will classify time intervals of traffic. The chosen method was spectral clustering. The technique itself consists in utilizing a measure of the quality of data partitions, the normalized cut (N_{cut}). Using spectral clustering in the graph (affinity matrix, W) will indicate the partition which minimizes the weight associated with removing edges to obtain the desired number of clusters. The number of clusters (k) is manually set and must be chosen in accordance with the problem at hand.

Chapter 3

Conclusions

At the end of all the steps described in the previous section, we can effectively say that we are grouping time segments of traffic images based on a notion of similarity. As a byproduct, we also define a method of describing the traffic pattern captured by a camera in a time segment as a linear combination of other time segments of the same dimensions. As there are no means to qualify the results of the classification a subjective analysis was performed.

This work is open for further development in most of its parts, namely the dataset, the estimation of the number of vehicles in a traffic image and the pre-processing which precedes the Sparse Subspace Clustering algorithm. Other clustering methods can also be employed and the results of their employment compared to understand if they have benefits or short-comings in this use-case. The analysed dataset would benefit from being larger and continuous instead of being composed of two distinct intervals. This could bring both more data and an understanding of the different patterns which occur throughout the year. As the neural network was not trained on the same dataset that was used for analysis, the inference results could benefit from using some form of domain adaptation. Another approach would be to simply train the network with images obtained from the same cameras that were used in the analysis. In the subspace representation step, the process is sensitive to time shifts which can make the whole process less robust. Adding a step before to take this flaw into consideration could have positive results.

Bibliography

[1] S. Zhang, G. Wu, J. P. Costeira, and J. M. F. Moura. Understanding traffic density from large-scale web camera data. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4264–4273, 2017. doi: 10.1109/CVPR.2017.454.

[2] C. of Tallinn. Tallinn cam network, 2020. URL `https://ristmikud.tallinn.ee/index.php/cams`.

[3] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL `http://arxiv.org/abs/1505.04597`.

[4] L. Ciampi, C. Santiago, J. P. Costeira, C. Gennaro, and G. Amato. Unsupervised vehicle counting via multiple camera domain adaptation, 2020.

[5] W. Xie, J. A. Noble, and A. Zisserman. Microscopy cell counting and detection with fully convolutional regression networks. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 6(3):283–292, 2018. doi: 10.1080/21681163.2016.1149104. URL `https://doi.org/10.1080/21681163.2016.1149104`.

[6] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013. doi: 10.1109/TPAMI.2013.57.