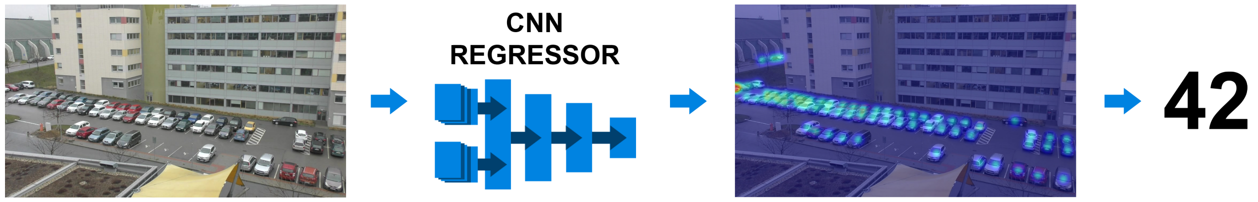
BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2020
http://excel.fit.vutbr.cz

Counting Vehicles in Static Images

Ondřej Zemánek



Abstract
This paper addresses the problem of counting vehicles in static images with no geometric information of the scene. Four convolutional neural network architectures were studied, implemented and trained as a main part of this work. Also, a dataset that consists of 19 310 images in total from 12 views that captures 7 different scenes were taken as part of this work. The trained networks map the appearance of the input sample to its corresponding vehicles density map, which can be easily translated to the vehicle count with keeping the localization of the vehicles in the input image. The main contribution of this work is in an application and a comparison of the state-of-the-art solutions to the problem of object counting. Most of them were mainly designed to count pedestrians in crowded scenes or for medicine images, so the major goal was to adapt these solutions for vehicle counting task. The implemented models were trained on TRANCOS dataset which is a popular benchmark for counting vehicles on annotated low quality highway pictures. Their performance is compared and the results are discussed.

Keywords: visual counting, vehicle counting in static images, car park dataset

Supplementary Material: *N/A*

*xzeman53@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Visual counting that aims to accurately estimate the number of vehicles is a hard problem. But its potential is huge in many applications across many industries.

For instance, it can help truck drivers, who need to plan their next break, by monitoring parking capacity near highways. Another application can be long-term analysis of a traffic density on main city roads and highways, so road closures, detours or road expansion can be planned easily and smoothly. Also, solving the vehicle counting problem can bring a cheaper solution for monitoring shopping center parking lots, so instead of using a physical sensor for each parking space, a few cameras can be used to monitor the parking lot.

The most recent state-of-the-art solutions are based mainly on the convolutional neural network model. Therefore, this work is focused only on these approaches. The best approaches can be divided into objects detection approaches and density map regression based approaches.

The first group uses classification of individual objects in YOLO-like (You Look Only Once) [1] style to detect and count the objects in the input image. Although these approaches can be fast and reliable in trivial cases, in very dense and overcrowded scenes like images with overlapping objects, low-resolution, partly visible objects, images with slightly unseen perspective, the overall performance of these models is limited.

The other group of solutions that reaches much better results in the target scenarios is based on a different idea. Instead of solving this problem by detection of each object in the image, these solutions transform the visual counting task into object density map estimation from the input image. In other words, they are using convolutional neural networks to transform an input image appearance into an object density map in a certain resolution. From the output of this transformation, the object count can be easily estimated by the output density map integration even with keeping the information of the objects localization.

The main contribution of this work is an application and a comparison of the existing solutions on parking lot dataset. To achieve this, I had to analyse the existing convolutional neural architectures, adapt these models to vehicle counting problem, create a large and diverse dataset for training, train them in with various parameters, and finally, evaluate them.

2. Implemented Architectures

The state-of-the-art approaches for visual counting are mainly built upon one of two concepts: density map regression and detection. Thus, the chosen architectures are designed in this way.

First three solutions are based on Density map regression. So, the input image is regressed into density map that represents spatial distribution of objects. Then, this map is integrated into a object count which corresponds to the input sample.

The last studied approach is somewhere in the middle of these two concepts. It combines density map regression with classification. More in subsection 2.4.

Tensorflow open source platform was used to implement these convolutional neural networks. The final models of the Counting CNN, the Hydra CNN and Spatial Division and Conquer Network were inspired by the original authors implementations, that were implemented in the Caffe framework and the PyTorch platform respectively. The authors of Stacked Hourglass model provides only brief description of the implementation, so the network implementation is based on this description only.

Despite the fact that the authors shares the implementation details, the training process implementation is tied its application and the dataset. Thus, the training process for each used network has to be created manually based on the deep analysis of the architecture.

2.1 From Human Pose Estimation to Visual Counting

The Stacked Hourglass architecture proposed by Newell at al. [2] is a composition of multiple modules called hourglass. Each Hourglass module processes

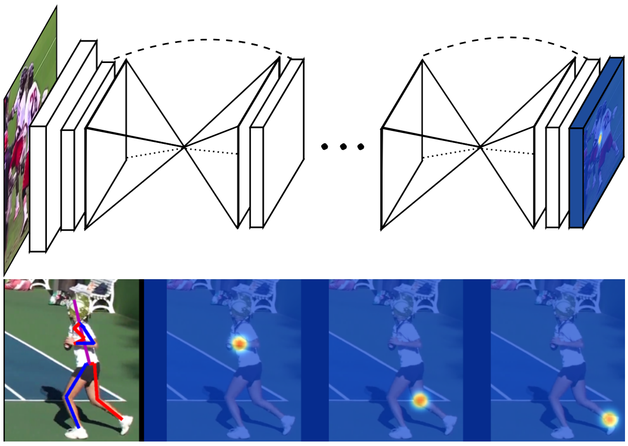


Figure 1. top: Stacked hourglass concept – input image is passed through multiple hourglass modules. Each hourglass model creates an intermediate prediction to improve the final result. **bottom:** Output of the multiple-level hourglass model for Human pose estimation problem [2].

the input features on multiple scales and consolidates them to best capture the object landmarks. This is done by repeated up-down/bottom-up process. Also, it uses an intermediate supervision process, i.e. the architecture uses intermediate prediction to improve the next prediction. This is done by skipping layers. This model (Figure 1) aims on the problem of human pose estimation, but as it is shown in the achieved results (Sec. 5), it also shows good results in the visual counting problem.

2.2 Single-Pipeline CNN with Great Results

The Counting convolutional neural network created by Oñoro et al. [3] is a simple sequence of 6 convolutional layers and two max-pooling layers, as can be seen in Figure 2. The input image patch with size 72×72 pixels is processed by this sequence and it returns a density map with 18×18 pixels size which represents the object spatial distribution in the image. Finally, the object count is gathered by integrating the density map.

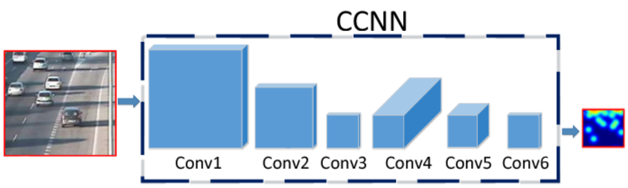


Figure 2. Counting CNN architecture – sequence of 6 convolutional layers in an up-down order [3].

2.3 Even Better Results with Multi-Scale Inputs

Next implemented architecture is called Hydra CNN, like the mythological creature Hydra with nine heads and a big body. It is mainly based on multiple Counting CNN's that are used as the input heads of the creature as illustrated in Figure 3. Each head processes the input sample on a different scale, so the final architecture is a scale-aware solution for visual counting. The patch scale (crop ratio of the original sample) for head H_i is defined as follows,

$$H_s = 1 - \frac{1}{C} \cdot H_i, \tag{1}$$

where C is the number of heads. In case of the first level head, the input patch corresponds to the original sample. The heads' intermediate outputs are fused by three fully connected layers, so even the input array contains the image in multiple scales, the output is a single density map like in the case of Counting CNN. The concept can be easily extended by adding a new head or simplified by removing one. The results are much better than in the case of the simple counting CNN.

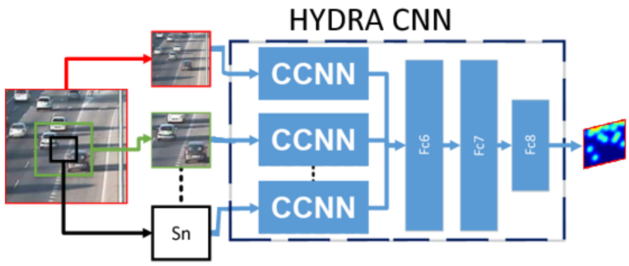


Figure 3. Hydra CNN scheme – input sample is up-sampled multiple times and processed by the Counting CNNs. The sub-results are merged into a single output map by fully connected layers.

2.4 Open-Set to Closed-Set Transformation as the State of the Art

Previously described methods are modeled in a regression manner. Xiong et al. [4] proposed a new different approach with their Spatial Divide-and-Conquer Network (S-DCNet) that is more complex and it uses multiple modules with different purposes. The main idea is spatial division of the input image into small regions, each with a closed set of a defined range, so they can transform quantity to intervals which the network can easily classify.

It is based on the main idea that the visual counting problem is an open-set problem by nature. But only limited and closed set labeled counts can be observed in reality. So the goal is to transform the original problem into a close-set one.

This is done by spatial division of the image until the count of every part is in a specified range. For example, the experimented range for vehicle counting I

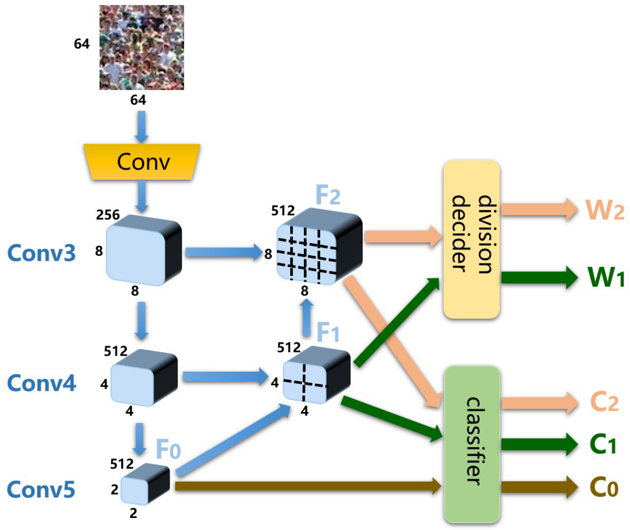


Figure 4. The architecture of Spatial division-and-conquer network. The first 5 layers are convolutional and follow the VGG16 concept [5]. These layers are than connected to the classifier and division decider straight forward or preceded by fusion with another layer. Finally, the output weights and classification are processed to get the object density map.

am using is $[0,5]$. That means the architecture spatially divides the input image, until every part reaches 5 object at max.

The division decider in Figure 4 is trained to decide whether it is necessary to divide the input or not. It returns W_i weights in range $[0,1]$, where greater value means higher need for division. The weight is than used to compute the division result DIV_i. So, higher resolution count maps C_i are applied if the division weight W_i is higher.

Next, the classifier module predicts the object counts C_i for each output feature map. The classification is done on specified intervals in the closed-set range, i.e. for each feature map, a single object count is obtained.

After the model prediction, following post-process is applied:

$$DIV_i = (1 - W_i) \circ avg(C_{i-1}) + W_i \circ C_i, \tag{2}$$

where "$\circ$" denotes the Hadamard product and avg is an averaging re-distribution operator. Finally, predicted object count is represented by last DIV_i map. The i level corresponds to max division time which is the value that limits the input image division. So, if the i is 2, than the model takes into account up-to 2 times divided input image.

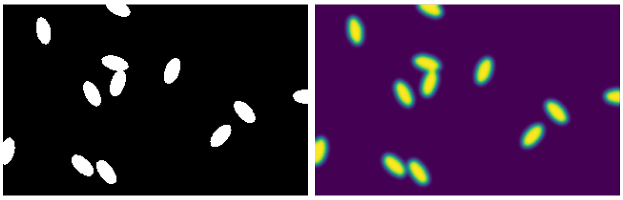


Figure 5. Toy dataset example of input sample (left) and output density map (right).

3. Training Details

The authors of the proposed solutions are proving their network performance with pretrained model that has been successfully trained on available benchmark dataset. Unfortunately, they do not always share training details.

For instance, the paper about proposed Spatial division-and-conquer architecture contains very vague information about the training process. Therefore, to train this model, it was necessary to understand the architecture in depth and experiment with model parameters.

3.1 Iterative Training Process With a Toy Dataset

To train the models on large and diverse dataset, it is necessary to know how the models performs with different training settings and parameters. As the fastest approach to make the training process work in the way how it was designed is training on a simplified, "toy" dataset.

Toy dataset, as can be seen in Figure 5 is a custom generated dataset which can be easily created in a short time. Its parameters, like Gaussian blur variance σ, resolution and object shape, can be modified for each convolutional neural network architecture. The main benefit of this simple dataset is that the training process is much faster than training on real images.

3.2 Output Activation and Loss Functions

Even with a fast training process, there is another important factor influencing training success. That is the combination of the last activation function and the loss function. The last activation function gives the transformation of the linear output value and the loss function is used to compute the trained model error.

Unfortunately, there is no general-purpose combination of these functions. So, as the implemented architectures do not have the same output format, it was necessary to understand the model pipeline and decide which combination is the best for each output.

For classification problem is common to use sigmoid or soft-max function as the last activation function and some type of cross-entropy function. Also, there are multiple cross-entropy loss functions for different purpose, like multi-class or binary classification.

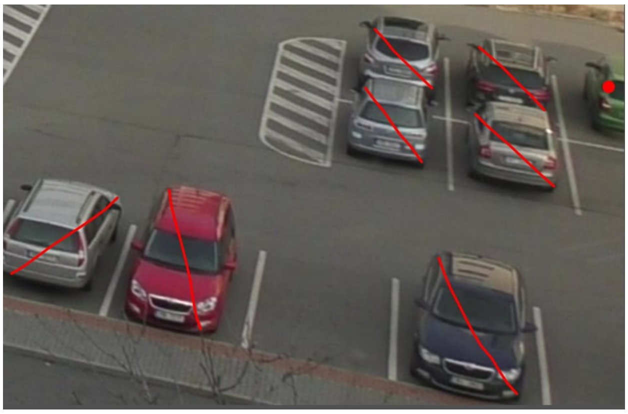


Figure 6. Example of the labeling style. Scribbles are used for high-resolution vehicles and dots for partially visible cars or vehicles in distant.

For regression problem, it is typical to use linear function as the activation function and use L1-norm (mean absolute error) or L2-norm (mean square error) as loss functions (density map). Otherwise, the activation function can be set to sigmoid with use of quadratic loss function so the regressed value is in range $[0, 1]$ (weight, normalized output)[1].

The Counting CNN, Hydra-CNN model and the Stacked Hourglass model are using a combination of linear activation function and mean square error loss function.

The Spatial Division-and-Conquer model is much more complex. The network output consist of the division weights and quantity interval classifications. The division weight is a regression to values between 1 and 0, so the combination of sigmoid activation function and mean square error loss function are implemented. The count interval problem is a multi-class classification, so the soft-max activation function with mean absolute error loss function is applied.

3.3 Ground Truth Labeling

For ground truth labels is used dotted annotation blurred by 2D Gaussian function. So, the vehicles in images are labeled by only single dot in first step. Then the dotted map is then blurred with Gaussian blur. Even after blurring the dots out, an integration of the blurred map still corresponds to the vehicle count. During the training process this map is used as the ground truth density map.

4. Diverse and Robust Training Dataset

To train the implemented network to count vehicles in images, it is crucial to have a big enough dataset

[1] https://medium.com/@phuctrt/189815343d3f

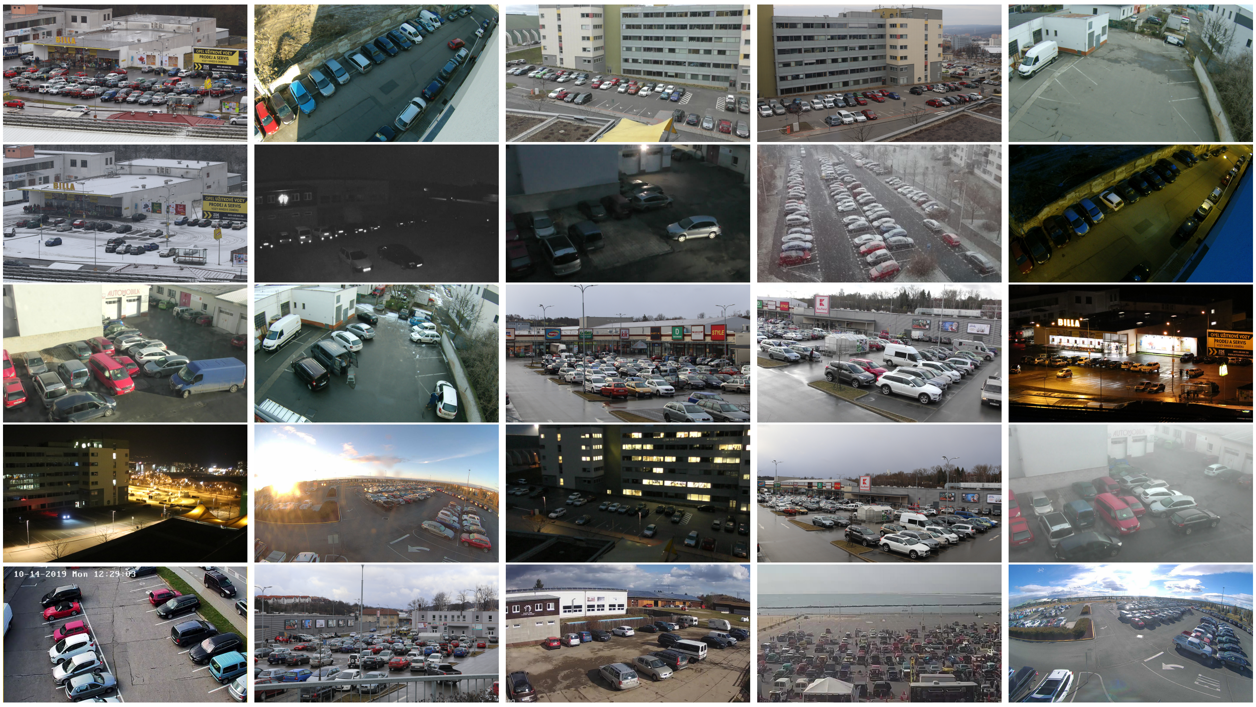


Figure 7. Custom dataset samples.

of images with similar parameters. As the parameters depend mainly on the final application, it is necessary to define it.

Parking lot occupancy monitoring was chosen as the main application. Therefore, the dataset images should capture parking areas or similar places like highways or streets with cars.

4.1 Existing Datasets

Currently, there are three suitable datasets for this application. The first one is the TRaffic ANd COngestionS (TRANCOS) [6] dataset with more than thousand labeled low-quality images from highway. The next dataset is called CNRPark+EXT[2] and it captures occupancy of parking lots with roughly 4 300 labeled images. Lastly, the CARPK[3] is a collection of drone images of huge parking lots with 1 500 annotated images where only a part of it is suitable for our application.

4.2 Newly Created Dataset

As the main goal of this work is an robust real-world application for vehicle counting problem, we need much more diverse and robust dataset to train the networks. Therefore, a new and more diverse parking lot dataset was collected as part of this work. Figure 7 shows examples of this custom dataset. It consist of 19 310 images in total from 12 views that capture 7 different scenes.

[2] http://www.cnrpark.it/
[3] https://lafi.github.io/LPN/

Each location was captured from a similar angle to the ground to simulate the common monitoring cam position. The recording process took place from September to March, so diverse weather and lighting conditions were captured. Also, three online webcams were recorded as part of the custom dataset. This adds another few thousands of images to this dataset.

4.3 Annotation

The training cannot be done without the ground truth labels for the dataset pictures. Several annotation styles were tested to label the images, like bounding box, silhouette, scribble. Although the bounding box annotation is common approach for object detection and silhouette annotation is even more precise, these two label styles are too time-consuming and unnecessary for our application. Thus, the faster and sufficient scribble and dots labeling styles were chosen as can be seen in Figure 6.

So far, more than 3 500 images were annotated as part of this work and the labeling process still continues.

5. Achieved Results

The presented networks were trained on the TRANCOS dataset to demonstrate their performance so far. The dataset contains training, validation and test sets, so the results can be accurately compared on samples that were not used for training. Visual comparison of the trained models on TRANCOS dataset can be seen in Figure 8.

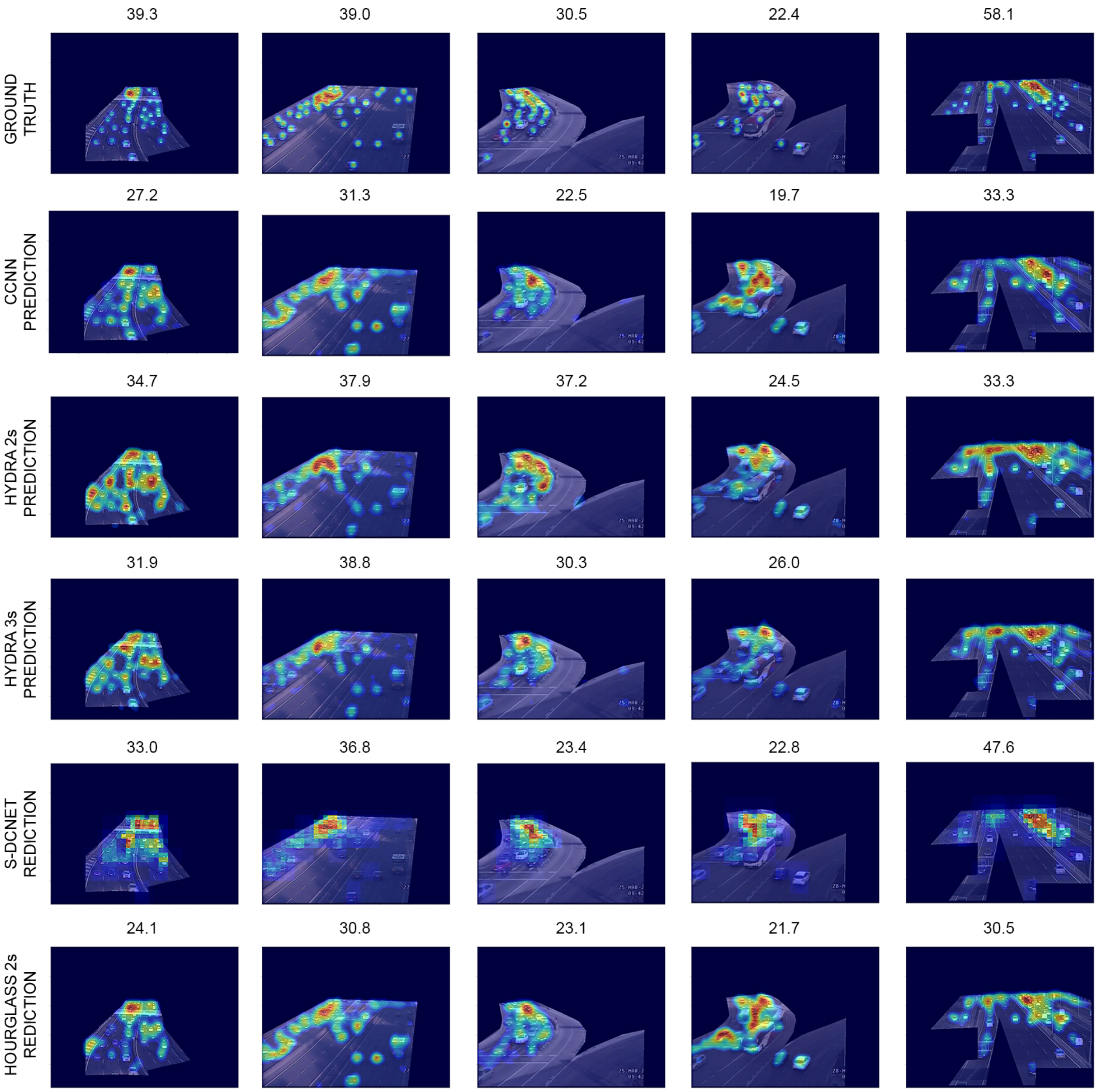


Figure 8. Five test samples from TRANCOS dataset with trained models predictions. Top row corresponds to the target image with ground-truth. "Hydra 2s", "Hydra 3s", "Hourglass 2s" stands for the Hydra CNN with 2 heads, 3 heads and the Stacked Hourglass model with 2 stacks respectively. The ground-truth counts are slightly different because Gaussian blur was applied to the label maps.

The comparison of the used architectures on full TRANCOS dataset can be seen in table 1. The evaluation was done with Grid Average Mean Absolute Error (GAME) [7] metric defined by

$$GAME(L) = \frac{1}{N}\sum_{n=1}^{N}\left(\sum_{l=1}^{4^L}|C_{pre}^{l} - C_{gt}^{l}|\right), \quad (3)$$

where N denotes the number of images, C_{pre}^{l} and C_{gt}^{l} are the predicted and ground-truth count of the L-th subregion, respectively.

Method	GAME 0	GAME 1	GAME 2	GAME 3
CCNN	12.18	16.44	20.35	22.97
Hydra 2s	10.77	14.28	17.69	21.13
Hydra 3s	11.02	14.37	17.01	20.64
SHG 2s	14.30	15.84	18.23	22.81
S-DCNet	**8.56**	**9.357**	**10.40**	**11.83**

Table 1. Trained model evaluation with GAME metric on the TRANCOS dataset. The best performance is in boldface. "SHG 2s" stands for the Stacked Hourglass model with 2 stacks

The Counting CNN achieves good results in total count prediction, but in higher levels on GAME metric shows some false prediction. In case of the Hydra CNN the results are better but the problem with noisy prediction is remains. The best predictions gives the Spatial Divide-and-Conquer Network, which has accurate prediction across all GAME levels, so the spatial prediction is very precise. Lastly, the Stacked hourglass model with 2 stacks shows great results in spatial prediction, but the total density of the predicted map is lower than ground-truth.

Next step in comparison of these architectures is the custom dataset evaluation with the GAME metric. However, the training process is still in progress and I don't want to present temporally results.

6. Conclusions

In this paper, I have shown four different architectures for visual counting that has been implemented, described the training details of these models. Also, The custom car park dataset with 19 300 images and 3 500 already labeled pictures have been presented.

The architectures were evaluated on the popular TRANCOS dataset and the results were shown in last chapter.

The newly created dataset is being continually updated with new locations and more importantly it is being labeled.

Also extended version of the Spatial division and conquer network has been recently released by the authors and I am finishing the implementation of this network.

Finally, all the presented networks are being trained on the custom dataset and will be evaluated.

Acknowledgements

I would like to thank my supervisor prof. Ing. Adam Herout, Ph.D. for great support and constructive criticism during my work on this paper. Also, I would like to thank to Ing. Jakub Špaňhel for his advice and help with a dataset acquisition.

Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015-042), is greatly appreciated.

References

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[2] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[3] Daniel Onoro-Rubio and Roberto J López-Sastre. Towards perspective-free object counting with deep learning. In *European Conference on Computer Vision*, pages 615–629. Springer, 2016.

[4] Haipeng Xiong, Hao Lu, Chengxin Liu, Liang Liu, Zhiguo Cao, and Chunhua Shen. From open set to closed set: Counting objects by spatial divide-and-conquer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8362–8371, 2019.

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

[6] Roberto López-Sastre Saturnino Maldonado Bascón Ricardo Guerrero-Gómez-Olmedo, Beatriz Torre-Jiménez and Daniel Oñoro-Rubio. Extremely overlapping vehicle counting. In *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, 2015.

[7] Ricardo Guerrero-Gómez-Olmedo, Beatriz Torre-Jiménez, Roberto López-Sastre, Saturnino Maldonado-Bascón, and Daniel Onoro-Rubio. Extremely overlapping vehicle counting. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 423–431. Springer, 2015.