Supervised Spatial Divide-and-Conquer Applied to Fish Counting

Gianna Arencibia-Castellanos
CITSEM
Universidad Politécnica de Madrid
Madrid, Spain
gianna.arencibia@upm.es

Alejandro González-Fernández
CITSEM
Universidad Politécnica de Madrid
Madrid, Spain
alejandro.gfernandez@alumnos.upm.es

María Castillo-Moral
CITSEM
Universidad Politécnica de Madrid
Madrid, Spain
maria.castillom@upm.es

Rubén Fraile
CITSEM
Universidad Politécnica de Madrid
Madrid, Spain
r.fraile@upm.es

Juana M. Gutiérrez-Arriola
CITSEM
Universidad Politécnica de Madrid
Madrid, Spain
juana.gutierrez.arriola@upm.es

Fernando Pescador
CITSEM
Universidad Politécnica de Madrid
Madrid, Spain
fernando.pescador@upm.es

***Abstract*—The estimation of fish biomass plays a crucial role in aquaculture. Performing this task automatically using machine learning algorithms has attracted the attention of the scientific community. This work describes the application of Supervised Spatial Divide-and-Conquer net to counting the number of larvae present in an image of an aquaculture tank. SS-DCNet is among the most robust object counters in the state of the art when applied to different datasets. It is trained with labeled images of turbots in breeding tanks, taking into account that the sizes can be variable and that they can be grouped and overlapped. Data augmentation is applied to obtain a greater number of training instances. The application of this model to counting turbots in images provides a mean relative error lower than 3.5%, which is an acceptable accuracy for this task. The main advantage of the model studied is its generalization ability, confirmed by its performance in counting objects in images where the density and the total number of objects are much higher than for the training images. Adapting the model for counting other types of fish, or turbot in other stages of growth, is straightforward since it is not necessary to build large training datasets.**

***Index Terms*—Image processing, Object detection, SS-DCNet, biomass estimation**

I. Introduction

Biomass estimation, that is, knowing the number of fish and their weight, allows fish farmers to optimize the amount of feed, plan later stages of farming, and make decisions at the right times. Traditionally, biomass estimation has been carried out by people using invasive procedures that are usually slow and laborious and require great expertise, experience, and knowledge of the conditions of the farm and the environment [1].

Technological advances in recent decades have allowed the development of systems that offer automatic estimation of biomass based on artificial vision, acoustic signals, environmental deoxyribonucleic acid (DNA), or resistivity counters. These methods are objective, noninvasive and produce repeatable and reliable results. In contrast, they can be expensive and not easily adaptable to variations in the environment [1].

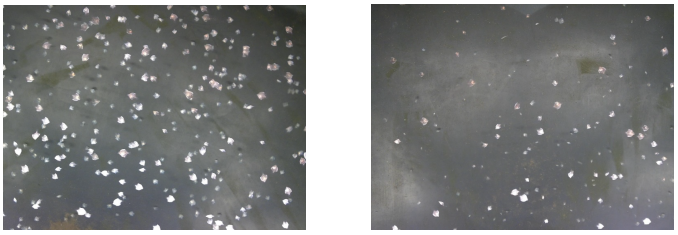
Recently, machine learning (ML) techniques have grown remarkably in applicability to the fields of industry, social networks, etc. In aquaculture, they have been used to predict water quality [2], identify and distinguish among fish types [3], diagnose diseases [4], estimate biomass [5], etc. Both image recording technology and computer services have been generalized and cheapened so that biomass estimation systems can currently be developed cheaply and reliably. The number of fishes in an image is among the parameters required for biomass estimation. For the purpose of estimating it, the algorithmic approaches used for counting objects in RGB images can be adapted.

To date, approaches used for counting objects in images can be grouped in roughly three types: counting by detection, regression, and density estimation [6]. Counting by detection is based on the position of each object in the image using the extracted image features. These methods have shown good results in datasets where the objects are separated from each other. However, in scenes where the objects are next to each other or even overlapping, the results have not been good. Some recent proposals in this area, using local features instead of global features, have improved counting results in images with high object density [6].

Alternatively, counting based on regression models attempts to establish a relationship between image features and the number of objects using supervised machine learning techniques. These models do not use datasets based on the location of individual objects but require only the total number of objects in the image. Thus, although the results of these models are generally better than those based on detection, they usually require large datasets to be trained [6].

The two model types previously described ignore the spatial information of the images; the solution proposed in [7] incorporates this information. In this work, a mapping of the features in the images and their corresponding density maps are developed that improves the accuracy of the counting

 ISBN: 978-1-68558-057-5



(a) (b)

Figure 1. Example of frames captured at a frequency of 15 f/s. (a) High density of turbots and (b) Low density of turbots.

results compared to previous approaches [8]. The advantages of this proposal are the following: the density maps provide more information about the distribution of the objects, and the algorithm is more adaptable to objects with different sizes and more tolerant to different images [8].

Aforementioned research suggests that the application of ML algorithms to images of fish larval tanks can enable the implementation of low-cost, accurate, and reliable biomass estimation systems. In this paper, we develop a system that allows obtaining an estimated number of turbot larvae present in RGB (red, green, blue) images. For this purpose, a deep learning algorithm is trained with labeled images of a fish larval tank, taking into account that fish sizes can appear to be variable in the image due to differences in depth, and that there can be grouped and overlapped objects.

The organization of the document is as follows: section II.A explains the experimental setting and construction of the dataset. Section II.B describes the implemented machine learning algorithm and the evaluation of several hyperparameters. Furthermore, the influence of different hyperparameter values on the prediction is measured with error metrics. The optimal values of the hyperparameters and the generalization capacity of the neural network were verified in section III: *Results and Discussion*. Finally, section IV presents the conclusions of the work.

II. Materials and Methods

This section describes the neural network used to count the number of turbot larvae in an image, as well as the dataset used to train and test the model. In addition, the parameters that characterize a neural network and the metrics used to evaluate its performance and generalization capacity are explained.

A. Dataset

The dataset consists of 156 RGB images with a resolution of 2560×1920 pixels. Two sample frames are shown in Figure 1. The images were manually annotated in the Group of Multimedia and Acoustic Applications (GAMMA) in our university with a Matlab® application specifically developed for this purpose.

Figure 1 shows two frames prototypical of two different cases: the left frame shows a high density of turbots while density is low in the right one. These images were captured in the same tank at different moments.The implemented algorithm must produce equally acceptable results in both cases, and also in intermediate ones.

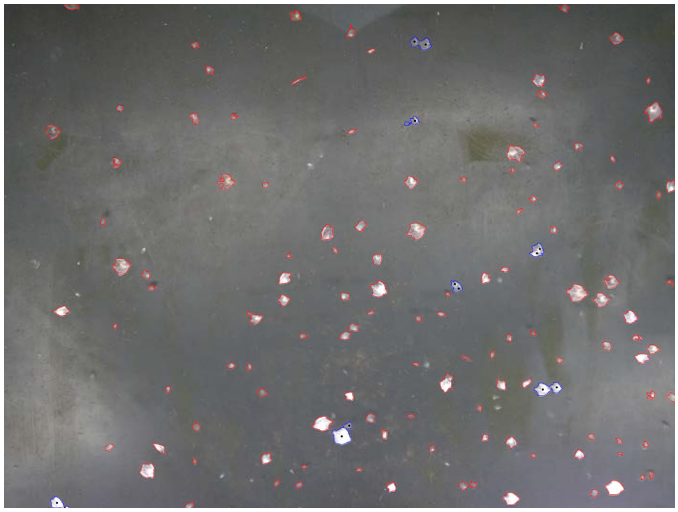


Figure 2. Segmentation of turbots, the red and blue boundaries stands out a single turbot and a group of turbots, respectively.

All images were taken from turbot larval tanks. The camera was located with the lens axis perpendicular to the water surface. In order to avoid the glaring of lighting reflections on the water, the camera focused only in part of the tank surface. Users of the annotation application were provided with images for which a segmentation by threshold had been applied to identify the objects present in the image (see Figure 2). Annotators were asked to check whether each object corresponded to a turbot larva or not. The process was made manually, and image by image, which is laborious and time consuming. But it is the most confident procedure to get a ground-truth fish count for each frame.

To train and test the neural network, the images were randomly divided into training and testing sets: 124 (80%) for training and 32 (20%) for testing. The distribution of turbots in both sets averaged 246 and 273 turbots per image, respectively.

B. Machine learning algorithm

1) Neural Network: The convolutional neural network model implemented in our proposal for counting objects shows the best results in the application of counting people [6] [8]. The chosen model is the Supervised Spatial Divide-and-Conquer for Object Counting model (SS-DCNet) because it has been reported to produce low errors [9] and the applicability of the model beyond counting people has already been assessed: for counting vehicles [10], and grains of corn [11]. Thus, it is expected to be adaptable to alternative datasets too.

SS-DCNet learns from a closed set of counts and it generalizes to scenarios with open sets. This model was designed to approach the problem that only finite local patterns (a closed set) can be observed, but new scenes in the reality have a high probability of containing out of range objects (an open set). Specifically, SS-DCNet (see Figure 3) uses a 16-layer deep neural network (VGG16) as encoder and a Convolutional Networks for Biomedical Image Segmentation (UNet) like decoder to generate multi-resolution feature maps in frames of 64×64 pixels. All feature maps share the same counter, in

 ISBN: 978-1-68558-057-5

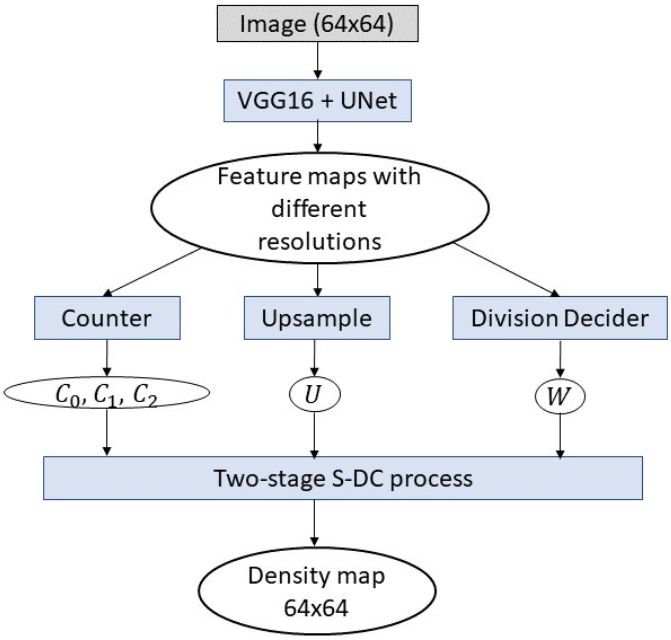


Figure 3. Diagram of SS-DCNet algorithm. C_0, C_1 y C_2 are the estimation counters for three different resolutions; the parameters U y W allows to combine the values of estimation counters to obtain the density map.

these are obtained C_0, C_1 y C_2 for three different resolutions. Then is applied two-stage spatial divide and conquer (S-DC) process to estimate the density map related to sub-image selected. The density map is used to calculate the local count. The final count of the image can be recovered by combining all sub-image counts into one count map with the same size as the test image. For each pixel, a normalization step is performed by dividing the number of sub-images that yield a prediction for the pixel [9]. In local counter modeling, one of the ways to define a counter in the closed set is [0, $C_{\max}$]. In practice, $C_{\max}$ should not be larger than the maximum local count observed in the training set. If the predicted counts are greater than $C_{\max}$, the predictions are simply truncated to $C_{\max}$.

Although the authors of the SS-DCNet model [9] have published source code to evaluate the accuracy of their model, their implementation only has the ability to evaluate an already trained model and does not have routines to train a model with a specific dataset. For this reason, the basic source code used in this project is that published by Dmitry Burdeiny [12] on the Github platform as free code. The code has been adapted to meet the design specifications and to make it compatible with the current dataset.

Analyzing the distribution of objects on 64×64 squares, it is observed in Table I that the 95$^{\text{th}}$ percentile corresponds to the value of 5 turbots per square. Therefore, following the recommendations of the model developers, a value of 5 was chosen as a starting point for model training. However, tests were performed with the lower and upper values to analyze their variation.

TABLE I. PERCENTILES OF TURBOTS COUNTED IN FRAMES OF 64×64 PIXELS

Percentil	Value
65	1
75	2
85	3
95	5

2) Density map: Density maps in SS-DCNet are generated using a Gaussian kernel. The density estimation based approach uses an adaptive geometric density mapping system. This implies that the standard deviation (σ) is calculated dynamically for each labeled point. This value is usually calculated as the product of the mean distance to nearest neighbors and a mitigation coefficient, usually 0.3 [13]. However, the adaptive calculation of the standard deviation is applied to images where the size of the object is evenly distributed among different image regions. For example, an image of a street where people's heads have similar size means that they are in the same image region (foreground, background, other). However, in our dataset, the turbot size is not distributed across the image regions, the size varies mainly with distance to the water surface. Turbots closer to the surface are larger than those in the depth, therefore neighbors in the same region can be in different planes. For this reason, a fixed standard deviation was chosen to create the density maps.

To measure how the value of σ affects the accuracy of the model, the density map was created with different values of σ between 3 and 15 in intervals of three, all with a kernel size of 30 pixels, as shown in Figure 4.

In Figure 4 can be seen that when the parameter σ is increased, the algorithm detects objects where there are none, while at a low sigma of 3 it detects fewer objects.

3) Train and validation test: A random division of the training dataset is made to apply double cross validation: 90% of images for training and 10% for validation. Note that this validation is different from the final evaluation of the error on the test set. The goal of this evaluation is to check during training the evolution of accuracy after certain training iterations.

Moreover, the technique of data augmentation or artificial data generation is used to obtain a larger number of training instances. The strategy followed is to generate nine sub-images with a quarter of the total image resolution, as in [9] [14]. Four sub-images are drawn from the four corners without overlapping, and the other five are drawn randomly from the image. These images need to be normalized, so the average pixel value was calculated for each RGB channel using all images set. The calculated average pixel subtracted from pixels of each RGB channel, and then divided by 255 was the normalization process implemented.

The Stochastic Gradient Descent (SGD) optimization algorithm is chosen as the learning algorithm of the model. The implementation uses an initial learning rate of 0.0001, which is divided by a factor of 10 for each iteration of

 ISBN: 978-1-68558-057-5

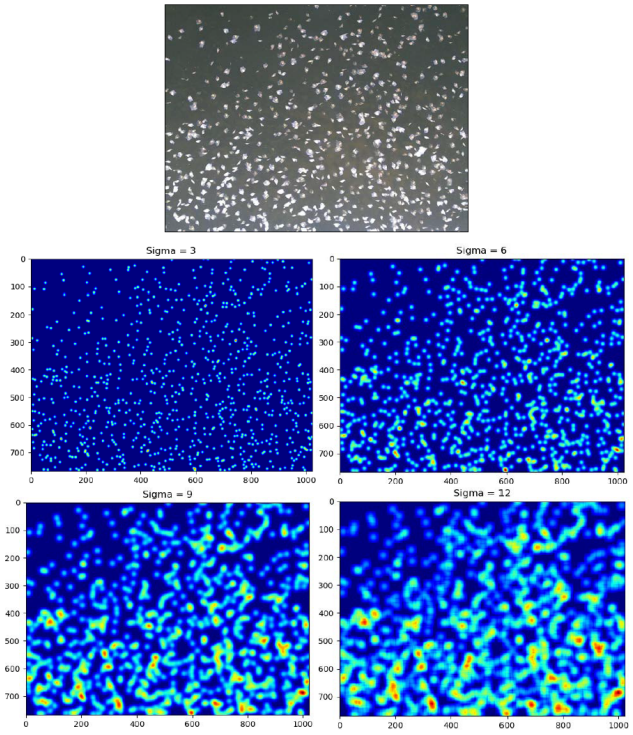


Figure 4. Examples of density maps for different values of σ. The original frame (top), maps with σ equal to 3 (center-left), 6 (center-right), 9 (down-left) and 12 (down-right)

the training process. A random Gaussian initialization with a standard deviation of 0.01 is used to compute the weights. The convolutional neural network is pre-trained with the ImageNet dataset and the batch size is equal to 1 in our proposal.

In addition, the following techniques are used to improve the SGD optimization algorithm:

- Momentum: It is used to reduce excessive fluctuations in the weight changes in successive iterations and thus improve the learning rate [15]. The value used for this parameter is 0.9.
- Weight decay: This is a regularization technique whose main goal is to avoid overfitting that would affect generalization for new data. This technique introduces a penalty in the cost function to reduce the weights during the backward propagation of the error [16]. The value used for this parameter is 10^{-4}.

III. Results and Discussion

In order to obtain the optimal parameters for the generation of the density maps and $C_{\max}$ of the classifier, experiments began with σ equal to 12 and $C_{\max}$ equal to 5. The impact of these parameters was analysed training the system with their extreme values to appreciate the change of these parameters.

1) Relationship between σ and density map: In order to evaluate how the choice of σ for the Gaussian kernel affects the accuracy of the model when generating density maps, it was trained with a value of $C_{\max}$ equal to 5 and the density maps were generated for different σ values, between 3 and 15 in steps of three. As can be seen in Table II, there is no significant effect on the model errors at small standard deviations.

TABLE II. ERRORS OBTAINED BY DIFFERENT DENSITY MAPS

σ	MAE	RMSE	MAPE (%)
3	9.00	18.82	3.52
6	11.66	19.46	4.04
9	11.05	19.22	3.56
12	9.66	18.20	3.48
15	10.62	18.09	3.69

A value of 12 was used for σ to create the density maps for the rest of tests. Although it has a slightly worse Mean Absolute Error (MAE) value than the map created with a $\sigma = 3$, the Mean Square Error (MSE) and Mean Absolute Percentage Error (MAPE) values are better and the deviations are therefore more homogeneous. The Root Mean Square Error (RMSE) is similar in all cases.

2) Selection of $C_{\max}$ value: The developers of the SS-DCNet model obtained the best model accuracy results for a $C_{\max}$ value corresponding to the 95$^{\text{th}}$ percentile of the objects distribution in 64×64 pixels. This value is 5 for the current dataset. The validity of that conclusion was verified training the model with $C_{\max}$ values below and above 5.

As can be seen in Table III, for $C_{\max} = 5$, the smallest errors are obtained for both MAE and MAPE. However, for $C_{\max} = 6$, the RMSE is slightly smaller, meaning that there is less variation. Nevertheless, the difference between MAE and MAPE is considered to be more significant than RMSE, so a value of $C_{\max}$ equal to 5 is used for the further tests.

TABLE III. ERRORS OBTAINED BY DIFFERENT $C_{\max}$

$C_{\max}$	MAE	RMSE	MAPE (%)
2	14.45	25.98	3.90
3	15.02	27.26	4.13
4	14.67	26.49	4.09
5	9.66	18.20	3.48
6	10.39	18.05	3.64

3) Generalization capability / ability: In order to evaluate how the model generalizes for frames with higher concentration of turbot larvae, it was re-trained with images that had a low density of individuals, less than 350 per frame, and tested with images that had a high density, between 350 and 898 individuals. For this experiment, 129 and 27 images were used for training and testing, respectively.

Figure 5 shows a low deviation for predictions in test images. Therefore, the model maintains an acceptable accuracy for images with a higher density and number of objects than that of the training set.

 ISBN: 978-1-68558-057-5

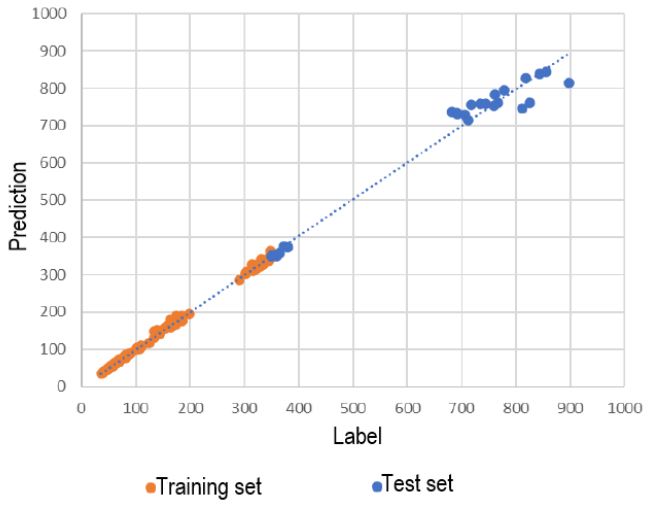


Figure 5. Generalization capability of SS-DCNet. The *Label* and *Prediction* axes represent the real and estimated number of turbots in a tank, respectively. The orange dots are the data of training set and the blue dots are the new data with high turbots density. The broken line shows the ideal estimation model

IV. Conclusions

Applying a convolutional neural network model to count turbot larvae in breeding tanks from images yields a mean error lower than 3.5%, which is acceptable accuracy for this task. Adaptation of the model to count other fish species or turbot at other growth stages is feasible, as it is not necessary to use large datasets for training. The evaluated model exhibits a remarkable generalization ability, providing good counting estimates even when the density and total number of objects in test images is much larger than in the training images.

While the characteristics of the dataset used do not allow the application of the adaptive geometry strategies used in people counting, other strategies for creating the density maps can be explored, such as adjusting the value of σ for each labeled point based on the morphological features extracted during the label segmentation process.

While using a pre-trained VGG16 encoding network helps in reducing the need for a large training dataset, it is possible that training the encoder from scratch with application specific images could improve accuracy, as there may be few or no images about larval turbot in the ImageNet dataset with that the encoder was pre-trained, despite its large expansion of images and categories.

Acknowledgment

This work has been funded by Ministerio de Agricultura, Pesca y Alimentación, Plan de Recuperación, Transformación y Resiliencia, NextGenerationEU. Project: *Aplicación de tecnologías de visión e inteligencia artificial a la mejora del proceso productivo (Acuicultura 4.0)*

References

[1] D. Li, Y. Hao, and Y. Duan, "Nonintrusive methods for biomass estimation in aquaculture with emphasis on fish: a review," *Reviews in Aquaculture*, vol. 12, no. 3, pp. 1390–1411, 2020.

[2] A. Najah Ahmed, F. Binti Othman, H. Abdulmohsin Afan, R. Khaleel Ibrahim, C. Ming Fai, M. Shabbir Hossain, M. Ehteram, and A. Elshafie, "Machine learning methods for better water quality prediction," *Journal of Hydrology*, vol. 578, p. 124084, 2019.

[3] V. Kandimalla, M. Richard, F. Smith, J. Quirion, L. Torgo, and C. Whidden, "Automated detection, classification and counting of fish in fish passages with deep learning," in *Frontiers in Marine Science*, 2022.

[4] M. S. Ahmed, T. T. Aurpa, and M. A. K. Azad, "Fish disease detection using image based machine learning technique in aquaculture," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, Part A, pp. 5170–5182, 2022.

[5] N. Abinaya, D. Susan, and R. K. Sidharthan, "Deep learning-based segmental analysis of fish for biomass estimation in an occulted environment," *Computers and Electronics in Agriculture*, vol. 197, p. 106985, 2022.

[6] B. Li, H. Huang, A. Zhang, P. Liu, and C. Liu, "Approaches on crowd counting and density estimation: a review," *Pattern Analysis and Applications*, vol. 24, no. 3, pp. 853–874, 2021.

[7] R. Perko, M. Klopschitz, A. Almer, and P. M. Roth, "Critical aspects of person counting and density estimation," *Journal of Imaging*, vol. 7, no. 2, 2021.

[8] W. Li, Z. Fangbo, and H. Zhao, "Crowd density estimation based on global reasoning," *Journal of Robotics, Networking and Artificial Life*, vol. 7, no. 4, pp. 279–283, 2021.

[9] H. Xiong, H. Lu, C. Liu, L. Liu, C. Shen, and Z. Cao, "From open set to closed set: Supervised spatial divide-and-conquer for object counting," *ArXiv*, vol. abs/2001.01886, 2020.

[10] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascón, and D. Oñoro-Rubio, "Extremely overlapping vehicle counting," in *Pattern Recognition and Image Analysis* (R. Paredes, J. S. Cardoso, and X. M. Pardo, eds.), (Cham), pp. 423–431, Springer International Publishing, 2015.

[11] H. Lu, Z. Cao, Y. Xiao, B. Zhuang, and C. Shen, "Tasselnet: counting maize tassels in the wild via local counts regression network," *Plant Methods*, vol. 13, no. 79, 2017.

[12] B. Dmitry, "Unofficial pytorch implementation of s-dcnet and ss-dcnet." https://github.com/dmburd/S-DCNet, 2020. (Accessed on 13/02/2023).

[13] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 589–597, 2016.

[14] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1091–1100, 2018.

[15] H. Shi, N. Yang, H. Tang, and X. Yang, "asgd: Stochastic gradient descent with adaptive batch size for every parameter," *Mathematics*, vol. 10, no. 6, 2022.

[16] H. Tessier, V. Gripon, M. Léonardon, M. Arzel, T. Hannagan, and D. Bertrand, "Rethinking weight decay for efficient neural network pruning," *Journal of Imaging*, vol. 8, p. 64, 03 2022.

 ISBN: 978-1-68558-057-5