DATA SCIENCE (資料科學)
SPRING, 2023

LECTURE 6:
CROWD ESTIMATION

PC: Midjourney

Shuai, Hong-Han (帥宏翰)
Associate Professor
Department of Electronics and Electrical Engineering
National Yang Ming Chiao Tung University

Week	Contents
8	Computer Vision (I)-Vision Transformer
9	Computer Vision (II)-Crowd Estimation [HW4: Crowd Estimation]
10	Computer Vision (III)-Domain Adaptation

Counting for Base Station Allocation

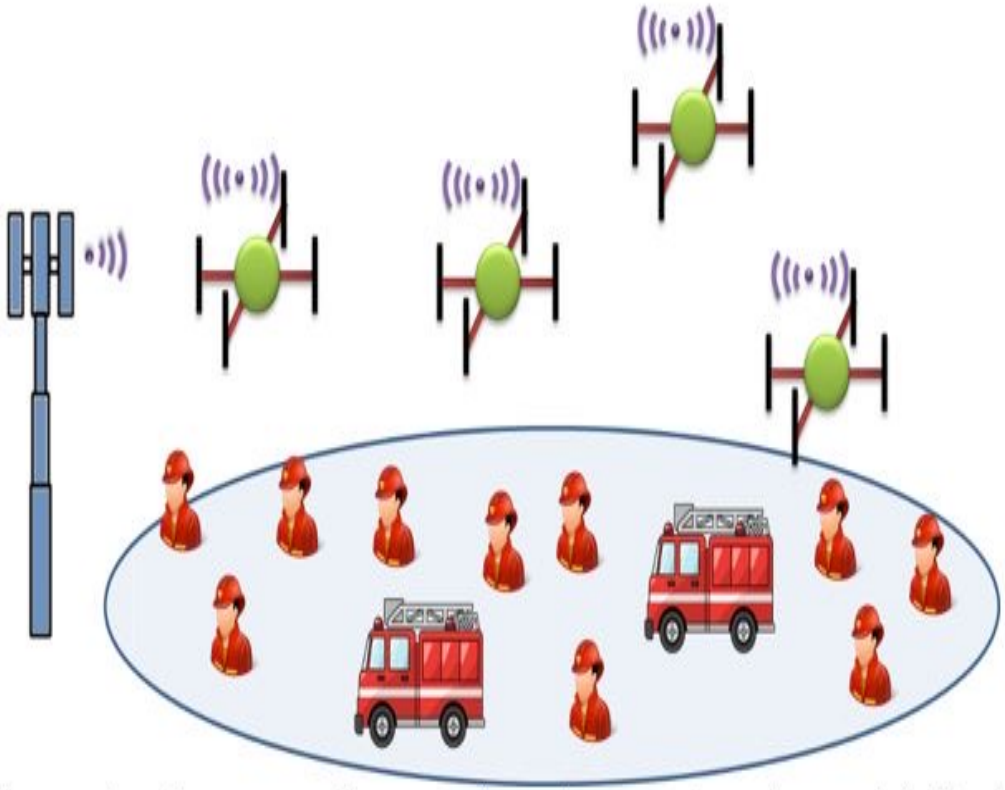


Figure 1: Airborne base stations are flown to provide 5G radio access to front-end personnel. The "flying" network can be viewed as an extension to an existing terrestrial 5G infrastructure. In essence, the clique of flying base stations is nothing but a manifestation of mobile ad hoc networks (MANET).

Deadly Halloween Crowd Surge in South Korea

Motivations

Examples of recent deadly stampedes and crowd disasters (see [1]).

Year	Place	Deaths
2010	Loveparade, Duisburg	21
2010	Water Festival, Phnom Phen	>380
2006	Stadium, Yemen	51
2006	Pilgrimage, Mena	363
2005	Religious Procession, Bagdad	>640
1999	Subway Station, Minsk	53
1990	Pilgrimage, Mena	1426
1989	Stadium, Sheffield	96
1982	Stadium, Moscow	340

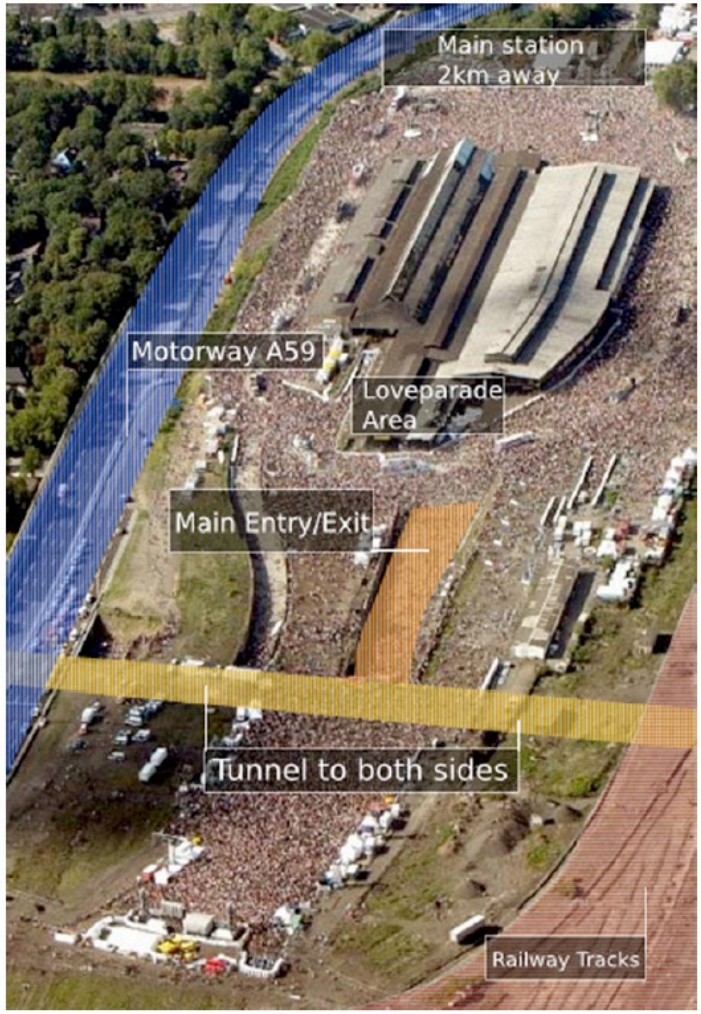


Fig. 1. Aerial view of the festival area [20].

Krausz, Barbara, and Christian Bauckhage. "Loveparade2010: Automatic video a
Vision and Image Understanding116.3 (2012): 307-319.

Detection-Based Crowd Counting

Limitations

(a) Occlusion

(b) Complex background

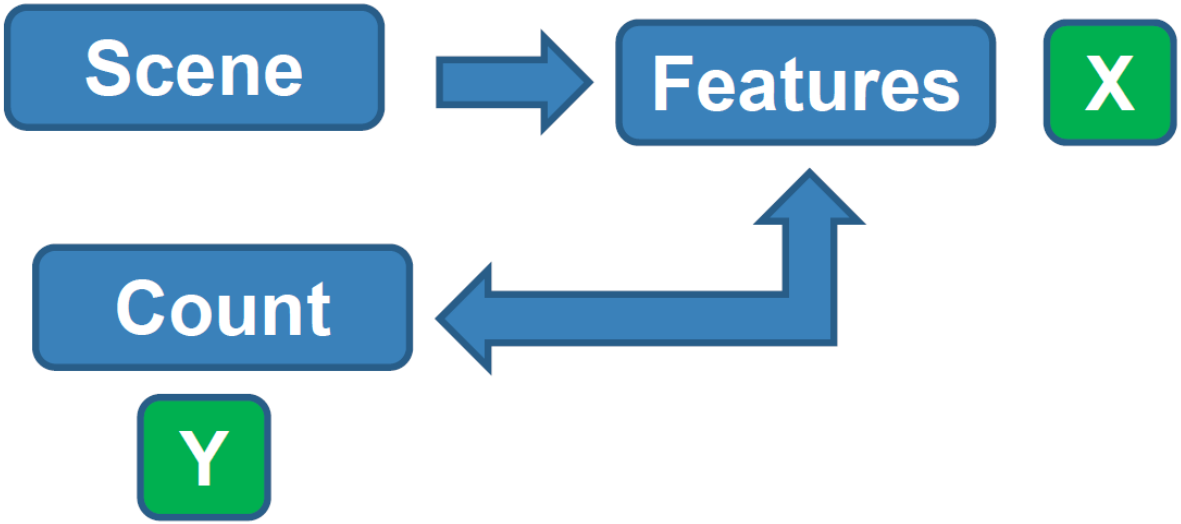
(c) Scale variation

(d) Non-uniform distribution

Extreme cases

Crowd Density Estimation

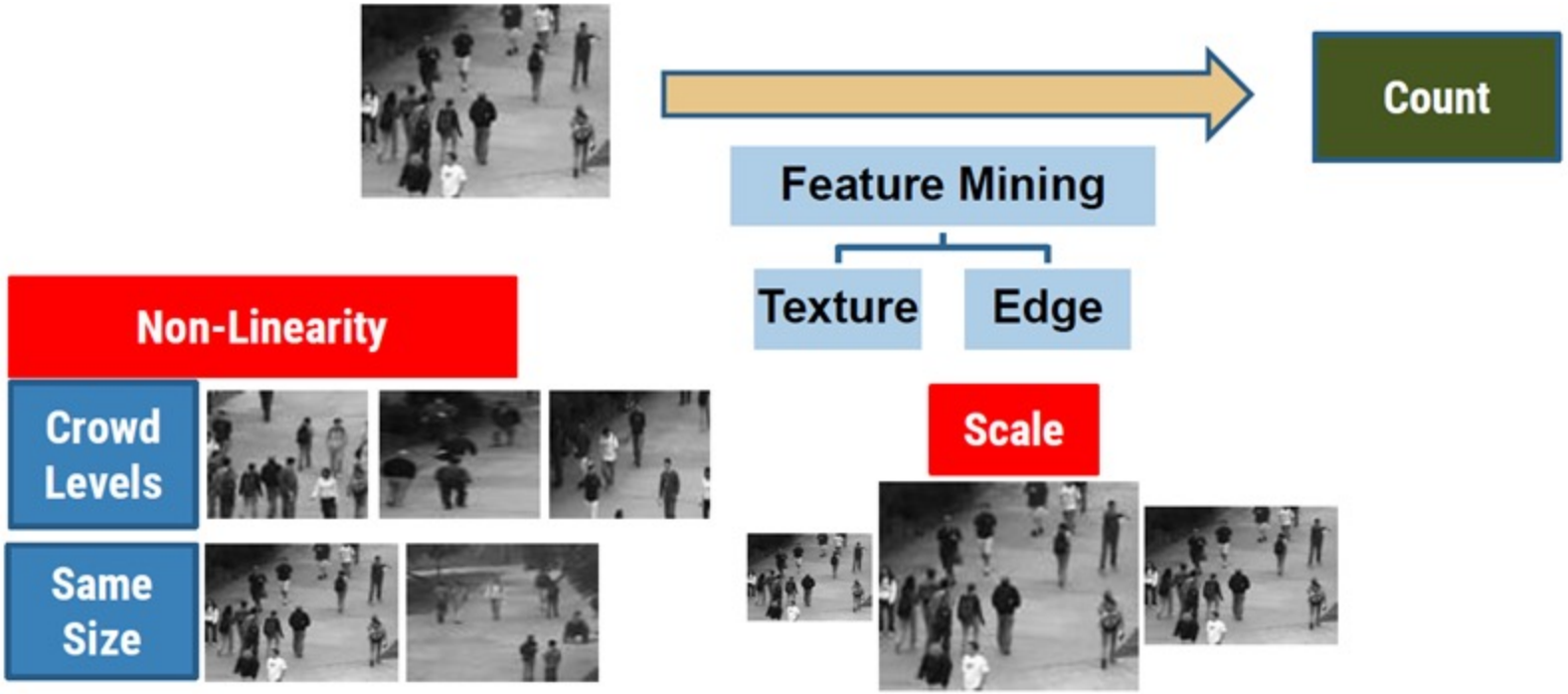
- Solves the requirements to detect and track objects
- Counting based on groups not individuals.



Loy, Chen Change, et al. "Crowd counting and profiling: Methodology and evaluation." Modeling, Simulation and Visual Analysis of Crowds. Springer, New York, NY, 2013. 347-382.

Basic Approach

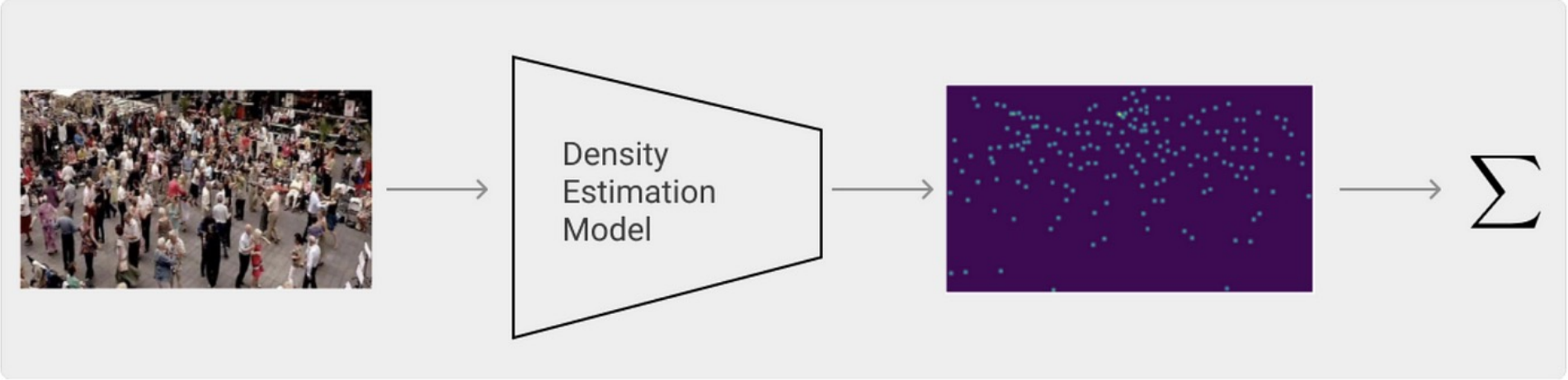
- Counting the people in a scene is not straight forward.



Density Map COnstruction

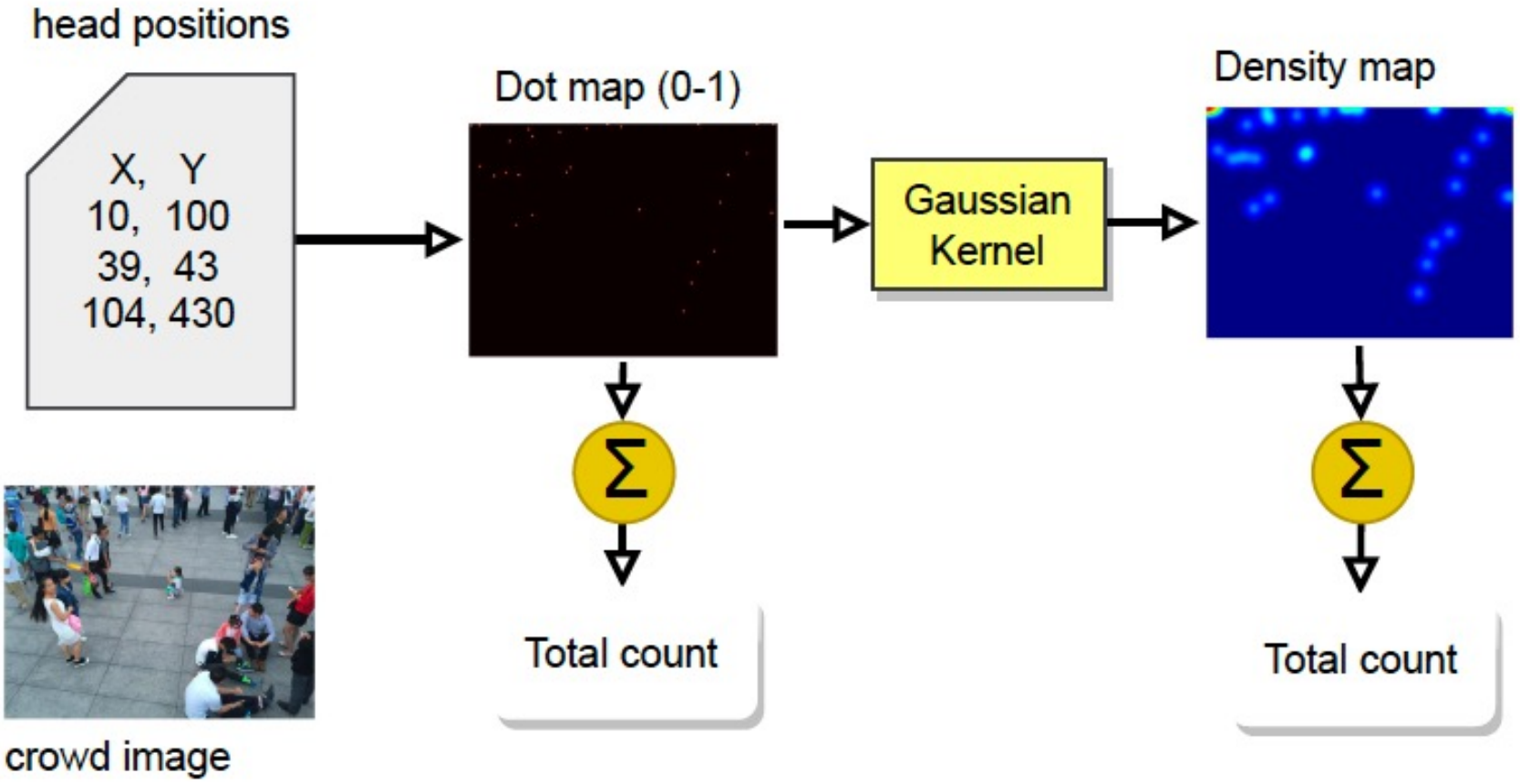
Crowd Density Map

- By estimating the density map, we can derive the locations. To calculate the total count, we summarize the density over the whole image.



More examples

Ground Truth generation for crowd counting models.

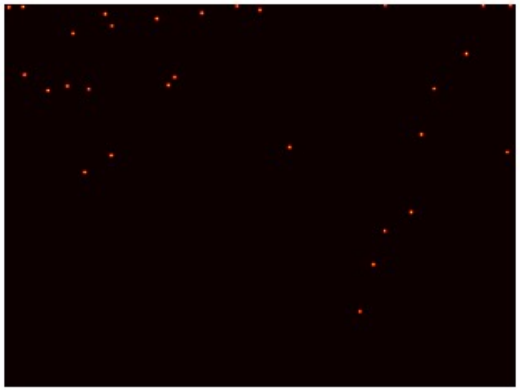


$$Y = \sum_{i=1}^{N} \delta(x - x_i) * G_\sigma$$

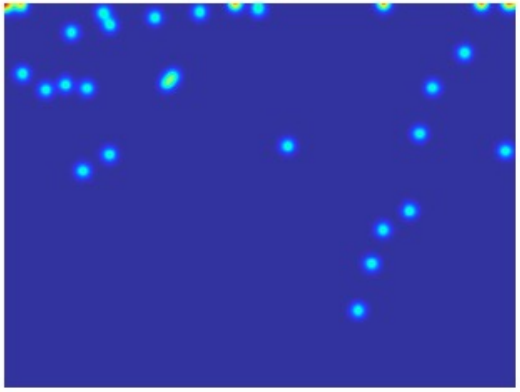
Density maps with different scale (σ) values.

Input image

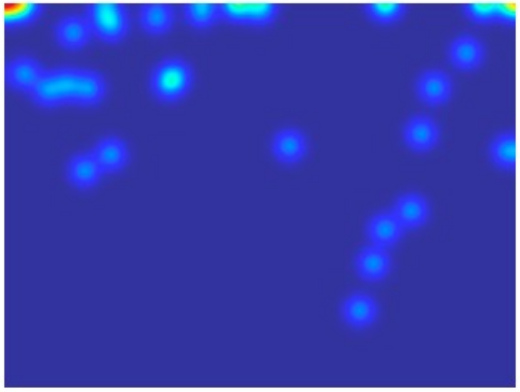
Dot map



Density map (σ=10)



Density map (σ=20)

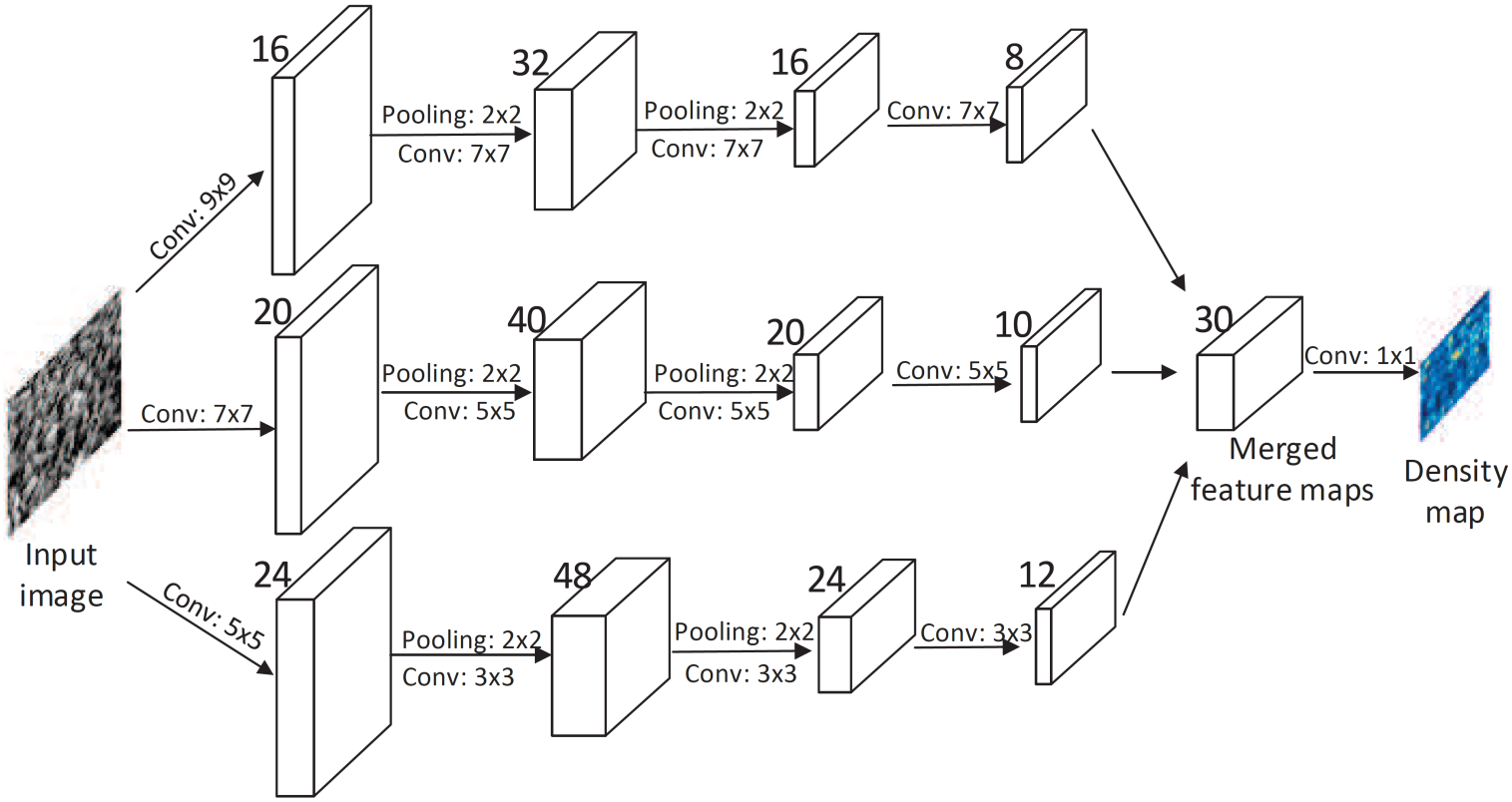


Typical values of σ used in earlier works are 15.

X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in ECCV, 2018.

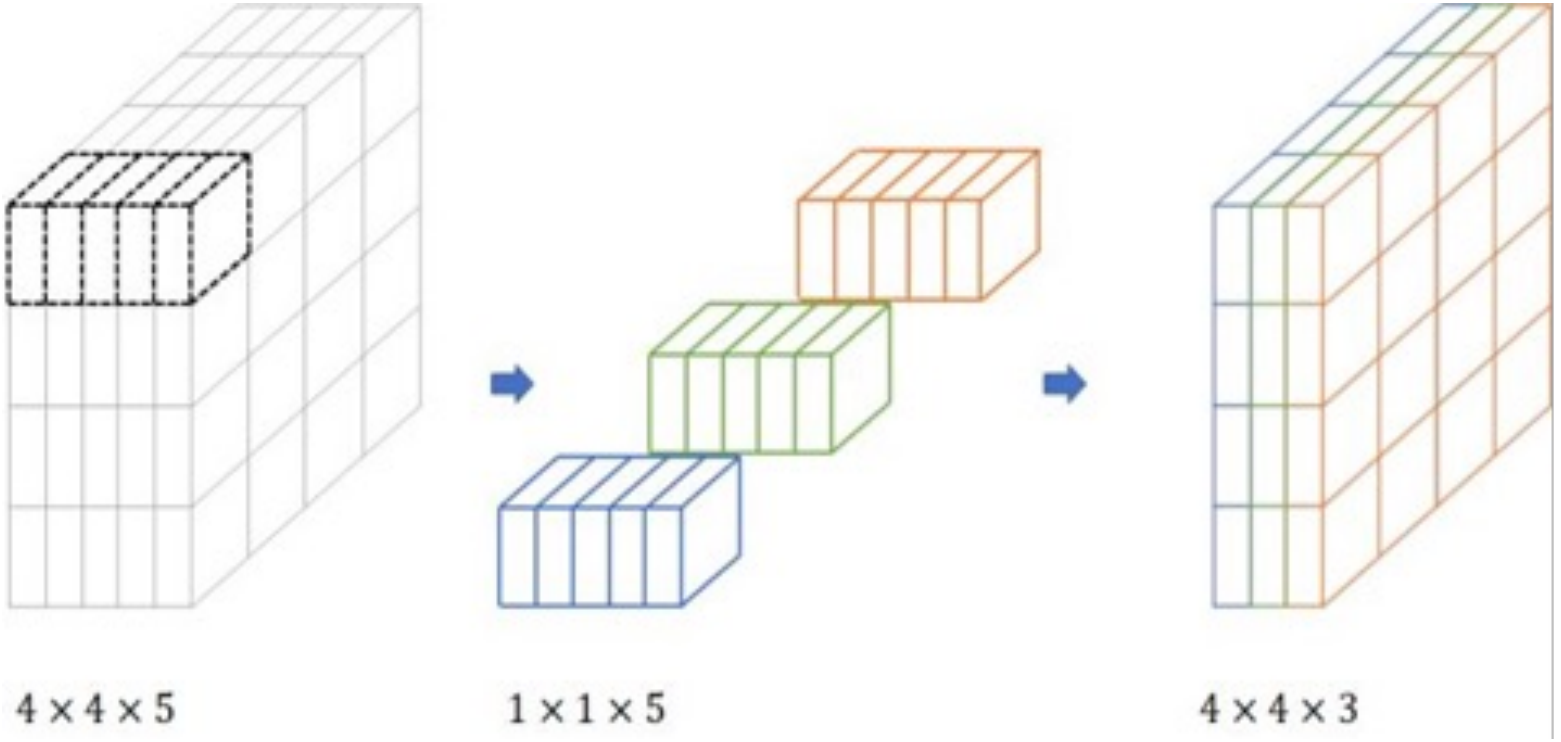
Single Image Crowd Counting via Multi Column Convolutional Neural Network (CVPR'16)

MCNN (CVPR'16)



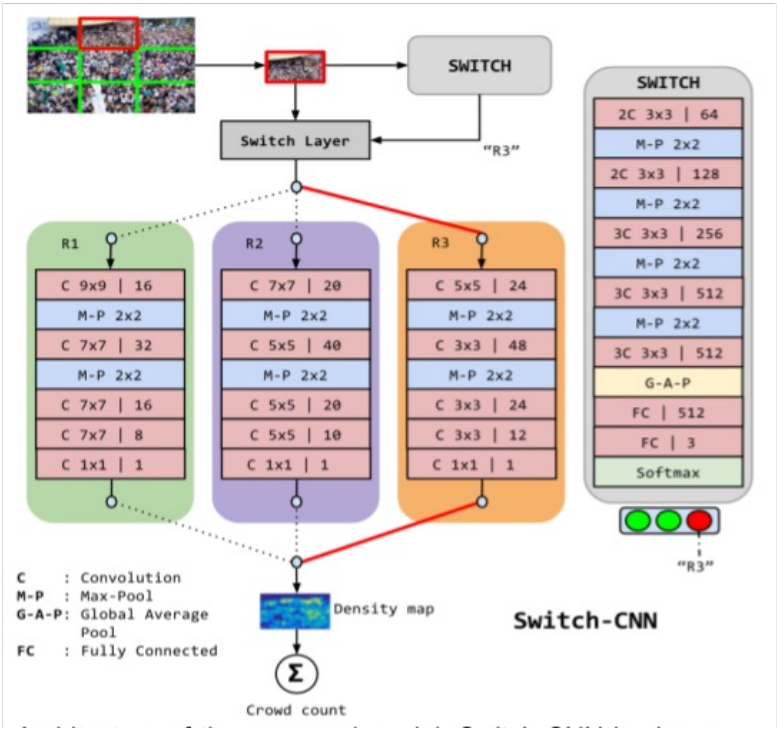
Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single Image Crowd Counting via Multi Column Convolutional Neural Network. CVPR'16.

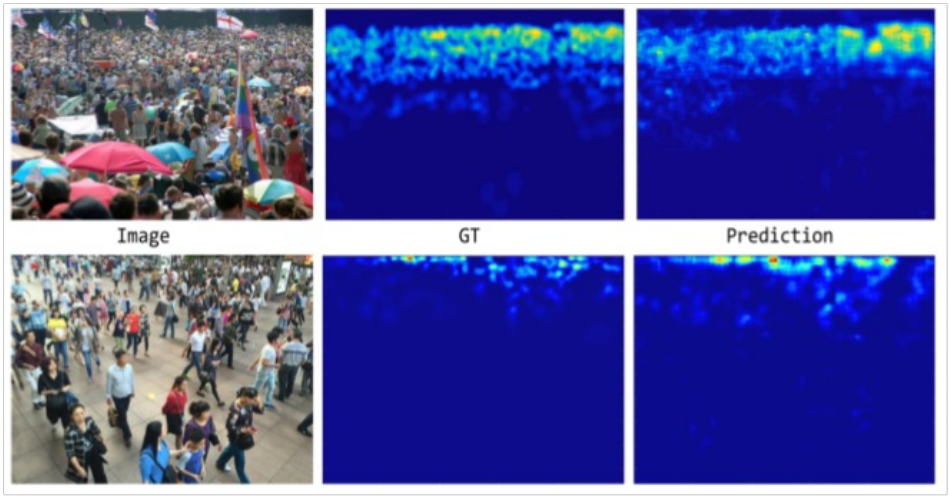
1x1 convolution



Switching Convolutional Neural Network for Crowd Counting (CVPR'17)

Model Architecture





Sample predictions by Switch-CNN for crowd scenes from the ShanghaiTech dataset is shown. The top and bottom rows depict a crowd image, corresponding ground truth and prediction from Part A and Part B of dataset respectively.

Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. Switching Convolutional Neural Network for Crowd

CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes (CVPR’18)

Motivations

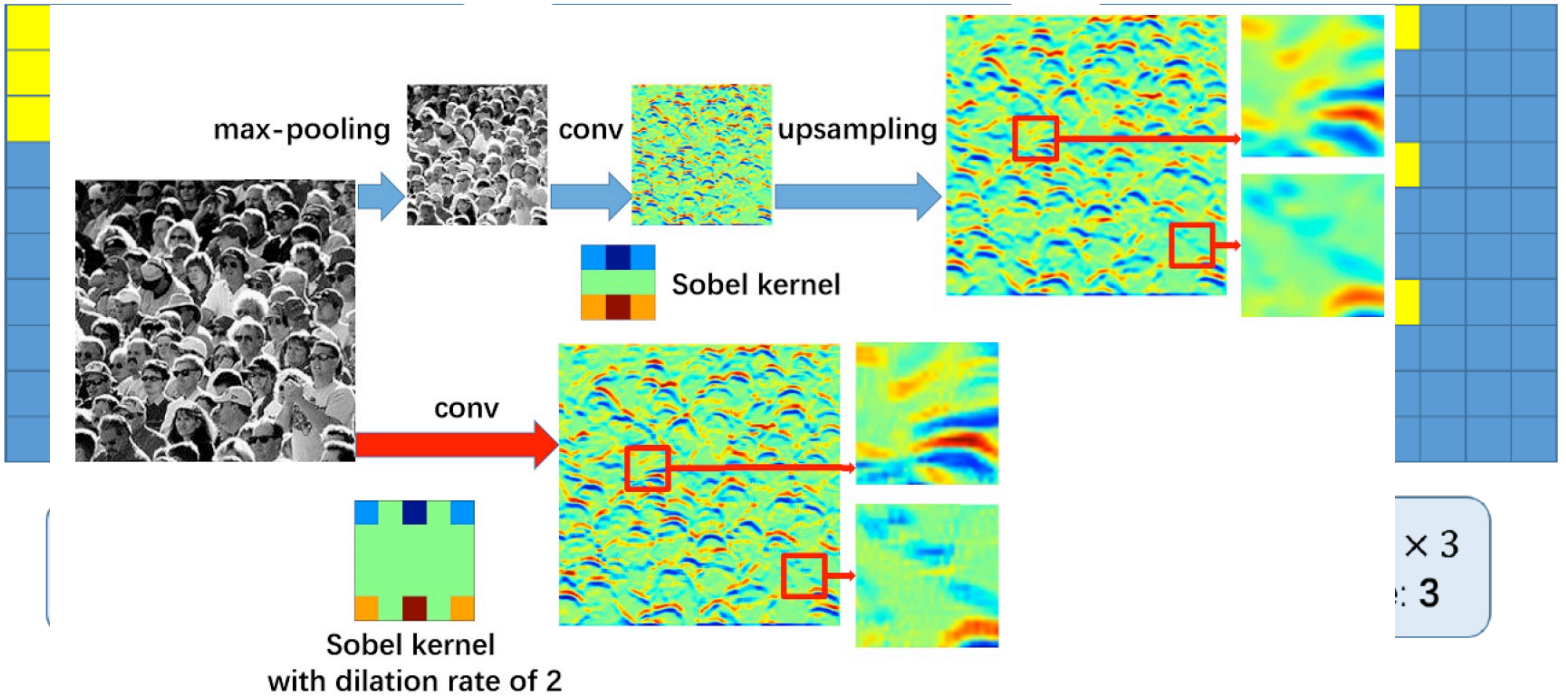
Method	Parameters	MAE	MSE
Col. 1 of MCNN	57.75k	141.2	206.8
Col. 2 of MCNN	45.99k	160.5	239.0
Col. 3 of MCNN	25.14k	153.7	230.2
MCNN Total	127.68k	110.2	185.9
A deeper CNN	83.84k	**93.0**	**142.2**

Testing Sample

Yuhong Li, Xiaofan Zhang, Deming Chen. CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. CVPR'18.

CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes

- VGG-16 as the front-end of CSRNet
- Dilated convolutional layers as the back-end

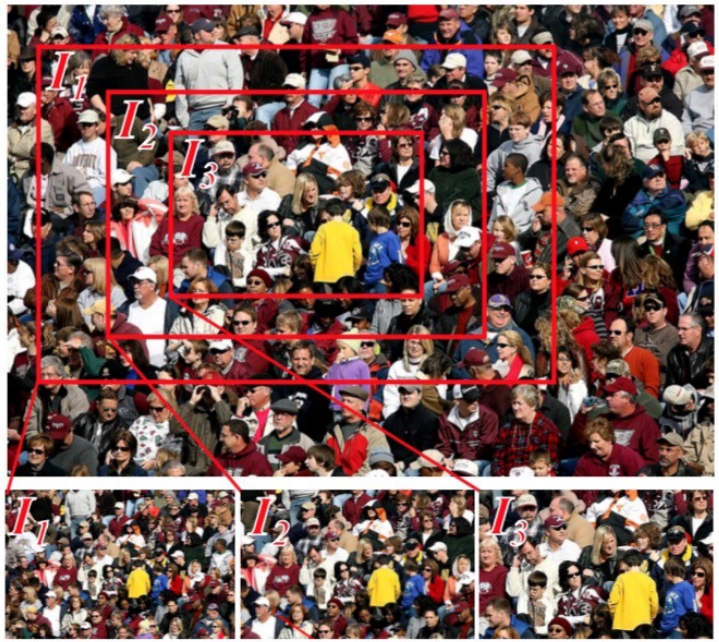


Yuhong Li, Xiaofan Zhang, Deming Chen. CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. CVPR'18.

Leveraging Unlabeled Data for Crowd Counting by Learning to Rank (CVPR'18)

Motivations

- The datasets of crowd counting are usually small since manual labeling is time-consuming.



Xialei Liu, Joost van de Weijer, Andrew D. Bagdanov. Leveraging Unlabeled Data for Crowd Counting by Learning to Rank. CVPR'18

Leveraging Unlabeled Data for Crowd Counting by Learning to Rank (CVPR'18)

Algorithm 1 : Algorithm to generate ranked datasets.

Input: A crowd scene image, number of patches k and scale factor s.

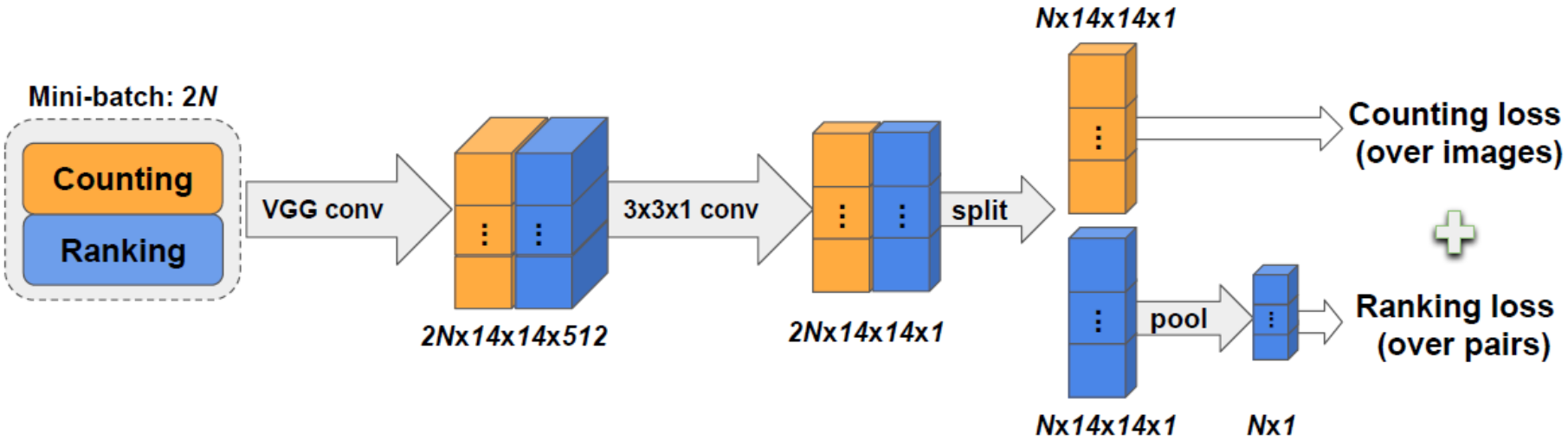
Step 1: Choose an anchor point randomly from the anchor region. The anchor region is defined to be $1/r$ the size of the original image, centered at the original image center, and with the same aspect ratio as the original image.

Step 2: Find the largest square patch centered at the anchor point and contained within the image boundaries.

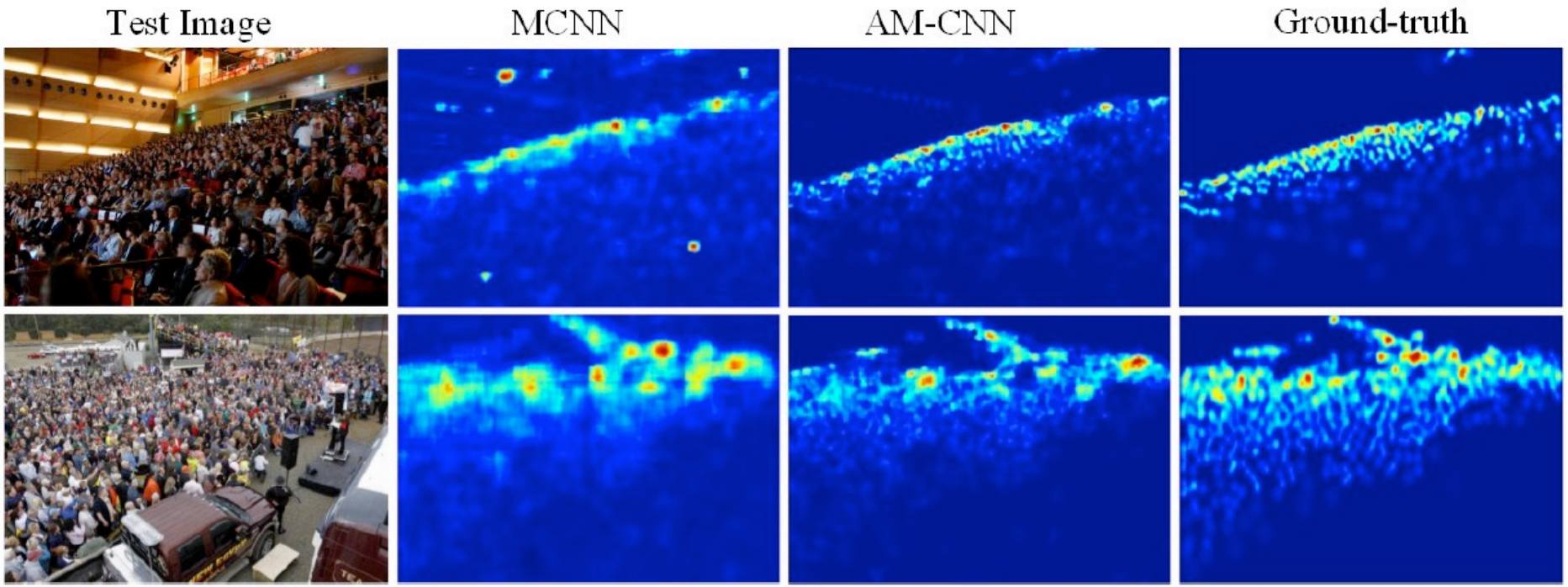
Step 3: Crop $k-1$ additional square patches, reducing size iteratively by a scale factor s. Keep all patches centered at anchor point.

Step 4: Resize all k patches to input size of network.

Output: A list of patches ordered according to the number of persons in the patch.



Attention to Head Locations for Crowd Counting (WACV'19)



$$S = \varphi(W \circledcirc f^i + b) \quad M_p = \frac{e^{S_p}}{\sum_{p' \in P} e^{S_{p'}}}$$

$$L_{ED} = \frac{1}{N}\sum_{i=1}^{N} \frac{(F(X_i, \Theta) - D_i)^2}{Pix_i} \qquad L_{RD} = \frac{1}{N}\sum_{i=1}^{N} (\frac{y_i - y_i^{'}}{y_i + z})^2$$

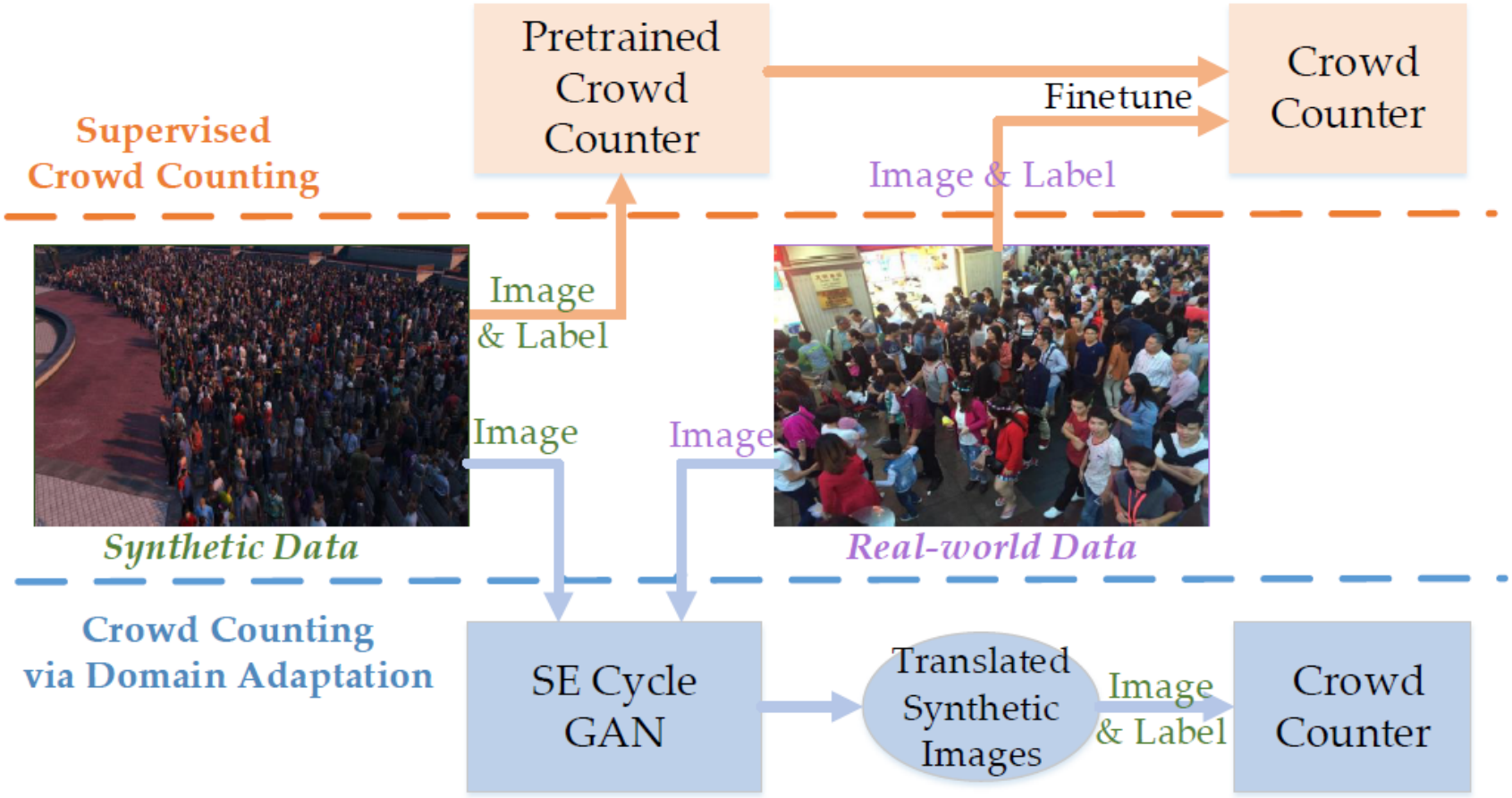
Youmei Zhang, Chunluan Zhou, Faliang Chang, and Alex C. Kot. Attention to Head Locations for Crowd Counting. WACV'19.

. Learning from Synthetic Data for Crowd Counting in the Wild (CVPR'19)

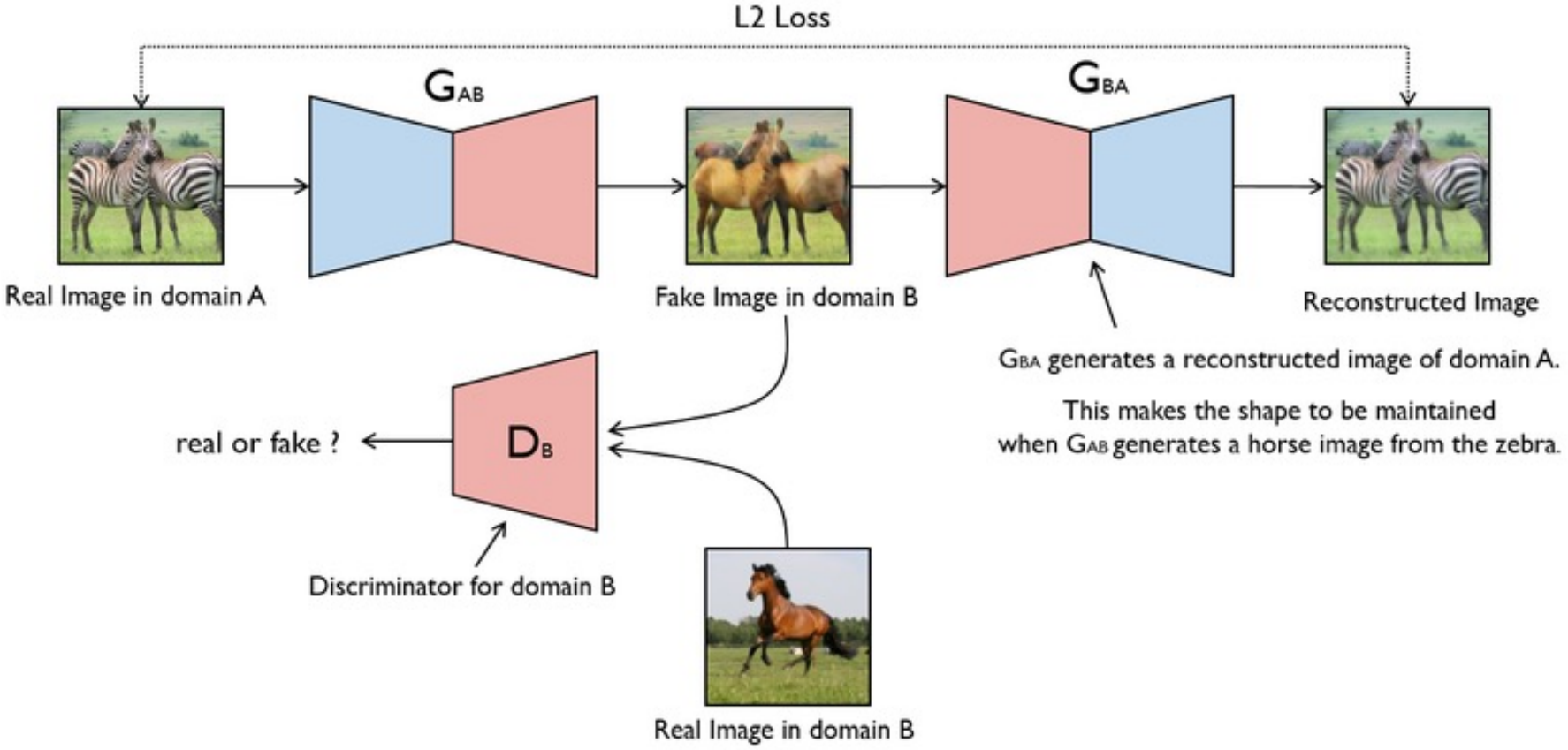
GCC dataset consists of 15,212 images, with resolution of 1080×1920, containing 7,625,843 persons. Compared with the existing datasets, GCC is a more large-scale crowd counting dataset in both the number of images and the number of persons.

Qi Wang, Junyu Gao, Wei Lin, Yuan Yuan. Learning from Synthetic Data for Crowd Counting in the Wild. CVPR'19.

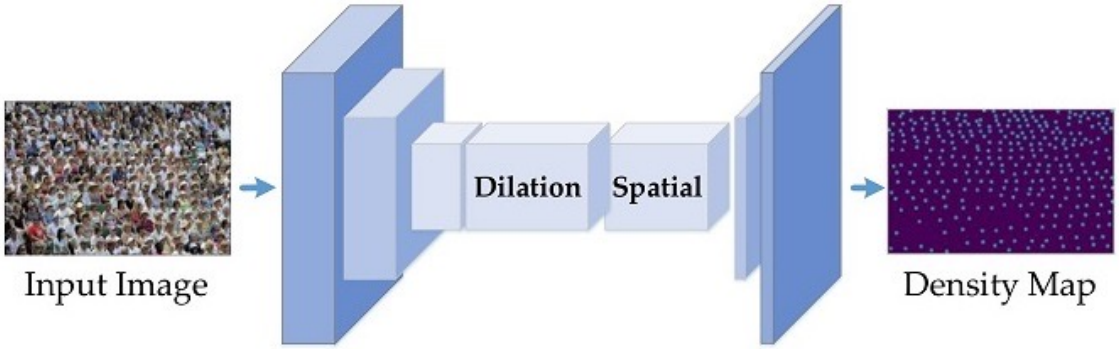
Using the GCC dataset



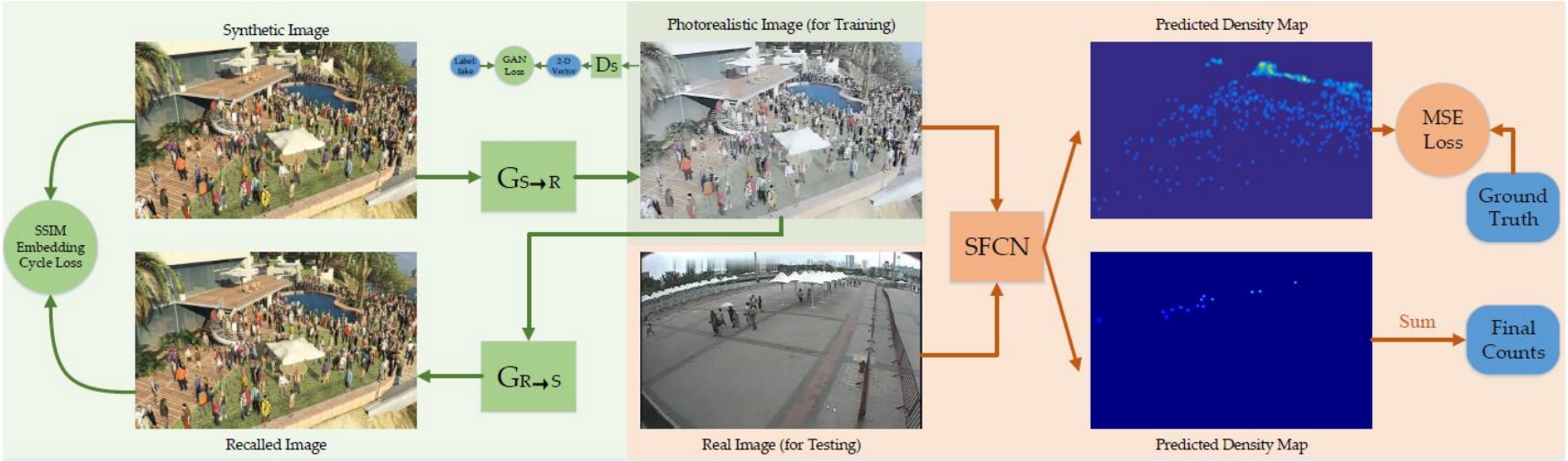
CycleGAN


L2 Loss
G_{AB}
G_{BA}
Real Image in domain A
Fake Image in domain B
Reconstructed Image
G_{BA} generates a reconstructed image of domain A.
This makes the shape to be maintained
when G_{AB} generates a horse image from the zebra.
real or fake ?
D_B
Discriminator for domain B
Real Image in domain B

Supervised Crowd Counting



Crowd Counting via Domain Adaptation



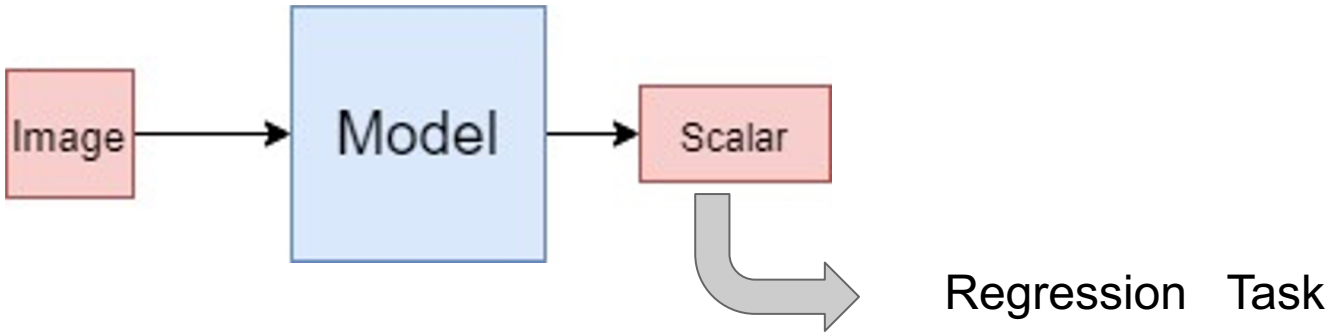
Qi Wang, Junyu Gao, Wei Lin, Yuan Yuan. Learning from Synthetic Data for Crowd Counting in the Wild. CVPR'19.

Point in, Box out: Beyond counting persons in Crowds (CVPR 2019)

Problem Formulation

Input : image or video

Output : scalar (number of people in ROI)



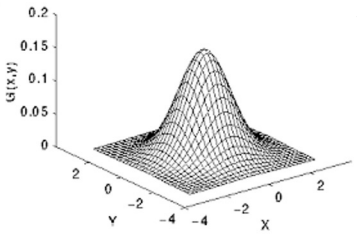
Crowd Density Map

(1) Has same size (H , W) as the input size

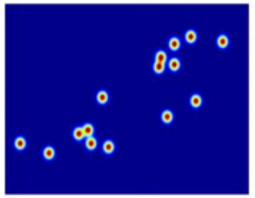
(2) More informative than crowd count (scalar number)

(3) We can compute the number of people in a particular region (or the entire region)

(4) We can adjust the parameters (variance) to deal with perspective distortion



$$\forall p \in I_i,\ F_i(p) = \sum_{P \in \mathcal{P}_i} \mathcal{N}(p; P, \sigma^2 I_{2\times 2})$$



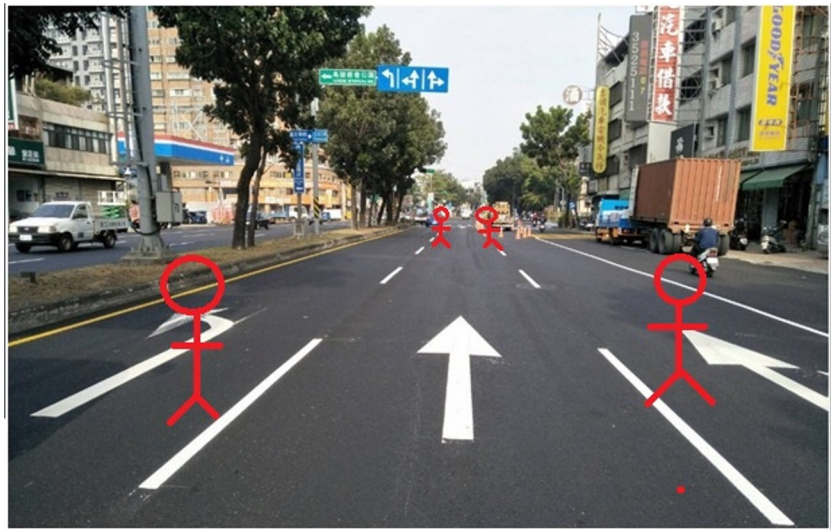
Ground truth: 15

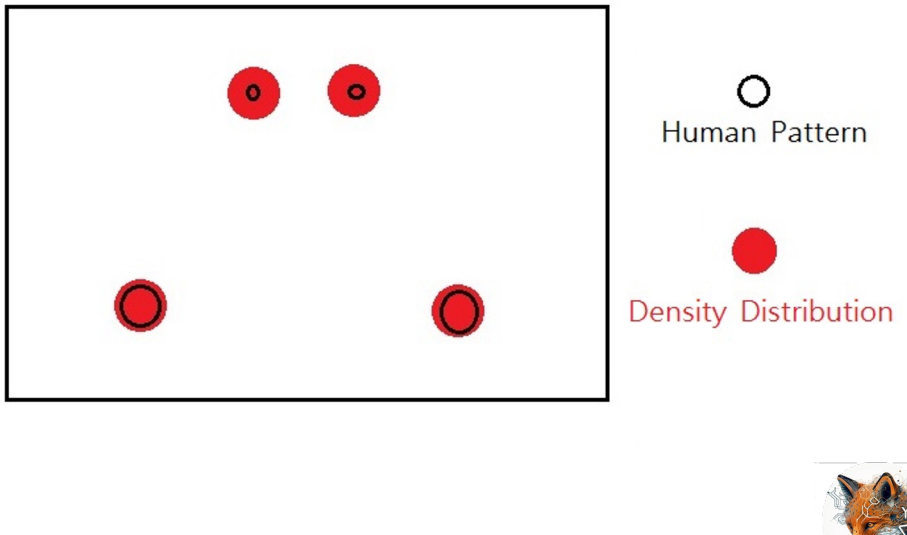
Perspective Distortion in Density Map

Therefore, we need to adjust the variance of gaussian distribution for one person.

遠景 = variance小

近景 = variance大





Related Work

. Detection-based Method

- detect individual person, then counting becomes trivial

Drawback: Hardly detect people due to the occlusion and deformation + box label is costly

. Scalar Regression-based Method

- mapping extracted features to a scalar number

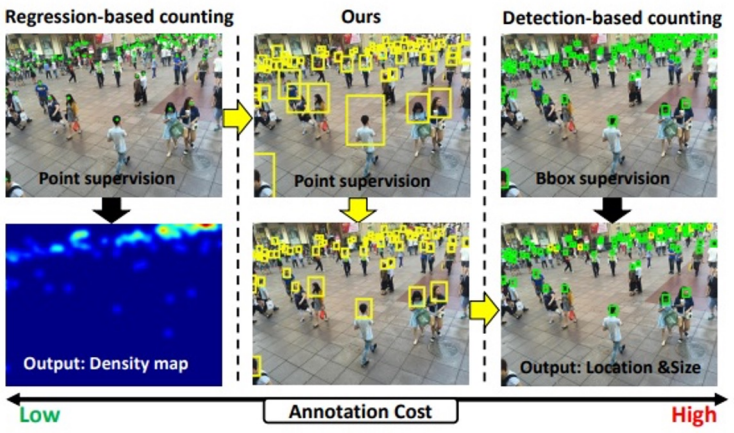
Drawback: Still getting poor performance in highly congested scene

. Density map Regression-based Method

mapping extracted features to density map

Advantage: More informative and more powerful due to the supervision in local region

Introduction



Propose a detection-based model in counting task with only point annotations

challenge 1 : No bounding box label , how to detect and train ?

⇨ Propose a pseudo ground truth updating scheme

challenge 2 : pseudo bounding box label may be unreliable ?

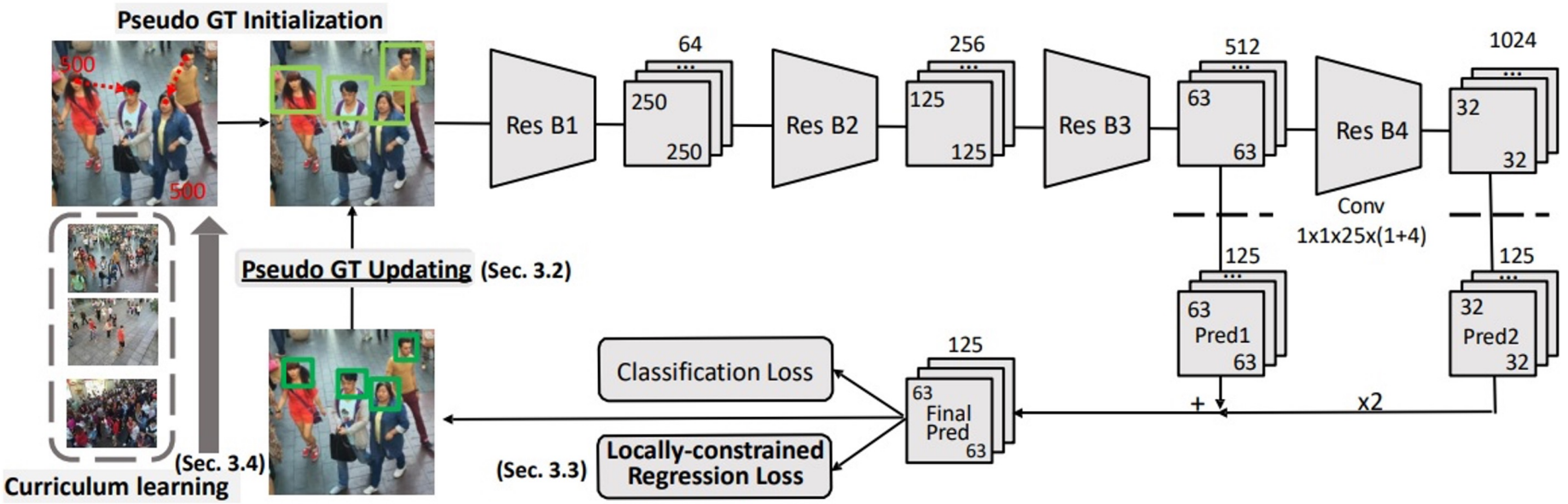
⇨ Propose the locally-constrained regression loss on the box label by intuitive observation

challenge 3 : Dense or Sparse , the training difficulty varies

⇨ Propose a curriculum learning strategy to improve the training process

Model Architecture

Based on anchor based detection framework



Scale Variations

Method 1-1 : Pseudo GT initialization

Now what we only have is point label ⇨ How to get bounding box label ?

An intuitive observation

⇨ When two people are close enough, their head distance reflects their head size

For a point label, the size of initial bounding box GT = $d(g, NN_g)$

Method 1-1 : Pseudo GT example

Does it make sense ?

YES !

Generally,

dense scene often has a smaller initialization

sparse scene often has a bigger initialization

Method 1-2 : Pseudo GT updating

positive and negative sample is defined by the IOU between predicted anchor box and GT box label.

Now we have bounding box label

⇨ But it seems like pseudo label isn't accurate enough

↓

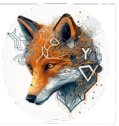
Epoch by epoch, iteratively update the pseudo box label

↓

In a new epoch, for one person in point label , over the positive anchor boxes , select the highest scored one whose size is smaller than previous pseudo box label .

↓

New pseudo box label (smaller width or height) ⇨ replace original pseudo box label

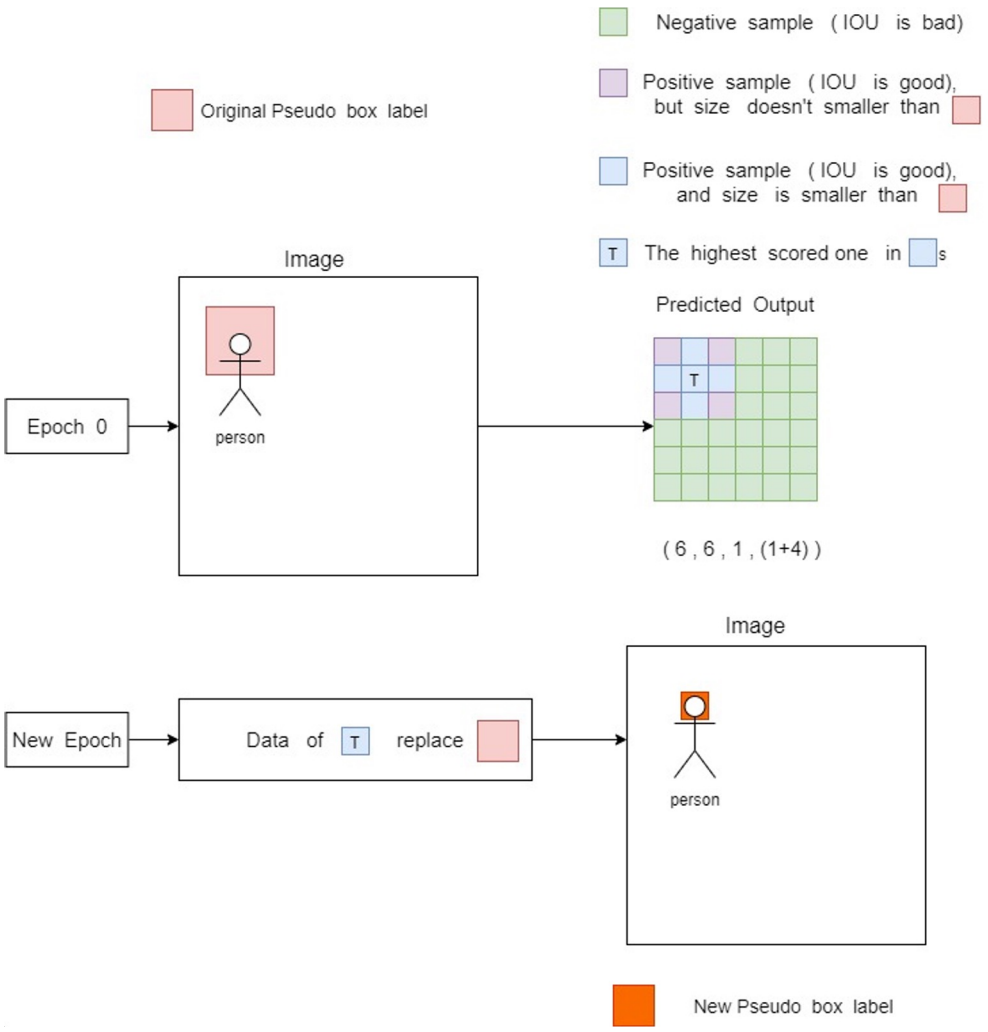


Method 1-2 :

Pseudo GT updating example

Because there are 25 anchor boxes in one grid,

it is dense, which can guarantee that label can be updated with suitable predictions iteratively.



Method 2: Locally-constrained regression Loss

Total Loss = L_{cls} + L_{reg}

L_{cls} = Binary cross entropy for every anchor box.

Because we have real point label, so the box center coordinates (gx , gy) are accurate

$$lxy = (gx - \widehat{gx})^2 + (gy - \widehat{gy})^2$$

But (gw , gh) are not very accurate ! We can not adapt the original L_{reg} for (gw , gh) in Faster R-CNN

GT box = (gx , gy , gw , gh)

anchor = (ax , ay , aw , ah)

predicted box data = (d_x , d_y , d_w , d_h)

$$\widehat{gx} = aw \cdot d_x(a) + ax,\ \widehat{gy} = ah \cdot d_y(a) + ay$$
$$\widehat{gw} = aw \cdot \exp(d_w(a)),\ \widehat{gh} = ah \cdot \exp(d_h(a))$$

Method 2: How to define L_{wh}

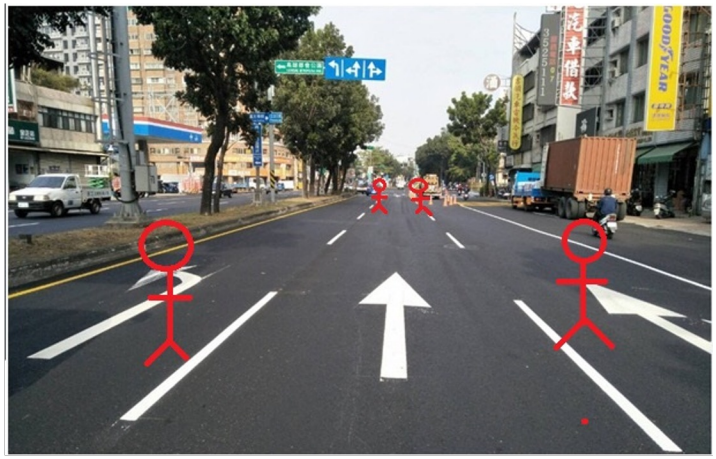
By another intuitive observation :

↓

Bounding box of people along the same horizontal line should have similar size

↓

Due to the perspective distortion !



Method 2: Locally-constrained regression Loss

Penalize $\hat{g}_w$ and $\hat{g}_h$ if it clearly violates the **Observation**

$$\mu w_i = \frac{1}{|G_i|} \sum_{mn \in G_i} gw_{mn}$$

$$\sigma w_i = \sqrt{\frac{1}{|G_i|} \sum_{mn \in G_i} (gw_{mn} - \mu w_i)^2}$$

$$lw_{ij} = \begin{cases} (\widehat{gw}_{ij} - (\mu w_i + 3\sigma w_i))^2 & \widehat{gw}_{ij} > \mu w_i + 3\sigma w_i \\ ((\mu w_i - 3\sigma w_i) - \widehat{gw}_{ij})^2 & \widehat{gw}_{ij} < \mu w_i - 3\sigma w_i \\ 0 & otherwise \end{cases}$$

GT box = (gx , gy , gw , gh)

anchor = (ax , ay , aw , ah)

predicted box data = (d_x , d_y , d_w , d_h)

$$\widehat{gx} = aw \cdot d_x(a) + ax, \ \widehat{gy} = ah \cdot d_y(a) + ay$$

$$\widehat{gw} = aw \cdot \exp(d_w(a)), \ \widehat{gh} = ah \cdot \exp(d_h(a))$$

i , j = the index of output map

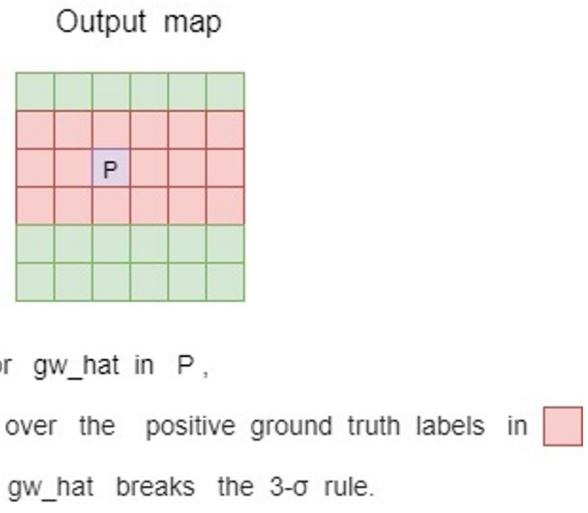
Method 2: Locally-constrained regression Loss

Penalize $\hat{g}_w$ and $\hat{g}_h$ if it clearly violates the **Observation**

$$\mu w_i = \frac{1}{|G_i|} \sum_{mn \in G_i} gw_{mn}$$

$$\sigma w_i = \sqrt{\frac{1}{|G_i|} \sum_{mn \in G_i} (gw_{mn} - \mu w_i)^2}$$

$$lw_{ij} = \begin{cases} (\widehat{gw}_{ij} - (\mu w_i + 3\sigma w_i))^2 & \widehat{gw}_{ij} > \mu w_i + 3\sigma w_i \\ ((\mu w_i - 3\sigma w_i) - \widehat{gw}_{ij})^2 & \widehat{gw}_{ij} < \mu w_i - 3\sigma w_i \\ 0 & otherwise \end{cases}$$



Method 2: Total Loss

$$Lreg = \sum_{ij \in G} \widetilde{lxy}_{ij} + \widetilde{lw}_{ij} + \widetilde{lh}_{ij},$$

lh_{ij} is similar to lw_{ij}

Method 3 : Curriculum learning

Sparse scene --- bigger initialization

↓

Dense scene --- smaller initialization

↓

Both are improper to train when model is a beginner

↓

Train the easier ones first

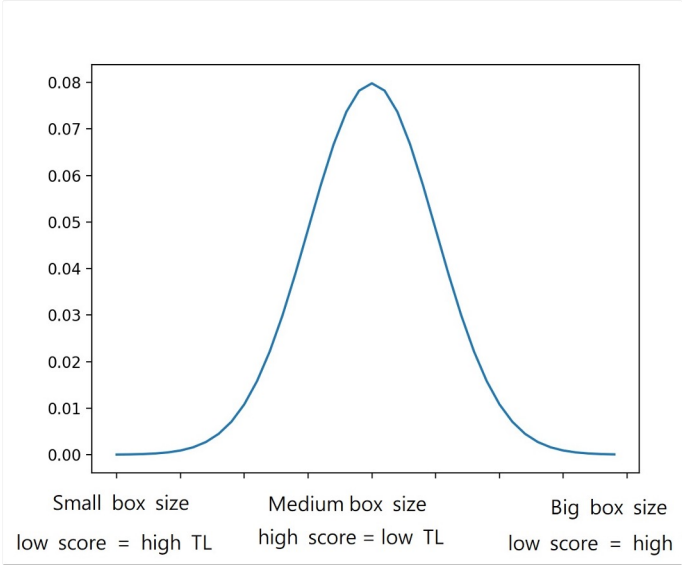
(not too big and too small initialized ones)

Method 3 :Curriculum learning

Define the learning difficulty :

Over entire training set,

compute μ and σ of $d(g , NN_g)$



For each image, compute the training difficult $\mathrm{TL} = 1 - \frac{1}{|G|} \sum_{g \in G} \Phi(d_g | \mu, \sigma)$

G is the set of bounding box set in the image

Φ is gaussian function whose mean = μ , std = σ

Method 3 : Curriculum learning

according to TL ,

split training set into Z folds

Train Z1 over some epochs……

Train Z1 + Z2 over some epochs…

And so on……

Experiments --- Counting task

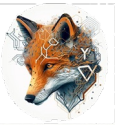
Dataset	SHA		SHB	
Measures	MAE	MSE	MAE	MSE
Pv0	168.6	268.3	69.8	98.1
Pv1	104.7	193.8	41.7	66.6
Pv2	89.8	169.5	19.1	42.4
Pv3(PSDDN)	85.4	159.2	16.1	27.9
PSDDN + [20]	**65.9**	**112.3**	**9.1**	**14.2**
Li et al. [20]	**68.2**	115.0	**10.6**	**16.0**
Ranjan et al. [31]	68.5	116.2	10.7	16.0
Liu et al. [24]	73.6	112.0	13.7	21.4
Liu et al. [22]	-	-	20.7	29.4
DetNet in [22]	-	-	44.9	73.2
Sindagi et al. [41]	73.6	**106.4**	20.1	30.1
Sam et al. [35]	90.4	135.0	21.6	33.4

Counting	UCF		
Measures	MAE	MSE	AP
Li et al. [20]	**266.1**	**397.5**	-
Liu et al. [24]	279.6	388.9	-
Sindagi et al. [41]	295.8	320.9	-
Sam et al. [35]	318.1	439.2	-
PSDDN	359.4	514.8	0.536

Methods	GAME0	GAME1	GAME2	GAME3	AP
Victor et al. [19]	13.76	16.72	20.72	24.36	-
Onoro et al. [27]	10.99	13.75	16.09	19.32	-
Li et al. [20]	**3.56**	5.49	8.57	15.04	-
PSDDN	4.79	**5.43**	**6.68**	**8.40**	**0.669**

Table 5: Results on TRANCOS dataset.

Pv0 : fixed GT label + classic regression Loss in faster R-CNN
Pv1 : iterative updated GT label + classic regression Loss
Pv2 : iterative updated GT label + new regression Loss in method
Pv3 : Pv2 + curriculum learning
PSDDN + : Pv3 + attention module proposed in DecideNet [22]



Experiments --- Detection task

Methods	Annotations	WiderFace		
		easy	medium	hard
Avg. BB	points(test)+ mean size	0.002	0.083	0.059
FR-CNN (ps)	points(train) + mean size	0.008	0.183	0.108
FR-CNN (fs)	bounding boxes (train)	**0.840**	**0.724**	0.347
PSDDN	points(train)	0.605	0.605	**0.396**

Dataset	Pv0	Pv1	Pv2	Pv3 (PSDDN)
SHA	0.308	0.491	0.539	0.554
SHB	0.015	0.241	0.582	0.663

Table 3: Person detection: ablation study of PSDDN on ShanghaiTech (SHA and SHB) dataset. AP is reported.

Experiments

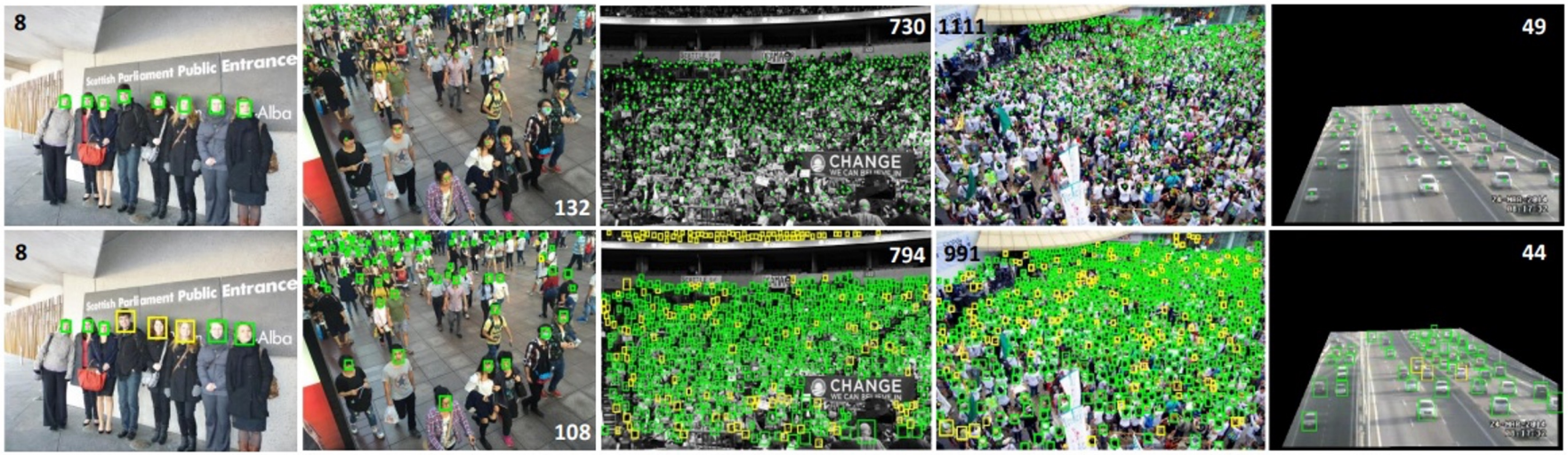


Figure 3: Examples from WiderFace, SHB, UCF, SHA, and TRANCOS datasets. The top row is test images with ground truth (bounding

Conclusion

The proposed detection-based model can achieve remarkable results compared with other density regression-based method.

Pseudo ground truth label updating scheme is effective when we don’t have real label.

Curriculum learning strategy can prevent the model from a bad training process.

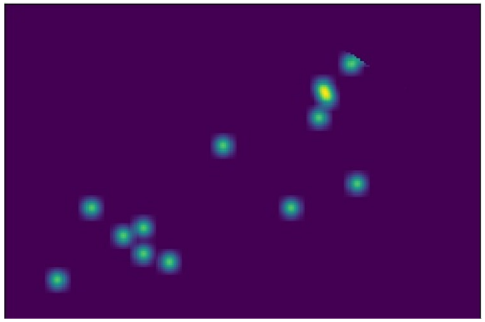
One-Shot Scene-Specific Crowd Counting (BMVC 2019)

Outline

- Problem Formulation
- Motivation
- Related Work
- Method
- Experiment
- Conclusion

Problem Formulation

Input :



Task : Pixel-wise Regression Problem

Motivation (Introduction)

Properties or Assumptions of Previous works :

Assumption 1 : The model can be general enough to handle all possible scenes when in the

real-world applications.

Assumption 2 : There is no any fine-tuning in the testing (inference) stage.

⇒ Unnecessary demands !

⇒ In many real-world applications, we can just fine-tune our model, and it only need to perform well in the target scene (specific scene).

Motivation (Introduction)

What we want : The model does perform well in the target scene when in the real-world application.

Intuitive solution : It's too easy, let's fine-tune !

⇨ Fails.

- Challenge1 : We may only have few images in the target scene
- Challenge2 : The training policy is too naive. Such brute-force approach may not perform very well.

Motivation (Introduction)

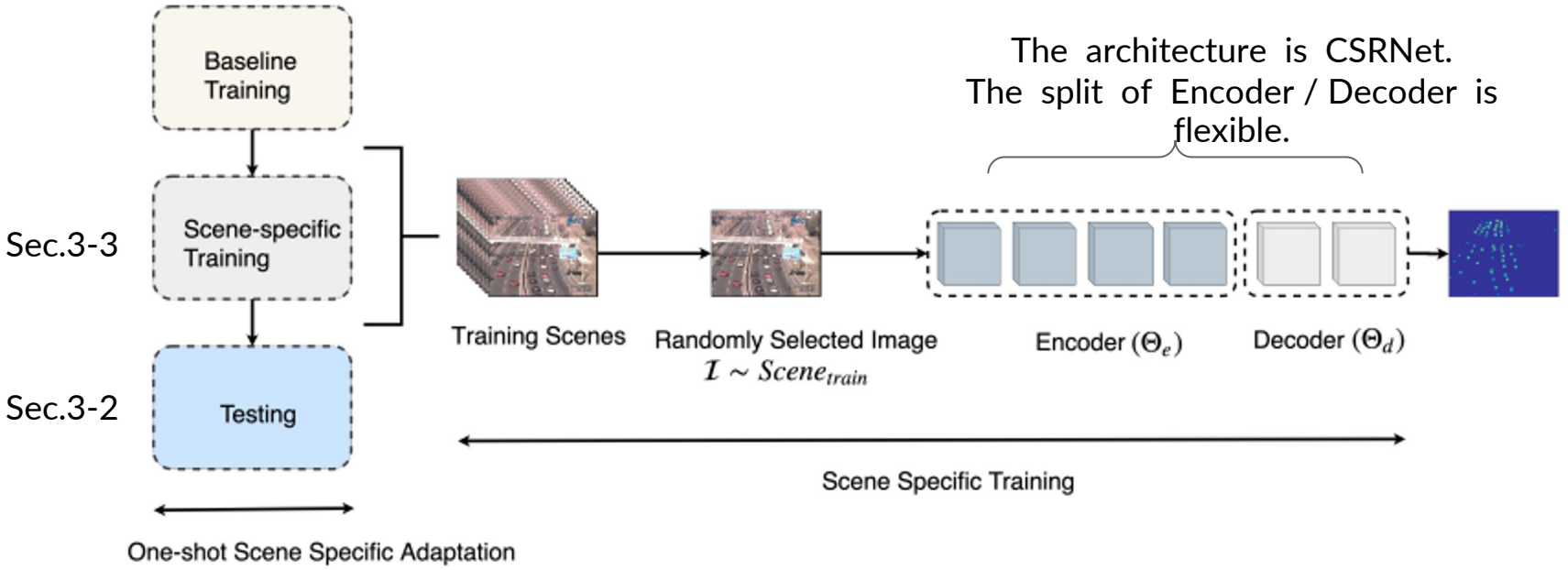
Abstract :

Propose a training strategy to help the model to have a good transferable fine-tune capability.

Only use 1 image in the target scene!

FEW-SHOT LEARNING/ META LEARNING!

Method


Baseline Training
Scene-specific Training
Testing
Sec.3-3
Sec.3-2
The architecture is CSRNet.
The split of Encoder / Decoder is flexible.
Training Scenes
Randomly Selected Image
$\mathcal{I} \sim Scene_{train}$
Encoder (Θ_e)
Decoder (Θ_d)
Scene Specific Training
One-shot Scene Specific Adaptation

Sec.3-2 : One-Shot Scene-Specific Adaption

During testing, we are given a collection of M images $\{x_i\}_{i=1}^{M}$ from the *same* scene.

We only have one labelled image for this scene (x_1, y_1)

Now, we fix the parameters of Encoder Network Θ_e

And only fine-tune the parameters of Decoder Network Θ_d

$$\Theta_d^* = \arg\min_{\Theta_d} L_d(\Theta_d), \quad \text{where } L_d(\Theta_d) = ||\text{vec}(F(x_1; \{\Theta_e, \Theta_d\})) - \text{vec}(y_1)||_2 \qquad (1)$$

Generally, if labelled images are little, then we choose the Decoder Network to have few layers.

Sec.3-3 : Scene-Specific Training

We have labelled training images from T scenes, there are N images for each Scene :

$$\{(x_i^{(t)}, y_i^{(t)})\}_{i=1}^{N} \quad (\text{where } t = 1, 2, \ldots, T)$$

Now, we hope Θ_e to have this property :

Given a target scene t, if we can learn Θ_d by using Eq.(1) and fine-tuning on $(x_1^{(t)}, y_1^{(t)})$

Then Θ_e can make the model to perform well on the remaining part of images for target scene t.

$$\Theta_d^* = \arg\min_{\Theta_d} L_d(\Theta_d), \quad \text{where } L_d(\Theta_d) = ||\text{vec}(F(x_1; \{\Theta_e, \Theta_d\})) - \text{vec}(y_1)||_2 \qquad (1)$$

Sec.3-3 : Scene-Specific Training

Whole Training Set : $\{(x_i^{(t)}, y_i^{(t)})\}_{i=1}^{N}$ (where $t = 1, 2, \ldots, T$)

First, we train Θ_d by $(x_1^{(t)}, y_1^{(t)})$ and Eq.(2)

$$\Theta_d^{(t)} = \arg\min_{\Theta_d} L_d^{(t)}(\Theta_d), \quad \text{where } L_d^{(t)}(\Theta_d) = ||\text{vec}(F(x_1^{(t)}; \{\Theta_e, \Theta_d\})) - \text{vec}(y_1^{(t)})||_2 \qquad (2)$$

Second, suppose the solution of Eq.(2), Θ_d^*, is unique for a fixed Θ_e,

then Θ_d is a function of Θ_e : $\Theta_d^{(t)}(\Theta_e)$

Now, according to the property, we train Θ_e by using the remaining part and Eq.(3)

$$L_e^{(t)}(\Theta_e) = \sum_{i=2}^{N} ||\text{vec}(F(x_i^{(t)}; \{\Theta_e, \Theta_d^{(t)}(\Theta_e)\})) - \text{vec}(y_i^{(t)})||_2 \qquad (3)$$

Sec.3-3 : Scene-Specific Training

Whole Training Set: $\{(x_i^{(t)}, y_i^{(t)})\}_{i=1}^{N}$ (where $t = 1, 2, \ldots, T$)

Total loss for all scenes (Note that it is a function of Θ_e)

$$L_{final}(\Theta_e) = \sum_{t=1}^{T} L_e^{(t)}(\Theta_e)$$

But it can not derive the gradient easily. Since there is another optimization problem (2) in (3)

$$\Theta_d^{(t)} = \arg\min_{\Theta_d} L_d^{(t)}(\Theta_d), \quad \text{where } L_d^{(t)}(\Theta_d) = ||\text{vec}(F(x_1^{(t)}; \{\Theta_e, \Theta_d\})) - \text{vec}(y_1^{(t)})||_2 \qquad (2)$$

$$L_e^{(t)}(\Theta_e) = \sum_{i=2}^{N} ||\text{vec}(F(x_i^{(t)}; \{\Theta_e, \Theta_d^{(t)}(\Theta_e)\})) - \text{vec}(y_i^{(t)})||_2 \qquad (3)$$

Sec.3-3 : Scene-Specific Training

Whole Training Set: $\{(x_i^{(t)}, y_i^{(t)})\}_{i=1}^{N}$ (where $t = 1, 2, \ldots, T$)

Alternative approach : if Θ_d is fixed (not the argmin), then the gradient of $L_{final}(\Theta_e)$ can be computed easily.

Therefore, using the coordinate-descent style approach :

Fix Θ_e, solve for $\Theta_d^{(t)}$ (where $t = 1, 2, \ldots, T$) using Eq. 2 for each scene;
Fix $\Theta_d^{(t)}$ (where $t = 1, 2, \ldots, T$), solve for Θ_e by minimizing $L_{final}(\Theta_e)$

} repeat

coordinate-descent approach

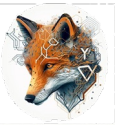
$$x_1^{(k)} = \arg\min_{x_1} f(x_1, x_2^{(k-1)}, x_3^{(k-1)}, \ldots x_n^{(k-1)})$$

$$x_2^{(k)} = \arg\min_{x_2} f(x_1^{(k)}, x_2, x_3^{(k-1)}, \ldots x_n^{(k-1)})$$

$$x_3^{(k)} = \arg\min_{x_3} f(x_1^{(k)}, x_2^{(k)}, x_3, \ldots x_n^{(k-1)})$$

$$\ldots$$

$$x_n^{(k)} = \arg\min_{x_n} f(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \ldots x_n)$$



Insights

Training Set $\{(x_i^{(t)}, y_i^{(t)})\}_{i=1}^{N}$ (where $t = 1, 2, \ldots, T$)

key insight : 在training時模擬某個property並且訓練之 or 模仿testing時會有的動作並且訓練之

Note that we hope Θ_e to have this property :

- Given a target scene t, if we can lear Θ_d by using Eq.(1) and fine-tunin $(x_1^{(t)}, y_1^{(t)})$
- Then Θ_e can make the model to perform well on the remaining part of images for target scene t.

Simple fine-tuning

- Training Stage : $\Theta_e^*, \Theta_d^* = \arg\min_{\Theta_e, \Theta_d} \sum_{t=1}^{T} \sum_{i=1}^{N} ||\text{vec}(F(x_i^{(t)}; \{\Theta_e, \Theta_d\})) - \text{vec}(y_i^{(t)})||_2$ (4)
- Testing Stage : $\Theta_d^* = \arg\min_{\Theta_d} L_d(\Theta_d)$, where $L_d(\Theta_d) = ||\text{vec}(F(x_1; \{\Theta_e, \Theta_d\})) - \text{vec}(y_1)||_2$ (1)

OSSS

- Training Stage :
  - $\Theta_d^{(t)} = \arg\min_{\Theta_d} L_d^{(t)}(\Theta_d)$, where $L_d^{(t)}(\Theta_d) = ||\text{vec}(F(x_1^{(t)}; \{\Theta_e, \Theta_d\})) - \text{vec}(y_1^{(t)})||_2$ (2)
  - $L_e^{(t)}(\Theta_e) = \sum_{i=2}^{N} ||\text{vec}(F(x_i^{(t)}; \{\Theta_e, \Theta_d^{(t)}(\Theta_e)\})) - \text{vec}(y_i^{(t)})||_2$ (3)
- Testing Stage : $\Theta_d^* = \arg\min_{\Theta_d} L_d(\Theta_d)$, where $L_d(\Theta_d) = ||\text{vec}(F(x_1; \{\Theta_e, \Theta_d\})) - \text{vec}(y_1)||_2$ (1)

Difference

The problem of this simple fine-tuning method is that there is a disconnect between learning and testing. During learning, the loss function in Eq. 4 does not explicitly enforce Θ_e to be transferable to a new scene. In contrast, our proposed approach learns Θ_e in a specific way, so that when we use Θ_e to get the fine-tuned decoder parameters $\Theta_d^{(t)}$ in a target scene, the final model will perform well.

Experiments

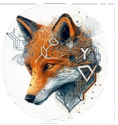
Method	WorldExpo'10	
	MAE	MSE
no fine-tuning	19.91	37.71
simple fine-tuning (2 layer)	10.62	13.98
ours (2 layer)	**8.23**	**12.08**

Table 2: Comparison of the performance (MAE and MSE) of our approach and the baselines on the WorldExpo'10 dataset using the standard split.

Experiments

Method	WorldExpo'10		Trancos	
	MAE	MSE	MAE	MSE
no fine-tuning	11.58	17.72	5.66	7.65
simple fine-tuning (1 layer)	7.82	10.87	5.51	6.11
ours (1 layer)	**7.7**	**10.62**	**5.45**	**6.04**
simple fine-tuning (2 layers)	8.39	11.41	4.98	5.55
ours (2 layers)	**7.53**	**10.42**	**4.46**	**5.18**

Table 1: Comparison of the performance (MAE and MSE) of our approach and the baselines on the WorldExpo'10 dataset and Trancos dataset. For "ours" and "simple fine-tuning", we consider either using the last layer or the last two layers of CSRNet as the decoder.



Experiments

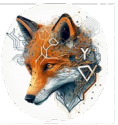
Method	(a) W→U		(b) W→M		(c) W→T		(d) T→W	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
no fine-tuning	23.84	24.9	18.76	19.29	31.19	35.67	50.14	61.77
simple fine-tuning (L-1)	4.45	5.29	5.17	6.34	18.09	19	41.33	46.36
ours (L-1)	**2.32**	**2.39**	**1.91**	**2.48**	**7.12**	**8.02**	**19.23**	**23.23**
simple fine-tuning (L-2)	4.42	5.26	4.29	5.12	7.14	8.31	21.03	26.12
ours (L-2)	**2.31**	**3.12**	**1.85**	**2.40**	**5.72**	**6.61**	**8.91**	**12.19**

Conclusion

. Propose a new problem called the one-shot scene-specific crowd counting.

. Simple fine-tuning has some issues in one-shot learning

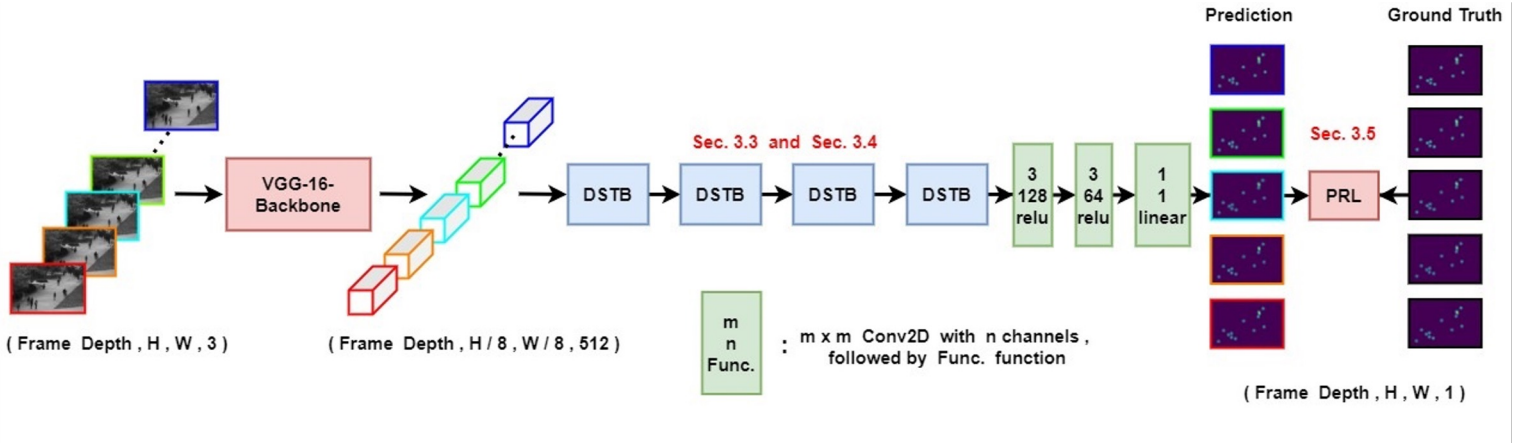
. Propose a novel algorithm to make the model learn how to fine-tune.



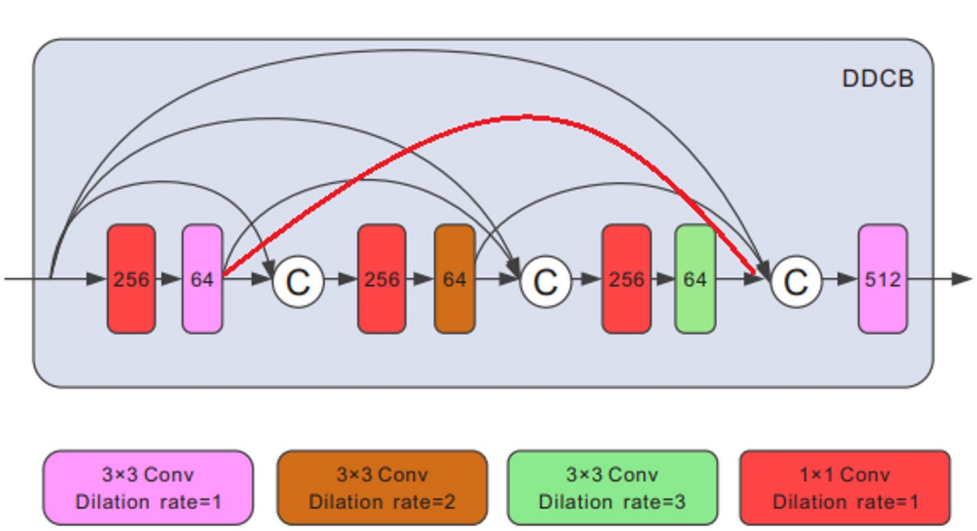
Spatio-Temporal Dilated Convolution with Uncertain Matching for Crowd Estimation (TMM 2021)

Spatio-Temporal Dilated Convolution with Uncertain Matching for Crowd Estimation

- Challenge
  - Temporal information are usually neglected in estimating crowd density since the learnable parameters become extremely large.
  - Labeling for crowds is difficult with errors (uncertainty).
- Key ideas
  - Propose a spatio-temporal dilated Convolution for better capture the long term and short tem information.
  - Propose a new loss function to deal with the uncertainty.



Spatial --- Scale Variation



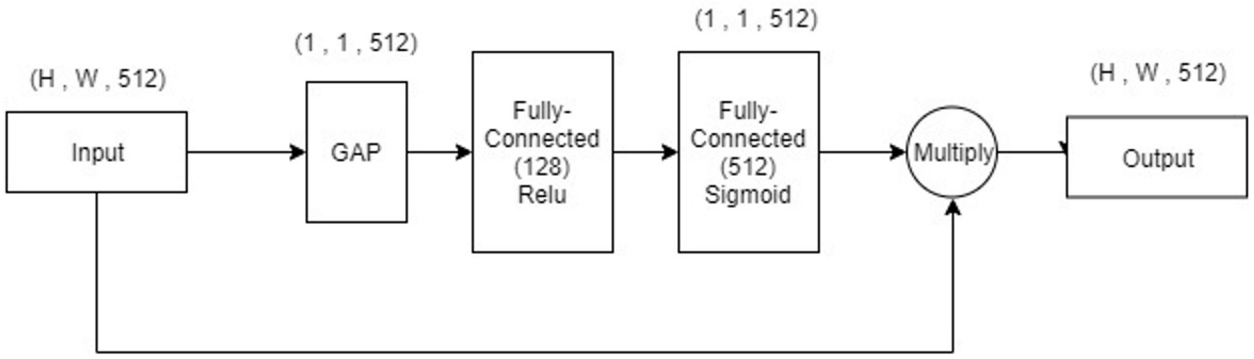
DSNet --- DDCB

Reception field
1
3
5
7
9
11
13

Spatial --- Channel Enhancement

Because it has rich scale-information on the channels ,

We devise the channel-attention block to enhance the importance of channels.



Temporal --- Long-term + Short-term information

Similar to DDCB ,
extend the dilated operation to temporal dimension

- Dilated Conv3D
- Refer to Pseudo 3D Network
- conv3D kernel size = (3 , 1 , 1)
  (time , H , W)

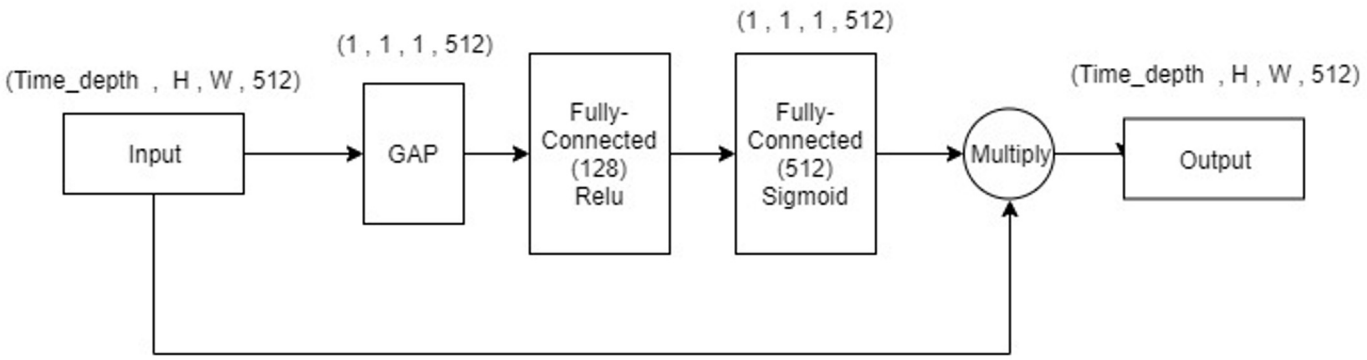
to reduce the computation cost and number of parameters

Temporal Reception field
1 (short-term)
3
5
7
9
11
13 (long-term)

Temporal --- Channel Enhancement

Because it has rich period-information on the channels ,

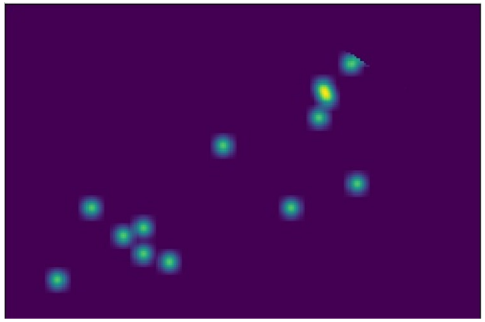
We devise the temporal channel-attention block to enhance the importance of channels.



Crowd Estimation

Input :

Output :



Model : CNN or LSTM .....

Task : Pixel-wise Regression Problem

Our Model

If video data is available, we can use Conv3D layer or LSTM cell....

But it leads to the difficulty of training process:

(1) training time

(1) # of parameters

(1) Inference speed

Method	Temporal	Training time (seconds/epoch)	# of parameters (in millions)	Inference speed (in fps)
CSRNet [20]		35	16.26	48
ConvLSTM [43]	√	530	40.61	13
STDNet (ours)	√	50	18.14	34

Our Model

Spatio-Temporal Modeling

(1) Direct use of the Conv3D layer leads to training difficulty

⇒ Using the 3D CNN decomposition method

(2) Continuous scale variations problem

⇒ Using dilated Convolution + Concatenate

(3) Multi-scale features are concatenated on channel dimension

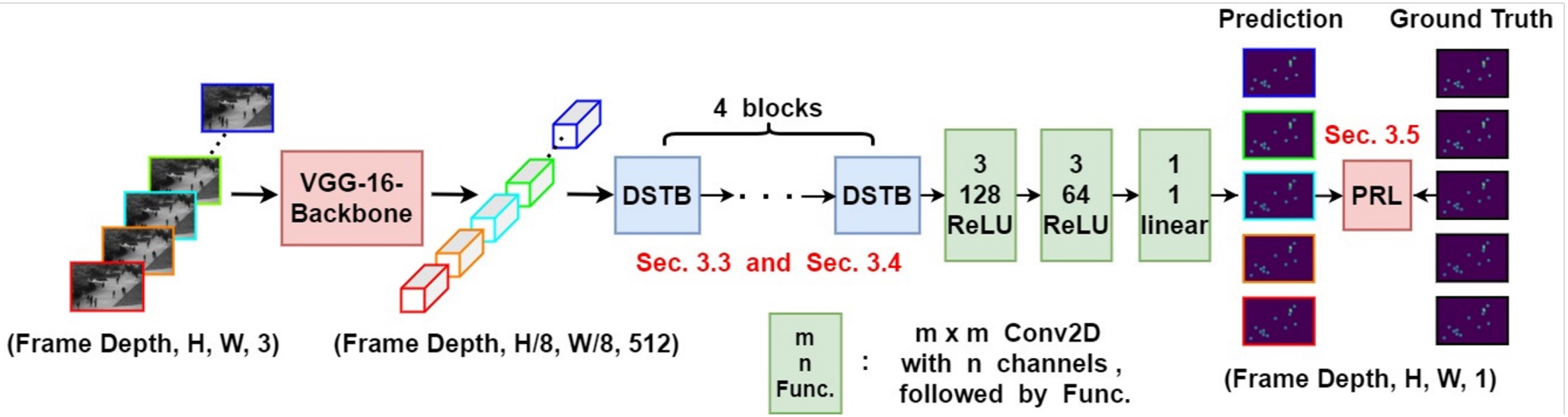
⇒ Using Channel Attention

New Loss function

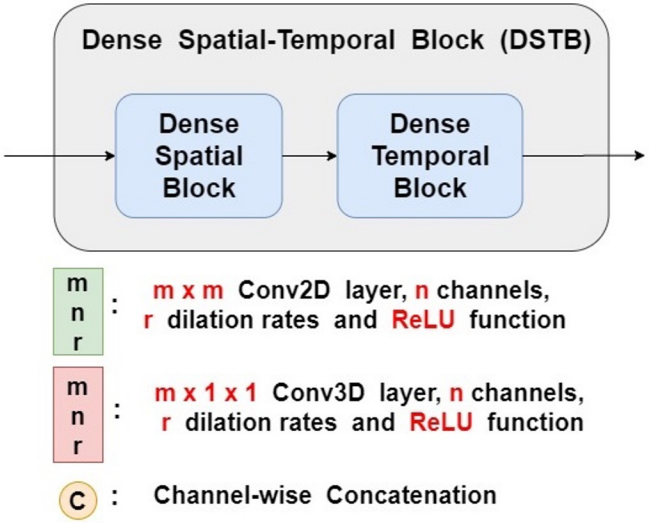
(4) Pixel-wise regression loss has some issues

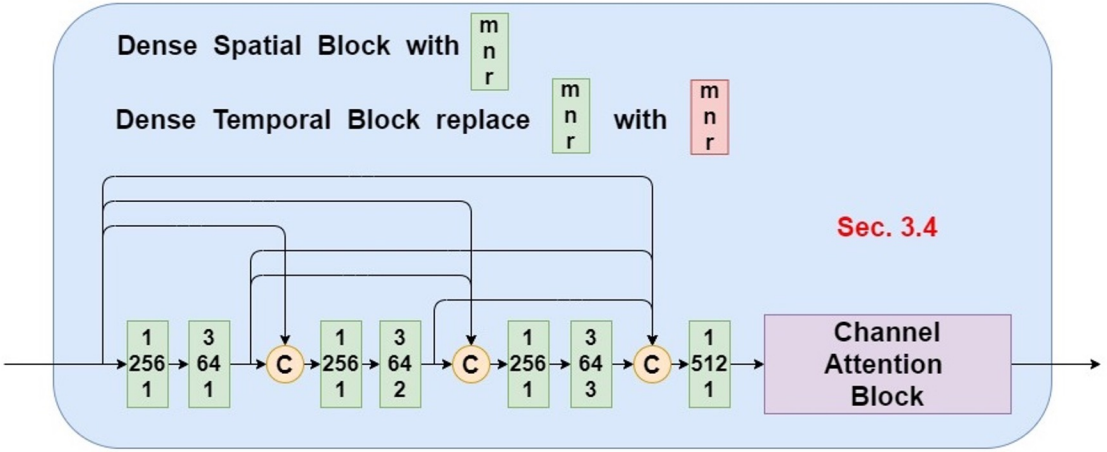
⇒ Use Patch-wise Regression Loss (PRL)

Model

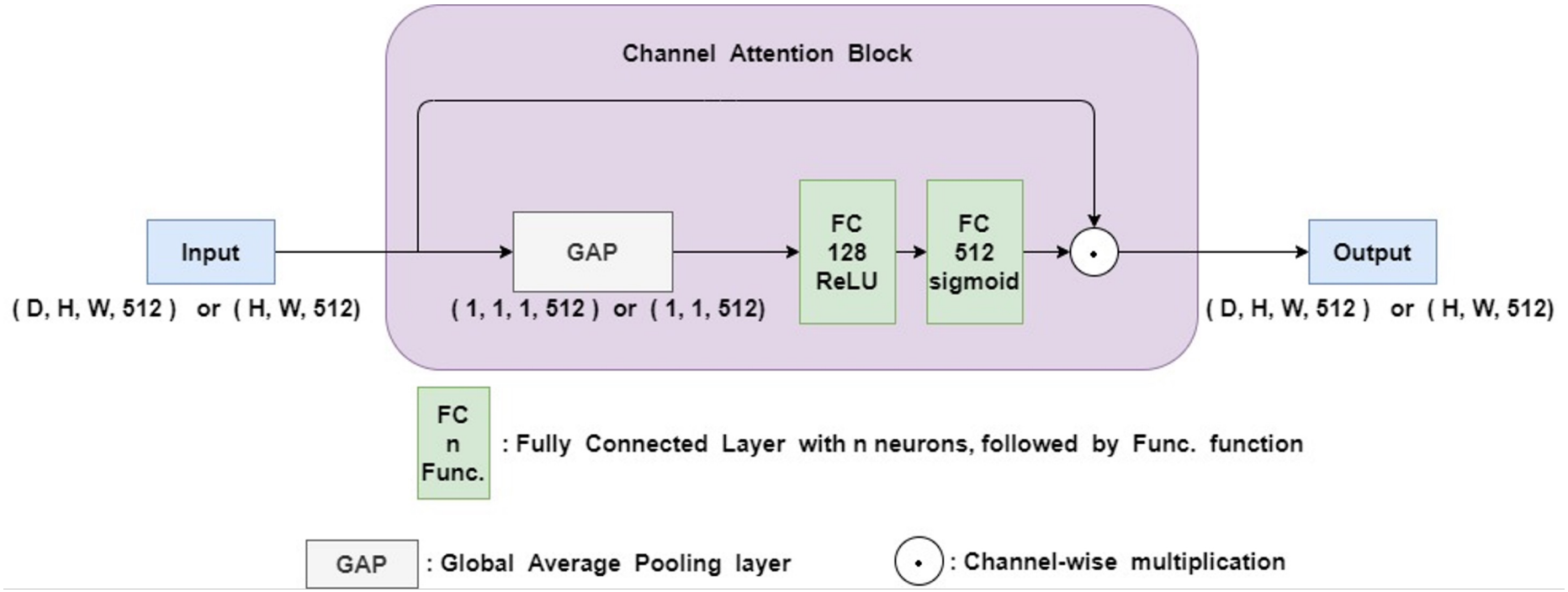

Prediction
Ground Truth
4 blocks
VGG-16-Backbone
DSTB
DSTB
3 128 ReLU
3 64 ReLU
1 1 linear
Sec. 3.5
PRL
Sec. 3.3 and Sec. 3.4
(Frame Depth, H, W, 3)
(Frame Depth, H/8, W/8, 512)
m n Func.
: m x m Conv2D with n channels, followed by Func.
(Frame Depth, H, W, 1)

DSTB

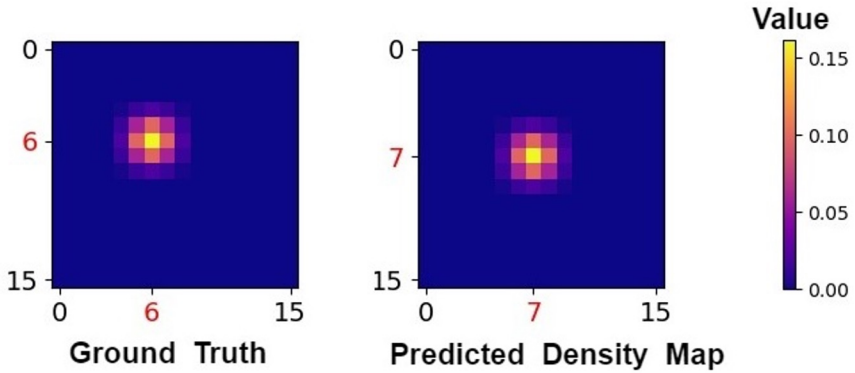

Dense Spatial-Temporal Block (DSTB)
Dense Spatial Block
Dense Temporal Block
m n r : m x m Conv2D layer, n channels, r dilation rates and ReLU function
m n r : m x 1 x 1 Conv3D layer, n channels, r dilation rates and ReLU function
C : Channel-wise Concatenation


Dense Spatial Block with m n r
Dense Temporal Block replace m n r with m n r
Sec. 3.4
1 256 1
3 64 1
C
1 256 1
3 64 2
C
1 256 1
3 64 3
C
1 512 1
Channel Attention Block

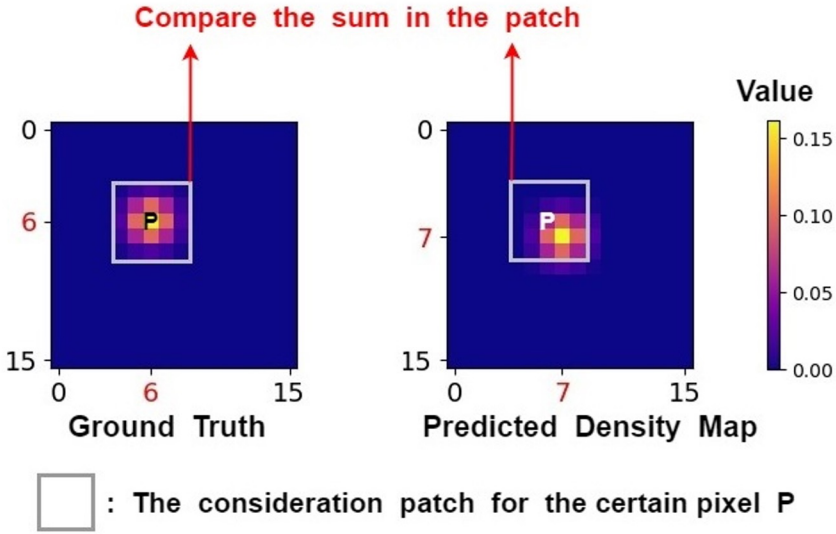
Channel Attention Block



Patch-wise Regression Loss



(a) 2D example with Pixel-level difference



(b) illustration for the Patch-wise Regression Loss

Comparison for Loss

original pixel-wise regression loss :

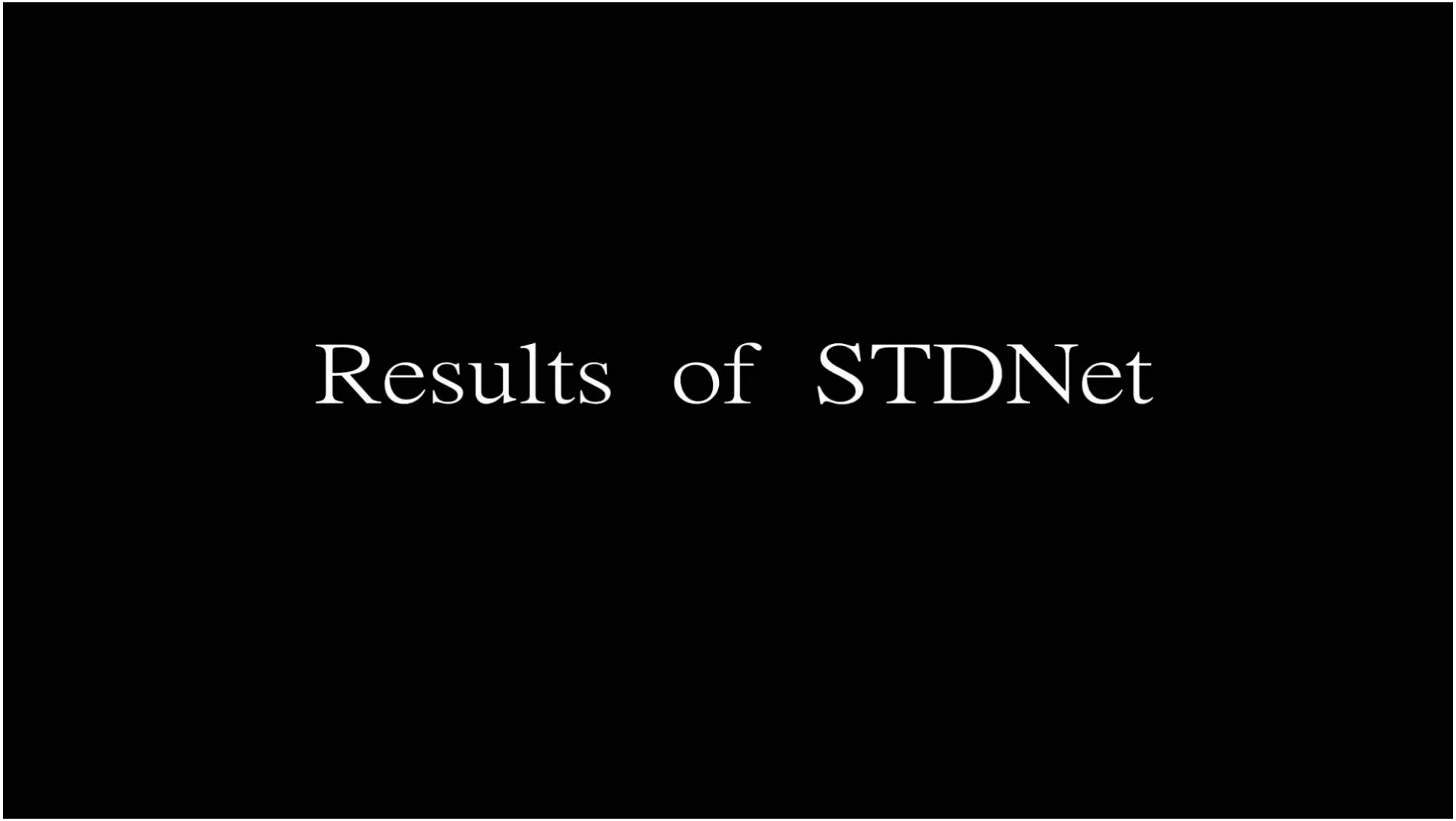
$$\mathcal{L}(\theta) = \frac{1}{2N_b} \sum_{i=1}^{N_b} \left\| f(X_i;\theta) - Y_i^{GT} \right\|_2^2$$

Patch-wise Regression Loss (PRL) :

$$\mathcal{L}_p(\theta) = \sum_{k=1}^{n_p} \lambda_k \cdot \mathcal{L}_p^{(k)}(\theta)$$

$$\mathcal{L}_p^{(k)}(\theta) = \frac{1}{N_b} \sum_{i=1}^{N_b} \left\| G_\sigma^{(k)} * f(X_i;\theta) - G_\sigma^{(k)} * Y_i^{GT} \right\|_1$$

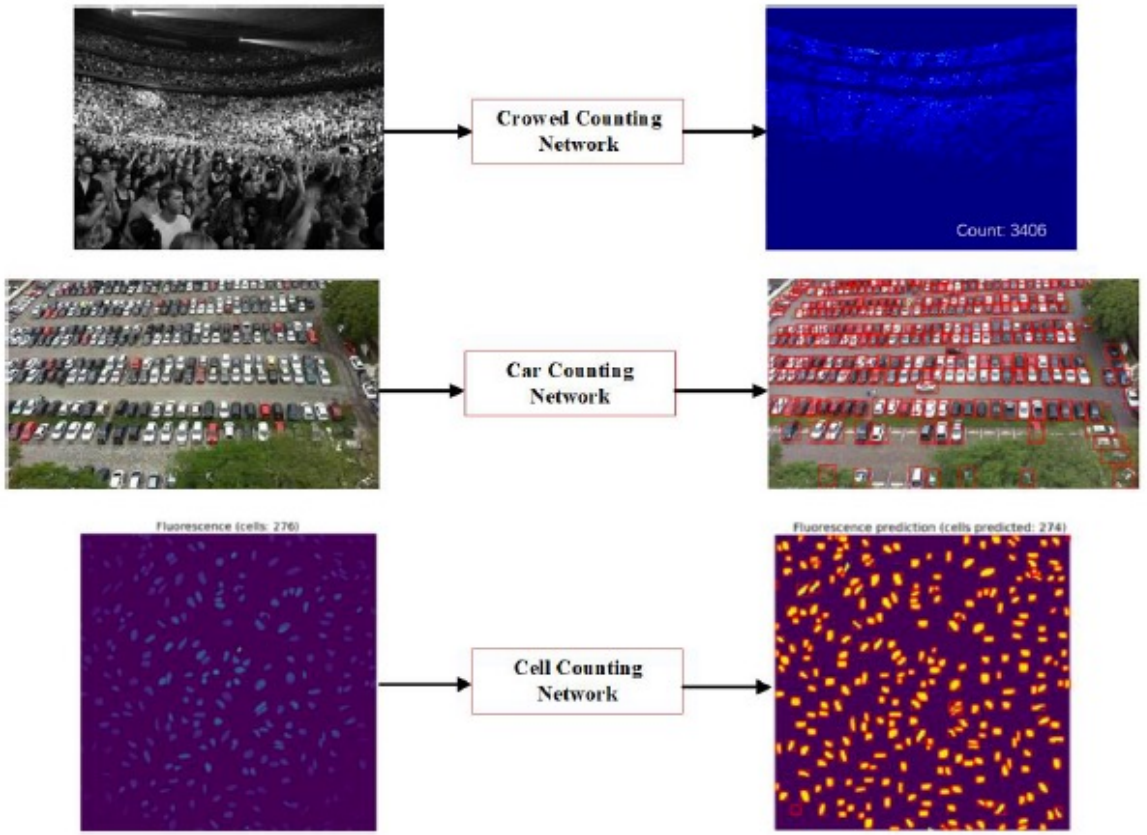
Demo video



Learning To Count Everything (CVPR 2021)

1. Introduction

Existing Works for Visual Counting

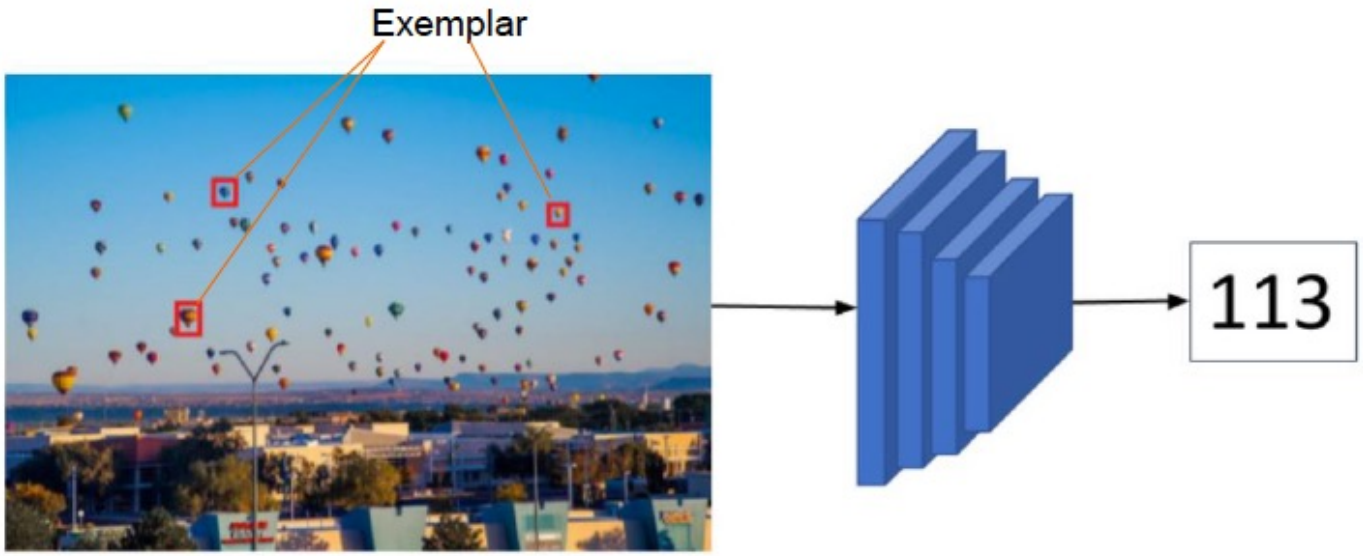


Existing Works for Visual Counting can NOT work for objects from another category

Can’t afford to train a network every category

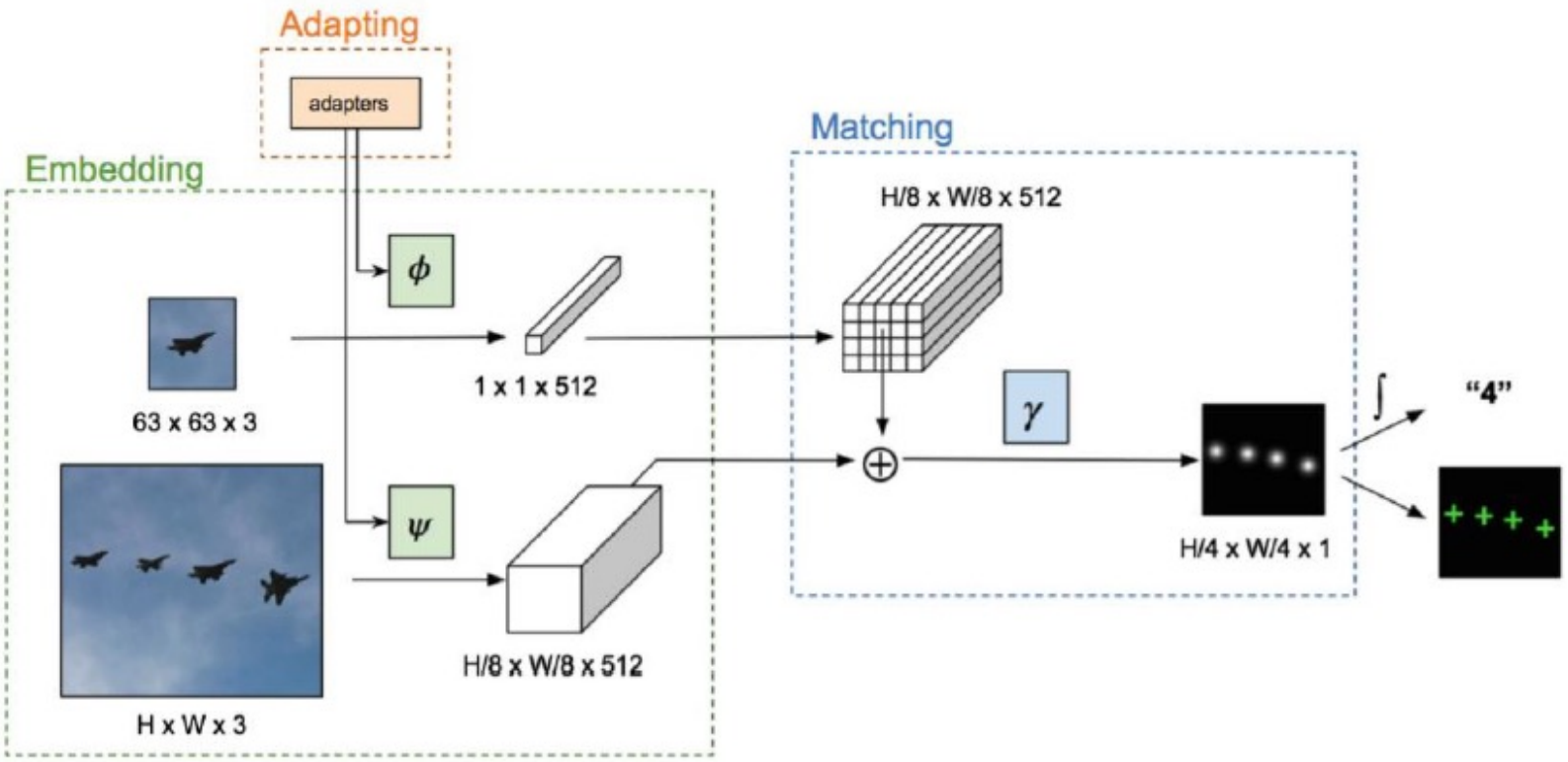
Name	#image	#People
NWPU-Crowd	5,109	2,133,238
JHU-CROWD	4,250	1,114,785
UCF-QNRF	1,535	1,251,642
ShanghaiTech Part A	482	241,677
UCF_CC_50	50	63,974

Goal



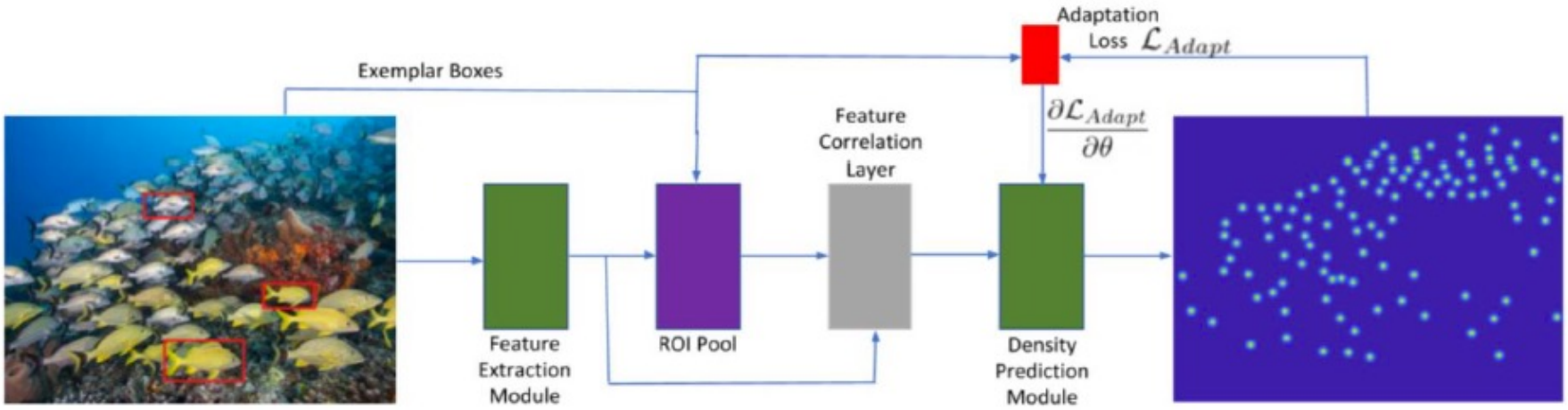
Learn a network that can count objects of interest from any category

Generic Matching Network (GMN)

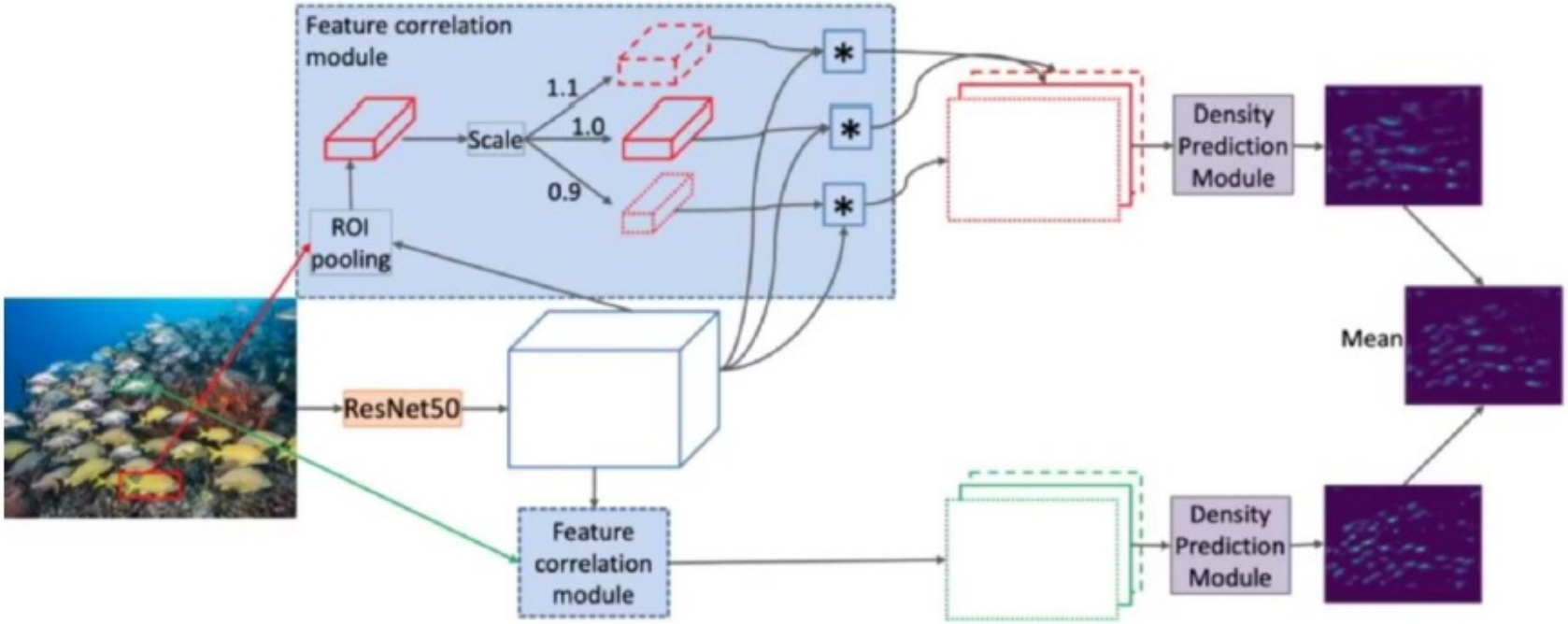


Erika Lu, Weidi Xie(B), and Andrew Zisserman. Class-Agnostic Counting. ACCV 2018

FamNet Processing Pipeline



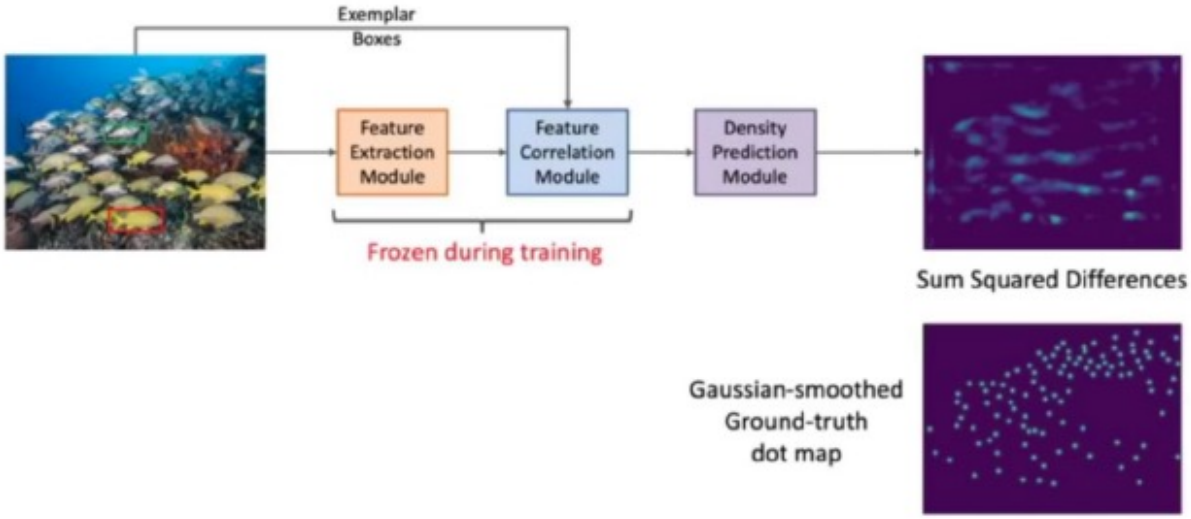
FamNet Processing Pipeline



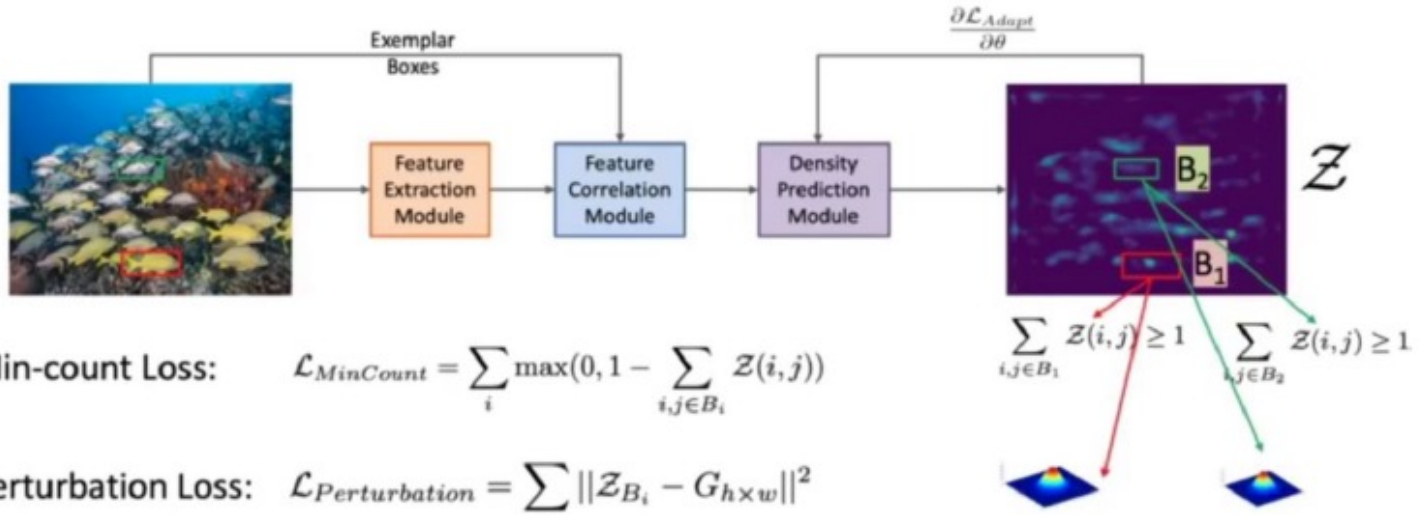
Density Prediction Module

Conv 196×7× 7
RELU
Upsample ×2
Conv 128×5×5
RELU
Upsample ×2
RELU
Conv 64×3×3
RELU
Upsample ×2
Conv 32×1×1
RELU
Conv 1×1×1
RELU

Training FamNet



Inference and Test-time Adaption



- Min-count Loss: $\mathcal{L}_{MinCount} = \sum_i \max(0, 1 - \sum_{i,j \in B_i} \mathcal{Z}(i,j))$
- Perturbation Loss: $\mathcal{L}_{Perturbation} = \sum_i ||\mathcal{Z}_{B_i} - G_{h \times w}||^2$
- Adaptation Loss: $\mathcal{L}_{Adapt} = \alpha \mathcal{L}_{MinCount} + \beta \mathcal{L}_{Perturbation}$

Few-shot counting Dataset (FSC-147)

- Image collection
  - Use Google, Flickr and Bing with *keyword: "many, multiple, lots of, stack of " + category names*
  - Filtering & verification
    - *High image quality*
    - *Large enough object count*
    - *Appearance similarity*
    - *No severe occlusion*
- Image Annotation
  - OpenCV Image & Video Annotation Tool

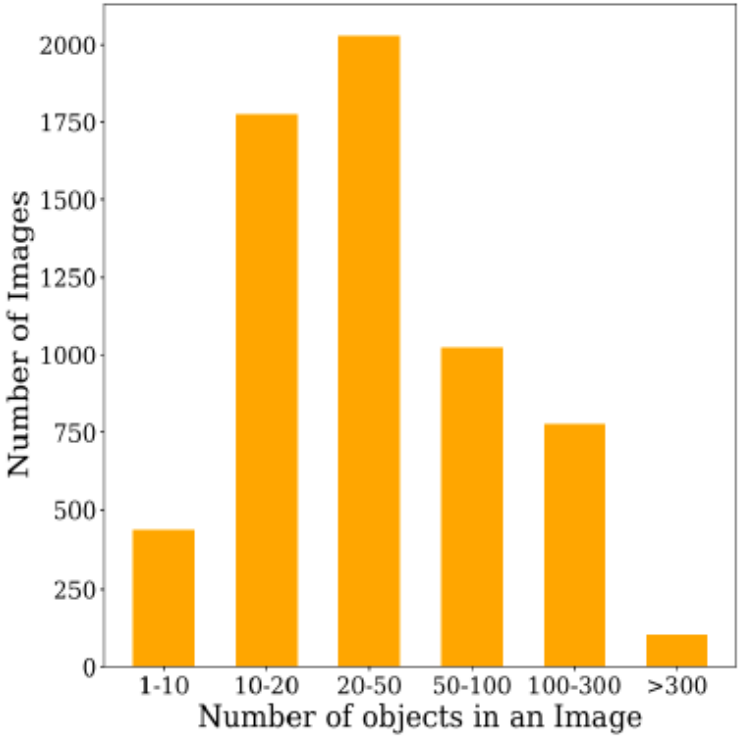
Categories and Super-categories

	Images	Categories
Total	6135	147
Train	3659	89
Validation	1286	29
Test	1190	29

Image categories and number of images for each category in our dataset.

Number of Objects in Images

- No of objects in all images: 343,818/6135
- No of objects in an image:
  - Min: 7
  - Max: 3701
  - Average: 56



Comparison to Other Visual Counting Datasets

Dataset	Images	Categories	Annotation type: Dot	Annotation type: Bounding Box
UCF CC 50 [15]	50	1	✓	✗
Shanghaitech [55]	1198	1	✓	✗
UCF QNRF [16]	1535	1	✓	✗
NWPU [49]	5109	1	✓	✗
JHU Crowd [43]	4372	1	✓	✓
CARPK [14]	1448	1	✓	✓
Proposed	6135	147	✓	✓

Few-shot counting Dataset (FSC-147)

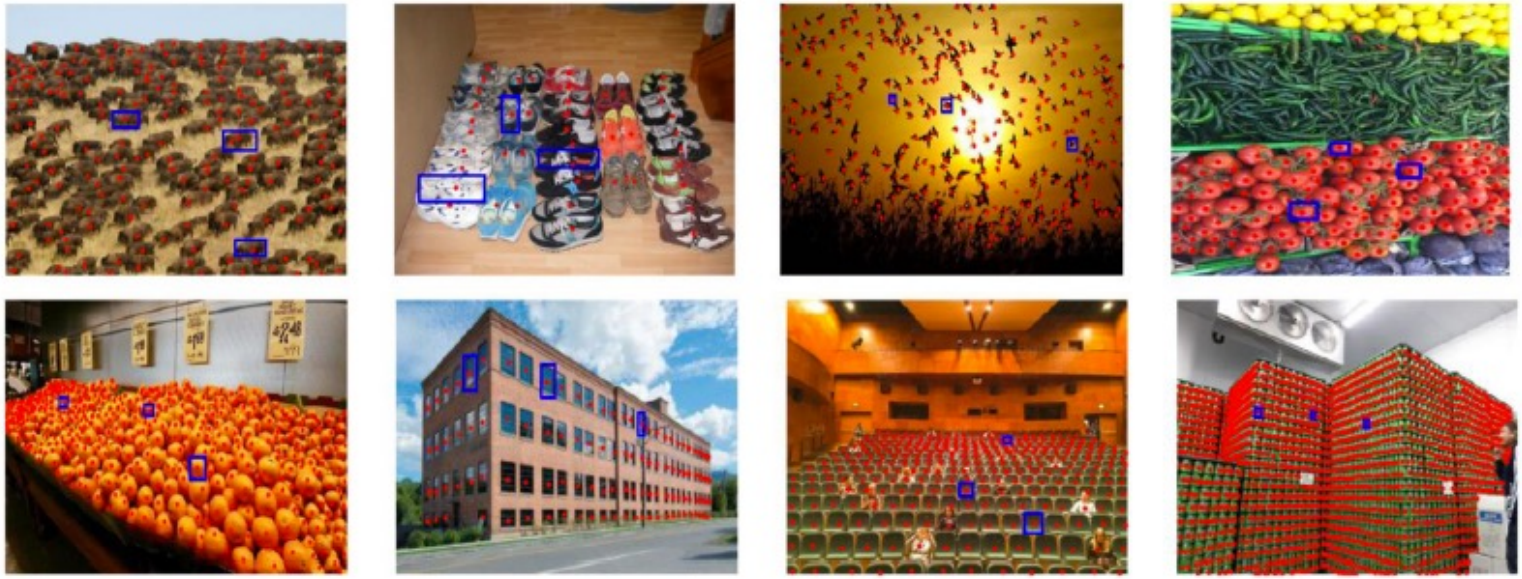


Figure 4: **Few annotated images from the dataset**. Dot and box annotations are shown in red and blue respectively. The number of objects in each image varies widely, some images contain a dozen of objects while some contains thousands.

Comparison with Few-Shot Approaches

	Val Set		Test Set	
Method	MAE	RMSE	MAE	RMSE
Mean	53.38	124.53	47.55	147.67
Median	48.68	129.70	47.73	152.46
FR few-shot detector [17]	45.45	112.53	41.64	141.04
FSOD few-shot detector [8]	36.36	115.00	32.53	140.65
Pre-trained GMN [28]	60.56	137.78	62.69	159.67
GMN [28]	29.66	89.81	26.52	124.57
MAML [9]	25.54	79.44	24.90	112.68
FamNet (Proposed)	**23.75**	**69.07**	**22.08**	**99.54**

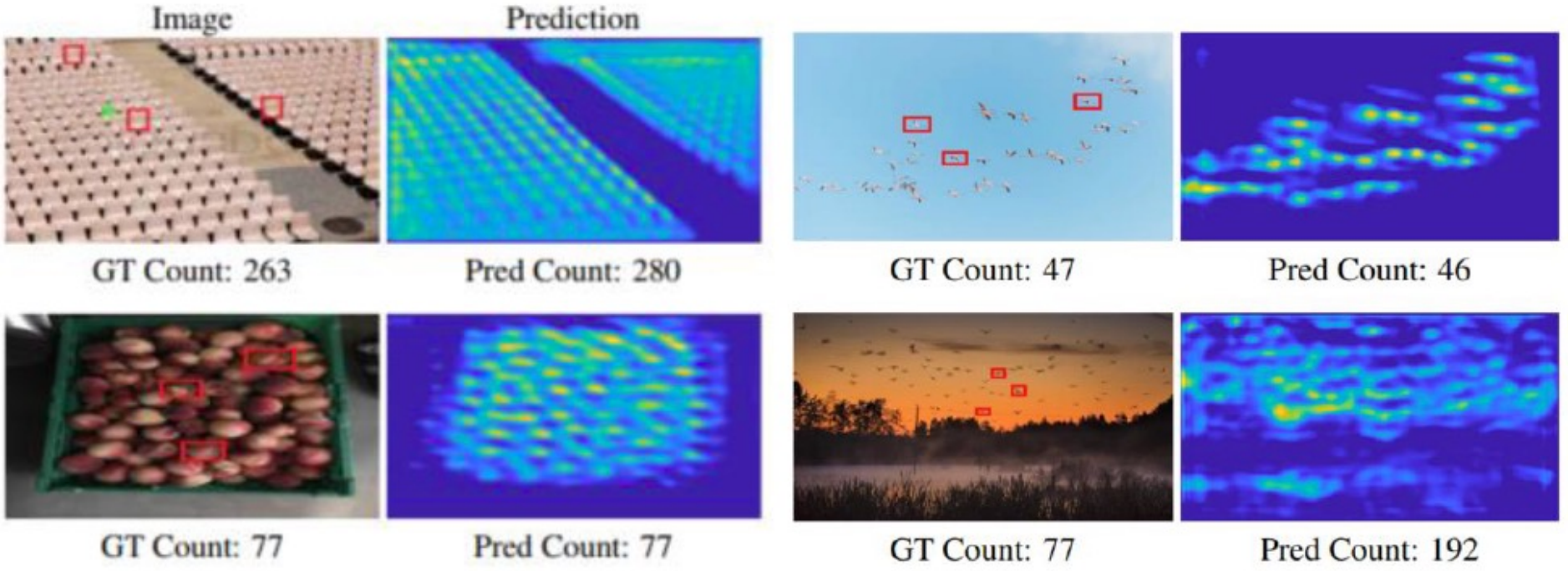
Table 1: Comparing FamNet to two simple baselines (Mean, Median) and four stronger baseline (Feature Reweighting (FR) few-shot detector, FSOD few-shot detector, GMN and MAML), these are few-shot methods that have been adapted and trained for counting. FamNet has the lowest MAE and RMSE on both val and test sets.

Comparison with Object Detectors

	Val-COCO Set		Test-COCO Set	
Method	MAE	RMSE	MAE	RMSE
Faster R-CNN	52.79	172.46	36.20	79.59
RetinaNet	63.57	174.36	52.67	85.86
Mask R-CNN	52.51	172.21	35.56	80.00
FamNet (Proposed)	**39.82**	**108.13**	**22.76**	**45.92**

Table 2: **Comparing FamNet with pre-trained object detectors**, on counting objects from categories where there are pre-trained object detectors.

Qualitative Results



Predicted density maps and counts of FamNet

Qualitative Results

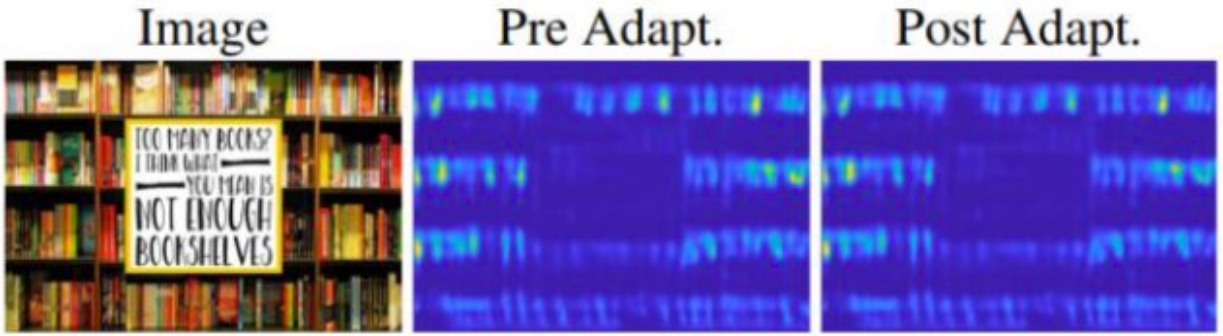
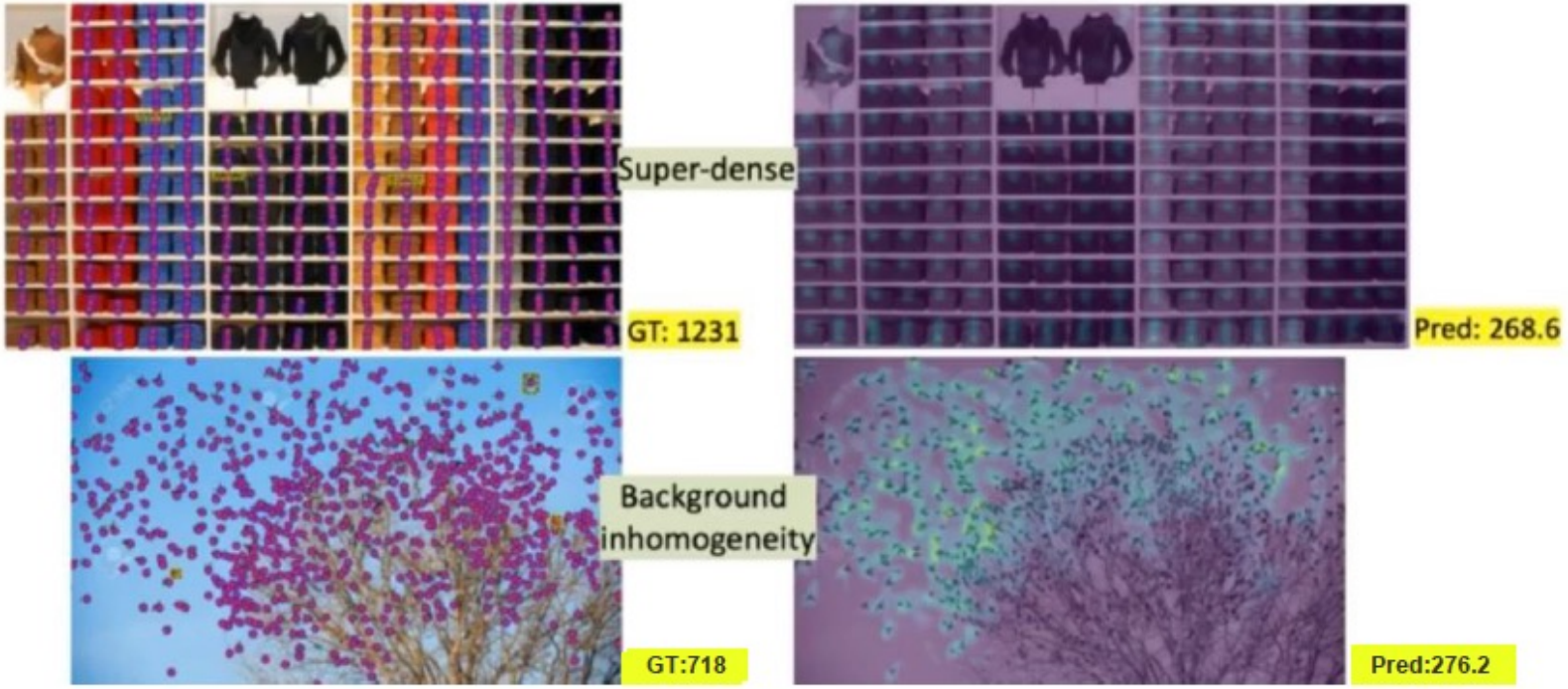
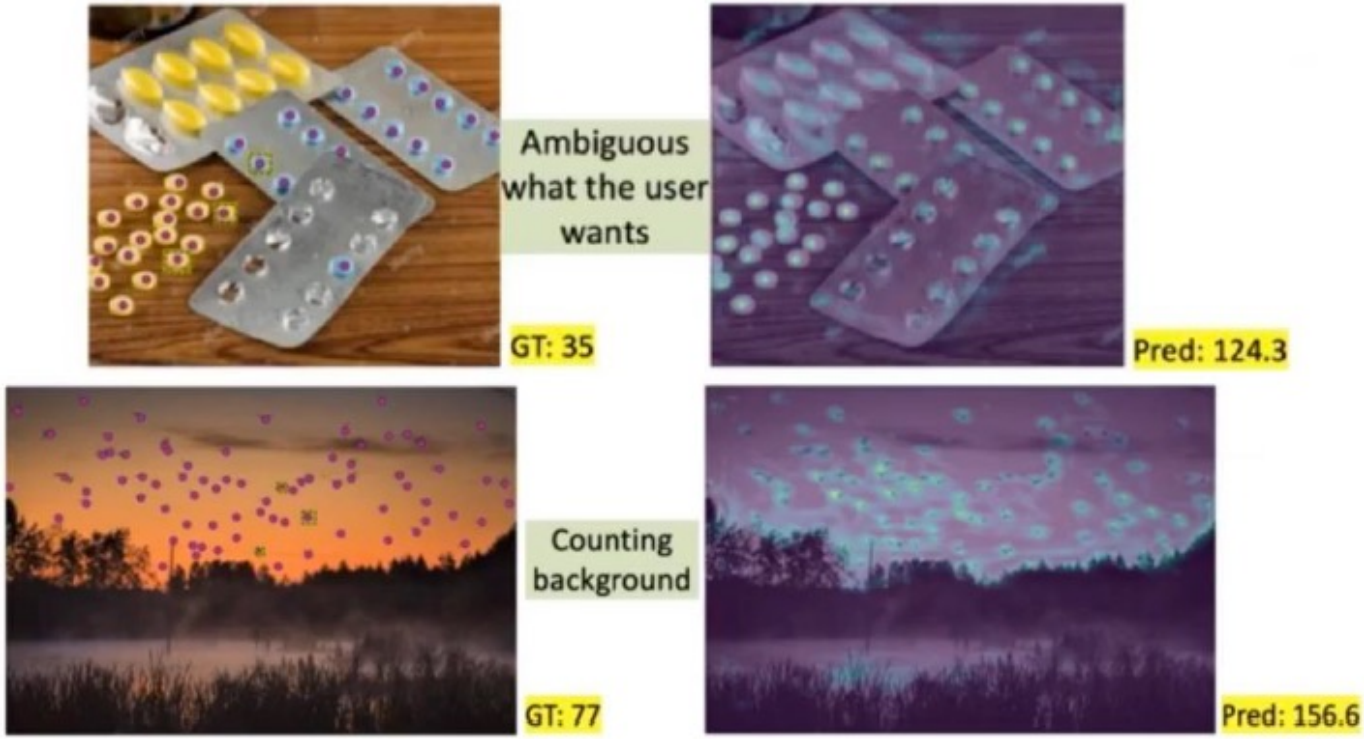


Figure 6: **Test time adaptation**. Shown are the initial density map (Pre Adapt) and final density map after adaptation (Post Adapt). In case of over counting, adaptation decreases the density values at dense locations.

Failure Cases – Under-counting

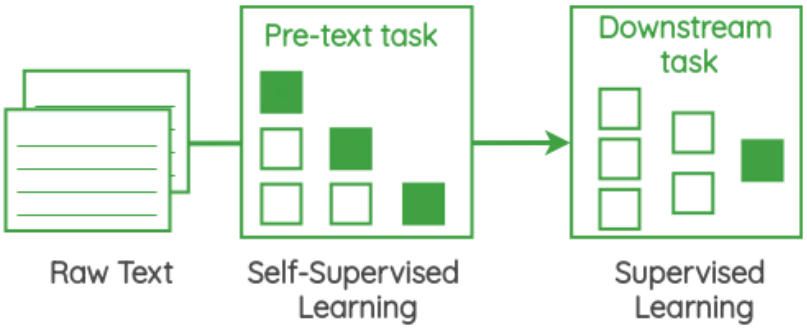


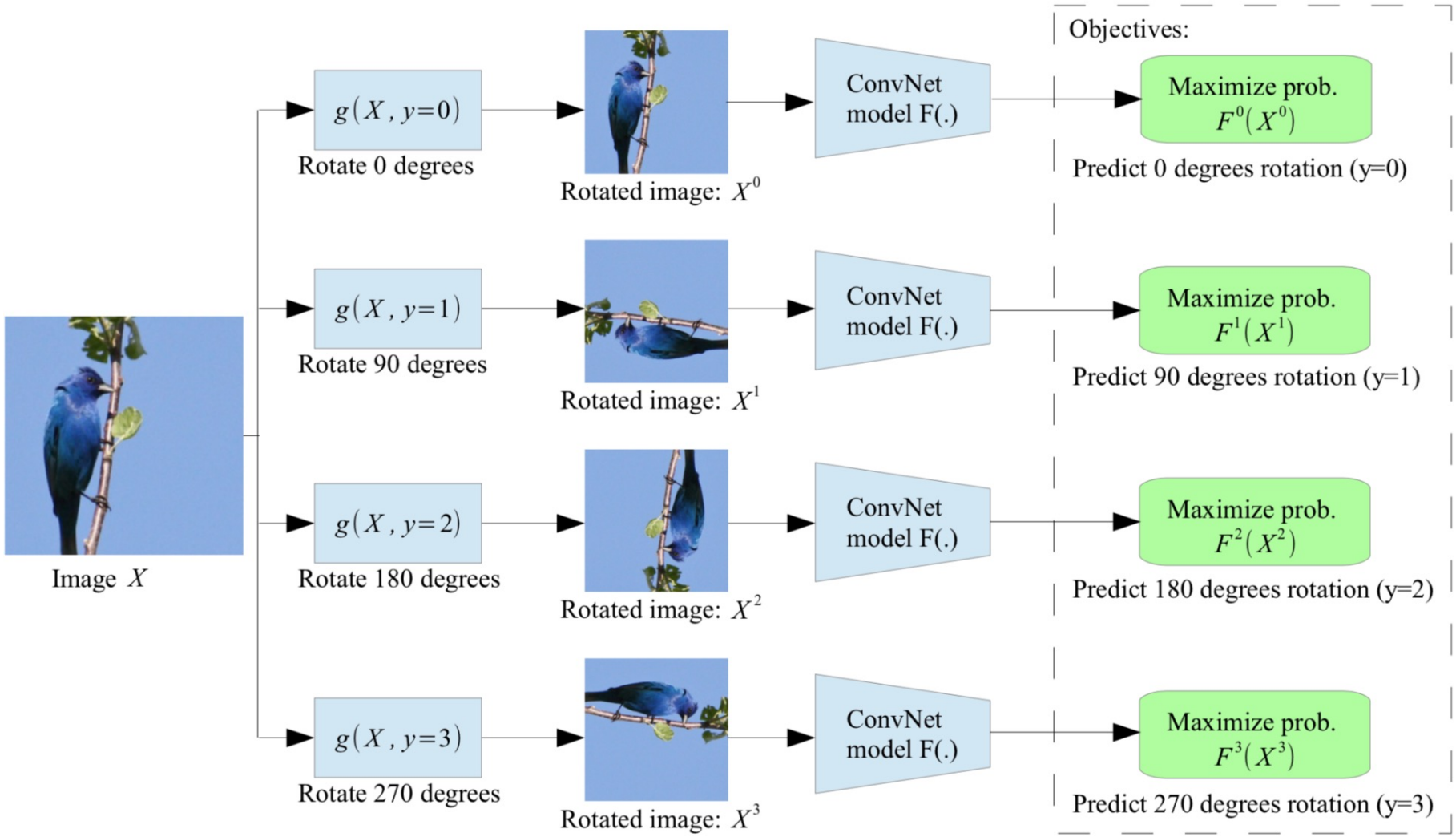
Failure Cases – Under-counting

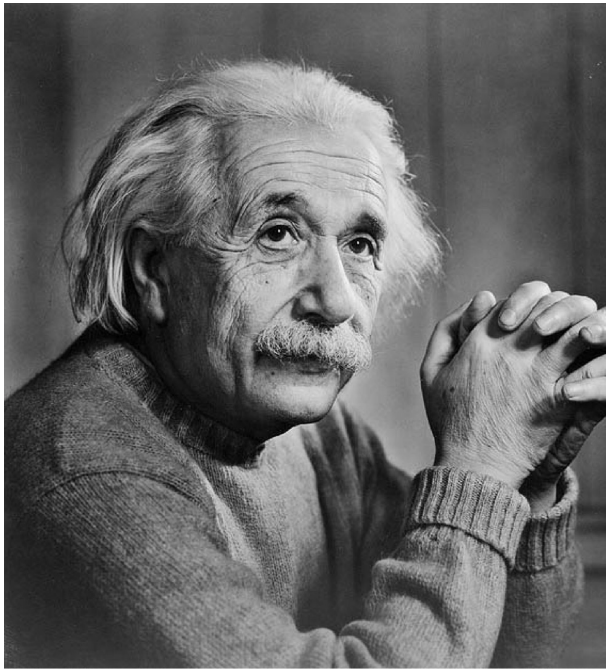


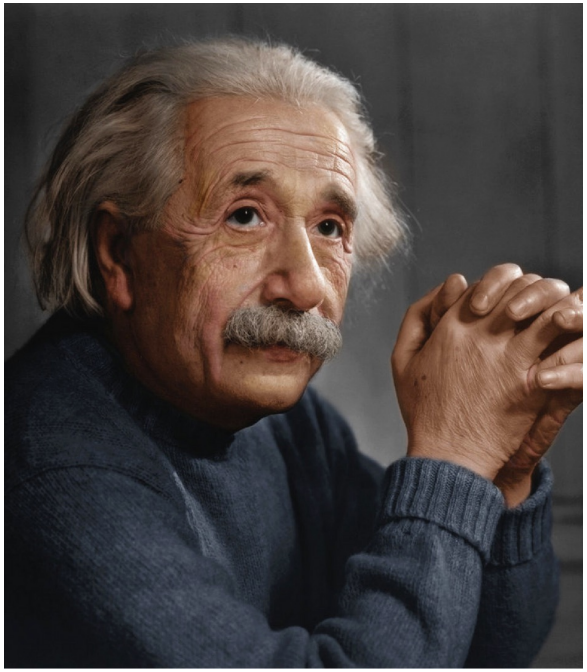
Crowd Counting by Self-supervised Transfer Colorization Learning and Global Prior Classification (arXiv:2105.09684, 2021)

- Key idea:
  - Semantics and local texture pattern as obtained in the coloration process of an image reflect the density of people in the region.
- Self-supervised Learning




Objectives:
Image X
$g(X, y=0)$
Rotate 0 degrees
Rotated image: X^0
ConvNet model F(.)
Maximize prob. $F^0(X^0)$
Predict 0 degrees rotation (y=0)
$g(X, y=1)$
Rotate 90 degrees
Rotated image: X^1
ConvNet model F(.)
Maximize prob. $F^1(X^1)$
Predict 90 degrees rotation (y=1)
$g(X, y=2)$
Rotate 180 degrees
Rotated image: X^2
ConvNet model F(.)
Maximize prob. $F^2(X^2)$
Predict 180 degrees rotation (y=2)
$g(X, y=3)$
Rotate 270 degrees
Rotated image: X^3
ConvNet model F(.)
Maximize prob. $F^3(X^3)$
Predict 270 degrees rotation (y=3)






I_3
I_4
I_3
I_4

- Classification-based priors
- Clustering-based priors
- Ranking-based priors

Google crowded scene

illustration | movie | pub | market | hong kong | puzzle | madrid spain | taipei | city | crowded market

Crowded Scene At Market With Crowd Of ...
dreamstime.com

The crowded scene - Exibart Street
exibartstreet.com

Eminonu ferry terminal....
alamy.com

Crowded Scene At Bailen Street, Madrid ...
dreamstime.com

Examples of crowded scenes: (a) a stage ...
researchgate.net

Gavin Hellier
naturepl.com

7 Crowded scene with two equal sized ...
researchgate.net

T-shirt in a crowded scene
thesun.co.uk

Crowded People Street Scene Ed...
dreamstime.com

crowded scene ...
robertharding.com

Google

street view scene with people

lugano | google maps | istanbul | crowd | helsinki | google earth | city | ticino switzerland | finland

People commuting in new york city on ...
pinterest.com

commuting background Stock Footage ...
pinterest.es

Stock Footage,#city#crowded#york#People ...
pinterest.com

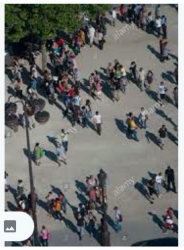
New york city street scene at sunset ...
pinterest.com

New York City streets scene. Crowd of ...
pinterest.com

Seoul Market Street Sc...
dreamstime.com

Penang, Malaysia - 2019, March 4th ...
123rf.com

Lugano, Ticino, Switzerland - April 19 ...
123rf.com



Paris, Champs Elysees ...
alamy.com

Lugano Center, Urban Scene, Switzer...
dreamstime.com

Neuron Linear Transformation: Modeling the Domain Shift for Crowd Counting (TNNLS 2021)

Outline

- Introduction
- Related Work
- Method
- Experiment
- Conclusion

Introduction

Domain-Shift issue in Crowd Counting :

We usually expect that our model has the good generalization ability,

however, given a specific training set,

when we test on the other Crowd Counting benchmark, the performance is poor

Introduction

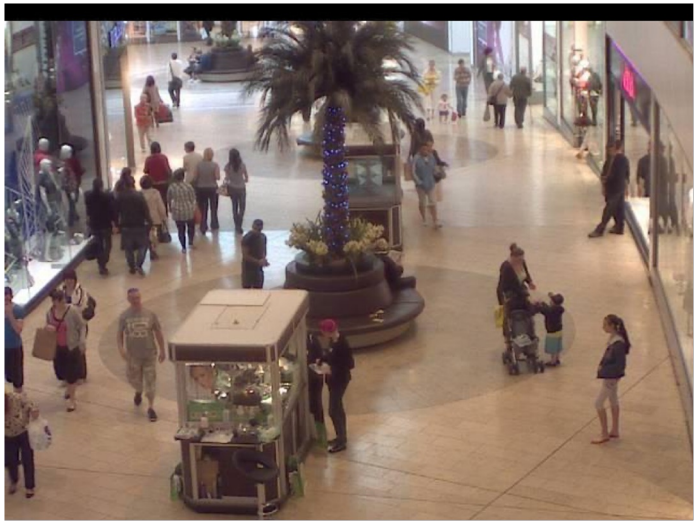
Type of Domain-Shift :

(1) Real scene vs. Synthetic data

Introduction

Type of Domain-Shift :

(2) Video 1 (Scene 1) vs. Video 2 (Scene 2)



Introduction --- A kind of few-shot learning

To deal with the domain-shift issue :

Trivial solution : Directly fine-tune

→ Failed !

→ Suffer from overfitting and perform poorly (due to few-shot data)

→ This paper proposes a few-shot learning method by using Neuron-Linear-Transformation to tackle the domain-shift issue.

Related Work

Few-Shot Learning / Meta Learning in Crowd Counting

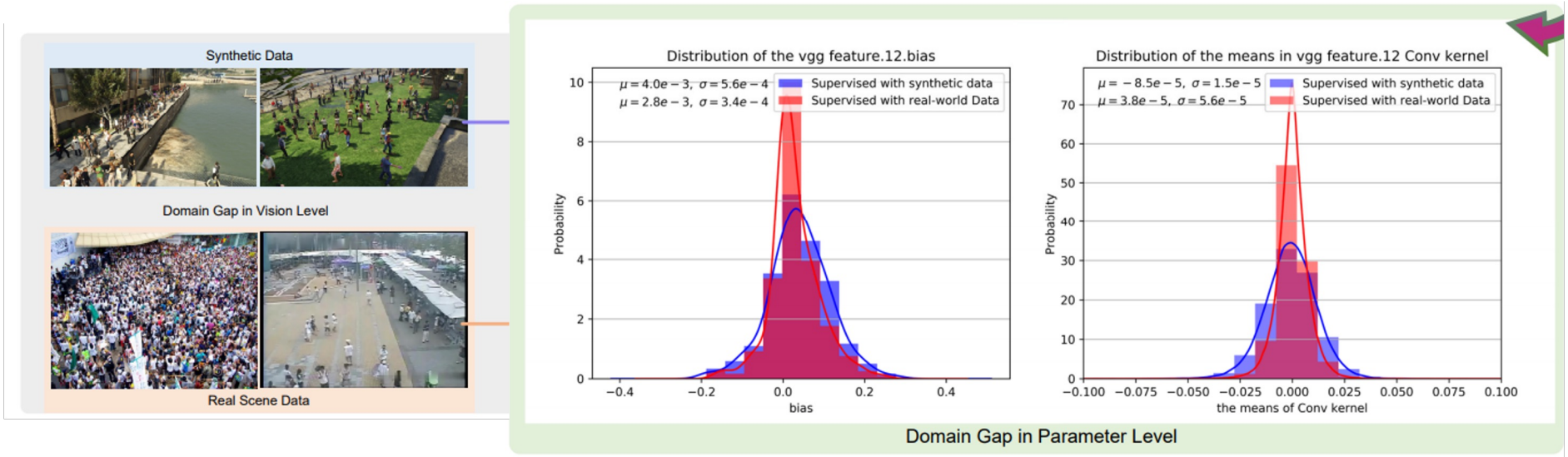
(1) One-Shot Scene-Specific Crowd Counting

: Propose a training strategy to make the model learn " how to fine-tune"

(2) Few-Shot Scene Adaptive Crowd Counting Using Meta-Learning

: Integrate MAML into Crowd Counting task

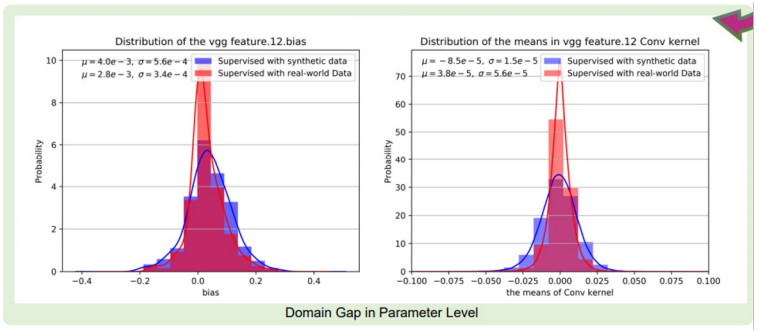
Method --- Observation of domain-shift



The domain-shift problem can be solved by adjusting the distribution of parameters from source-domain to target-domain

Method --- Neuron Linear Transformation (NLT)

Given a source domain Neuron parameters $A^{\mathcal{S}} \in \mathbb{R}^{c \times h \times w}$.



Domain Gap in Parameter Level

Our goal : $F \in \mathbb{R}^{C \times 1 \times 1}$

Find a domain-fa $B \in \mathbb{R}^{C \times 1 \times 1}$.
and

domain-bias

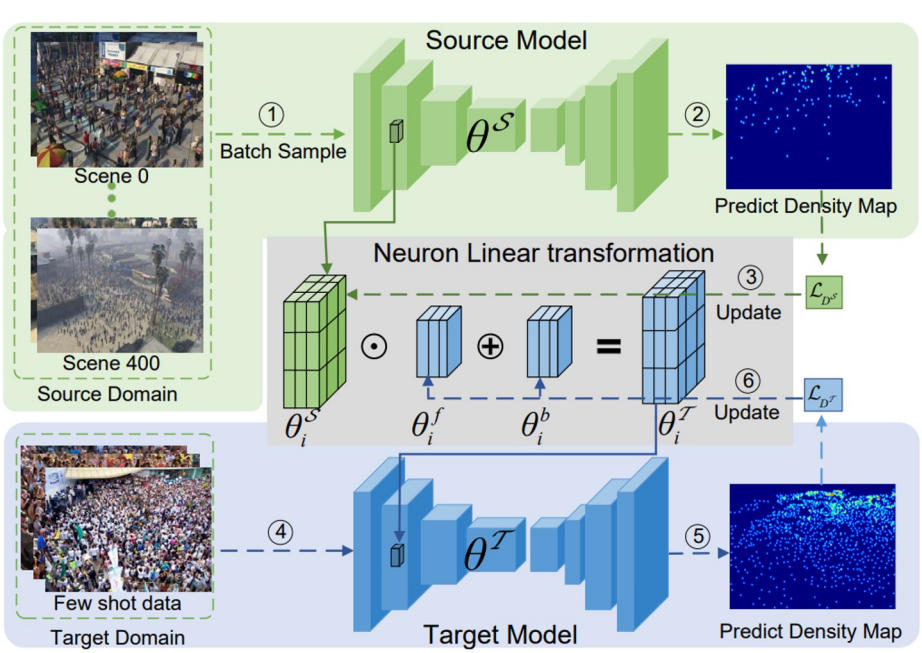
adjusted paramel $A^{\mathcal{T}} = A^{\mathcal{S}} \odot F \oplus B$

$$A^{\mathcal{T}} = \left[f^1 \times \begin{bmatrix} a^1_{11} & \cdots & a^1_{1w} \\ \vdots & \ddots & \vdots \\ a^1_{1h} & \cdots & a^1_{hw} \end{bmatrix} + [b^1], \cdots, \right.$$
$$\left. f^c \times \begin{bmatrix} a^c_{11} & \cdots & a^c_{1w} \\ \vdots & \ddots & \vdots \\ a^c_{1h} & \cdots & a^c_{hw} \end{bmatrix} + b^c \right],$$

to adjust the distribution of parameters such that they

can perform well on the target domain

Method



Source-domain Training (①、②、③) :

Data :

$$D^{\mathcal{S}} = \{I_i^{\mathcal{S}}, Y_i^{\mathcal{S}}\}_i^{\hat{N}_{\mathcal{S}}}$$

Loss :

$$\mathcal{L}_{D^{\mathcal{S}}}(\theta^{\mathcal{S}}) = \frac{1}{2n}\sum_i \left\|\mathcal{S}(I_i^{\mathcal{S}};\theta^{\mathcal{S}}) - Y_i^{\mathcal{S}}\right\|_2^2$$

$$\tilde{\theta}^{\mathcal{S}} = \theta^{\mathcal{S}} - \alpha\nabla\mathcal{L}_{D^{\mathcal{S}}}(\theta^{\mathcal{S}})$$

Training :

$$D^{\mathcal{T}} = \{I_i^{\mathcal{T}}, Y_i^{\mathcal{T}}\}_i^{N_{\mathcal{T}}}$$

Target-domain Fine-tuning (④、⑤、⑥) :

$\theta_i^{\mathcal{S}}$ θ_i^f θ_i^b

Data :

$$\theta_i^{\mathcal{T}} = \theta_i^{\mathcal{S}} \odot \theta_i^f \oplus \theta_i^b, (i = [1,k] \in N)$$

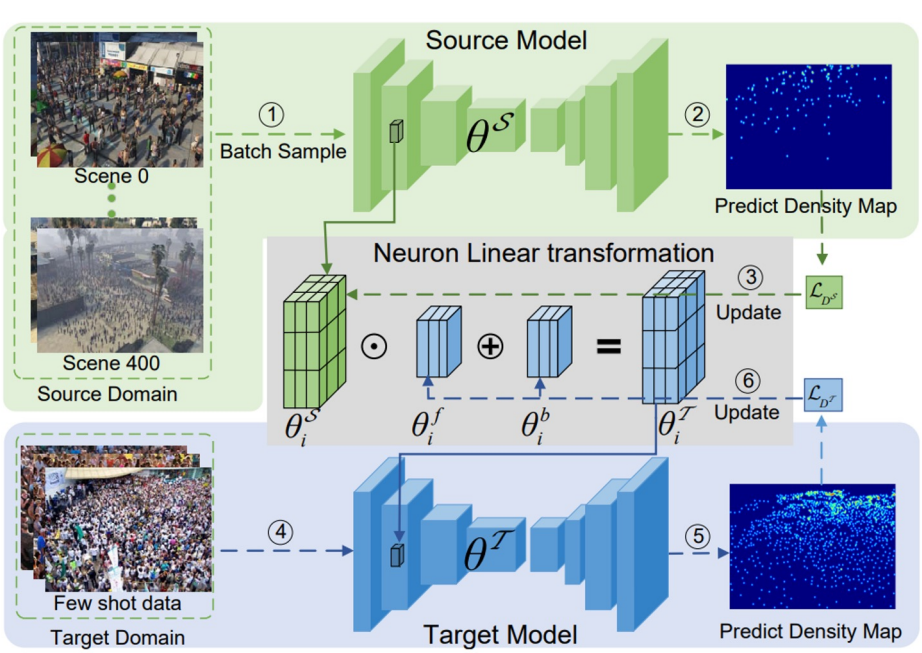
Parameter : fixed 、 learnable

$$\mathcal{L}_{D^{\mathcal{T}}}(\theta^{\mathcal{T}}) = \frac{1}{2n}\sum_i \left\|\mathcal{T}(I_i^{\mathcal{T}};\theta^{\mathcal{T}}) - Y_i^{\mathcal{T}}\right\|_2^2 + \lambda(\sum_{i=1}^{k}(\theta_i^f - 1)^2 + (\theta_i^b)^2),$$

$$\tilde{\theta}^{\mathcal{T}} = \theta^{\mathcal{T}} - \beta\nabla\mathcal{L}_{D^{\mathcal{T}}}(\theta^{\mathcal{T}})$$

Fine-tuning :

Method --- Overview



Training phase (source domain):

$\theta^{\mathcal{S}}$

Learnable parameters :

Testing phase (target domain):

θ^f θ^b

$$\theta_i^{\mathcal{T}} = \theta_i^{\mathcal{S}} \odot \theta_i^f \oplus \theta_i^b, (i = [1, k] \in N),$$

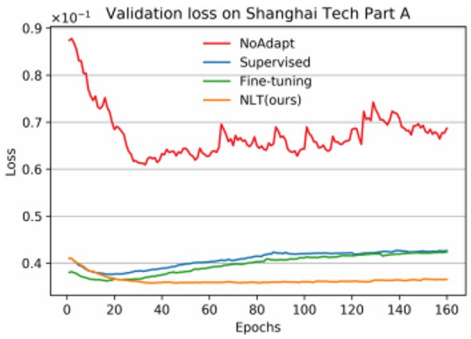
Learnable parameters :

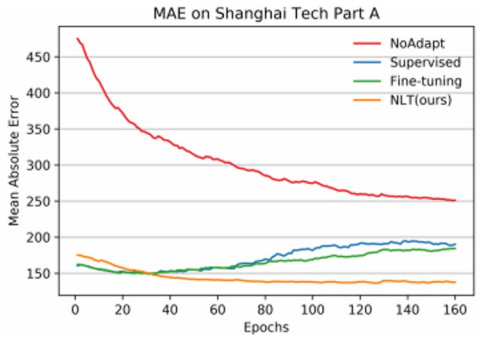
NLT :

Experiments

Training set : GCC dataset (synthetic data from GTA 5)
Testing set : 6 Crowd Counting benchmark

Method	DA	FS	Shanghai Tech Part A				Shanghai Tech Part B			
			MAE	MSE	PSNR	SSIM	MAE	MSE	PSNR	SSIM
NoAdpt	✗	✗	188.0	279.6	20.91	0.670	20.1	29.2	26.62	0.895
Supervised	✗	✔	107.2	165.9	21.53	0.623	16.0	26.7	26.8	0.932
Fine-tuning	✔	✔	105.7	167.6	21.72	0.702	13.8	22.3	27.0	0.931
NLT (ours)	✔	✔	93.8	157.2	21.89	0.729	11.8	19.2	27.58	**0.937**
IFS [26]+NLT (ours)	✔	✔	**90.1**	**151.6**	**22.01**	**0.741**	**10.8**	**18.3**	**27.69**	0.932





NoAdpt : Train on GCC , no fine-tu

Supervised : Train on Testing set wit few data

Fine-tuning : Train on GCC and fine tune on
Testing set with few dat

NLT

Experiments --- Ablation study

Method	DA	FS	Shanghai Tech Part A				Shanghai Tech Part B			
			MAE	MSE	PSNR	SSIM	MAE	MSE	PSNR	SSIM
NoAdpt	✗	✗	188.0	279.6	20.91	0.670	20.1	29.2	26.62	0.895
Fine-tuning	✔	✔	105.7	167.6	21.72	0.702	13.8	22.3	27.0	0.931
Factor (θ^f)	✔	✔	109.2	161.3	21.49	0.758	13.5	23.5	27.26	0.921
bias (θ^b)	✔	✔	107.8	169.9	21.14	0.796	12.8	20.6	27.17	0.916
NLT ($\theta^f + \theta^b$)	✔	✔	**93.8**	**157.2**	**21.89**	**0.729**	**11.8**	**19.2**	**27.58**	**0.937**

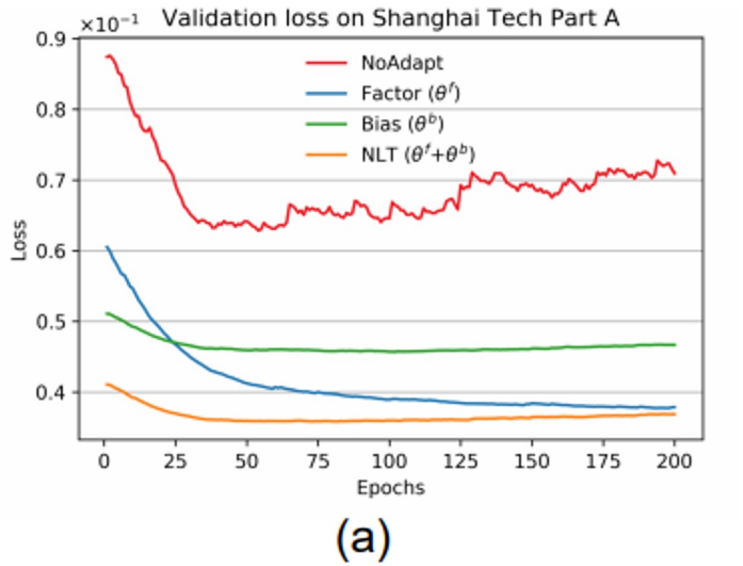
Experiments --- Comparison

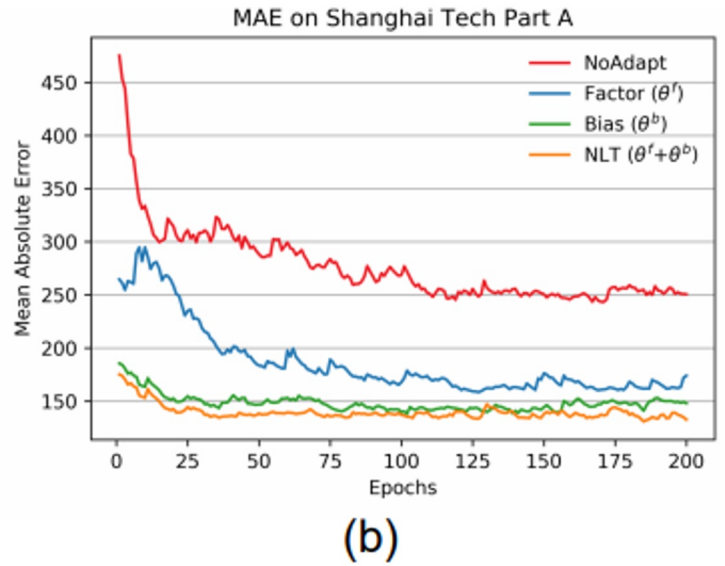
Method	DA	FS	Shanghai Tech Part A				Shanghai Tech Part B				UCF-QNRF			
			MAE	MSE	PSNR	SSIM	MAE	MSE	PSNR	SSIM	MAE	MSE	PSNR	SSIM
CycleGAN [46]	✔	✗	143.3	204.3	19.27	0.379	25.4	39.7	24.60	0.763	257.3	400.6	20.80	0.480
SE CycleGAN [25]	✔	✗	123.4	193.4	18.61	0.407	19.9	28.3	24.78	0.765	230.4	384.5	21.03	0.660
FA [28]	✔	✗	-	-	-	-	16.0	24.7	-	-	-	-	-	-
FSC [27]	✔	✗	129.3	187.6	21.58	0.513	16.9	24.7	26.20	0.818	221.2	390.2	**23.10**	0.7084
IFS [26]	✔	✗	112.4	176.9	21.94	0.502	13.1	19.4	**28.03**	0.888	211.7	357.9	21.94	0.687
NoAdpt (ours)	✗	✗	188.0	279.6	20.91	0.670	20.1	29.2	26.62	0.895	276.8	453.7	22.22	0.692
NLT (ours)	✔	✔	93.8	157.2	21.89	0.729	11.8	19.2	27.58	**0.937**	172.3	307.1	22.81	0.729
IFS[26]+NLT (ours)	✔	✔	**90.1**	**151.6**	**22.01**	**0.741**	**10.8**	**18.3**	27.69	0.932	**157.2**	**263.1**	23.01	**0.744**

Experiments --- Comparison

Method	DA	FS	WorldExpo'10 (only MAE)						UCSD				MALL			
			S1	S2	S3	S4	S5	Avg.	MAE	MSE	PSNR	SSIM	MAE	MSE	PSNR	SSIM
CycleGAN [46]	✔	✗	4.4	69.6	49.9	29.2	9.0	32.4	-	-	-	-	-	-	-	-
SE CycleGAN [25]	✔	✗	4.3	59.1	43.7	17.0	7.6	26.3	-	-	-	-	-	-	-	-
FA [28]	✔	✗	5.7	59.9	19.7	**14.5**	8.1	21.6	2.0	2.43	-	-	2.47	3.25	-	-
IFS [26]	✔	✗	4.5	33.6	**14.1**	30.4	4.4	17.4	1.76	2.09	24.42	0.950	2.31	2.96	25.54	0.933
NoAdpt (ours)	✗	✗	5.0	89.9	63.1	20.8	17.1	39.2	12.79	13.22	23.94	0.899	6.20	6.96	24.65	0.879
NLT (ours)	✔	✔	2.3	22.8	16.7	19.7	3.9	13.1	1.58	1.97	25.29	0.942	1.96	2.55	26.92	**0.967**
IFS[26]+NLT (ours)	✔	✔	**2.0**	**15.3**	14.7	18.8	**3.4**	**10.8**	**1.48**	**1.81**	**25.58**	**0.965**	**1.86**	**2.39**	**27.03**	0.944

Experiments --- Observation





Conclusion

· Summarize that domain-shift issue comes from the difference of parameter distribution between source domain and target domain

· Propose NLT to deal with the domain-adaptation task.

· Crowd Counting is just a demonstration, this few-shot learning strategy as well as NLT can be applied to other tasks