Crowd Counting: Towards Perspective-Free Object Counting with Deep Learning

Computer Vision

Mattia Paolacci

Counting people in the images

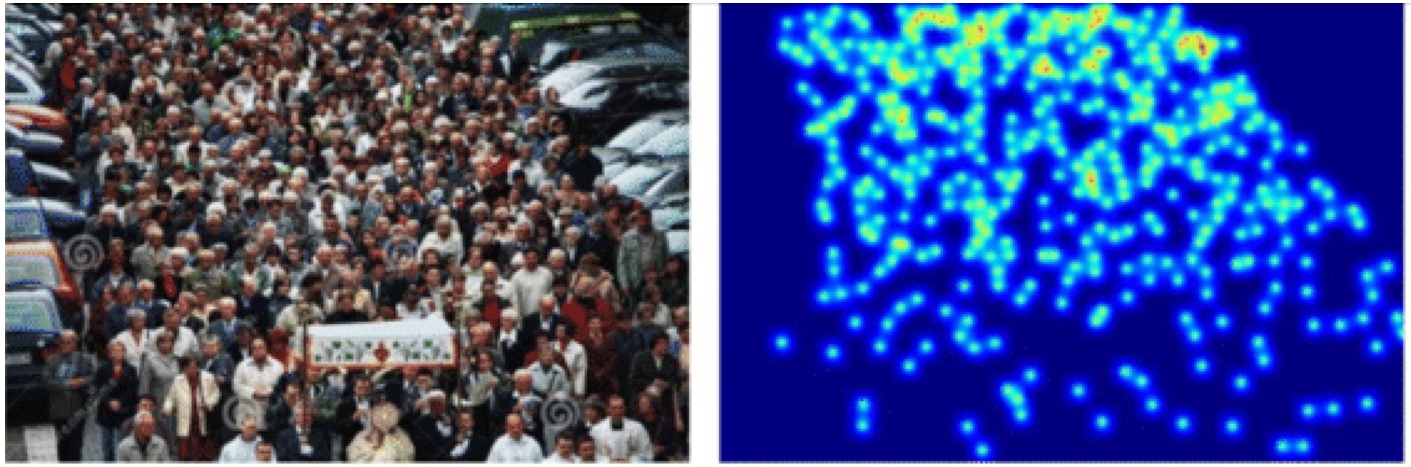
- The goal can be accomplished following three methodologies[1]:
  - *Counting by detection – by detecting the instances of person in a given image, then counting them.*
  - *Counting by clustering – assumes a crowd to be composed of individual entities, each of which has unique yet coherent motion patterns that can be clustered to approximate the number of people.*
  - *Counting by regression – counts people in a crowd by learning a direct mapping from low-level image features to crowd density.*

This paper focuses on counting by regression models.

[1] Crowd Counting and Profiling: Methodology and Evaluation. Loy, C., Chen, K., Gong, S., Xiang, T.

Counting by regression

- Essentially, this method works by defining a map from the input image features to the object count.



The image on the right represents the density map, in which the more the colour tends towards the red colour, the higher the people's density is in this area.

How to build a density map

- The density map is what we are going to guess, starting from an image.
- Given an image I, the density map D_I is defined as a sum of *Gaussian functions* centered in each dot annotation:

$$D_I(p) = \sum_{\mu \in \boldsymbol{A}_I} \mathcal{N}(p; \mu, \Sigma)$$

Where $\boldsymbol{A}_I$ is the set of 2D points, that is the points where the heads are in the images; $\mathcal{N}$ is the evaluation of a 2D Gaussian function, with mean μ and isotropic covariance matrix Σ evaluated at pixel position defined by p.

How to build a density map (2)

- Note that by building such density map we can obtain the *total object count* N_I just by integrating the density map values in D_I :

$$N_I = \sum_{p \in \mathrm{I}} D_I(p)$$

The Counting CNN (CCNN)

- The main goal of the work is to design a NN that can learn the non-linear regression function $\mathcal{R}$ such that:

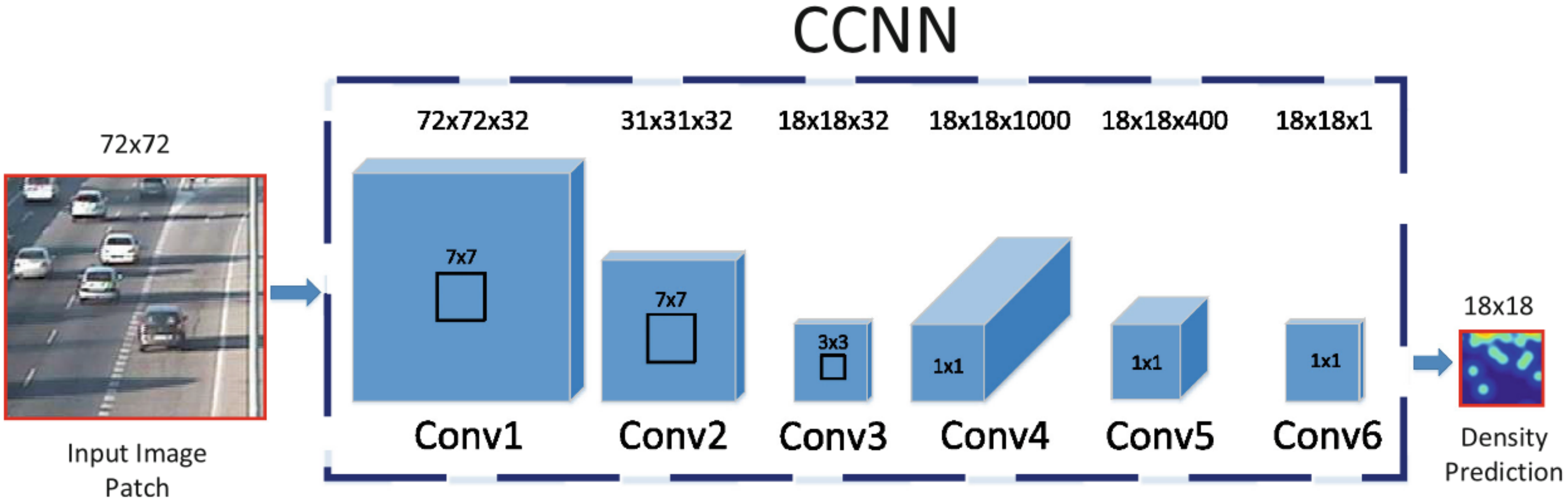
$$D_{pred}^{(P)} = \mathcal{R}(P|\Omega)$$

Where Ω is the set of parameters of the CNN, P is the image patch and $D_{pred}^{(P)}$ is the predicted density map.

- The loss is a MSE between a predicted density map and the ground truth density map.

The Counting CNN (CCNN) (2)

The NN architecture appears as the following:

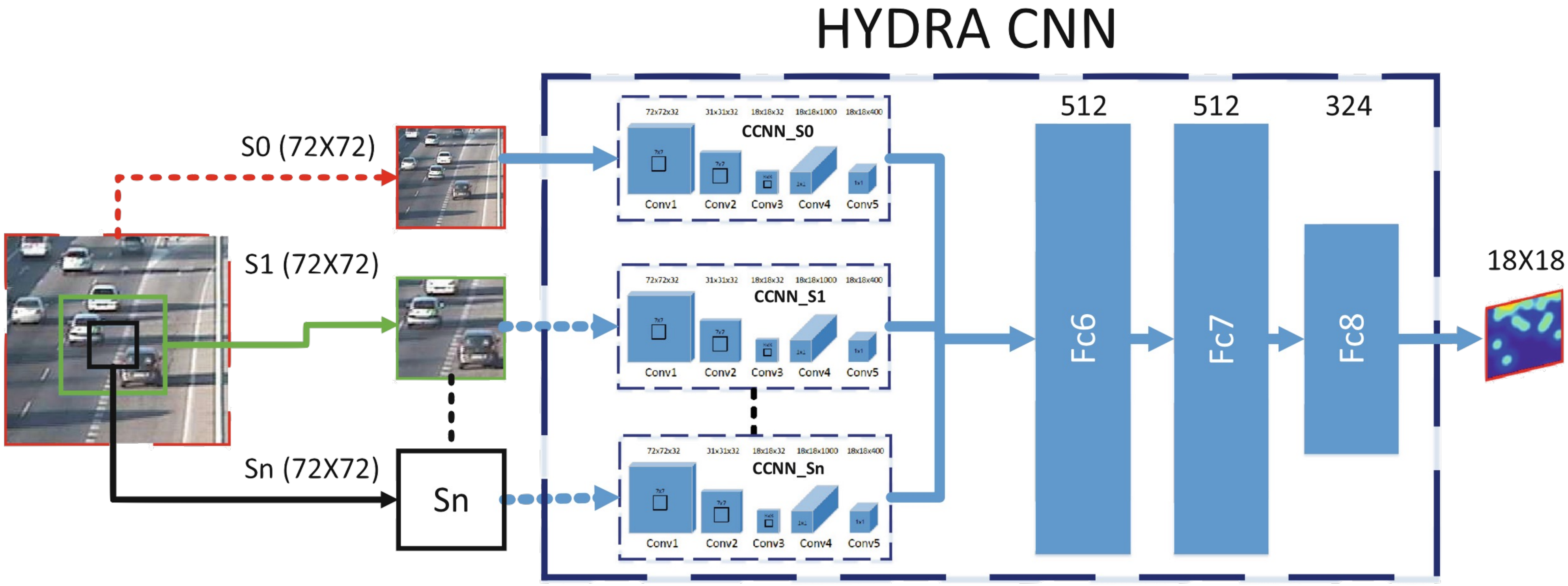


The Hydra CNN

- The perspective distortion exhibited by an image causes the features extracted from same object but at different scene depths to have a huge difference in values. Hence the CCNN shown above, alone is not enough.
- The idea is to learn the non-linear regression mapping by integrating the informations, without any geometric correction of the scene, from multiple scales simultaneously.

The Hydra CNN (2)

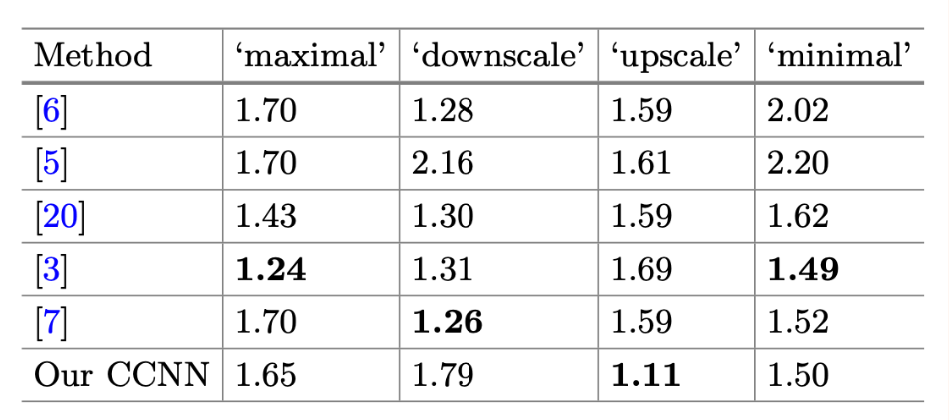
- Each *head* of the Hydra model is in charge of learning features for a given scale s_i from the input image
- The *body* concatenates all the features output from the heads by a fully connected-layers.



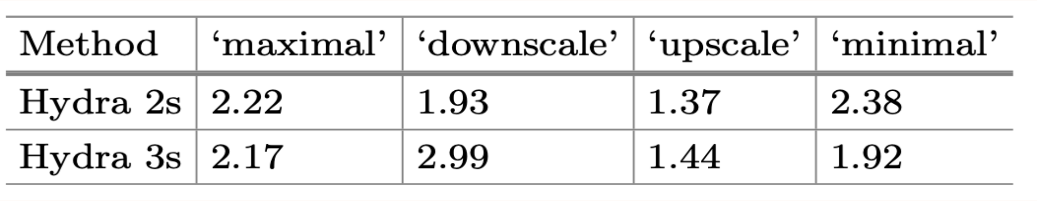
Evaluation of the proposed architecture

Authors of the paper carried out two types of experiments to compare the performances of the CCNN model to those of the Hydra model:

- The former follows [5-7,26] and has been done by using the *UCSD dataset* which contains the video frames captured by a single camera (*single scene*), the results (MAE) are the following:



Method	'maximal'	'downscale'	'upscale'	'minimal'
[6]	1.70	1.28	1.59	2.02
[5]	1.70	2.16	1.61	2.20
[20]	1.43	1.30	1.59	1.62
[3]	**1.24**	1.31	1.69	**1.49**
[7]	1.70	**1.26**	1.59	1.52
Our CCNN	1.65	1.79	**1.11**	1.50

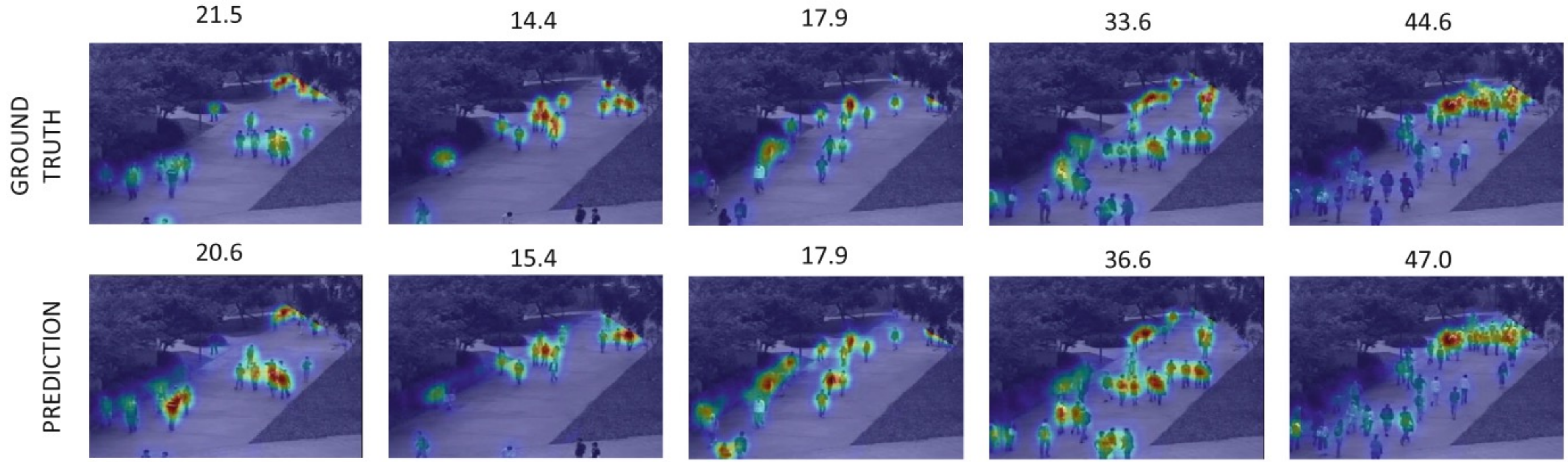


Method	'maximal'	'downscale'	'upscale'	'minimal'
Hydra 2s	2.22	1.93	1.37	2.38
Hydra 3s	2.17	2.99	1.44	1.92

The columns are set of frames, the sets have the following instances: *maximal* 160 frames, *downscale* 79 frames, *upscale* 59 frames, *minimal* 9 frames.

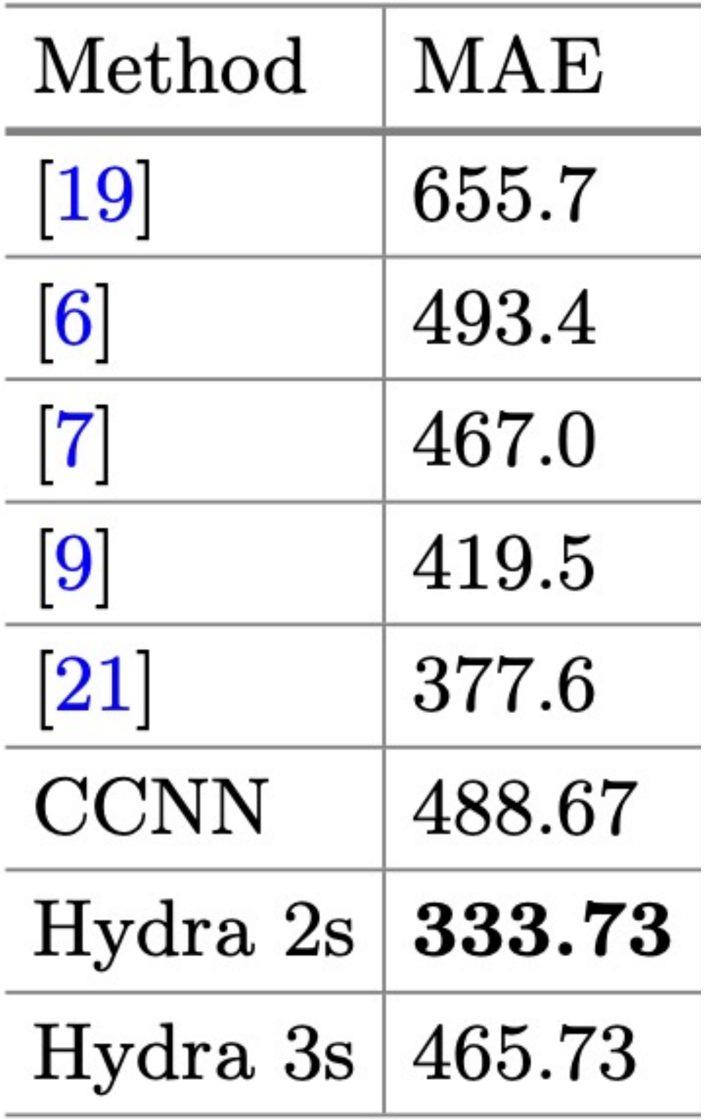
Evaluation of the proposed architecture (2)

A qualitative result of the CCNN model.



Evaluation of the proposed architecture (3)

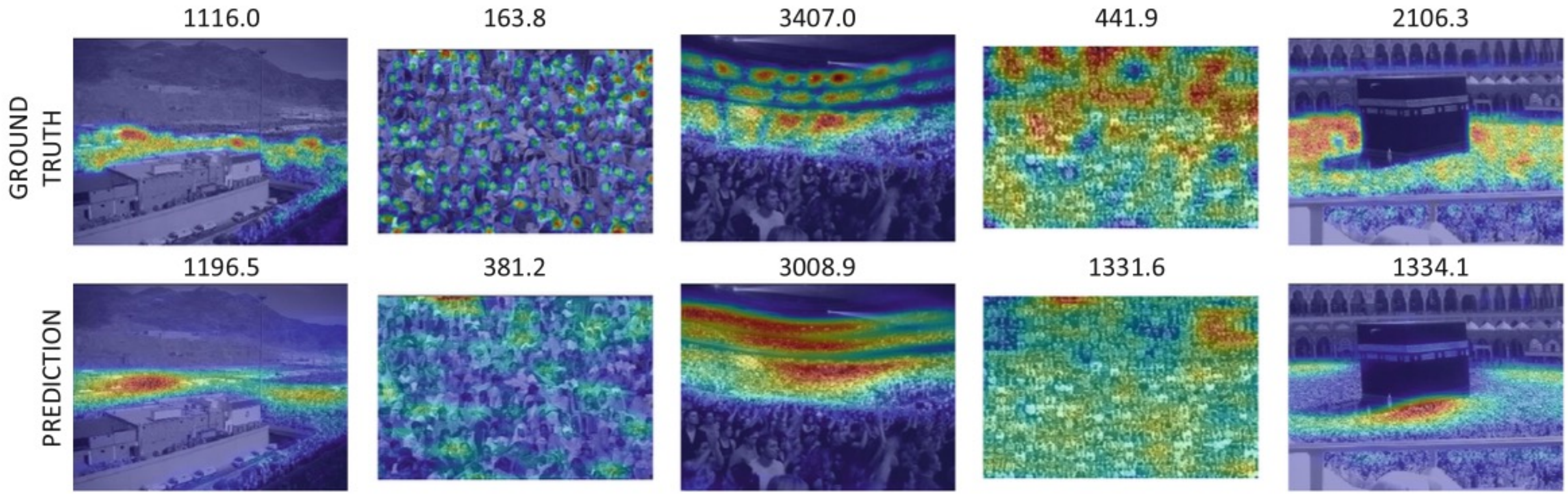
- The latter experiment consists in measuring the performances of the Hydra model on a dataset of 50 pictures with a range of people between 94 and 4543, and a mean of 1280 people per image.
- In this case the images have different scenes with an hight density of people: concerts, protests, stadiums, marathons and pilgrimages.
- By observing the table alongside, the Hydra model seems to outperform with respect to the others.



Method	MAE
[19]	655.7
[6]	493.4
[7]	467.0
[9]	419.5
[21]	377.6
CCNN	488.67
Hydra 2s	**333.73**
Hydra 3s	465.73

Evaluation of the proposed architecture (4)

A qualitative result of the Hydra model with two scales.



References

- [3] Arteta, C., Lempitsky, V., Noble, J.A., Zisserman, A.: Interactive object counting. In: ECCV (2014)
- [5] Fiaschi, L.,K˙othe,U., Nair,R., Hamprecht, F.A.: Learning to count with regression forest and structured labels. In: ICPR (2012)
- [6] Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: NIPS (2010)
- [7] Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: CVPR, June 2015
- [9] Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. In: CVPR (2013)
- [19] Rodriguez, M., Laptev, I., Sivic, J., Audibert, J.Y.: Density-aware person detection and tracking in crowds. In: ICCV (2011)
- [20] Pham, V.Q., Kozakaya, T., Yamaguchi, O., Okada, R.: COUNT forest: CO-voting uncertain number of targets using random forest for crowd density estimation. In: ICCV (2015)
- [21] Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: CVPR, June 2016
- [26] Ryan, D., Denman, S., Fookes, C., Sridharan, S.: Crowd counting using multiple local features. In: DICTA (2009)

Following the order of the paper.

Thanks for the attention! ☺