© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Counting People based on Linear, Weighted and Local Random Forests

Helia Farhood, Xiangjian He, Wenjing Jia Global Big Data Technologies Centre University of Technology Sydney Sydney, Australia Helia.Farhood@student.uts.edu.au {Xiangjian.He, Wenjing.Jia}@uts.edu.au

Michael Blumenstein

Centre for Artificial Intelligence
University of Technology Sydney
Sydney, Australia
Michael.Blumenstein@uts.edu.au

Hanhui Li

School of Data and Computer Science
Sun Yat-sen University
Guangzhou, China
lihanhui@mail2.sysu2.edu.cn

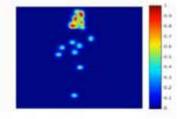
Abstract-Recently, many works have been published for counting people. However, when being applied to real-world train station videos, they have exposed many limitations due to problems such as low resolution, heavy occlusion, various density levels and perspective distortions. In this paper, following the recent trend of regression-based density estimation, we present a linear regression approach based on local Random Forests for counting either standing or moving people on station platforms. By dividing each frame into sub-windows and extracting features with ground truth densities as well as learned weights, we perform a linear transformation for counting people to overcome the perspective problems of the existing patch-based approaches. We present improvements against several recent baselines on the UCSD dataset and a dataset of CCTV videos taken from a train station. We also show improvements in speed compared with the state-of-the-art models based on detection and Deep Learning.

Keywords—Density estimation; crowd counting; Random Forest; linear regression

I. INTRODUCTION

Crowd counting and density estimation has become one of the most challenging tasks in intelligent visual surveillance systems. Automated crowd counting and density estimation plays an essential role in many real-world applications such as estimating people in public places, counting various objects (e.g., trees, cars or cells), as well as measuring the dynamic crowd to control overcrowd disasters and understanding crowd behavior. Therefore, the analysis of crowd size and crowd behaviors has been a topic of great interests in computer vision research community. Recently published works [1-7] show that crowd counting may have already been reasonably addressed. However, when being applied to some real-world scenarios such as train station CCTV videos (see Fig. 1, for example), the performance of existing solutions have dropped significantly. We have observed that some real-world problems are the contributors, including low resolution of images/video, heavy occlusion of subjects, diverse crowd densities, various people's sizes, unusual situations in large scale, time-consuming computation of adopting complex network models, etc. In one word, existing solutions are not sufficient to handle these realworld problems.





Estimated Count: 17.1 True Count: 18.0

Fig. 1. Counting people based on crowd density estimation.

Among various existing solutions, during the past two or three years, there have been several influential publications [15, 26, 28] adopting Deep Learning techniques for crowd density estimation. In this paper, in order to address the above mentioned problems, we propose a regression-based density estimation approach without having to use the resource-expensive Deep Learning techniques. The main innovation and difference between our approach and the existing regression-based approaches is that we use weighted Local Random Forests for a linear regression, of which the performance has demonstrated to outperform the-state-of-the-arts. The main contributions of this paper are listed as follows.

- We propose to count people with high accuracies in public CCTV scenes such as a train station based on crowd density estimation, as shown in Fig 1.
- Our model is learned with a simple linear regression based on weighted and Local Random Forests for estimating crowd density in a frame, and avoids the tasks of people detection, as well as gathering huge learned data for learning deep networks.
- Unlike the existing methods, our method can estimate the count of people accurately without going through a complex computation.
- Instead of using features based on randomly selected patches, which are time consuming to use and have perspective problems, we divide each image into fixed sub-windows.

Our goal is to estimate the density of crowd on a platform no matter if they are moving or not, and then estimate the total count by integrating the estimated densities over the whole frame. Given a set of training images in different situations such as high density, medium density and low density, our regression model based on Local Random Forests is learned. The best match between the estimated density function for the training images and the ground truth densities are learned. Then, we estimate the count in testing images as a summation of learned Local Random Forests with learned weights. Our experimental results show its effectiveness especially in the scenes where people have overlapping.

The rest of the paper is organized as follows. We first discuss related works in Section II. Section III describes our model based on Local Random Forests. Section IV introduces the datasets and presents experimental results. The paper concludes in Section V.

II. RELATED WORKS

In general, the existing people/crowd counting approaches can be grouped into two broad categories [1]: direct and indirect.

The direct approaches, based on human detection, rely on detecting each person (either the whole body or head-shoulder) in the scene using various classifiers and then counting them [3-5]. In these approaches, counting people can be modeled at a global scene level as well as correctly segmented scene level. With the advance and success in human detection, counting people becomes a by-product once each individual is correctly detected. The features that may be used include body, head, shoulder, skin, and hair [9, 10]. The benefit of these approaches is that they have a high level of accuracy. However, for the situations with highly dense crowd (such as stadium) where there are heavy people overlapping and occlusion [5], detecting individuals can become very difficult and not applicable when there are more crowed people with overlapping or low resolution of images. Therefore, detection-based approaches become helpless and are not suitable for large crowds. Such detection problems can be addressed using a head-only detector. Some of detection-based approaches by using human shapes attempt to segment or detect every single person and then count them [11]. Nevertheless, some other detection-based approaches try to detect each independent motion in the image via clustering interest points on people tracked spanning time and then count the people [10, 12-16].

Indirect approaches are based on detecting features and map them onto the count value instead of detecting individuals. It seems more logical. Many features of images have been used such as foreground areas [7] and texture features [2]. These approaches include edge counting and regression-based crowd density estimation. One advantage of these approaches is that they can be scaled to high-density scenes. When handling highly crowded scenes, indirect, regression-based or feature-based approaches are considered to be more powerful and robust, and therefore have become dominant. In these approaches, counting people is equivalent to density estimation and the integral over the whole frame/image produces the total estimated count. Density is provided by extracting some features using a learning algorithm or through a statistical analysis on an entire image [6-8]. These approaches have a problem of handling significant perspective distortions. Some techniques have been presented in the literature to deal with the perspective problem, e.g., a geometric correction to conduct all the objects at different distances to the same scale [17].

Counting by Regression: In [18] the authors presented a simple model like a regression model for counting objects such as people and cells. They used Maximum Excess over Sub Arrays (MESA) distance and used Random Forest for creating their codebook. In this paper, we improve the learning step via a weighted regression based on Local Random Forests for getting more accurate results. Chen et al. [19, 20], for counting objects, performed a regression based on the low-level imagery features. The authors of [20] used a ridge regression, and their inputs are local features from local regions and their output is the count of people. The authors of [19] presented the concept of cumulative attribute for regression and solved the problems of feature inconsistency and sparse data. In their work, by defining a set of training images, they extracted low-level features of images, and the numeric output value such as people count is converted into a binary cumulative attribute vector. Then, the cumulative attribute vector is provided into a single output regression model to estimate the output value as numerical. Loy et al [21] used semi-supervised learning for regression model.

Using Random Forests: Pham et al. [22] and Fiaschi et al. [23] used Random Forest regressions and non-linear models based on patch features for prediction by majority voting. Using image patches in the learning step has shown its strength but it also has some limitations. For example, in the situations where the video frames being investigated do not have enough resolution, the accuracies of these approaches are very limited. Furthermore, a patch is a sliding window and it is selected randomly.

In this paper, we replace the randomly selected patches by fixed sub-windows. The sub-windows are divided from a frame and used to extract dense SIFT [27] features. Then, we used a linear regression in our approach to achieve better results.

III. THE PROPOSED APPROACH

Fig. 2 shows the framework of our approach with the example of a crowded scene on a train station. The goal of the proposed system is to count the number of passengers,

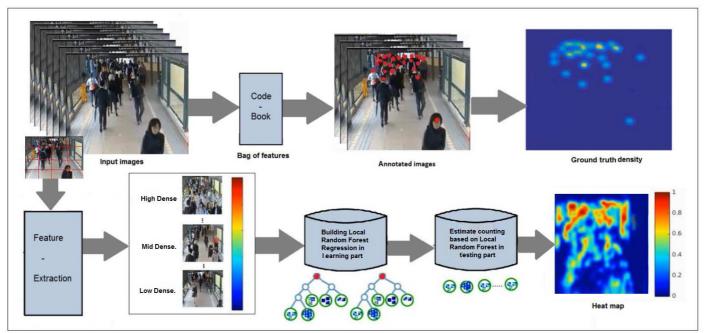


Fig. 2. The framework of our approach with input images, learning step and testing step.

Including moving and standing people in each platform. Input of this system includes the extracted features of raw images that are divided to N sub-windows (in our work N=8), annotated images and a codebook. After a learning step using a weighted and Local Random Forest for each sub-window, we can estimate the number of people based on a summation, using a linear function for the weighted and Local Random Forests in the testing step. The details of our approach are illustrated in the following subsections.

A. Learning Decision Tree

Given a set of M training images, denoted as I_i $i \in 1,2 \dots M$, with annotations, where each annotation is one pixel of dot located at the center of each passenger. Note that, an annotation is to specify the position of a person by putting a single dot on the person (roughly the center of the head; known as "dotting") in each frame of video, and it is easier than the bounding-box annotation. Dotting is a convenient and less difficult way for humans to count objects compared with other methods such as using a bounding box, especially when the number of objects in images is large. The density function is integrated over the whole image region to create the object count. The ground truth density function for each pixel $p \in I_i$ is defined as a kernel density estimation function centered at the annotated dots:

$$F^{0}(p) = \sum N(p; Y, \sigma^{2}), \quad Y \in A.$$
 (1)

Here, A is a set of annotated pixels and $N(p; Y, \sigma^2)$ shows a 2D Gaussian kernel generated on each dot $Y \in A$ with a small variance σ (in our work, $\sigma = 2.5$ pixels), which is a smoothness parameter. Note that the results obtained do not very much rely on the setting of σ and also when we normalize the base functions, the total number of objects in an image can be computed by Eq. (1). Given a set of training images together with their ground truth densities presented in Eq. (1), we first divide each frame to N sub-windows, and then extract the scale-invariant feature transform (SIFT) features on each sub-window and learn a decision tree F to project the features $x \in R^{h \times w}$ to a density map $d \in [0,1]^{h \times w}$, where h and w denote the height and width of each sub-window, respectively. The count of pedesitran c of each sub-window, can be obtained via performing the intergral operation over d:

$$c = \sum_{i=1}^{h} \sum_{j=1}^{w} d_{ij} .$$
(2)

As shown in Fig. 3, we estimate the density map of the testing frames based on the ground truth densities and the extracted features.

B. Building Local Random Forests

The robustness of a decision tree can be further improved by assembling multiple decision trees into a Random Forest. More specifically, a Random Forest is a mixture of decision trees such that each tree relies on the values of a random vector sampled separately and with the same scattering, for all trees in the forest.

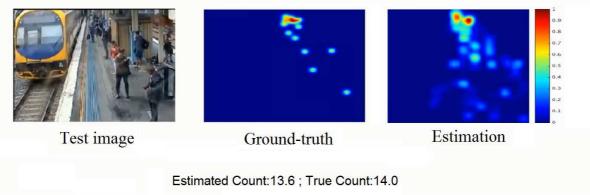


Fig. 3. Estimate the density function based on ground truth densities and extracted features.

As shown in Fig 4, a Random Forest regression is organized by developing trees relying on the associates of sequential predictions with various inputs. We learn a Local Random Forest for each sub-window via an approach similar to [22]. That is, we first extract the dense SIFT features for each sub-window. After that, we build the trees on a randomized subset of the training examples belonging to the same sub-window. The learning step proceeds repetitively, by splitting the training samples set S_{τ} , appearing at a node τ , into a left and right subsets S_{l} , S_{R} . In the test phase, we choose the split point with threshold value τ similar to that in [3]:

$$S_{L} = \{i \in \{1, ..., |S|\} | f(x_{i}) < \tau\}$$

$$S_{R} = S \setminus S_{L}.$$
(3)

Here, S denotes the set of test instances and |S| denotes the cardinality of S. Note that our method differs from conventional methods that our Random Forests are defined on the sub-window level (local). The local property of our method is vital for tackling the counting problem, since in real-life

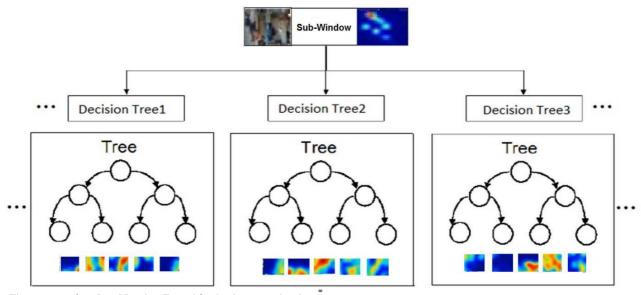
applications the density levels of patches in the same frame can vary dramatically. Conventional methods neglect that and attempt to tackle the counting problem via a single model, thus suffer from the problem of inconsistency of features.

C. Prediction step

In this section, we illustrate how to predict the number of people using a Local Random Forest regression in our approach. In comparison to the ordinary Random Forest, after the training process, we propose to use the linear combination of counts in sub-windows, to estimate the total count of people.

$$C = \sum_{n=1}^{N} c_n w_n. \tag{4}$$

Here, w_n is the weight of the n-th sub-window and is learned by minimizing the distance between the predicted numbers and ground-truth numbers on the training sets.



 $Fig.\ 4.\ \ The\ structure\ of\ our\ Local\ Random\ Forestd\ for\ density\ map\ estimation.$

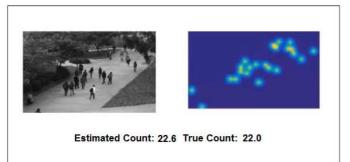


Fig. 5. A sample frame from the UCSD dataset and the ground truth.

D. Creating codebook

For the purpose of getting better result and also saving time, we create a bag of features of training samples, instead of using dense SIFT features directly. We perform K-Means on the training images to generate a codebook consisting of *K* terms. Then, the features of each sub-window are represented by the frequencies of the individual terms in the codebook. In this way, we convert the original high dimensional dense SIFT features into a feature vector of length *K*. Consequently, by using this codebook, we can easily save time and by creating codebook for one time it is possible to compare it with input features quickly.

I. EXPERIMENTS

In order to demonstrate the effectiveness of our proposed idea, we tested our approach on the widely used UCSD dataset [2], and our own Train Station dataset. Figs. 5 & 6 give the typical examples of each dataset. In this section, we present the detailed experiments and comparative results.

A. Datasets

We conducted experiments on two different datasets. Table 1 provides some information about the two datasets.

The UCSD Dataset: In order to compare our algorithm with the state-of-the-art approaches, we firstly ran experiments on the widely adopted UCSD dataset. The authors of [2] also published their annotated images for those frames, the positions of the annotation, and the regions of interests.

TABLE 1: DETAILS OF THE TWO DATASETS USED IN OUR EXPERIMENTS

Dataset	# of Frames	Resolution	FPS	Count per Frame	Total Counts
UCSD	2000	238*158	10	11- 46	49885
Train station	2000	256*256	4	1-53	62581

An example frame of this dataset and its ground truth density heat map obtained using our method (as shown in Eq. 1) are shown in Fig. 5. In our experiment, we selected 2000 fames from a relatively busy pedestrian street and used 400 frames for training with the rest for testing.

Train Station Dataset: In addition to the UCSD, we conducted comparative experiments on our own train station data, comprising real CCTV surveillance video footages from a train station. Note that, due to public privacy concerns, the train station data cannot be made publicly available. The details of the dataset are also shown in Table 1. Fig. 6 shows two example frames from this dataset and their estimated density heat maps using our approach.

As Fig. 6 shows, this dataset is very challenging because of the following three reasons: 1) due to the high compression ratio applied, the resolution of the video is very limited, which results in that much of the subjects' details, especially at a certain distance away, are unable to be recognized by human eyes; 2) the mounting angle of the surveillance camera results in there is heavy occlusion among most of the people, not to mention various obstacles that block the camera view; and 3) in different times of day, passengers of significantly different density may appear on the platforms, moving or still.

Similarly, we selected 2000 frames video of different camera views of the train station and annotated some of them as ground truth in order to compare the estimated count generated using our algorithm with the true count. We used 500 frames for learning. However, note that the initial 500 frames contain different situations including high density, medium density and low density. A region-of-interest (ROI) was selected for the platform only where there are passengers either standing or moving. Moreover, we proposed to address the problem of perspective distortion using sub-windows instead of image patches, as shown in Eq. (5). Comparative results of heat maps obtained on the train station dataset using our approach in different situations are shown in Fig. 6.

B. Evaluation Metrics

In order to quantitatively measure and compare the performance of various approaches, two metrics, i.e., the mean absolute error (MAE) and the mean squared error (MSE), which have been widely used in literature, were adopted. These two metrics are defined as below:

$$MAE = \frac{1}{N} \sum_{1}^{N} |z_{i} - \hat{z}_{i}|, MSE = \sqrt{\frac{1}{N} \sum_{1}^{N} (z_{i} - \hat{z}_{i})^{2}}.$$
 (6)

Here, N is the number of testing images, z_i is the true count of people in the i_{th} image, and \hat{z}_i is the estimated number of people in the i_{th} image. From their definitions, MAE also indicates the accuracy of the estimations, and MSE represents the robustness of the estimations.

Table 2: Comparative experimental results of different methods on the UCSD dataset

Methods	MAE	MSE
Ridge Regression [20]	2.25	7.82
Gaussian Process Regression [2]	2.24	7.97
Cumulative Attribute Regression [19]	2.07	6.86
Our Method	1.89	3.19

C. Comparing with the state-of-the-art

The comparative experimental results are presented in Table 2 in comparison with the existing approaches on the UCSD dataset, where existing approaches' results are directly cited from their publications. As can be seen from this table, compared with other regression-based methods, our approach has achieved the best accuracy.

For Train Station dataset, we compare the results obtained using our approach with two other approaches implemented by ourselves, i.e., Lempitsky et al. [18] and Fiaschi et al. [23], as shown in Table 3. As shown in this table, our method is faster and more accurate than the other two methods.

TABLE 3: COMPARATIVE EXPERIMENTAL RESULTS OF DIFFERENT METHODS ON THE TRAIN STATION DATASET.

Method	MAE	MSE	Runtime
Regression based [23]	2.57	3.25	82ms
Lempitsky et al. [18]	2.11	2.51	57ms
Our Method	1.67	1.86	41ms

From the above results, we can see that, our proposed approach has demonstrated promising results in real-world low-resolution video of various densities and significant perspective distortion and occlusion.

II. CONCLUSION

In this paper, we have presented an improved, regressionbased approach for crowd counting in low-resolution surveillance public spaces. We have focused our work on tackling a real-world problem using train station CCTV data, where 1) the resolution of frames was poor due to a high compression ratio, and 2) the densities of people on the platform at different times of the day vary significantly. In particularly, in order to improve the estimation accuracy, we have proposed to use the Local Random Forests for learning. Instead of using feature patches (i.e., sliding windows) selected randomly and used in a loop in order to compare with other parts of an image causing a time consuming process, we have divided each frame to fixed sub-windows and learned the density in each subwindow based on a Local Random Forest for faster process. We have tested our approach on a widely adopted dataset and a private train dataset and have achieved promising results. Compared with the recent Deep Learning-based solutions, our approach is much simpler and does not require a huge amount of training data. Furthermore, due to the simplicity of our approach, the processing speed is also satisfactory.

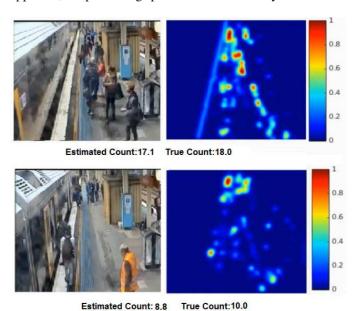


Fig. 6. Two examples of the estimated density heat maps of the train platform scenario. It shows that, with our approach, we have achieved very high accuracies.

For future work, by extending this approach to a more powerful estimator, it is expected to have further performance improvement.

References

- [1] S. A. M. Saleh, S. A. Suandi, and H. Ibrahim, "Recent survey on crowd density estimation and counting for visual surveillance," *Engineering Applications of Artificial Intelligence*, vol. 41, pp. 103-114, 2015.
- [2] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008, pp. 1-7.
- [3] B. Zhou, X. Wang, and X. Tang, "Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 2012, pp. 2871-2878.
- [4] X. Liu, P. H. Tu, J. Rittscher, A. Perera, and N. Krahnstoever, "Detecting and counting people in surveillance applications," in Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on, 2005, pp. 306-311.
- [5] Y.-L. Hou and G. K. Pang, "People counting and human detection in a challenging situation," IEEE transactions on systems, man, and cybernetics-part a: systems and humans, vol. 41, pp. 24-33, 2011.

- [6] Z. Zhang and M. Li, "Crowd density estimation based on statistical analysis of local intra-crowd motions for public area surveillance," Optical Engineering, vol. 51, pp. 047204-047204, 2012.
- [7] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Crowd counting using multiple local features," in Digital Image Computing: Techniques and Applications, 2009. DICTA'09., 2009, pp. 81-88.
- [8] A. Albiol, M. J. Silla, A. Albiol, and J. M. Mossi, "Video analysis using corner motion statistics," in IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, 2009, pp. 31-38.
- [9] J. Xing, H. Ai, L. Liu, and S. Lao, "Robust crowd counting using detection flow," in Image Processing (ICIP), 2011 18th IEEE International Conference on, 2011, pp. 2061-2064.
- [10] R. Khatoon, S. M. Saqlain, and S. Bibi, "A robust and enhanced approach for human detection in crowd," in Multitopic Conference (INMIC), 2012 15th International, 2012, pp. 215-221.
- [11] G. J. Brostow and R. Cipolla, "Unsupervised bayesian detection of independent motion in crowds," in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, 2006, pp. 594-601.
- [12] M. Li, Z. Zhang, K. Huang, and T. Tan, "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection," in Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, 2008, pp. 1-4.
- [13] V. Rabaud and S. Belongie, "Counting crowded moving objects," in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, 2006, pp. 705-711.
- [14] A. M. Cheriyadat and R. J. Radke, "Detecting multiple moving objects in crowded environments with coherent motion regions," ed: Google Patents, 2013.
- [15] X. Zeng, W. Ouyang, M. Wang, and X. Wang, "Deep learning of scene-specific classifier for pedestrian detection," in European Conference on Computer Vision, 2014, pp. 472-487.
- [16] H. Ma, C. Zeng, and C. X. Ling, "A reliable people counting system via multiple cameras," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 3, p. 31, 2012.
- [17] A. C. Davies, J. H. Yin, and S. A. Velastin, "Crowd monitoring using image processing," Electronics & Communication Engineering Journal, vol. 7, pp. 37-47, 1995.
- [18] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in Advances in Neural Information Processing Systems, 2010, pp. 1324-1332.
- [19] K. Chen, S. Gong, T. Xiang, and C. Change Loy, "Cumulative attribute space for age and crowd density estimation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 2467-2474.

- [20] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature Mining for Localised Crowd Counting," in BMVC, 2012, p. 3.
- [21] C. Change Loy, S. Gong, and T. Xiang, "From semisupervised to transfer counting of crowds," in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2256-2263.
- [22] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, "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, 2015, pp. 3253-3261.
- [23] L. Fiaschi, U. Köthe, R. Nair, and F. A. Hamprecht, "Learning to count with regression forest and structured labels," in Pattern Recognition (ICPR), 2012 21st International Conference on, 2012, pp. 2685-2688.
- [24] T. Van Gestel, J. Suykens, B. De Moor, and J. Vandewalle, "Automatic relevance determination for least squares support vector machine regression," in Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on, 2001, pp. 2416-2421
- [25] S. An, W. Liu, and S. Venkatesh, "Face recognition using kernel ridge regression," in *Computer Vision and Pattern Recognition*, 2007. CVPR'07. IEEE Conference on, 2007, pp. 1-7.
- [26] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 833-841.
- [27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91-110, 2004.
- [28] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 589-597.