Crowd Counting Approaches: Density-based vs. Regression-based

Agata Garbin
agata.garbin@studenti.unipd.it

Alessia d'Addario
alessia.daddario@studenti.unipd.it

Abstract

In this study, we compared two crowd counting methods: density-based using CSRNet, with VGG16 and VGG19 as backbones, and regression-based using InceptionV3, ResNet50, VGG16 and MobileNetV2. ResNet50 demonstrated superior performance across models, achieving a MAE of 1.56 on the Mall dataset, while CSRNet with VGG-19 yielded a MAE of 1.57. These findings show ResNet50's effectiveness for regression-based crowd counting and CSRNet's potential for density-based crowd estimation.

1. Introduction

The term "crowd counting" refers to the practise of counting the number of people present in a certain area, photo, or more in general in a "frame" of some sort, that finds essential applications in fields like public safety and urban planning. Some of the challenges encountered in problems of this type include occlusions, distortion, non-uniform distribution and very high density. One of the most used methods are convolutional neural networks, which automatically learn from the visual patterns of images, adapting to various conditions such as lightning and perspective distortion. Other innovative approaches involve objects density maps, that makes it easier to identify areas of higher density, and direct regression based methods that analyze the global characteristics of the image to predict the final count, as they look at entire scene and use the distribution patterns to calculate the finale result. It is also possible to integrate clustering techniques and intensity maps to divide the images into regions or groups, providing a detailed estimate of local patterns. In this project we mainly focused on making a comparison between regression based and density based models, where the first approach directly estimates the number of people using a deep CNN that performs regression, and the second indirectly estimates the count using density maps. We tested several models, such ad InceptionV3, ResNet50, VGG16, MobileNetV2 and CSRNet with various backbones, both in ShangaTech Part-B dataset and in Mall dataset, two of the most famous dataset for this kind of problem, reaching pretty good results for both methods. In particular, we were impressed to see that simple regression based models on neural networks could perform as well as a density based models, with a best overall MAE of 1.56, thanks to ResNet50 strong architecture, compared to a MAE of 1.57 coming from CSRNet density-based approach. Our GitHub Repository.[1]

2. Related Work

Since crowd counting is a interesting research area with many real-world appications, this lead to the creation of a huge amount of research on automated crowd counting using image and video analysis methods through the years. An overview on the progressive developments and contributions over time and the current state-of-the-art of the Crowd Counting field can be found in [4]:

- Traditional methods for crowd counting were all based on **total count approach**. These methods employ image processing techniques to detect hand-crafted features e.g., body appearance [7], [12], or body parts [13], [6] and then use machine learning models to find the total head count in an image. However, these methods' accuracy in dense crowds strongly decreased due to occlusions, low resolution, and perspective issues.

- To overcome these challenges, **regression-based methods** were introduced. Instead of detecting body parts or shapes, they used global image features like texture [1] and gradients [2] to estimate the total count from an image or its patches. While this approach addressed some issues, it performed poorly in high-density crowds.

- Recent advances have seen the use of **convolutional neural networks** (CNNs) for crowd counting[15],[5], due to their strong automatic feature extraction capabilities. This method is used widely and aoutperformed all the traditional methods employing crowd density estimation, producing a density map that includes both location information and total head count.

[1] https://github.com/agatagarbin/Crowd-counting-problem

- First introduced in [14], **density estimation using CNNs** has been adopted in all subsequent research contributions. However, there have been major architectural improvements to achieve optimum performance. Moreover many crowd datasets were also published over time introducing more challenges in terms of high density, scale variation, scene variation.

In our analysis we mainly focused on two methods, Density estimation using CSRNet and total head count estimation (with a regression based approach) using InceptionV3, VGG16, ResNet50 and MobileNetV2.

3. Dataset

A number of benchmarking datasets have been introduced over time for evaluating crowd counting models. These datasets vary in size (number of samples), annotations, and image attributes. Here we put our attention on two of them used in a variety of paper thanks to their useful characteristics:

3.1. Mall Dataset

First introduced in [1], Mall Dataset is a collection of 2000 RGB images with a resolution of 480x640 pixels 1. Every frames is taken from a fixed security camera video and the dataset itself includes annotations for the head positions of over 60,000 pedestrians across all frames.

Figure 1. Example of images in the Mall Dataset

We divided the entire dataset in train and test keeping a percentage of images for the test set of 20% resulting in a total of 400 frames. The remaining 1600 images were then split in actual training and validation sets resulting in 1280 and 320 images respectively. For reproducibility purpose and for assuring comparability, we did this division at the beginning of each experiment keeping a fixed seed (16).
An exploration of the data is necessary to better understand the results:

As can be seen, the images in the Mall Dataset contain a wide range in the number of people, moving from a minimum of 13 to a maximum of 53 individuals per frame with a mean of 31.16, value that will be useful to compare with the models result.

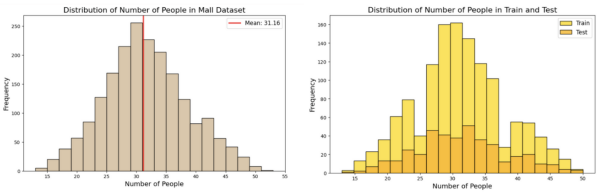

Figure 2. Data Distribution Mall Dataset

3.2. ShanghaiTech Dataset

ShanghaiTech Dataset is a large-scale crowd counting dataset firstly introduced in [15]. It consists of 1198 annotated crowd images, for a total of 330,165 annotated people, and it is divided into two parts: Part-A, containing 482 images collected from the Internet representing dense scenes, and Part-B, collected on the busy streets of Shanghai, containing 716 images and presenting more sparse scenes. Here some exaples: Figure 4.

Figure 3. Part A

Figure 4. Part B

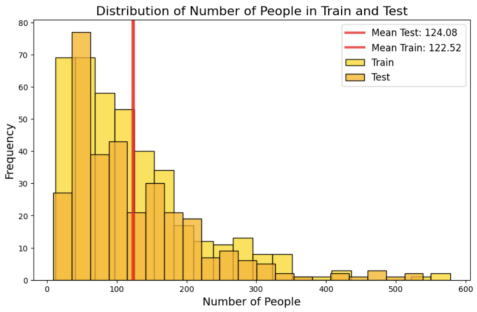

Figure 5. Data Distribution ShanghaiTech Part-B Dataset

In particular we focused on Part-B, since it showed more similarity with the sparse nature of the Mall dataset images. Part-B is already divided into train and test subsets, consisting of 400 and 316 images respectively. Each person in a crowd image is annotated with one point close to the center of the head.

4. Method

4.1. Evaluation Metrics and Losses

As general base, in order to evaluate the goodness of the two models under analysis we opted for three different evaluation metrics:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|C_i - C_i^{GT}| \tag{1}$$

$$\text{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|C_i - C_i^{GT}|^2} \tag{2}$$

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{C_i - C_i^{GT}}{C_i}\right| * 100 \tag{3}$$

with:

- N number of images in the train/test set
- C_i^{GT} the Ground Truth of counting for image i
- C_i the estimated count for image i

These are the most common evaluation metrics [4] due to the fact that they are particularly suited for this task.

Regarding the choice of the loss function to be utilized during the training phase, as a common baseline for all models, we opted for the MSE loss function, which is the most commonly used loss function in crowd counting research [4]. Furthermore, although not actually being used in paper like [5], eager to see the comparison wrt MSE loss, we decided to also use the MAE loss.

4.2. Density Based Approach

The main idea in the Density Based Approach is to indirectly estimate the number of people in an image, making use of the so called density maps. In our context, the density map of a image is essentially a 2D array, interpreted as image itself, where each pixel value indicates the estimated number of people in a small region around that pixel. To be able to generate a density map, the annotation of the heads has to be given. This annotation can be seen as a sparse matrix with dimensions equal to the corresponding image, containing zeros everywhere except at the positions corresponding to the centers of the heads, where the entry in equal to one. For both the dataset utilized, the head annotation were given.

4.2.1 Ground Truth Generation

To actually generate the density maps, representing the ground truth that has to be given to the models, we followed the method described in [15] using geometry-adaptive kernels. Called F the density maps we want to produce, it is obtained by:

$$F(x) = \sum_{i=1}^{N}\delta(x - x_i) * G_{\delta_i}(x) \text{ with } \sigma_i = \beta\bar{d}^i \tag{4}$$

Basically, for each head x_i in a given image, $\bar{d}^i$ is the average distance from it's k nearest neighbors $d_1^i, d_2^i, \ldots, d_m^i$. Thus, the pixel associated with x_i corresponds to an area on the ground in the scene roughly of a radius proportional to $\bar{d}^i$. Therefore, to estimate the crowd density around the pixel x^i, we need to convolve $\delta(x - x_i)$ with a Gaussian kernel with variance σ_i proportional to $\bar{d}^i$, where $\delta(.)$ is the discrete version of the delta function: $\delta(0) = 1$ and 0 everywhere else. Following the configuration of [5], based also on [15], we put $\beta = 3$ and $k = 3$.

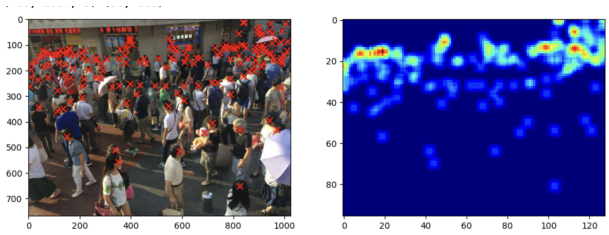

Figure 6. Example of GT density map from a ShanghaiTech image

4.2.2 CSRNet Architecture

Among the various modification of CSRNet presented in [5], we chose to apply the one that was indicated as best in performance, highlited in Figure 7.

The main features of the architecture of CSRNet are represented by a Convolutional Neural Network (VGG16) as front-end and Dilatated Convolutional layers as the back-end:

- The choice of VGG16 was lead by its strong transfer learning ability and its flexible architecture useful for easily concatenating the back-end for density map generation. We used VGG16 pre-trained on Imagenet, keeping only the first 10 layers and removing the last 6 costituted by the Fully Connected Layers. One think to be careful about is that the output size of this front-end network is 1/8 of the original input size. This means that during the process of generating the density maps, that would correspond to the target for the model, a scaling factor of 8 is needed. This particular can be seen in Figure 6, were the density map present a resolution of 1/8 wrt the original image.

- The Convolutional layer were chosen for extracting deeper information of saliency as well as main training the output resolution. Dilated convolution utilizes sparse kernels, which allow it to alternate between pooling and convolutional layers. This approach enlarges the receptive field without increasing the number of parameters or the computational load. For these layers, the initial values are generated using Gaussian initialization with a standard deviation of 0.01.

Configurations of CSRNet			
A	B	C	D
input(unfixed-resolution color image)			
front-end (fine-tuned from VGG-16)			
conv3-64-1 conv3-64-1			
max-pooling			
conv3-128-1 conv3-128-1			
max-pooling			
conv3-256-1 conv3-256-1 conv3-256-1			
max-pooling			
conv3-512-1 conv3-512-1 conv3-512-1			
back-end (four different configurations)			
conv3-512-1	conv3-512-2	conv3-512-2	conv3-512-4
conv3-512-1	conv3-512-2	conv3-512-2	conv3-512-4
conv3-512-1	conv3-512-2	conv3-512-2	conv3-512-4
conv3-256-1	conv3-256-2	conv3-256-4	conv3-256-4
conv3-128-1	conv3-128-2	conv3-128-4	conv3-128-4
conv3-64-1	conv3-64-2	conv3-64-4	conv3-64-4
conv1-1-1			

Figure 7. Configurations of CSRNet [5]

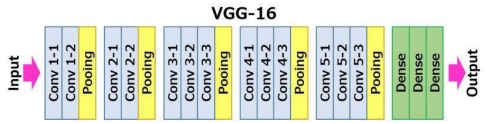

Figure 8. VGG16 Model Architecture

4.2.3 CSRNet with VGG-19 as Backbone

As mentioned, the CSRNet presented in [5] uses VGG-16 as a backbone, relying on its intrinsic characteristics.
To further investigate the relevance that the chosen front-end has on the overall model, we decided to also implement VGG-19 for comparison.
First introduced in [10], this model has the same basic idea as the VGG-16 model, with the exception that it supports 19 layers, leading to a larger number of parameters. This additional depth in VGG19 allows it to learn more complex features and representations from the input data, potentially leading to better performance on certain tasks.

4.3. Regression Based Approach

Regression-based models are a class of machine learning algorithms used to predict continuous values. These models establish a relationship between independent variables and a dependent variable (input ad output respectively) through a regression function. In modern contexts, especially with the use of pre-trained CNNs, regression models are powerful tools for complex applications such as crowd counting, where the goal is to estimate the number of people in images or videos by analyzing features extracted from the inputs visual.

We used models such as InceptionV3 [11], ResNet50 [3], VGG16 [9] and MobileNetV2 [8], pre-trained on ImageNet, and applied them to both the Mall dataset and Shanghai Tech Part B.

We started by selecting an appropriate base model for regression-based estimation and then we added a dense layer at the end of each model to improve the counting ability. To further improve generalization, we integrated data augmentation techniques and increased the complexity of the model. We explored different activation functions and, during the training process, varied parameters such as the optimizer, learning rate, batch size and number of epochs. We also implemented advanced techniques such as early stopping and diminishing learning rate to optimize performance. We finally performed pre-training on the ShanghaiTech dataset, followed by a further training phase and a testing phase on the Mall dataset, finding disappointing results probably due to the fact that these models in their base form struggle to adapt to a complex (density speaking) dataset as Shanghai Tech Part B.

4.3.1 InceptionV3

InceptionV3 [11] is a deep convolutional neural network model that contains 48 distinct layers. The model starts with a series of initial convolutions and pooling operations to reduce image size and increase feature depth. Next, it uses a series of ”Inception modules” that combine convolutions of various sizes (1x1, 3x3, 5x5) within the same block. The model also includes reduction modules to decrease the spatial dimensions of features and increase their depth. Finally, there are final convolutions, a global pooling operation, and a fully connected dense layer for classification.

4.3.2 VGG16

VGG16 [9] 8 is a convolutional neural network composed of 16 weight layers, and all its convolutions are 3x3 in size. The model is organized into five main blocks containing a series of convolutions, each followed by a max pooling operation. Finally, the model ends with three fully connected layers used for classification.

4.3.3 ResNet50

ResNet50 [3] is a deep convolutional neural network model featuring 50 layers. It uses ”residual blocks”, which con-

tain skip connections that bypass one or more convolutional layers. Its architecture starts with an initial convolution followed by a pooling operation. Then it follows a series of residual blocks organized into four main groups, each composed of various blocks with 1x1, 3x3 and 1x1 convolutions. After the residual blocks, the model ends with a global pooling operation and a fully connected layer for classification.

4.3.4 MobileNetV2

MobileNetV2 [8] is a convolutional neural network with an architecture based on "inverted residuals" and "linear bottlenecks". It starts with an initial convolution followed by batch normalization and ReLU and then it develops through a series of 17 blocks of inverted residual. The blocks include 1x1 and 3x3 convolutions and a jump connection that passes through a 3x3 convolution. Finally the model ends with a 1x1 convolution followed by global average pooling and a fully connected layer for classification.

5. Experiments

5.1. Experiments on Mall dataset

5.1.1 Density based models

During the training of our models we tried different configuration of parameters, in order to obtain our best overall result. In particular, other than experiment with the different losses 4.1, we tried changing optimizer (SGD / Adam), as well as learning rates. In general the epochs were kept constant to 100 controlled by an implemented early stopping of patient equal to 10.
It is worthwhile to notice that the training in the original paper carries a configuration with 400 epochs, difficult to replicate with our resources.

Here we report our best experiments on the Mall Dataset using CSRNet with VGG-16 (4) and VGG-19.

OPTIMIZER	LOSS	LR	MAE	MSE	MAPE
SGD	MAE	1e-2	3.79	4.36	11.73
SGD	MSE	1e-2	4.53	5.17	14.12
ADAM	MAE	1e-5	1.90	2.27	5.84
ADAM	MSE	1e-5	1.60	2.02	5.08

Table 1. Experiments using VGG-16 as Backbone

OPTIMIZER	LOSS	LR	MAE	MSE	MAPE
SGD	MAE	1e-2	2.39	2.94	7.35
SGD	MSE	1e-2	4.10	4.86	12.68
ADAM	MAE	1e-5	1.57	2.04	4.86
ADAM	MSE	1e-6	2.88	3.40	8.96

Table 2. Experiments using VGG-19 as Backbone

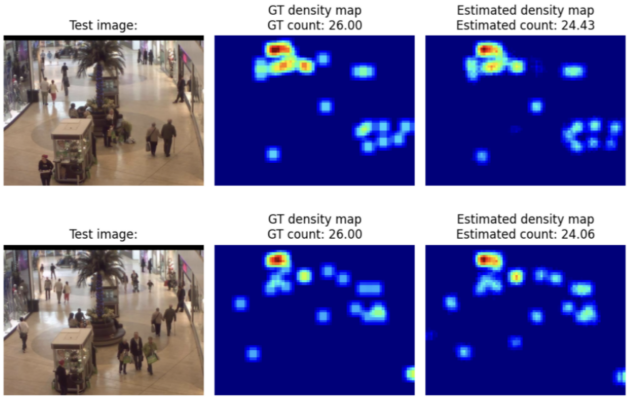

Figure 9. Exaple of Best model with VGG-16 Backbone

As we can see, the results are quite satisfactory, especially if we compare the best result of the two model with the ones reported in this comprehensive collection Repository.
It is notable that in both cases the best results were achieved when using the Adam optimizer, and in general the MAE loss. Furthermore, we can argue that, when the two approaches are indeed comparable, the second one slightly outperforms the first one. This can probably be attributed to the deeper nature of VGG-19, that results in the capacity of learning more complex relationships.

5.1.2 Regression based models

We compared four regression-based models: InceptionV3, VGG16, ResNet50 and MobileNetV2. During training and testing, we varied parameters such as the number of epochs, the learning rate, the activation function, the optimizer and the batch size. ResNet50 showed the best overall performance, with MAE=1.56, MSE=2.01 and MAPE=5.14.
Below we report the optimal configurations and related metrics for each model, where we are considering the number of epochs equal to 100 and a decay of $5e-4$:

- **ResNet50**: optimizer: Adam; activation: ReLU; learning rate: $1e-4$; loss function: MAE; metrics: MSE, batch size: 32. Results: MAE = 1.56, MSE = 2.01, MAPE = 5.14.
- **InceptionV3**: optimizer: Adam; activation: ReLU; learning rate: $1e-4$; loss function: MSE; metrics: MAE, batch size: 16. Results: MAE = 1.77, MSE = 2.29, MAPE = 5.65.
- **VGG16**: optimizer: SGD; activation: linear; learning rate: $1e-7$; loss function: MSE; metrics: MAE, batch size: 16. Results: MAE = 2.46, MSE = 3.10, MAPE = 7.96.

- **MobileNetV2**: optimizer: Adam; activation: ReLU; learning rate: $1e-5$; loss function: MSE; metrics: MAE, batch size: 32. Results: MAE = 2.49, MSE = 3.14, MAPE = 8.10.

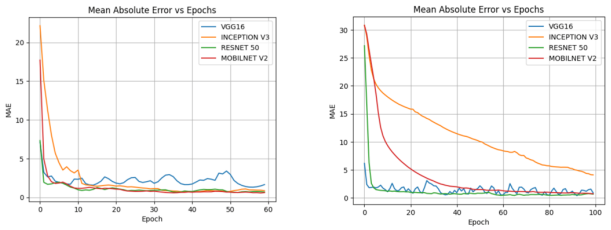

Figure 10. Comparison of all models in two of the optimal configurations

For each model, we added a global pooling layer and a dense layer with linear activation to improve performance on the crowd counting task. This additional structure improved the models' ability to generalize and provide more accurate predictions. Let's finally notice that for InceptionV3 model we used images of size 299x299, while for the others we used images rescaled to size 224x224, as requested by the models' architectures.

5.2. Experiments on ShanghaiTech Part-B dataset

Regarding the ShanghaiTech Part-B Dataset we repeat the same training pipeline as described in 5.1.1.
We report here our best results and an example of the best model performance:

OPTIMIZER	LOSS	LR	MAE	MSE	MAPE
SGD	MAE	1e-2	24.01	42.25	14.22
SGD	MSE	1e-2	43.51	56.66	34.31
ADAM	MAE	1e-5	18.21	27.29	14.23
ADAM	MSE	1e-5	31.32	38.89	27.02

Table 3. Experiments using VGG-16 as Backbone

OPTIMIZER	LOSS	LR	MAE	MSE	MAPE
SGD	MAE	1e-2	32,24	45,12	23,38
ADAM	MAE	1e-5	22.25	32.32	17.45
ADAM	MSE	1e-5	20.27	29.32	16.54

Table 4. Experiments using VGG-19 as Backbone

Examining the output images for the density maps, it can be seen how the model tend to make more mistakes when provided with congested scenes. This characteristic is one of the features of the ShanghaiTech Part-B dataset expecially if compared with the Mall dataset, therefore it is not so strange to have higher MSE values compared to the one obtained from the latter. Nevertheless, we still consider

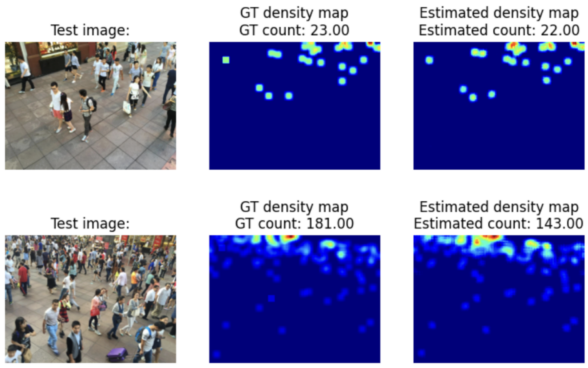

Figure 11. Exaple of Best model with VGG-16 Backbone

our result satisfactory, considering also that, as previuosly showed, the mean value of people distribution is near 122. Furthermore, the author of the paper introducing the CSRNet architecture reported values of 10.6 for MAE and 16.0 for MSE, which are not too distant from our best results.

5.2.1 Regression based models

We compared the same four regression based models using ShanghaiTech Part-B dataset. However, due to the high density of people present in the images and the fact that we did not use density maps, the obtained results were not satisfactory nor usable in real scenarios. The "best" obtained result were MAE = 65.47, MSE = 92.96, MAPE = 58.02, with the following configuration: optimizer: Adam; activation: linear, learning rate: $1e-4$; decay: $5e-4$; loss: MSE; metrics: MAE; batch size: 16; epochs: 100.

6. Conclusion

We focused on two distinct crowd counting methods: density-based and regression-based approaches. The density-based approach with CSRNet with VGG-19 as a backbone resulted in a surprising MAE of 1.57, MSE of 2.04 and MAPE of 4.86. For the regression-based method, we utilized several pre-trained deep convolutional neural networks, including InceptionV3, ResNet50, MobileNetV2, and VGG16. Among these, we discovered that ResNet50 achieved the highest performance on the Mall dataset, with a MAE of 1.56, MSE of 2.01, and MAPE of 5.14 showing that simple regression based approach on neural networks could perform as well as a density-based approach.
Further work should consider a Cross-scene counting and domain adaptation, i.e research on model generalization using model pre-trained on a dataset and the fine-tuned on another dataset. This topic could be generalized in a range of diverse scenarios from drone images to indoor places.

References

[1] Ke Chen, Chen Change Loy, Shaogang Gong, and Tao Xiang. Feature mining for localised crowd counting. 01 2012.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 06 2016.

[4] Muhammad Asif Khan, Hamid Menouar, and Ridha Hamila. Revisiting crowd counting: State-of-the-art, trends, and future perspectives. *Image and Vision Computing*, 129:104597, 2023.

[5] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1091–1100, 2018.

[6] Sheng-Fuu Lin, Jaw-Yeh Chen, and Hung-Xin Chao. Estimation of number of people in crowded scenes using perspective transformation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 31:645 – 654, 12 2001.

[7] Zhe Lin and Larry Davis. Shape-based human detection and segmentation via hierarchical part-template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:604–618, 04 2010.

[8] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. pages 4510–4520, 06 2018.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[11] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and ZB Wojna. Rethinking the inception architecture for computer vision. 06 2016.

[12] Oncel Tuzel, Fatih Porikli, and Peter Meer. Pedestrian detection via classification on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1713–1727, 2008.

[13] P. Viola and M. Jones. Robust real-time face detection. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 747–747, 2001.

[14] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 833–841, 2015.

[15] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, 2016.