Robust Hierarchical Deep Learning for Crowdedness Regression

Qi Wang, *Senior Member, IEEE,* Jia Wan, Xuelong Li, *Fellow, IEEE*

***Abstract*—Crowdedness regression such as congestion detection and crowd counting is important to protect public security. However, most of the existing algorithms are insufficient for real applications. Traditional features are not discriminative which results in rather poor performance under complex scenarios. The deep features can better represent high-level information, but the training of deep network for regression is difficult. To promote the crowdedness regression, a robust hierarchical deep learning is proposed for the task. In this method, a deep network is designed for hierarchical semantic feature extraction. Different from traditional deep regression networks which usually directly utilize mean squared error as loss function, a robust metric learning is employed to effectively train the network. Based on this, multiple networks are combined together to further improve the generalization ability. Extensive experiments are conducted and the proposed model is confirmed to be effective.**

***Index Terms*—Deep learning, visual surveillance, regression, congestion detection, crowd counting, ensemble learning, metric learning**

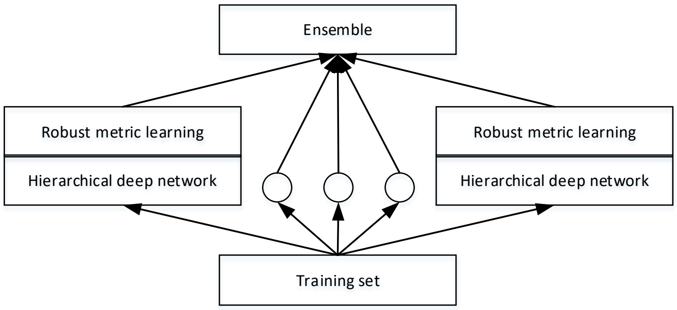


Fig. 1. The pipeline of the proposed system. The features of multi-level are extracted by a hierarchical network. To effectively train the network, a robust metric learning is utilized as loss function. Diverse deep networks are combined together for better generalization.

I. Introduction

In recent years, public security has become a serious problem in modern cities [1]. Many accidents caused by traffic congestion or extremely crowd result in death every year [2]. How to automatically detect congestion level and count crowd number is essential to address the problem. Since the cameras are very popular nowadays, automatically detecting congestion or counting crowd number is possible with those cameras.

The congestion detection and crowd counting have been researched for years. Traditional methods treat congestion analysis as a classification problem which divide congested videos into 2-5 classes. Since more congested images contain more objects, the moving blobs in traffic videos and their speed are usually used as features. Recently, a new regression perspective of congestion is proposed which utilizes a real continuous value from $[0, 1)$ as the accurate level of congestion. That makes congestion detection a regression task. Crowd counting is more popular than congestion detection as there exist more datasets. It's a typical regression problem which aims to predict the crowd number in an image. Traditional approaches include local information mining and holistic feature extraction. Recently, the deep learning based methods have shown potentials on the task.

Though both of the tasks have achieved great progress, they are still insufficient for real applications. There exist some problems which limit the performance and generalization ability. The first problem is the feature extraction. Most traditional methods extract low-level, mid-level or high-level feature to represent the images in simple scenario. However, there are many different scenes in real world. Under this circumstance, the performance of are limited. Though the deep learning based methods can achieve better performance, the training of these networks are difficult [3]. Based on the label (congestion level or crowd number), most of them need additional information to guide the training, like the position of each person which is very time-consuming to obtain. Moreover, the generalization ability is not adequate for real applications. Since large variation exists in different scenes, the generalization ability is quite important to improve the performance. Though ensemble learning [4] is a good choice to increase the generalization ability, how to train diverse network with limited samples is another challenge.

To address the existing problems, a robust hierarchical deep learning method is proposed for crowdedness regression. First of all, a hierarchical deep network which combines different level of features is designed for better feature extraction. Then, since the metric learning [5] is effective to embed structural information, a robust metric learning is proposed as the loss function to better optimize the hierarchical network. Finally, multiple networks are combined to improve the generalization ability.

To sum up, the contributions of this paper are as follows:

- To better represent congestion related proprieties, a hierarchical deep network is designed to combine different level of features together. The proposed network can be used to extract more discriminative feature.

The authors are with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China.

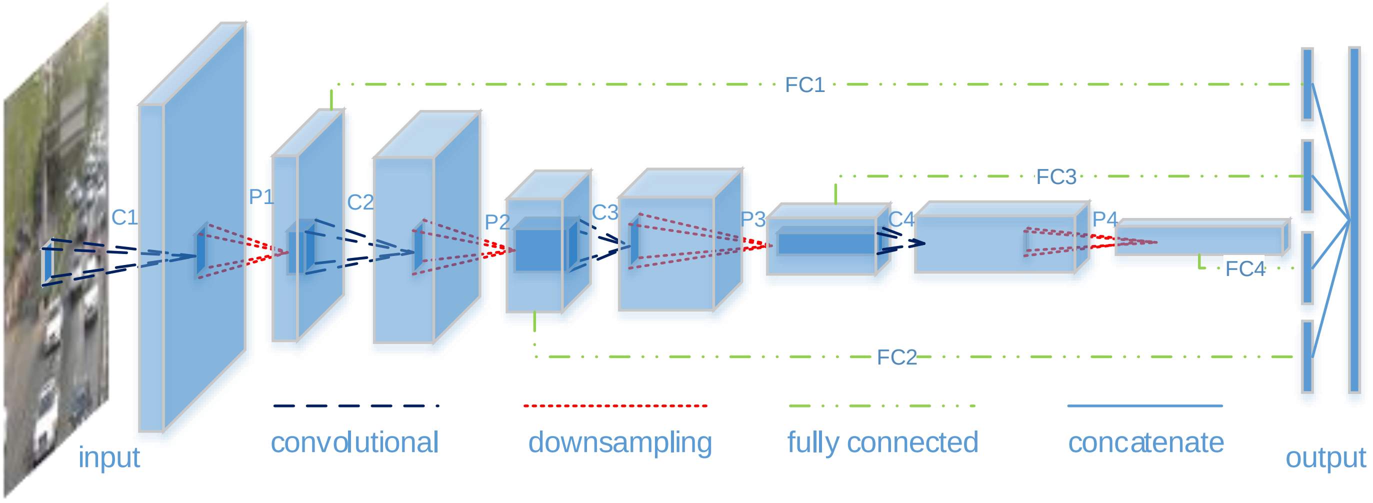


Fig. 2. The structure of hierarchical network. The input of this network is an image and the output of the network is the corresponding hierarchical feature. The detail parameters are shown in Table I.

- To efficient train the proposed network, a robust metric learning is utilized as the loss function. Comparing to traditional methods, the deep learning for regression is more effectively optimized with the proposed loss.
- To improve the generalization ability, multiple deep networks are combined together. Since the diversity of networks is preserved with our training method, the combination is effective.

The reminder of this paper is organized as follows. Section II reviews the relevant works of congestion detection and crowd counting. Section III elaborates the proposed method in detail. After the experimental results are reported and discussed in Section IV, the conclusion and future works are presented in Section V.

II. Related Work

In this section, we briefly review the related works of deep learning for regression, congestion detection and crowd counting.

A. Deep Learning for Regression

In recent years, deep learning has shown its powerfulness in many tasks such as person re-identification [6], image classification [7], face verification [8] and multi-modal learning [9]. Many regression based deep learning methods are highly correlated with specific tasks including object detection [10], pose estimation [11], landmark detection [12], *et al.*.

In object detection, Szegedy *et al.* [13] propose a multi-scale framework with an L_2 error to predict the mask of ground truth. Redmon *et al.* [14] propose to localize the objects in the cell with a multi-part loss function. With the preprocessing, the bounding box regression in [15] is very accurate. In pose estimation, Ouyang *et al.* [16] propose to combine part-based model [17] with deep learning, and the loss function is the sum of square error with the L_1 norm regularization term. Instead of utilizing part of body for estimation, Toshev *et al.* [18] propose to estimate the human pose directly with a cascade pose regressor. In landmark detection, Sun *et al.* [19] propose a three-level convolutional neural network to detect facial point in which multi-level regression is proposed. Zhang *et al.* [20] propose to detect facial points with a deep multi-task learning.

B. Congestion Detection

Traditional congestion detection can be roughly assorted into two categories. The first category is based on the analysis of moving objects in traffic videos. Another category is based on the holistic feature extraction and classification.

To classify congested videos of a single scene, the most direct way is to count the number of vehicles on the road. However, the vehicle detection techniques can not detect all vehicles especially under congested scenarios. Thus, the key points and moving blob in videos are used to represent the vehicles in many algorithms. He *et al.* [21] propose a method which represent the vehicles by moving blobs. The background subtraction algorithm [22] is first utilized to detect moving blobs. Then, the speed of these blobs is calculated by Optical Flow [23]. Finally, the fuzzy logical is used for final decision. Sobral *et al.* [24] propose to represent the vehicles by key points. In this work, the speed of key point is calculated by Kanade-Lucas-Tomasi (KLT) algorithm [25]. The performance of them are limited by preprocessing algorithms like background subtraction, key point detection and tracking.

To avoid the preprocessing, many algorithms based on density related features [26] are proposed. Derpanis *et al.* [27] propose to detect congestion videos through the visual dynamics. In particular, the proposed representation is extracted by a set of 3D Gaussian filters which considers spatial and temporal information simultaneously. Riaz *et al.* [28] propose to classify traffic congestion using motion vector statistical properties. K-Nearest Neighbor and Artificial Neural Network are evaluated for classification. Dallalzadeh *et al.* [29] propose a symbolic representation and the corresponding symbolic

TABLE I
THE DETAIL PARAMETERS OF THE HIERARCHICAL NETWORK.

Layer	Type	Input	Filter size	Filter number	Output
C1	Convolutional	$64 \times 64 \times 3$	$3 \times 3 \times 3$	32	$64 \times 64 \times 32$
P1	Pooling	$64 \times 64 \times 32$	2×2	1	$32 \times 32 \times 32$
C2	Convolutional	$32 \times 32 \times 32$	$3 \times 3 \times 32$	64	$32 \times 32 \times 64$
P2	Pooling	$32 \times 32 \times 64$	2×2	1	$16 \times 16 \times 64$
C3	Convolutional	$16 \times 16 \times 64$	$3 \times 3 \times 64$	128	$16 \times 16 \times 128$
P3	Pooling	$16 \times 16 \times 128$	2×2	1	$8 \times 8 \times 128$
C4	Convolutional	$8 \times 8 \times 128$	$3 \times 3 \times 128$	256	$8 \times 8 \times 256$
P4	Pooling	$8 \times 8 \times 256$	2×2	1	$4 \times 4 \times 256$
FC1	Fully connected	$32 \times 32 \times 32$	$32 \times 32 \times 32 \times 10$	1	10
FC2	Fully connected	$16 \times 16 \times 64$	$16 \times 16 \times 64 \times 10$	1	10
FC3	Fully connected	$8 \times 8 \times 128$	$8 \times 8 \times 128 \times 10$	1	10
FC4	Fully connected	$4 \times 4 \times 256$	$4 \times 4 \times 256 \times 10$	1	10

method to detect congestion videos. These methods achieve good performance for one specific scene. However, these methods can not used for real applications since the congestion detection with multiple scenes is still challenging.

C. Crowd Counting

Crowd counting algorithms can be divided into two classes: holistic and local methods. Holistic algorithms count crowd number from the whole image directly while the local algorithms from patches.

Most holistic methods use textures, foreground pixels and edges as features which can distinguish different crowd size efficiently. Marana *et al.* [30] propose to measure the crowd density with a Gray Level Cooccurrence Matrix (GLCM) based features. Regazzoni *et al.* [31] propose a crowd counting method based on the edge features. Cho *et al.* [32] propose a novel background subtraction techniques to model a human observer. The performance of these algorithms is limited since the crowd behavior has large variations.

Instead of taking the whole image as input, the local approaches first split image into patches. Then, the crowd estimation is performed on the patch. Finally, the crowd number of these patches are accumulated to final crowd size. Chen *et al.* [33] propose a method in which the feature is extracted over the grid cells. Lempitsky *et al.* [34] propose to estimate the crowd size of each pixel and then all pixels are accumulated to final count. Fiaschi *et al.* [35] propose to utilize random forest to promote the regression of crowd count. Chen *et al.* [36] propose to aggregate the local and global information to better represent the crowd scene. Tang *et al.* [37] propose an multiview people counting method based on different camera views.

Recently, the deep learning has show great potential on many tasks as well as crowd counting. Zhang *et al.* [38] propose a deep network to estimate the crowd size by iteratively learning the density map and the global number. Zhang *et al.* [39] propose to count the crowd size with a multi-column convolutional neural network (MCNN). Though the deep learning based methods achieve better performance, the training of these networks needs additional information which is hard to label.

III. OUR METHOD

In this section, the proposed hierarchical deep robust metric learning ensemble is presented in detail. A deep network is first designed for hierarchical feature extraction. Then, the robust metric learning is utilized as loss function. Finally, multiple deep networks are combined to further improve the generalization ability.

A. Hierarchical Feature Extraction Network

To extract hierarchical features to better represent the congestion, a deep network is designed. Traditionally, the deep learning is utilized to extract high-level semantic representation. However, we find that the low-level and mid-level features are also helpful for congestion detection. Thus, a hierarchical feature extraction network is designed to combine the features from different levels.

The proposed Hierarchical Network (HNet) consists of 4 convolutional layers and 4 fully connected layers. The structure of the proposed network can be seen in Figure 2 and the detail parameters are summarized in Table I.

Formally, given an image x as input, the hierarchical feature $f_h(x)$ is calculated by concatenating the feature of each level $f_i(x)$ as follows:

$$f_h(x) = f_1(x)||f_2(x)||f_3(x)||f_4(x), \tag{1}$$

where $||$ indicates concatenation. To calculate the features of different levels, the Convolutional Neural Network (CNN) is utilized in which the output of the first layer is:

$$h_i(x) = \phi(W_1^c x + b_1^c), \tag{2}$$

where W_1^c is the projection matrix of the first layer and b_1^c is the bias. ϕ is the activation function. Based on the output of the previous layer h_i, the output of next layer can be calculated as:

$$h_{i+1}(h_i) = \phi(W_{i+1}^c * h_i + b_{i+1}^c), \tag{3}$$

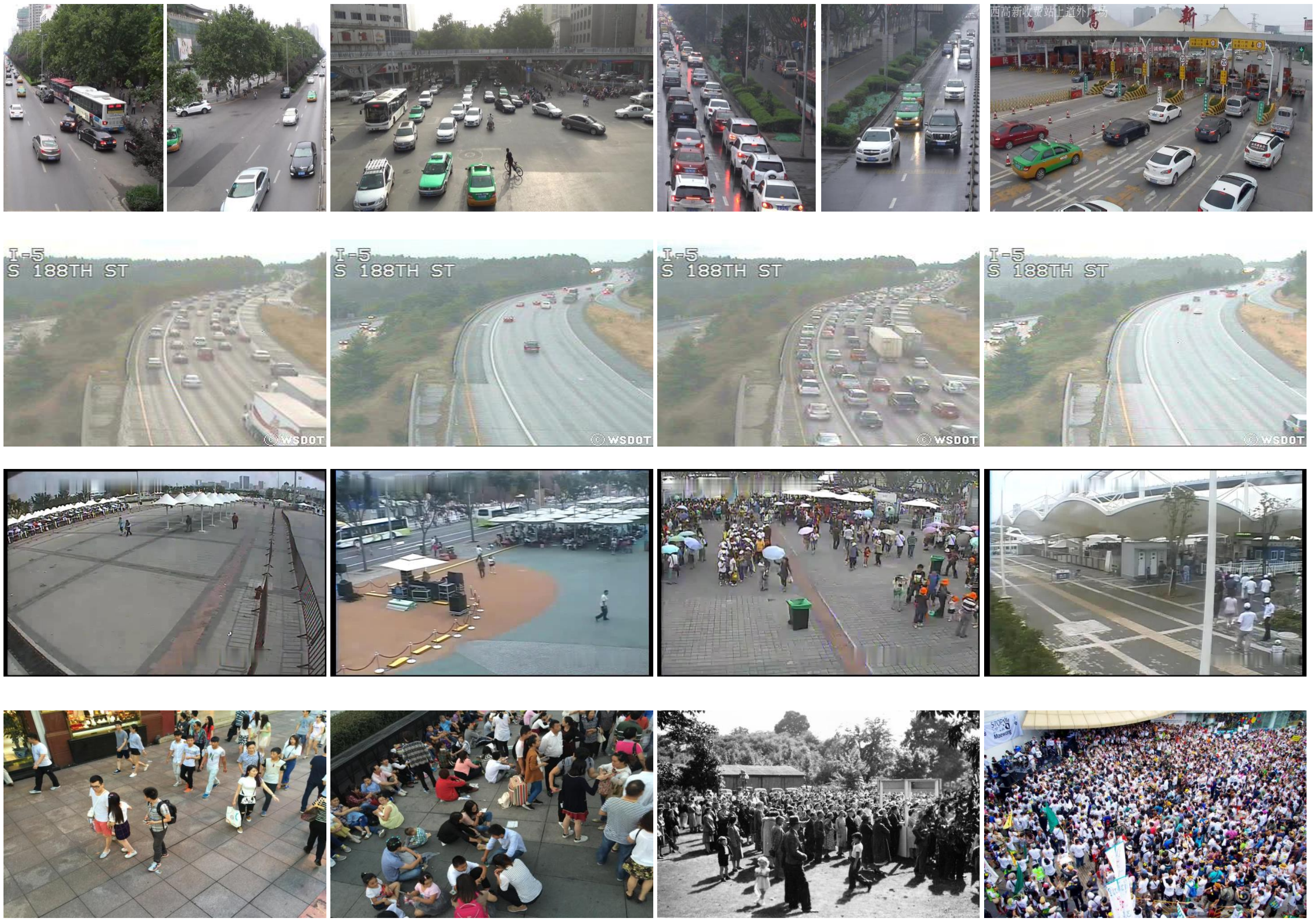


Fig. 3. The datasets used for evaluation. The first to last raw are NWPU_Congestion dataset, traffic video database, WorldExpo dataset and shanghai_Tech respectively.

where the W^c_{i+1} is the projection matrix of the $(i+1)$-th layer, and the b^c_{i+1} is the corresponding bias. With the output of each convolutional layer, $f_i(x)$ are defined as below:

$$f_i(x) = W^f_i h_i(x) + b^f_i, \tag{4}$$

where W^f_i and b^f_i are parameters of fully connected layers.

B. Robust Metric Learning

Since the only label of congestion is a real value, the optimization of the proposed network is relatively complex. The mean squared error is usually utilized as the loss function traditionally. However, the performance is very limited from the experiment. The metric learning [40] is a technique to improve the distance measurement among samples which is effective to embed the structural information. To effectively train the network, a robust metric learning is employed as the loss function to embed more structural information and guide the training of the network.

Specifically, given n training examples $\{x_1, x_2, ..., x_n\}$ and the corresponding labels $\{y_1, y_2, ..., y_3\}$, the prediction $\hat{y}$ of a new sample x is:

$$\hat{y} = \frac{\sum_1^n y_i k(x, x_i)}{\sum_1^n k(x, x_i)}, \tag{5}$$

where $k(x, x_i)$ is the weight based on the distance of x and x_i which can be computed as:

$$k(x, x_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{d(x, x_i)}{\sigma}), \tag{6}$$

where $d(x, x_i)$ is the similarity between x and x_i. To measure the similarity, a Mahalanobis Distance is used instead of traditional Euclidean distance which is defined as:

$$\begin{aligned} d(x, x_i) &= (f_h(x_i) - f_h(x))^\top M (f_h(x_i) - f_h(x)) \\ &= (f_h(x_i) - f_h(x))^\top L^\top L (f_h(x_i) - f_h(x)) \\ &= \|L(f_h(x_i) - f_h(x))\|_2^2, \end{aligned} \tag{7}$$

where $f_h(x)$ and $f_h(x_i)$ are hierarchical features of x and x_i. M is the distance metric which can be decomposed into $L^\top L$ where L is the feature transformation. Since $f_h(x)$ is a hierarchical non-linear mapping, more discriminative information can be exploited. To eliminate the outliers, the parameters in the proposed model can be learned by optimizing the following

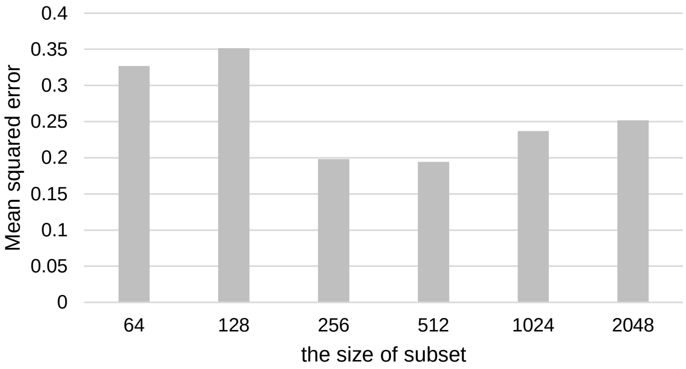
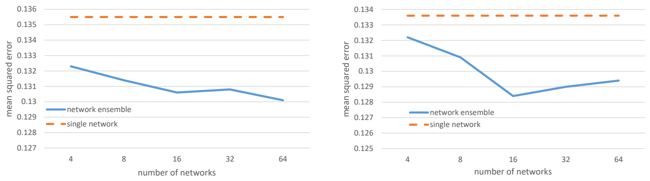


Fig. 4. The mean squared errors with respect to different sizes of subset. The horizontal axis is the size of subset and the vertical axis is the mean squared error.



(a) 2 covolutional layers (b) 3 covolutional layers

Fig. 5. The mean squared error with respect to the number of networks for ensemble. The horizontal axis is the different number, and the vertical axis is the mean squared error. The left diagram is evaluated on a two level network and the right diagram is evaluated on a three level network.

Algorithm 1 Hierarchical deep robust metric learning ensemble

```
Input:  $X \in \Re^{n \times 64 \times 64 \times 3}$: the training images
        $Y \in \Re^{n \times 1}$: the labels of $X$
        $n \in \Re$: the number of epochs
        $\alpha \in \Re$: the learning rate
// For each deep network
for $i = 1, 2, ..., m$ do
    Split $(X, Y)$ into $(X_s, Y_s)$, $(X_t, Y_t)$ and $(X_v, Y_v)$
    Initialize $W$, $b$
    // For each training epoch
    for $j = 1, 2, ..., n$ do
        for $(X_{batch}, Y_{batch})$ in $[X_t, Y_t]$ do
            // Forward propagation
            Calculate $f(X_s)$ via Equation 1
            Calculate $f(X_{batch})$ via Equation 1
            // Compute the loss
            Calculate $\mathcal{L}$ via Equation 10
            // Compute gradients
            Calculate gradients $\frac{\partial \mathcal{L}}{\partial L}$, $\frac{\partial \mathcal{L}}{\partial W}$ and $\frac{\partial \mathcal{L}}{\partial b}$
            // Update parameters
            $W = W - \alpha \frac{\partial \mathcal{L}}{\partial W}$
            $b = b - \alpha \frac{\partial \mathcal{L}}{\partial b}$
        end for
    end for
    // Test the network
    Calculate $err_i$ with $(X_v, Y_v)$
end for
Output: $m$ deep models and the corresponding error $err$.
```

robust loss function:

$$\mathcal{L} = \sum \phi_{hub}(x, x_i), \tag{8}$$

where $\phi_{hub}(x, x_i)$ is defined as:

$$\phi_{hub}(x, x_i) = \begin{cases} (x - x_i)^2 & \text{if} |x - x_i| \le \tau \\ \tau(2|x - x_i| - \tau) & \text{if} |x - x_i| > \tau, \end{cases} \tag{9}$$

To avoid over-fitting, the $l2$ regularization term is added on parameters and then the loss function is written as:

$$\mathcal{L} = \sum \phi_{hub}(x, x_i) + \frac{\alpha}{2} \sum_{W} W^2. \tag{10}$$

To sum up, the details of proposed model is shown in Algorithm 1.

C. Deep Learning Ensemble

Since the there exists many scenes different from the training set in real applications, the generalization ability is desired. In our experiments, we find the ensemble learning is very useful to improve the generalization ability. Thus, multiple networks are trained and combined together as shown in Figure 1.

Traditionally, all training samples are used to predict the congestion level of a sample. To increase the divergence of different deep networks. A subset of training set is used for prediction instead of the whole training images. Specifically, before each deep network is training, we first randomly select s samples in training set as X_s. Then, the prediction of a new sample x can be rewritten as:

$$\hat{y} = \frac{\sum_{x_i \in X_s} y_i k(x, x_i)}{\sum_{x_i \in X_s} k(x, x_i)}. \tag{11}$$

After the X_{sub} is selected, the rest of the training set is split into two parts: the training set (X_t, Y_t) and the validation set (X_v, Y_v).

After multiple networks are optimized, we combined the results as follows:

$$\hat{y}_{final} = \frac{\sum w_i \hat{y_i}}{\sum w_i}, \tag{12}$$

where w_i is the weight with respect to the validation error of the i-th network which defined as:

$$w_i = \frac{1}{err_i}, \tag{13}$$

where err_i is the validation error of the i-th network.

D. Implementation

In this section, the implementation details of the proposed method is presented.

1) Activation Function: The activation function is used to define whether a node is activated. In this work, the Rectified

Linear Unit (ReLU) is utilized as the activation function since it is easy to computed and propagate. It is defined as below:

$$\phi(x) = max(0, x) \tag{14}$$

2) Initialization: Following [41], b is initialized to 0 and the W is initialized to a uniform distribution:

$$W \backsim U\left[-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}\right], \tag{15}$$

where d indicates the input dimension.

3) Optimization: The network is trained for n ($= 500$) epochs and the learning rate α is set as 0.0005. m ($= 64$) deep networks are trained and combined for better generalization.

IV. Experimental Evaluation

In this section, the experimental results are reported and discussed to demonstrate the effectiveness of the proposed method.

A. Evaluation Metrics

Two metrics are used for evaluation, the mean absolute error (MAE) and the mean squared error (MSE) which are defined as follows:

$$MAE = \frac{1}{N}\sum_{1}^{N}|y_i - \hat{y}_i|, \;\; MSE = \sqrt{\frac{1}{N}\sum_{1}^{N}(y_i - \hat{y}_i)^2}, \tag{16}$$

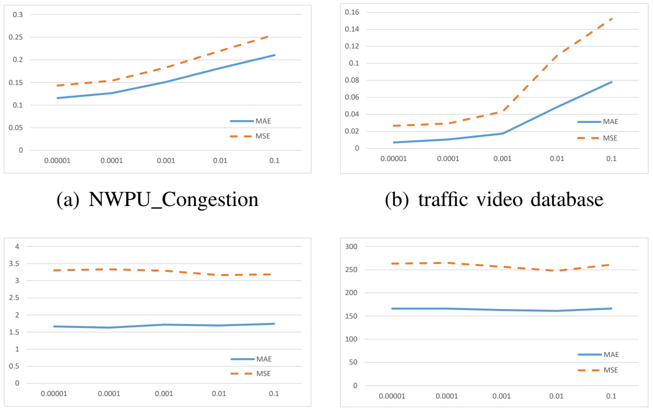
where N is the total number of validation/testing samples, y_i and $\hat{y}_i$ are the label (congestion level / crowd number) and the corresponding prediction of the i-th sample.

B. Congestion Detection

In this section, the experiments about congestion detection are conducted. The comparison methods are first listed. Then, the experimental settings and results on each datasets are reported.

For clarity, the comparison methods are summarized as follows:

- Random Guess: Randomly choose a value from $[0, 1)$ as the prediction.
- Traditional approach: The texture feature (LBP) [42] with metric learning for regression.
- Deep Learning: The deep network with an linear regression.
- Deep Metric Learning (DeepML): The proposed deep network with metric learning as loss function.
- Hierarchical Deep Metric Learning (H-DeepML): Deep metric learning with multi-level information encoded.
- Hierarchical Deep Metric Learning Ensemble (H-DeepMLE): The proposed method presented in Section III.



(a) NWPU_Congestion

(b) traffic video database

(c) WordExpo

(d) shanghai_Tech

Fig. 6. The selection of regularization parameter on different datasets. The horizontal axis is the different parameters and the vertical axis is the mean squared error.

1) NWPU_Congestion dataset: The first dataset is the NWPU_Congestion dataset which consists of 26 different scenes. There are 5585 samples in training set and 1492 samples in testing set. 6 different scenes in testing set are unseen in training set which can be used to evaluated the generalization ability of the algorithms. Typical images are shown in Figure 3.

In practice, the first parameter should be determined is the size of X_s. As shown in Figure 4, the best performance is achieved when we select 512 samples as the subset X_s. After that, the rest data is split into two parts: training set and validation set. 80% are used for training, and 20% are used for validation. Then, we train different networks for combination. The number of the networks is set as 16 to balance the performance and the training time based on the results shown in Figure 5. The regularization parameter is set as 0.00001 according to the results shown in Figure 6.

As shown in Table II and Figure 7, the proposed method achieves best performance which confirms its effectiveness. First of all, the performance of traditional method is similar to random guess due to the large variation among different scenes. The deep learning without careful design is worse than traditional method because the optimization is hard with a real value as label. Comparing deep learning to DeepML, we can see that the performance is increased after the metric learning is utilized to guide the training of the network. To evaluate the hierarchical feature extraction, the DeepML and H-DeepML are compared. The result show that more discriminative features can be learned with hierarchical network. Finally, the best performance is achieved by combining multiple networks together (H-DeepMLEs) which indicates that the deep learning ensemble is effective to increase the generalization ability.

2) traffic video database: Another dataset is traffic video database [43] which contains only one scene. It is used to evaluate the performance of different methods under simple circumstance. The database contains 254 highway video clips in which the resolution is 320×240. There are different light conditions and weathers in the videos. However, all videos are recorded by the same camera and angle. It is a popular dataset used for congestion classification which is divided into 3 classes. We first turn it into a regression task. Specifically, the congestion thresholds for the light, medium and heavy class are set as 0.165, 0.5, 0.83. Similar to NWPU_Congestion

TABLE II
COMPARISON OF DIFFERENT METHODS FOR CONGESTION DETECTION ON NWPU_CONGESTION DATASET.

Methods	MAE	MSE
Random Guess	0.356	0.434
Traditional Method	0.316	0.385
Deep Learning	0.367	0.398
DeepML	0.124	0.158
H-DeepML	0.112	0.134
H-DeepMLEs	**0.102**	**0.108**

TABLE III
COMPARISON OF DIFFERENT ARCHITECTURES FOR CONGESTION DETECTION ON TRAFFIC VIDEO DATASET.

Methods	MAE	MSE
Random Guess	0.3414	0.4190
Traditional Method	0.0325	0.0841
Deep Learning	0.2420	0.29.8
DeepML	0.0236	0.0555
H-DeepML	0.0092	0.0274
H-DeepMLEs	**0.0090**	**0.0253**

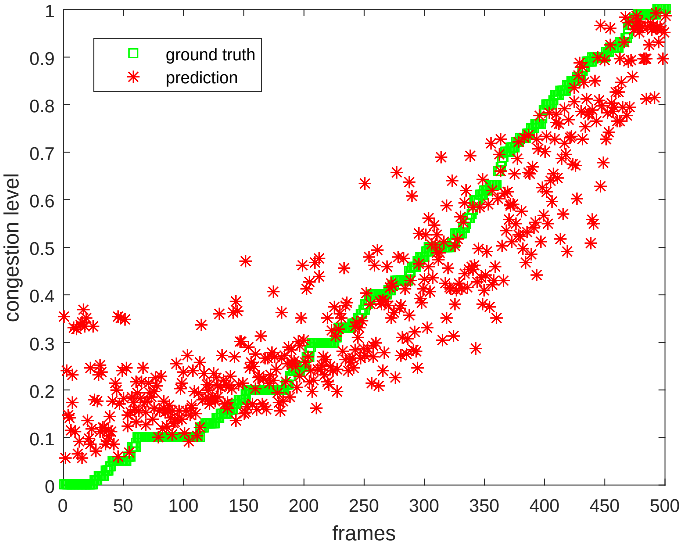


Fig. 7. The visualization of congestion detection performance on NWPU_Congestion dataset. The horizontal axis is the different testing samples and the vertical axis is the congestion level.

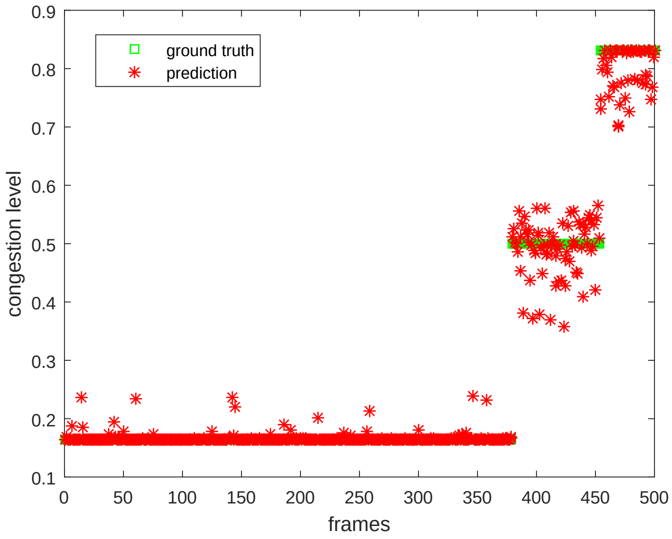


Fig. 8. TThe visualization of congestion detection performance on traffic video database. The horizontal axis is the different testing samples and the vertical axis is the the congestion level.

dataset, the size of subset is set as 512. 80% of samples are used for training and 20% of samples for testing. The regularization parameter is set as 0.00001 according to the results shown in Figure 6. Typical images can be see in Figure 3.

The experimental results are shown in Table III and Figure 8. Different from complex scenarios, the traditional approach is effective to detect congestion as the result shown that the performance of traditional approach is better than random guess and deep learning. The deep learning is still hard to optimize, as the result is similar to random guess. As same as the complex condition, the metric learning is effective to embed the structural information and guide the training of deep regression. Comparing DeepML and H-DeepML, the hierarchical feature is more effective since the representation is more complete and discriminative. However, under the simple condition, the increase of deep learning ensemble is limited since the H-DeepML is very effective already.

C. Crowd Counting

In this section, the experiments about crowd counting are conducted and the results are reported. We first present the dataset and the experimental settings. Then, the experimental results are discussed.

1) The WorldExpo dataset: The first dataset is the WorldExpo dataset [38] which consists of 3980 labeled frames come from 108 different surveillance cameras in which the resolution is 576 × 720. In this experiment, the image is split into 24 128 × 128 patches and then resized to 64 × 64 for training. Similarly, the testing image is split into patches and the summarized to a full image. In this experiment, the regularization parameter is set as 0.001. Typical images are shown in Figure 3.

The first comparison method is texture feature (LBP) [44] with ridge regression (LBP+RR). The second method is proposed by Fiaschi *et al.* [35] which utilizes random forest for prediction. Two deep learning based methods proposed by Zhang *et al.* [38] and Zhang *et al.* [39] are also used for comparison. Note that, the training of these deep networks needs additional information (the position of all human head) which is hard to obtain.

The results of this experiment are shown in Table IV and Figure 9. First of all, the proposed method achieves best performance in the methods without additional information.

TABLE IV
COMPARISON OF DIFFERENT METHODS FOR CROWD COUNTING ON WORLDEXPO DATASET.THE MAE IS USED AS THE METRIC FOR EVALUATION. THE BEST RESULTS WITH AND WITHOUT ADDITIONAL INFORMATION ARE INDICATED IN **BOLD**.

Methods	Scene 1	Scene 2	Scene 3	Scene 4	Scene5	Average	Additional info
LBP+RR	13.6	59.8	37.1	21.8	23.4	31.0	No
Fiaschi *et al.* [35]	**2.2**	87.3	22.2	16.4	**5.4**	26.7	No
H-DeepMLEs	9.3	**34.6**	**12.7**	**11.9**	18.5	**17.4**	No
Zhang *et al.* [38]	**2.0**	29.5	**9.7**	**9.3**	**3.1**	**10.7**	Yes
Zhang *et al.* [39]	3.4	**20.6**	13.0	13.0	8.0	11.6	Yes

TABLE V
COMPARISON OF DIFFERENT METHODS FOR CROWD COUNTING ON SHANGHAI_TECH DATASET. THE BEST RESULTS WITH AND WITHOUT ADDITIONAL INFORMATION ARE INDICATED IN **BOLD**.

	Part_A		Part_B		Average		
Methods	MAE	MSE	MAE	MSE	MAE	MSE	Additional info
LBP+RR	303.2	371.0	59.1	81.7	148.3	233.5	No
H-DeepMLEs	**129.1**	**186.0**	**32.1**	**53.1**	**67.5**	**120.1**	No
Zhang *et al.* [38]	181.8	277.7	32.0	49.8	86.7	172.5	Yes
Zhang *et al.* [39]	**110.2**	**173.2**	**26.4**	**41.3**	**57.0**	**109.8**	Yes

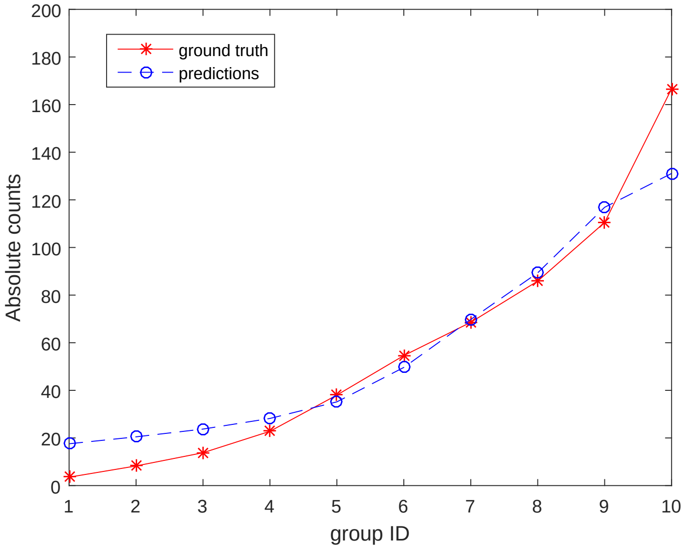


Fig. 9. The visualization of crowd counting on WorldExpo dataset. The horizontal axis is the different groups and the vertical axis is the crowd number.

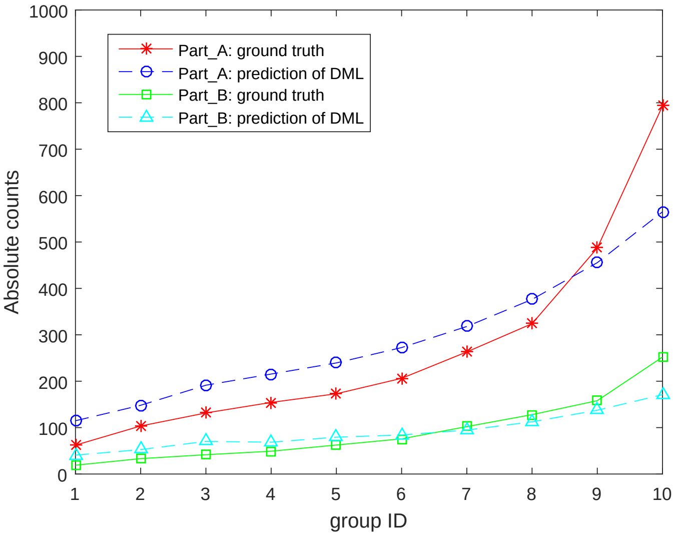


Fig. 10. The visualization of crowd counting on shanghai_Tech dataset. The horizontal axis is the different groups and the vertical axis is the crowd number.

As we can see that H-DeepMLEs outperforms the other ones (LBP+RR and Fiaschi *et al.* [35]) which confirms that the H-DeepMLEs is effective to count the crowd number. The deep learning based methods (Zhang *et al.* [38] and Zhang *et al.* [39]) which trained with additional information (i.e., density map) achieve top performance. However, these methods are hard to be applied to large-scale data since the additional information (human heads in crowd) is hard to label.

2) The Shanghai_Tech dataset: The shanghai_Tech dataset [39] is a more challenging dataset which contains two parts of images. The first part (Part_A) is the extremely crowd images collected from Internet which contains 482 images, and the second part (Part_B) is recorded from the busy streets which contains 1198 labeled images. 300 images are used as training set and the rest are used as testing set in Part_A. 400 images are used as training set and the rest as testing set in Part_B. Similarly, the images are first split into patches and then accumulated as a whole image. In this experiment, the regularization parameter is set as 0.0001. Typical images can be see in Figure 3.

The comparison methods include texture (LBP) feature with a ridge regressor and two deep learning based algorithms [38], [39] in which additional information is utilized for training.

The results are shown in Table V and Figure 10 which are very similar to the WordExpo dataset. The H-DeepMLEs outperforms the traditional method (LBP+RR) and achieves comparable result with deep learning methods (Zhang *et al.* [38] and Zhang *et al.* [39]) which indicates that H-DeepMLEs can be effectively optimized to count crowd number.

V. Conclusion

In this paper, a robust hierarchical deep learning method is proposed for two crowdedness regression tasks: congestion detection and crowd counting. To extract multi-level features, a hierarchical deep network is designed. To effectively train the network, we propose to guide the training of the network with a robust metric learning by structural information embedding. Then, multiple deep networks are combined together to increase the generalization ability. With extensive experiments, we demonstrate that the hierarchical feature is more discriminative and the deep learning ensemble is effective under complex scenarios.

However, it is time-consuming to train a deep network which increase the difficulty to the research of deep learning ensemble. Thus, the efficient training method should be exploited in the future.

References

[1] K. W. Cheng, Y. T. Chen, and W. H. Fang, "Gaussian process regression-based video anomaly detection and localization with hierarchical feature representation," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5288–5301, 2015.

[2] Y. Yuan, D. Wang, and Q. Wang, "Anomaly detection in traffic scenes via spatial-aware motion reconstruction," *IEEE Transactions on Intelligent Transportation Systems*, 2016.

[3] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab, "Robust optimization for deep regression," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2830–2838.

[4] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12, no. 10, pp. 1399–1404, 1999.

[5] K. Q. Weinberger and G. Tesauro, "Metric learning for kernel regression," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2007, pp. 612–619.

[6] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4766–4779, 2015.

[7] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 26, no. 10, pp. 2222–2233, 2015.

[8] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1875–1882.

[9] D. Hu, X. Lu, and X. Li, "Multimodal learning via exploring deep semantic similarity," in *Proceedings of the 2016 ACM Conference on Multimedia Conference, MM 2016, Amsterdam, The Netherlands, October 15-19, 2016*, pp. 342–346.

[10] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu, "Background prior-based salient object detection via deep reconstruction residual," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 8, pp. 1309–1321, 2015.

[11] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, J. Dong, L. Lin, and S. Yan, "Deep human parsing with active template regression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2402–2414, 2015.

[12] Y. Zheng, D. Liu, B. Georgescu, H. Nguyen, and D. Comaniciu, "3d deep learning for efficient and robust landmark detection in volumetric data," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2015, pp. 565–572.

[13] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems*, 2013, pp. 2553–2561.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[16] W. Ouyang, X. Chu, and X. Wang, "Multi-source deep learning for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2329–2336.

[17] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[18] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1653–1660.

[19] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3476–3483.

[20] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *European Conference on Computer Vision*. Springer, 2014, pp. 94–108.

[21] S. Hu, J. Wu, and L. Xu, "Real-time traffic congestion detection based on video analysis," *Journal of Information and Computational Science*, vol. 9, no. 10, pp. 2907–2914, 2012.

[22] C. Zhan, X. Duan, S. Xu, Z. Song, and M. Luo, "An improved moving object detection algorithm based on frame difference and edge detection," in *International Conference on Image and Graphics*, 2007, pp. 519–523.

[23] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[24] A. Sobral, L. Oliveira, L. Schnitman, and F. D. Souza, "Highway traffic congestion classification using holistic properties," in *International Conference on Signal Processing, Pattern Recognition and Applications*, 2013.

[25] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

[26] Y. Yuan, J. Wan, and Q. Wang, "Congested scene classification via efficient unsupervised feature learning and density estimation," *Pattern Recognition*, vol. 56, pp. 159–169, 2016.

[27] K. G. Derpanis and R. P. Wildes, "Classification of traffic video based on a spatiotemporal orientation analysis," in *IEEE Workshop on Applications of Computer Vision*, 2011, pp. 606–613.

[28] A. Riaz and S. A. Khan, "Traffic congestion classification using motion vector statistical features," in *International Conference on Machine Vision*, 2013, pp. 90 671A–90 671A.

[29] E. Dallalzadeh, D. Guru, and B. Harish, "Symbolic classification of traffic video shots," in *Advances in Computational Science, Engineering and Information Technology*, 2013, pp. 11–22.

[30] A. N. Marana, M. A. Cavenaghi, R. S. Ulson, and F. L. Drumond, "Real-time crowd density estimation using images," in *Advances in International Symposium on Visual Computing*, 2005, pp. 355–362.

[31] C. Regazzoni, A. Tesei, and G. Vernazza, "A bayesian network for automatic visual crowding estimation in underground stations," in *Image Technology*, 1996, pp. 203–230.

[32] S. Cho, T. W. S. Chow, and C. Leung, "A neural-based crowd estimation by hybrid global learning algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 29, no. 4, pp. 535–541, 1999.

[33] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting," in *British Machine Vision Conference*, 2012, pp. 1–11.

[34] V. S. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, 2010, pp. 1324–1332.

[35] L. Fiaschi, U. Köthe, R. Nair, and F. A. Hamprecht, "Learning to count with regression forest and structured labels," in *Proceedings of the International Conference on Pattern Recognition*, 2012, pp. 2685–2688.

[36] K. Chen, C. C. Loy, S. Gong, and T. Xiang, “Feature mining for localised crowd counting,” in *British Machine Vision Conference*, 2012, pp. 1–11.

[37] N. C. Tang, Y. Y. Lin, M. F. Weng, and H. Y. M. Liao, “Cross-camera knowledge transfer for multiview people counting,” *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 80–93, 2015.

[38] 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.

[39] 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.

[40] K. Q. Weinberger, J. Blitzer, and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” in *Advances in Neural Information Processing Systems*, 2005, pp. 1473–1480.

[41] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[42] F. Lu and J. Huang, “An improved local binary pattern operator for texture classification,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 1308–1311.

[43] A. B. Chan and N. Vasconcelos, “Classification and retrieval of traffic video using auto-regressive stochastic processes,” in *Proceedings of the Intelligent Vehicles Symposium*. IEEE, 2005, pp. 771–776.

[44] M. Pietikäinen, “Local binary patterns,” *Scholarpedia*, vol. 5, no. 3, p. 9775, 2010.