TÉCNICO LISBOA

Computer Vision Based Approach to Mackerel Population Counting from Single Images

Diogo Costa Ravasco

Computer Science and Engineering

Supervisors: Prof. Alexandre José Malheiro Bernardino
Prof. Helena Sofia Andrade Nunes Pereira Pinto

October 2024

Acknowledgments

Abstract

In an oceanarium context, the health of the ecosystem is paramount.
A litmus test that is employed to monitor the environment is the stability of its housed species, one of them being mackerel. To do so at different points throughout the year staff will capture photographs of the school and perform manual counts. This gives a view of the evolution of the population and a window into the overall well-being of the tank, but is a arduous, time consuming, and often difficult task.
Recently computer vision approaches to crowd counting have shown impressive results when it pertains to human settings, with its use becoming widespread in riot and protest events. Its application to other domains, however, remains relatively unexplored.
In this document we present a dataset, curated by us, of mackerel schools which habitate the main tank in Oceanário de Lisboa. And make use of it to train a variety of patch-level regression methods and a CLIP based model that performs blockwise classifications.
Current regression methods, however, use ground-truth generation methods which may be erroneous given our domain setting. As such in addition to the aforementioned contributions we also propose a novel ground-truth generation method that we believe is better suited to this specific problem.
Our final proposed solution is able to accurately estimate mackerel population within a mean error of six percent.

Keywords

Fish; Mackerel; Dataset; Crowd Counting; Crowd Localization; Machine Learning; Deep Learning;

Resumo

No contexto de um oceanário, a saúde do ecossistema é essencial.
Um dos indicadores empregado, que permite monitorizar o ambiente. é a estabilidade populacional das espécies alojadas, entre elas um exemplo é o das cavalas. Para efetuar este teste ao longo do ano várias fotografias são capturadas e funcionários do oceanário efetuam contagens manuais. Isto permite avaliar a evolução da população e possibilita visualizar o bem-estar do ambiente como um todo. É no entanto uma tarefa díficil, demorada e frequentemente erroneosa.
Recentemente avanços em técnicas de visão computacional permitem em domínios humanos efetuar contagens bastante fidedignas. Tendo a sua adoção em manifestações e protestos sido difundida. A sua aplicação em outros domínios é embora um sujeito de pouca exploração.
Neste documento apresentamos um conjunto de dados, coletado por nós, de cardumes de cavalas que habitam o tanque principal do Ocenário de Lisboa. E servimo-nos dele para treinar um conjunto de modelos que efetuam regressão ao nível do pixel e um modelo baseado na arquitetura CLIP que efetua classificação da imagem por blocos.
Para além das contribuições mencionadas préviamente, neste documento reconhecemos que o processo de geração de dados de raíz não se traduz diretamente do domínio humano para o domínio marinho e propomos uma nova metodologia que acreditamos ser mais fidedigna e introdutora de menos ruído.
A solução final proposta é capaz de estimar o tamanho de populações de cavalas com um erro médio de seis porcento.

Palavras Chave

Peixes; Cavalas; Dataset; Contagem de População; Localização de População; Machine Learning; Deep Learning;

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Contents

1.1 Motivation

Oceanário de Lisboa opened in the World Exhibition of '98, is the single most popular cultural attraction in Lisbon. Its importance nationally and internationally as an institution that strives to educate the public about the oceans, marine life, and current environmental issues can not be understated.

Not only important as a social and cultural landmark. The oceanarium plays a major role in allowing marine biologists and ocean conservationists to study aquatic species in a controlled environment distinct from the wild where species are too sparsely distributed to be reliably observed and tracked. Achieving these goals requires maintaining a healthy and thriving ecosystem, which is a complex, difficult task that requires constant monitoring on the part of biologists. One of the observations undertaken that serves as a litmus test for the overall health of each tank is the population stability of its housed species. Such a task is trivial when it comes to some solitary species that do not exist in too great of a number, such as sharks, rays, devil fish, among others, but it does present some challenges in species that exist in abundance and group in tightly packed schools.

Current methods of counting mackerel populations rely on taking stills of the school at specially chosen perspectives so that the maximum possible amount of fish is visible, and then having two or more persons count each fish. Though time-consuming this task must be performed regularly. Keeping a record of the evolution of a species' population is crucial to determine if the current allocation of food per individual is sufficient, and if any other factors may be adversely impacting the regular and proper development of the school, such as predation or disease, or on the contrary if the population is growing past a sustainable level.

Using current research in crowd counting, the various photographs of the school can instead be analyzed automatically, sparing staff of this task and freeing them to focus on others which are of more use to the oceanarium.

1.2 Project Scope and Challenges

Safety in numbers is a survival strategy that is adhered to by many species in nature. In the case of the mackerels, schooling is a strategy that offers many benefits to the individual fish. It means increased success in finding food, access to potential mates, increased protection from predators, and also presents hydrodynamic benefits [24].

Much like human crowds, when fish group into schools they become almost like a superorganism with a collective will, responding to threats, stimuli and displaying behaviors that are not necessarily dependent on any individual fish. Though beneficial to mackerel as a whole and as individuals, both schooling behavior and the fact our images are of an underwater environment that tries to mimic a real wildlife scenario presents many complications to computer vision tasks.

Schools of fish are not homogeneous masses, some areas may display a higher amount of fish while others are more sparse. This presents an issue when counting the population, as some regions are so densely packed that it may be hard to recognize individual fish while others are sparse enough to conduct a simple head count. As well as being non-homogeneous, schools have dimension. This rather obvious fact does in fact impair computer vision tasks as it introduces scale variability. Human vision is binocular meaning each eye will have a slightly different view and this disparity introduced by the eyes' horizontal parallax gives us depth perception. The brain then accounts for this and estimates the actual size of the fish. Photographs returned from our setup up however are monocular and fish that are more distant from the camera will appear smaller in size to fish that are closer.

Lastly, the closeness of the fish creates an additional obstacle, as fish overlay on top of each other those above will cause shadows to be cast on the ones below, this illumination change makes them appear different between themselves, as they effectively display different coloring.

Besides problems introduced from the schooling structure, some are introduced by the fact the photos are taken underwater. The mackerel are housed in the main and largest tank within the oceanarium which has a single overhead illumination source. As light travels through the water it progressively loses its higher wavelength colors, meaning the further the light has to traverse the more "blue-shifted" it will appear. This impacts images since more distant and sunken fish look differently from fish that are closer to the camera and the surface, causing very distant fish close to the bottom to blend almost entirely into the background.

Besides housing the mackerel many other species of fish are found in the main tank which share some similarities, like color, shape and size, making it harder to distinguish which fish we are currently observing and whether they belong to the school. Another issue that may make it harder to recognize a mackerel is the layout of rocks and plant life within the tank. This is meant to both provide cover and camouflage opportunities, meaning our objects of interest may be partly or completely occluded or blended into the background.

The challenges stated in this section present significant hurdles to the crowd counting problem but are also common in the usual setting where this task is performed. Most scenes where this problem is tackled are of human crowds, and through 3D, the bird's-eye view of the camera constructs the problem as a 2D plane of heads. Fish present a more challenging proposition. Their distribution cannot be mapped by a plane, there is no homography that can directly translate their location to the image plane. As such the proximity of two fish in the image plane, has no implication on the actual real-world distance they share between themselves, or the camera. The importance of this fact in usual human counting is cleared in section 2.3. Lastly, another problem that further differentiates this problem from usual crowd-counting tasks is the shape of our targets. Usually, the shape of target objects is circle-like or can be approximated by one. However, fish are oblate and their orientation is not axiomatic.

These two problems, of shape and loss of spatial information between fish, are those that most distinguish our work from the usual research focusing on human crowd counting.

1.3 Contributions

In this work, we develop a system capable of localizing and counting mackerels. Development which, due to the lack of publicly available datasets, mandated the creation of our very own. This dataset was made possible through the assistance of Oceanário de Lisboa who permitted the capturing of images within their facility. The final curated and annotated dataset is, to the best of our knowledge, the only existing crowd-counting dataset targeting schools of mackerel.

We also devised a new ground-truth generation method that is specific to the domain of schooling fish. Wherein we build upon the knowledge of how schools, due to their very nature, exhibit a structured formation and therefore give us the ability to infer information about individual fishes from their neighbors.

A comparative analysis is also presented, both between different ground truth generation methods - including our own - and between several different architectures and crowd-counting paradigms. To evaluate these models standard benchmark metrics are used alongside additional ones which are crucial for understanding their performance on our specific dataset. This will help us compare the models not only against each other but also in terms of their overall suitability for our needs.

Lastly, seeing as we envision this project as a foundation for future works, such as flow analysis or anomaly detection, we thought fit to also evaluate our solution's capacity to not only count but to localize mackerel.

1.4 Organization of the Document

This document is organized as follows. Chapter 2 focuses on the presentation of previous research in crowd counting. It goes from different labelling techniques, through traditional methods to the current state-of-the-art (SoTA) deep learning methods. We give special focus to three different areas we believe are the most promising or novel in regards to this, the crowd-counting task. Chapter 4 presents an overview of our chosen solution. It goes through ground-truth generation, typical solutions, and our new approach which aims to exploit school behavior and impart this information into the selected model's training process. Chapter 3 presents our created dataset. Which we believe is the first of its kind in our target domain of mackerels as it pertains to crowd-counting. We explain the pitfalls of previous collections and how we designed our collection setup to fix the observed limitations. Followed by an overview of our dataset statistics alongside an explanation of our newly designed annotation, the vector through which school information is captured. In Chapter 5 we present our training procedure. We go

over the count and localization metrics we calculate and how each model fared concerning these metrics. Due to computing limitations, a fairer comparative analysis is also performed, but only between a smaller subset of the trained models. And lastly, to close the document we present the limitations of our solution and how different directions for future work could further improve our system, in Chapter 6.

Chapter 2

Related Work

Contents

This chapter serves the purpose of reviewing state-of-the-art solutions in the field of crowd counting.

Crowd counting as a problem emerges in many different subjects, that span more than purely human crowds, work as been developed with the objective of counting vehicles, plants and some as been developed with fish in mind, mainly in the aquaculture industries [15, 18, 39, 58].

This section will include a review of current data labelling techniques, traditional crowd counting methods and after, a more in-depth analysis of current deep learning based methods of crowd counting.

2.1 Data Labelling

Usual crowd counting applications rely on supervised learning or semi-supervised learning methods, this means labeled data has to be provided from which the models can learn.

Different ways of labeling data have been proposed for the purposes of computer vision tasks related with crowds. This choice limits both the models that can be employed and how reliable are the results obtained.

1. **Total count annotation:** This type of annotation is simply a whole number that represents the total count of objects of interest in the image. It is used in detection-based, and regression-based approaches discussed more at length in subsection 2.2. As a label it describes no spatial aspect, carrying no information about the distribution of individuals within the crowd.

2. **Dot annotation:** Each object inside the image is annotated as a single dot. In humans this dot is typically located on the head, as it is the body part that is best detected and in many crowded scenarios severe occlusions mean much of the body has a high likelihood of being hidden. This type of annotation creates a dot-map (localization map), which is a matrix of all zeroes except the positions where the dots are located which are flipped to one. With the introduction of this annotation models now have spatial information about where the target objects are located. This gives models an insight into the distribution of the crowd and therefore lends itself to density estimation-based approaches discussed further in the next section. A drawback of this annotation is the lack of scale, regardless of apparent size a target object will only have a single pixel in the dot-map. The extreme sparsity of the dot-map, having all zeros except for the head positions also makes it very hard to train neural networks.

3. **Bounding box annotation:** A rectangle is drawn around each object in the image. This type of annotation provides both spatial and scale information at the cost of being more expensive to compute.

Most recent research in the area of crowd counting tries to estimate the density of the crowd and how it is distributed in space, generating a density-map. The total count of the crowd is then performed by

integrating the density map. These types of approaches are discussed further in subsection 2.2 and 2.3.

To be able to conduct the training a ground-truth density-map is needed. This is obtained by starting out with a dot-map and passing a Gaussian kernel (G_σ) convolution, where σ is its variance [31]. The resulting effect is a smoothing of the dot-map, instead of jumping from zero to one when at the boundary between nothing and the target object there is now a gradual increase, this type of data is easier to train neural networks with.

The value of a pixel y at location $\mathbf{x}$ in an image with N annotations $\{\tilde{D}_1, ..., \tilde{D}_N\}$ is then given by placing a Gaussian at each annotation:

$$y(\mathbf{x}) = \sum_{i=1}^{N} \mathcal{N}(\mathbf{x}|\tilde{D}_i, \sigma\mathbf{I}) = \sum_i \frac{1}{2\pi\sigma} exp\left(-\frac{1}{2}\|\mathbf{x} - \tilde{D}_i\|^2_{\sigma\mathbf{I}}\right) \tag{2.1}$$

Where $\tilde{D}_i$ is the mean, $\sigma\mathbf{I}$ is the covariance matrix, and $\|\mathbf{x} - \tilde{D}_i\|^2_{\sigma\mathbf{I}}$ as the squared Mahalanobis distance.

Most basic approach is to maintain a fixed σ. This method does not take scale variability into account. Regardless of the distance and size of the object the smoothing effect will be the same for each head position. This will impair the model's ability to learn scale variability in images. To mitigate this issue adaptive kernels were proposed [57]. The adaptive kernel also performs convolutions over the image using a Gaussian kernel, with the added behavior of dynamically altering σ.

Imagining a constant density crowd, as the targets appear smaller due to distance from the camera, so too do the distances between objects. From this we can infer, the more clustered head positions are, the smaller we should expect them to be in the original image. Much like a fixed kernel, this approach will visually blur the dot-map, but instead of constant, dense parts of crowds will be less perturbed, than sparser regions [59].

2.2 Traditional Methods

2.2.1 Detection-based Approaches

Early research efforts, treated crowd counting as a detection problem, where in a sliding window detector is used to detect individuals and the overall crowd count would be a byproduct of those detections.

Typical pedestrian detections are performed in what is called a monolithic style where full body features [16, 43, 49] are extracted and used to train a classifier [30, 34]. Although these approaches are successful when dealing with sparse crowds they fail when confronted with highly dense crowd scenarios where occlusions are more frequent and severe. To mitigate these issues researchers adopted parts-based detection [21, 54] where only specific parts such as the head are detected. Another approach to the problem of monolithic style detection was introduced by Zhao et al. [60] where based on a knowledge of

the camera model and making the assumption that humans only move on a ground plane, several 3D human model hypotheses are proposed and matched to foreground blobs extracted through Background Subtraction, since these human shape hypotheses are in 3D they model the real world, meaning occlusions are inherently built into the system and their reasoning is straightforward.

With the advent of deep learning more robust models were implemented, which use state-of-the-art detectors [42] that outperform traditional detection methods.

2.2.2 Regression-based Approaches

Although efforts were made to design models robust enough to deal with occlusions in dense crowds, by performing part-based detection, shape matching or more recently by employing deep learning detection methods, these models still failed when confronted with the problems of highly dense crowds, high levels of background clutter, occlusions and low resolution. As these present an insurmountable and intrinsic issue with the detection-based framework where a total count estimation is made and not with the detectors themselves [25]. Due to the aforementioned limitations regression-based models were adopted.

Regression-based models create a direct mapping from features extracted from the image to a crowd count. Features such as area and perimeter of foreground blobs are extracted and used to encode information about the crowd. Further work as been done to also use features such as edges, histogram of oriented gradients (HOG), gradient, local binary pattern (LBP) and texture to take into account crowd information at a lower level. The use of holistic features alongside these more local ones has shown improved results [8, 9]. After the extraction of features different regression techniques can be modelled to create the mapping [8, 11].

This approach avoids the need to train a detector, and is more computationally efficient by not having to run the pre-trained detector at several scales.

More recently, Idrees et al. [25] observed that in extremely dense crowds, due to low resolution, perspective and sever occlusions, no single feature or detection method is reliable enough to provide an accurate count.

As stated above texture is a feature that can be extracted from the image to carry information about the crowd, work has been done in the direction of analysing these textures as harmonic signals and applying Fourier Analysis [2, 38]. This analysis however as stated in [25] displays two main limitations, the spatial arrangement is very irregular, i.e. crowds are not homogeneous and it is not useful in localizing repeating elements. Idrees et al. propose a novel idea to combat the issues stated above, where different methods are combined to capture different information, namely Fourier analysis, head detection and interest points [25].

2.2.3 Density Estimation-based Approaches

Regression-based methods deal with the problems of crowd counting, namely low resolution, severe occlusions, perspective, background clutter, etc., better than detection-based approaches, however they fail to display spatial information as they regress a global count. Crowds however are bound by constraints like bottle necking, physical boundaries, points of attraction and even actions at the level of the individuals, as such, their distribution is not homogeneous. Some points may display higher density while others may have very few persons. In many applications like crowd analysis this type of information is valuable.

A solution was proposed by Lempitsky et al. to create a linear mapping between local image patch features and their respective density maps, the total count of objects within a region will then be the integral of the learnt map within that corresponding region, making spatial information an intrinsic part of the process [31]. This approach means there is no need for individual detection to still maintain information about their distribution.

Later Pham et al. recognizing that a linear mapping is a hard learning task propose a non-linear map between local patch features and density maps. Random forest regression is used to vote for densities of multiple target objects to learn non-linear mapping from multiple image patches. The authors also observe the large discrepancy between crowded and non-crowded image patches by suggesting a crowdedness prior and training different random forests corresponding to this prior [40].

Further advancements were made to improve computational efficiency [53] and to use a larger set of features as was done in other computer vision subjects, namely face recognition [55].

2.3 Deep Learning

Like many other computer vision fields, such as, object detection, classification and segmentation, most of crowd counting's recent research has focused in on convolutional neural networks (CNN) for their performance in automatic feature extraction. First introduced by Zhang et al. density estimation using CNNs as been the bedrock of all following research [57]. Their use has been cemented, as they outperform all traditional methods, but different architectures have been proposed to best mitigate usual crowd data problems such as scale variation, high density, severe occlusions, perspective, illumination inconsistencies, to name a few.

The CrowdCNN model proposed by Zhang et al. would require the construct of a perspective map to carry information about scale variations due to perspective distortions. This map is used to dynamically size image patches so they cover a $3 \times 3m$ square in the actual scene, meaning they maintain a similar scale. The pathces then serve as input of the model which iteratively switches between two objectives, predicting a total count and estimating a density map. These are related and help each other obtain better local optima. So the model is capable of performing cross-scene counting, it first has to be fine-tuned.

This is done through training epochs with images that share similar density distributions and viewing angle to that of the target scene. The latter restraint requires a perspective map of the unseen scene, and the former requires a rough estimation of the target scene's density map. Performed by counting people in each image patch through the non-tuned model. After selecting similar images the model is fine-tuned and the target scene is reevaluated.

In [59] Zhang et al. propose a multi-column neural network (MCNN), first introduced in the field of image classification [14]. This as opposed to the previous model does not require perspective information and performs whole-image inference. Mitigating the issue of needing apriori geometric information about the scene and the issue of image distortion due to the resizing the CrowdCNN model performed. Each branch possesses differently sized filters, that carry semantic information referent to different scales. The final prediction of the whole network is then be the average of each individual branch, i.e. a fusion of features at different scales. This work also marks the introduction of adaptive Gaussian kernels to the density map generation process which better approximates the scale variations of individuals.

This type of architecture does incur the drawback of computation overhead, SwitchCNN is another model that proposes a multi-column network with 3 columns, but instead of concatenating the result from each branch, a CNN denominated as classifier is used to select only the output of one column based on the crowd density on inference time [3].

Other models build on this multi-column paradigm creating multi-task networks where each column is responsible for learning different tasks. One such model is introduced by Gao et al. The authors' proposed model has two fully convolutional branches. The first is responsible for density map estimation (DME) and for performing foreground/background segmentation (FBS). The second performs high-level classification over random patches of the image (Random High-Level Density Classification R-HDC), classifying them over 10 labels that relate to that patch's density level [22]. The density map column is responsible for low-level feature extraction and the R-HDC column is related with extracting global contextual information, which the DME column overlooks. Finally the foreground/background segmentation helps alleviate the error that emerges during the generation of density maps. It is expected for there to be variability between the size of heads and the kernel, meaning the highlighted areas in the ground-truth density map do not contain all contextual information. Since both the DME and FBS tasks share a base network the minimization of the calculated loss of the FBS task helps the DME task learn more semantic information.

Gradient boosting machine (GBM) is a machine learning technique where the prediction of a model is the result of an ensemble of weaker prediction models and its performance is boosted by sequentially adding models to the ensemble that optimize any differentiable loss function, learning to map the error of the previous ensemble. Walach et al. introduce the concept of gradient boosting of CNNs to the crowd counting problem [50]. In their work at each step a new CNN model is fitted to the error of the last round and the ensemble is updated accordingly. The output of the network is then the result of a sum

layer that takes all models in the ensemble into consideration. This method was used on a data set of bacterial microscopy images [31] and the UCSD crowd counting benchmark [8]. Though their model was outperformed by a twice as deep network they did show that boosting significantly outperforms increase in the network depth, as a network three times as deep performed worse than their original model. The authors also compare their method to an ensemble of CNNs with different starting points and again boosting outperforms this approach.

In [32] Li et al. demonstrate that the MCNN has a non-effective branch structure that requires a large amount of training time, and has no better performance compared to deeper regular networks. Demonstrated is also the counter intended effect of the MCNN where each branch in the network finds itself learning nearly identical features as the others. This means the model where each column is meant to learn different semantic information, becomes redundant. In their work the authors propose a deeper network CSRNet. Following other works [6] the model employs transfer learning, choosing the first 10 layers of VGG-16 [44] as its front-end, this choice was due to its architecture, whose flexibility allows to easily concatenate the back-end for the generation of the density map. Due to the convolutional and pooling layers, the output of the front-end is one eighth of the original input. To address this issue the authors employ dilated convolutional layers [56] as the back-end to maintain the output resolution as well as to extract deeper information. Dilated convolutions have been demonstrated to significantly improve accuracy in other computer vision tasks [56]. These layers have the benefit of not reducing the spatial resolution, meaning no contextual information of the feature map is lost. By enlarging the receptive field without requiring more parameters an aggregation of multi-scale semantic information is performed, without shrinking the input, incurring higher computational costs or increasing the complexity of the model. Different dilation strides can be chosen to capture information at different scales.

Chen et al. propose a novel Scale Pyramid Network (SPN) architecture that employ dilated convolutions in parallel, each with different dilation strides [12]. As stated above these allow the construction of multiple receptive fields, without the need for extra parameters or computation. The architecture is made up of a simple single deep column structure, using the VGG-16 [44] as backbone. The fully connected layers are removed to provide input flexibility and the last layer is a 1×1 convolution to map the density map. Between the 4^{th} and 5^{th} layer the authors place their Scale Pyramid Module. The dilated convolutions in this module are performed in parallel with dilated rates of 2, 4, 8, 12 all having the same number of channels as the input features. Each dilated convolution is responsible for extracting high level features at different scales. These are then merged, through addition or concatenation, and fed to the following convolutional layer. Both strategies of merging obtain similar MAE, but concatenation manages lower MSE scores.

In [36] Liu et al. propose a novel method that does not require perspective/geometric information, and assumes that scale changes are continuous and not different discrete bins. Their, CAN/ECAN,

model extracts features at multiple scales and learns to adaptively combine them maintaining a sense of continuous scale. As opposed to having the density map at a certain scale solely dependent on features at that scale. This model uses the VGG-16 [44] as backbone, but to combat its constant receptive field during the feature extraction process the authors use Spatial Pyramid Pooling [23]. Spatial Pyramid Pooling is computed by dividing the VGG [44] computed feature maps into fixed amount of bins. Namely 1×1 blocks, i.e. the whole feature map, 2×2 blocks, 3×3 blocks, 6×6 blocks, and then performing average pooling on each individual bin. This allows the capture of multi-scale properties, but instead of concatenating these different levels of resolution, in effect discretizing the continuously varying scale of objects in the scene, the authors propose learnt weights that set the relative importance of each scale at different spatial locations. Meaning the architecture models the smooth transitions of scale.

In [28] Jiang et al. introduced a new trellis encoder-decoder network (TEDNet). Most state-of-the-art methods employ backbones such as the aforementioned VGG-16 [44], these backbones were originally designed for classification, their deep architectures take low-level features and use them to construct high-level semantic information. The authors note however that this process lowers the resolution of the feature maps and ultimately the accuracy of spatial information is lost. This type of architecture is compared to an hourglass, where the low-level spatial information and high-level semantic information that exist at the two ends of the architecture are separated by a gap. To mitigate this issue some networks have employed skip connections, but the authors note these lack hierarchical fusion of multi-scale features making them non-optimal for the purposes of density map generation. Their own introduced architecture has two different stages the encoding and decoding.

1. **Encoding:** This process, like other models employs multi-scale convolution kernels, to learn features at varying sizes.

2. **Decoding:** This stage resembles an ensemble of the aforementioned hourglass networks. At different encoding stages a decoding path is instantiated that aggregates the encoded features. To fuse features and integrate spatial and semantic information across scales dense skip connections are established. Each of these decoding paths have an intermediate output allowing distribution of supervision across the network and boosting the gradient flow.

Another novel proposal from the authors was the use of a combinatorial loss, composed by *Spatial Abstraction Loss (SAL)* and *Spatial Correlation Loss (SCL)*. These allow to capture the real world axiom that pixels in an image share a relation and are not independent. Local coherence and spatial correlation are integrated in the final density map. SAL computes the squared error loss across abstraction levels between the predicted and ground-truth map. Abstraction levels which are the result of progressively applying max pooling layers with down-sampling strides.

$$L_{SA} = \sum_{k=1}^{K} \frac{1}{N_k} ||\phi_k(\hat{Y}) - \phi_k(Y)||_2^2 \tag{2.2}$$

Where ϕ_k represents the k-th abstraction level and N_k the number of pixels for the corresponding density map.

SCL complements the SAL's patch-wise supervision by representing the difference between two density maps based on normalized cross correlation similarity.

$$L_{SC} = 1 - \frac{\sum_p^P (\hat{Y}_p \cdot Y_p)}{\sqrt{\sum_p^P (\hat{Y}_p^2) \cdot \sum_p^P (Y_p^2)}} \tag{2.3}$$

Where, P is the pixels in the density map. Final loss is then, $L = L_{SA+SC}$.

Lei et al. propose the first model that makes use of full supervision (dot annotations) for a small number of samples and what the authors call "weak supervision" (count annotation) for the remaining data is Multiple Auxiliary Tasks Training (MATT) [29]. The weakly-supervised training section of the model looks at the integral of the predicted density map and the ground-truth total count. Simply observing the error between the two holds a very weak constraint on the map generation process. To overcome this the model has multiple auxiliary branches, only trained through weak supervision. Each is a density map regressor, with the constraint that each estimates a map with varying smoothing levels. Meaning all maps are equivalent, but distinct. This constraint prevents the convergence of the branches and therefore their redundancy. The method requires the extracted features to be able to support the generation of multiple diverse density maps, thereby enforcing that the feature extraction process encodes spatial information.

More recently, due to their self-attention mechanism, transformers have been employed to automatically learn semantic information through, again what is called in this task's domain, weak supervision. Most recent application of a transformer to the crowd counting problem space was introduced by Liang et al. [33]. This approach has the added benefit of having a global receptive field, meaning it relates low-level information between image locations, a mechanism which had to be carefully considered and constructed in a typical CNN architecture.

A transformer architecture takes a sequence of feature embeddings $E \in \mathbb{R}^{N \times D}$, N being the length of the sequence and D being the dimensionality of the input channel.

This means the transformation of the input. Each image is divided into a grid of N patches with size $K \times K$, resulting in $X = \left[x^{(1)}, ..., x^{(N)}\right]^\intercal$, where $x^{(i)} \in \mathbb{R}^{K^2 \times 3}$, assuming 3 channels for RGB. After dividing the images a mapping has to be made from X into a D-dimensional embedding feature through a projection L that can be learned.

This latent variable displays a lower dimensionality than that of X. This projection however does not capture the positional encodings of the data, therefore the model would lose spatial information. Taking

all this into account the final feature embeddings are as follows:

$$E = X \times L + P = \left[x^{(1)}L + p_1, ..., x^{(N)}L + p_N\right] \tag{2.4}$$

These serve as input of the transformer encoder. That contains L layers of multi-head self-attention and multi-layer perceptron blocks. Each layer contains layer normalization and residual connections, a type of skip connection that allows gradients to flow and the model to converge faster.

The authors propose two different models TransCrowd-GPA and TransCrowd-Token, these are similar until this point only differing in the initial embeddings and what is fed into the regressor.

In the TransCrowd-Token model a learnable embedding named regression token is prepended to the input feature embeddings. This much like the CLS token in NLP-centered transformers [17] is a token that captures semantic crowd information. This is due to the self-attention mechanism spreading information between the regression token and patch tokens.

In the TransCrowd-GPA model, global average pooling is applied to shrink the feature embeddings dimensionality, while still maintaining useful crowd information in patch tokens.

The regressor employed in both methods is the same, but the authors note that pooling achieves better results than the addition of the regression token.

This model marks the first deep learning crowd counting architecture that learns from purely count annotations, and showcases competitive results when compared with fully supervised state-of-the-art methods.

2.3.1 Gaussian Kernels and Relaxing Spatial Invariance

Previous work, some of which was overviewed before in this section, believes that the key to crowd counting is spatial invariance, i.e applying the same filter bank to input patches at all locations. This is a safe assumption as locally connected layers which are identical to convolutional layers except in regard of their spatial invariance typically perform poorly in practice, while convolutional neural networks are among the most successful [4]. Locally connected layers which should be able to converge on the convolution solution, tend to overfit the data.

This direction of research makes the rather sensible assumption that regions in an image are related and it should be useful to apply the same filters on neighbouring regions.

These approaches however reached bottlenecks in performance, Elsayed et al. found that this is a result of too strict pixel-level spatial invariance causing overfitting to annotation noise, and therefore not allowing the model to properly generalize [20]. To address this problem a solution is proposed where in the feature extraction process tries to mimick the density map generation process [13] relaxing the level of spatial variance which may be a better inductive bias [20].

Explained in subsection 2.1, ground-truth density maps are generated through the smoothing of a

dot-map with an adaptive Gaussian kernel. This is a necessary process that turns a discrete counting problem in to a continuous regression problem, but it does also incur two different types of noise [51].

1. Error between the annotated center point and actual center point, ϵ.

2. The overlay of two different Gaussian kernels when objects are occluded.

Wan et al. supposes that the labelling error ϵ is independent and identically distributed and follows a Gaussian distribution [51]. Their work attempts to model the noise introduced during the process of density map generation.

The map generation process remains identical, but for the fact that the terms of the Gaussian distribution now take the expected value of the error and its variance as factors. The value of a pixel in density map at location $\mathbf{x}$ is given by:

$$\Phi = \sum_{i=1}^{N} \mathcal{N}(\mathbf{x} \mid \mathbf{D}_i, \sigma\mathbf{I}) = \sum_{i=1}^{N} \mathcal{N}(\mathbf{x} \mid \tilde{\mathbf{D}}_i + \epsilon_i, \sigma\mathbf{I}) = \sum_{i=1}^{N} \mathcal{N}(\mathbf{q}_i \mid \epsilon_i, \sigma\mathbf{I}) \tag{2.5}$$

Where $\mathbf{q}_i$ is the difference between the pixel location $\mathbf{x}$ and the labelled point $\tilde{\mathbf{D}}_i$, and ϵ_i is a random variable describing the error of the i^{th} annotation.

Distribution Φ can be approximated by a Gaussian distribution, due to the central limit theorem [51] and the authors note this approximation is made stronger with more annotations.

The whole density map is then an application of Φ to each pixel location, this however assumes independence between each spatial location. So a method is proposed by Wan et al. that takes correlation between locations via a multivariate Gaussian approximation to the joint likelihood of the density map $\Psi = \left[\Phi^{(1)}, ..., \Phi^{(P)}\right]$. The mean, and covariance matrix are both calculated through the mean and variance of its marginal distribution Φ.

However the covariance matrix of Ψ has dimension $\mathbf{P} \times \mathbf{P}$, where $\mathbf{P}$ is the amount of pixels. This does not scale well in storage or computation for large images. Noting that the position ii in the diagonal of the matrix is $var(\Phi^{(i)})$ and most off-diagonal elements are zero if far from an annotation, Wan et al. create a low-rank approximation of the covariance matrix. The authors tested both a version with only the diagonal of the covariance matrix and a full covariance matrix and found the latter had better results, this proves the importance of correlations between pixels [51].

Though the map generation still follows a Gaussian smoothing, this method better depicts both the center point, taking error in labelling into consideration as well as taking occlusions and shape into consideration, by correlating different spatial locations.

The method proposed by Cheng et al. builds further on the noise modelling and loss function optimization done by Wan et al., in their work the authors propose replacing the convolution filters with Gaussian kernels. Their novel convolution is as follows:

$$\mathbf{Y}_s = \sum_{i=o}^{N} G(\mu_i, \Sigma_i) * \mathbf{X}_s + \mathbf{b}_s \tag{2.6}$$

All N Gaussian kernels are employed in the convolution as to simulate the map generation process. As to not use such massive Gaussian convolutions, smaller amount of Gaussian kernels (K) are used to approximate the N amount of kernels, where $K \ll N$. Instead of following the method proposed by Chan et al. where the covariance matrix is calculated from the variance of the marginal distribution Φ, the authors use an attention mechanism to learn their correlations.

In this method a large amount of kernels are randomly sampled. Principal Component Analysis is then performed to obtain the K kernels associated to the non-zero eigenvalues. The authors choose to ignore the mean of the kernels to decompose them and accelerate computation. The weight associated with each kernel is initialized and then normalized through a softmax function. The new Gaussian convolution is performed as such:

$$\mathbf{Y}_s = \sum_{j=o}^{K} \left(\mathbf{w}_j \circ \sum_{i=0}^{K} (G(\mu_i, \Sigma_j) * \mathbf{X}_s) \right) + \mathbf{b}_s \tag{2.7}$$

Where the resulting value is now the linear combination of K different Gaussian kernel convolutions. As stated the authors do not infer the mean of the Gaussian, as their kernels are translation invariant a more efficient method is implemented where the Gaussians are described as having a $\mathbf{0}$ mean and are translated to the pixels location.

2.3.2 Point-based Approaches

As discussed in previous sections solving the crowd counting problem can be viewed as solving its parent task, that of crowd localization. Where most methods do so through density maps. Song et al. propose that such procedures where the learning target is an intermediary representation, are error-prone [46]. To tackle this issue and solve crowd counting problems by directly localizing individual target objects, the authors propose a purely point-based framework.

This solution does not rely on intermediary representations, instead making direct use of the dot-annotations $\mathcal{P} = \{p_i \mid i \in \{1, ..., N\}\}$ as its learning targets, outputting a set of predicted points $\hat{\mathcal{P}} = \{\hat{p_j} \mid j \in \{1, ..., M\}\}$ and their respective confidence scores $\hat{\mathcal{C}} = \{\hat{c}_j \mid j \in \{1, ..., M\}\}$, in which M represents the number of estimated head centers.

The proposed model consists of a VGG16 backbone which outputs a feature map $\mathcal{F}_s$ of size $H \times W$, followed by a two parallel branch architecture. One branch is responsible for point coordinate regression, and the other for proposal classification. The latter outputs the aforementioned confidence scores $\hat{\mathcal{C}}$ with a Softmax normalization. While the first, due the intrinsic translation invariant property of convolutional networks, resorts to producing the offsets of proposed points as opposed to the holistic coordinates. In

simple terms each pixel in feature map $\mathcal{F}_s$ represents an $s \times s$ patch in the original image.

The authors take this patch and introduce a set of fixed reference points $\mathcal{R} = \{R_k | k \in \{1, ..., K\}\}$ with predefined positions $R_k = (x_k, y_k)$.

In total the regression branch should produce $H \times W \times K$ proposed point offsets $(\Delta^k_{jx}, \Delta^k_{jy})$. Where K must be of high enough value to ensure that, $M > N$. This is required to ensure enough points for the matching strategy discussed further ahead.

The coordinates of proposed point $\hat{p_j}$ are then calculated as:

$$\begin{aligned} \hat{x}_j &= x_k + \gamma \Delta^k_{jx} \\ \hat{y}_j &= y_k + \gamma \Delta^k_{jy} \end{aligned} \tag{2.8}$$

Where γ is simply a normalization term.

This solution's approach allows for localization of individual entities as opposed to localization through the observation of a mass's position and shape.

To evaluate the performance of this novel approach the authors introduce normalized Average Precision (nAP) which considers both the localization of individual entities and consequentially the overall count.

The nAP is derived from the Average Precision. Which is obtained by sequentially, in decreasing order of confidence score, evaluating points $\hat{p_i} \in \hat{\mathcal{P}}$ as being a True Positive (TP) or False Positive (FP) according to a density aware criterion.

The criterion is a derivation of the pixel-level Euclidean distance, so as to mitigate the problem of density variation, and can be defined as:

$$\mathbb{1}(\hat{p}_i, p_i) = \begin{cases} 1 & if\ ||\hat{p}_i - p_i||_2 / d_{kNN}(p_i) < \delta \\ 0 & otherwise \end{cases} \tag{2.9}$$

Where δ serves as a threshold to control the desired localization accuracy.

The association between proposed point $\hat{p}_i$ and ground-truth point p_i is done through a mutually optimal one-to-one pixel-level distance matching strategy. Where positive proposals are pushed toward their targets and negative proposals are classified as backgrounds. Other matching proposals that do not enforce one-to-one matching were determined to require a threshold that serves as a region boundary where any proposed point would require a target point to fall within that region to be classified as a positive proposal.

This type of threshold method could produce 1 proposed point to N target point matching, or vice-versa. One-to-one matching however enforces that unmatched proposals are automatically remained as negatives, without introducing hyper parameter tuning.

The previously mentioned matching strategy is the result of an application of the Hungarian algorithm, $\Omega(\mathcal{P}, \hat{\mathcal{P}}, \mathcal{D})$ where $\mathcal{D}$ is a pair-wise matching cost matrix of shape $N \times M$ the authors define as:

$$\mathcal{D}(\mathcal{P}, \hat{\mathcal{P}}) = (\tau||p_i - \hat{p_j}||_2 - \hat{c_j})_{i \in N, j \in M} \tag{2.10}$$

This association is necessary not only to evaluate the final model's result, but most importantly to formalize the loss function. Say that ξ, a permutation of $\{1, ..., M\}$, represents the optimal matching result, meaning $\xi = \Omega(\mathcal{P}, \hat{\mathcal{P}}, \mathcal{D})$. Then ground truth point p_i is matched to the proposal point $\hat{p}_{\xi(i)}$. And the loss function follows:

$$\mathcal{L}_{cls} = -\frac{1}{M}\left\{\sum_{i=1}^{N} \log\ \hat{c}_{\xi(i)} + \lambda_1 \sum_{i=N+1}^{M} \log(1 - \hat{c}_{\xi(i)})\right\} \tag{2.11}$$

$$\mathcal{L}_{loc} = \frac{1}{N}\sum_{i=1}^{N} ||p_i - \hat{p}_{\xi(i)}||_2^2 \tag{2.12}$$

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{loc} \tag{2.13}$$

Where λ_1 is a reweighing factor for negative proposals, and λ_2 normalizes the regression loss to balance the effect of both loss terms.

Chen et al. expand on this framework [10]. Noting the instability in the optimization of the matching process, as a potentially limiting factor.

The authors propose an Auxiliary Point Guidance framework. Wherein they create for each target point a set of what are denoted as positive and negative points. These are sets of random, uniformly distributed points. The so called positive points have a Manhattan distance to the target point lesser than η and the negative points have one greater than η. These points are then used to compute the loss formerly defined in equation 2.13.

This framework was evaluated on the benchmark data sets, ShangaiTech Part A and B [59], UCF_CC_50 [25], UCF-QNRF [26] and JHU-CROWD++ [45]. It is the state-of-the-art in Shanghai-Tech Part A, NWPU-Crowd and UCF_CC_50 in both evaluation metrics MAE and MSE. And obtains SoTA results on the JHU-CROWD++ and on the UCF-QNRF datasets in regards to MSE.

2.3.3 Counting through Classification

The majority of methods presented thus far present an encoder-decoder architecture, where spatial information is lost during the process of turning low-level pixel information into high-level semantic pixel information. Liu et al. note that these do not take into account the long-tail distribution of count val-

ues [35]. Areas with large values, i.e. a high concentration of target points are severely undersampled, while representing a higher percentage of total count compared to other less populated areas. Another major drawback from these approaches also noted by Liu et al. was that of inaccurate generations of ground-truth density maps. The authors note the importance of the kernel used for Gaussian smoothing and how different values for σ can result in either the background being covered or the target objects remaining uncovered. This noise results in an unstable training process.

Liu et al. [35] were the first to reframe the crowd counting task as one of classification.

Their solution involves not providing an exact count for a certain pixel or patch, but to provide for each region a count that falls within a certain interval with high confidence, as they put it, blockwise count level classification. The training target for this approach is then not a density map, but rather a class map.

A count map is first produced by integrating the density map in blocks, and then assigning each block a class. Classes which are generated, by quantizing the count map. To account for the aforementioned long tail distribution of the counts, bins are generated, ranging from 0 to the maximum count following a logarithmic rule. Thus alleviating the problem of class imbalance. Wherein undersampled classes fall in larger range bins, meaning lower more seldom represented classes are grouped.

The network designed uses the encoding module in the VGG16 [44] as backbone to encode visual information and to account for resolution variations. The frontend of the network is a softmax layer to calculate the probability of each counted class for each block. Hence the output of the network is a matrix - P - where each cell is a probability vector for the counts of a 32×32 pixel block within the VGG16 backbone.

Another measure the authors take to account for class imbalances is to employ a variation of cross-entropy loss. Where for each point in P the typical formula is scaled by a information-entropy-based regularization factor.

$$IW(GT) = -\log \frac{\sum_{j=1}^{N}(gt(j) == GT)}{N} \tag{2.14}$$

Where $gt(i)$ represents the ground-truth class for point i in P. Classes which are underrepresented will then have a larger IW and consequentially a disproportionate contribution during training.

Lastly to account for objects that fall into two different patches the authors propose a type of redundant counting. Several different offsets of the same image are processed and then an average is conducted to obtain the final estimation.

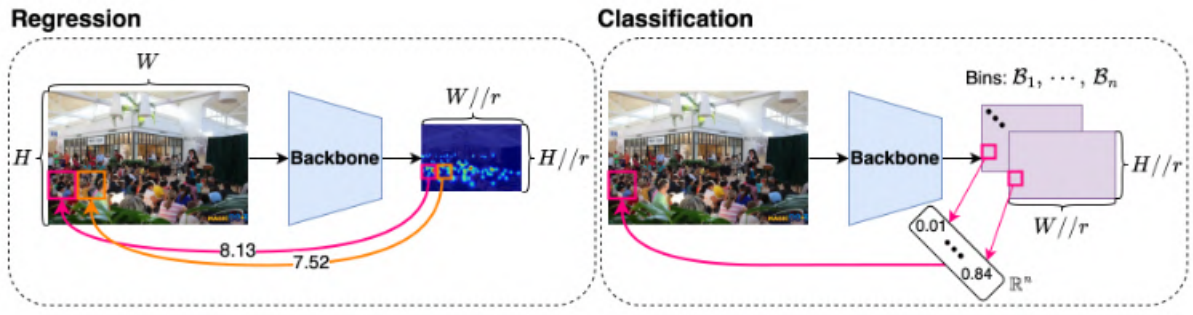


Figure 2.1: Comparison between typical regression-based methods (left) and the reframed blockwise-classification method (right) [37].

Multiplying the probability vectors at each location by the representative value of the corresponding bin, which is equivalent to its mid-point, produces a density map, see figure 2.1. The final count is then calculated as if it were a regression-based method, where the density map is integrated.

In [37], Ma et al. propose a framework named CLIP-EBC. The authors take inspiration from the work proposed by Liu et al. [35], noting however that the use of Gaussian smoothing, even though it is later segmented, still introduces noise. In fact the segmentation used to create the count and later class map, means the discrete count values are transformed into a continuous space, due to truncation at the borders of blocks.

Ground-truth noise is not solely restricted to the process of density map generation. In fact the basic dot-annotation itself, as mentioned before, is inherently noisy. The authors note that this makes any pixel-level prediction challenging.

To address both these types of noise the authors propose an approach that borrows from the YOLO family. Instead of performing Gaussian smoothing, each block is compelled to predict the presence of targets, if and only if, the dot-annotation falls within its boundaries. This removes the noise introduced through smoothing all the while maintaining the discreteness of the count. Avoiding the problem the aforementioned blockwise classification method neglected.

The authors propose a network that as of the writing of this document is the only solution to fully employ a CLIP-based model which generates crowd density maps.

The final pooling and linear projection layers are removed from CLIP's image encoder, to suit the problem of blockwise prediction, and the remaining backbone is used to extract the image feature map.

For text feature extraction, first each bin has a text prompt of the form, "There is/are n person/people.", "There are between n and m people." or "There are more than n people.", assigned depending on its count class. These prompts are tokenized and inputted into the original text encoder, which is frozen during training.

The cosine similarity is then calculated between the feature maps obtained from each encoder. Sub-

sequently the similarities are normalized through the softmax to obtain the final probabilities map P^*.

Much like the work proposed by Liu et al. this map is used to compute a density map Y^*, but with the slight variation of using the average count for each bin instead of its mid point. This is done to account for the non-uniform distribution of counts within a class. A problem which is accentuated by more severe class imbalances in merged bins.

The last contribution made is a loss function defined in 2.15. Previous classification-based crowd counting methods neglect the overall error in crowd count, focusing solely on classification of individual blocks. The authors note that two probability distributions can produce the same classification error but possess different count expectations. Hence the Distance-Aware-Cross-Entropy (DACE) Loss is proposed.

$$\begin{aligned}\mathcal{L}_{DACE} &= \mathcal{L}_{class}(P^*, P) + \lambda \mathcal{L}_{count}(Y^*, Y) \\ &= -\sum_{i=1}^{H//r} \sum_{j=1}^{W//r} \sum_{k=1}^{n} \mathbb{1}(P_{k,i,j} = 1) \log P^*_{k,i,j} + \lambda \mathcal{L}_{count}(Y^*, Y)\end{aligned} \tag{2.15}$$

where $\mathbb{1}$ is the indicator function, P, Y are the ground-truth probability and density maps, respectively, and P^*, Y^* are their predicted counterparts. The reduction factor - image encoder's patch size - is represented by r, and n represents the number of bins. The authors do not restrict $\mathcal{L}_{count}$ to any particular loss, instead any function that measures the difference between two density maps can be used.

This solution is currently the SoTa on the NWPU-Crowd dataset, and achieves competitive measures of the MAE and MSE score on typical crowd counting benchmarks.

2.3.4 Applied to the Marine Domain

Some work has been done in the field of fish counting through computer vision, mostly in the industry of aquaculture [15, 39, 58] and ecology [18].

In [35], a paper discussed in subsection 2.3.3, Liu et al. employ their solution to a variety of different domains namely the problem of fish counting. This represents the sole solution found that approached the problem of fish counting through the lens of crowd counting. However the dataset employed is one curated by authors themselves that used sonar-based collection instead of photographed images.

This variation poses some differences between our use case and theirs. Though the problem of occlusion, size and density variations persist, others are mitigated. Namely we see that in a sonar-based dataset we have no light attenuation causing blueshift and no visual impact from how the light is blocked or shines directly on mackerel.

In [18] Ditria et al. propose a method to count the abundance of fish through object detection. Their solution consists of a Mask R-CNN, a two-stage detector that classifies and localizes target objects in the Region of Interest (ROI), with the added behavior of predicting a segmentation mask. The annotations needed to train this model are more expensive than dot annotations, by requiring a hand-crafted segmentation mask in the ROI. The scenes evaluated by the model are of videos recorded in the wild, which poses new limitations such as varying light and visibility conditions. This method performs well under the conditions stated in their domain, outperforming experts and citizen scientists, both in accuracy and false positive metrics. However, the challenges posed in their problem case differ from ours. As the schooling effect is not always a limitation. The purpose of the tool is to detect and localize fish, namely luderick, that many times do not appear in as large a structure as mackerels.

In [58] Zhang et al. propose a method to estimate the number of Atlantic salmon in a population through a hybrid DNN model. The images are strategically taken in a net cage from bottom up, at a specific angle to avoid the interference of vertical light. The photos were then preprocessed, to enhance contrast and following the grey world hypothesis the images are color corrected to fill the whole luminance scale. Dot-map is used to construct a ground-truth density map, by applying an adaptive kernel Gaussian smoothing. The model consists of a multi-column CNN for the front-end to capture scale information and to maintain spatial information a deeper and wider DCNN is used with dilated convolutions as back-end to maintain feature map dimensionality and the output is an estimate of the density map. This model can accurately realize fish counting in a stable manner without big variations in the evaluation metrics.

In [15] Connolly et al. attempt to propose a model to count snappers in oceanic waters. Again the authors chose a detection-based approach using the two-stage detector Faster R-CNN. Much like the aforementioned work the authors use a ResNet50 configuration [18]. The variation of confidence thresholds, sequential non-maximum suppression and the application of statistical correction, were the main work developed, that differentiated this work from the one mentioned above.

In [39] Pai et al. Again employ a detection approach, employing the YOLOv5 model with CSPDarknet as feature extracting backbone. And make use of Optical flow to detect erratic behavior. The main focus of the work is behavior analysis. In the detection portion however the limitations of detection-based approaches are noted, with the the model over and under-detecting targets.

2.3.5 Summary

This section presented an overview of the current crowd counting literature. The presented models try to take scale into account and preserve spatial resolution and correlation to perform an estimation of the total head count within a crowd. Our domain presents challenges which are present and taken into consideration in the reviewed literature, such as occlusion, perspective distortion, low resolution, illumination changes, and others which are specific to our problem space like the ones mentioned in

subsection 1.2.

The current research directions we believe show the most promise, i.e. relaxing spatial invariance, point-based approaches and reframing counting as a classification task, had sections devoted solely to their approach. We believe the focus of these solutions on either denoising the ground-truth density map generation or breaking free from it all together in favour of other representations, makes them perfect candidates for our specific domain. Where the problems of gaussian interference and occlusions are exacerbated.

Chapter 3

Dataset

Contents

This chapter presents an overview of the collection and annotation processes employed to create the dataset used to train and evaluate the models mentioned in Chapter 4. This endeavor originated due to the absence of publicly available datasets on schools of fish, specifically mackerel, with the intended purpose of population counting, given the constraints of the aquarium. Already stated in Chapter 2 previous work conducted by Liu et al. already tackle population counting of fish schools, however, they do not make use of photographs, but instead of sonar images [35], a type of data collection not possible given our setting within the oceanarium facilities, and would not give us the ability to differentiate between mackerels and other species with which they cohabitate. As mentioned previously in this document our work was done with the support of Oceanário de Lisboa and as such all images were obtained within the oceanarium facilities.

3.1 Previous Image Captures

Previous data collection processes were conducted on the same environment [1, 7, 47]. However, due to conflicting objectives, none of the resulting images were found to have the required quality to support the training of the selected models. Seeing as the images previously captured were intended to capture sharks and rays that habitually reside lower within the water column and closer to the viewing pane, the camera was positioned nearer the ground, and set to focus on objects in the foreground. This resulted in images that displayed low resolution on the areas most predominantly occupied by mackerel, which reside higher within the tank and further back, closer to the tank wall. Moreover, as schools were not the intended target for these shootings most images showcase truncated objects, where parts of mackerel fall outside the captured frame.

Lastly the impact of having our camera set lower than the intended targets was twofold. First, our view of the fish is of their ventral side which is more oblong and feature lacking. Second, they were positioned between the camera and the floodlights above the aquarium, resulting in our observation of the fish's white underside against the white ceiling lights. Or nearer to noon when the lights are more intense the fish's shadow rather than the fish itself. These issues highly impacted the annotation process and helped inform the setup for future captures which focused on smaller fish that display schooling behaviour. One such image is shown in Figure 3.1. Another issue that can be observed in the previous recordings, is that of reflections. A result of having the camera outside the tank is that some incident angles of the lights on the tank wall produce very bright reflections, these pose little to no issue when capturing large entities, such as sharks or rays. However distant mackerel can be nearly totally occluded, again we refer to Figure 3.1.

Figure 3.1: Image collected with the intended purpose of detecting and tracking sharks and rays in the main tank at Oceanário de Lisboa [7].

Figure 3.2: Image collected with the intended purpose of counting mackerel.

3.2 Collection and Image Properties

The images were collected through a static GoPro Hero Session, mounted inside the main tank. To address the challenges mentioned earlier in section 3.1, the camera was placed slightly above the typical mackerel habitation plane, two meters below the surface tied to a feeding platform close to the back wall. This placement allowed us to capture both a side view of the fish and to get them in the brightest light and enabled a compromise between capturing as most of the school as possible all the while maintaining

sufficient resolution. An image captured during our amended setup recording session can be seen in Figure 3.2. We can note the better contrast between targets and background, the lack of truncations, the increased resolution of the mackerel, and the capture of the broad side of the fish.

As a whole, ninety-one images were captured and found to be of high enough quality to conduct a reliable annotation process. These ninety-one images consist of 25,714 mackerels, with the annotations being made on the center of their bodies. Annotating the center mass of the targets was a conscious one that was intended to produce the best possible results when applying our novel ground-truth generation process. The basic descriptive statistical measures of the dataset can be observed in table 3.1.

Resolution	Num Images	Min	Avg	Max	Total
$2,432 \times 3,264$	91	125	282.6	410	25,714

Table 3.1: Listing of dataset statistics. Total refers to the amount of mackerel in the whole dataset, while Min, Avg, and Max refer to counts in single images.

We also give the entity histograms of the images in the dataset in Figure 3.3. As can be seen, the images span a wide range of counts, ranging from 125 targets to 410. Though the distribution of total counts across images bears no weight on our particular task, seeing as what impacts the models is density and size variations, we inferred a correlation from a simple analysis of the dataset. Images that show a lower total count usually display a lower overall density of mackerel and a higher apparent size. This is a consequence of most captures which contain a smaller portion of the school being taken when the mass was close to the camera where most targets fell outside the scene captured and as a result of their closeness appear much larger compared to other stills. Due to the wide-ranging distribution, we believe there is a considerable variation in mackerel and school characteristics, again density and size discrepancies.

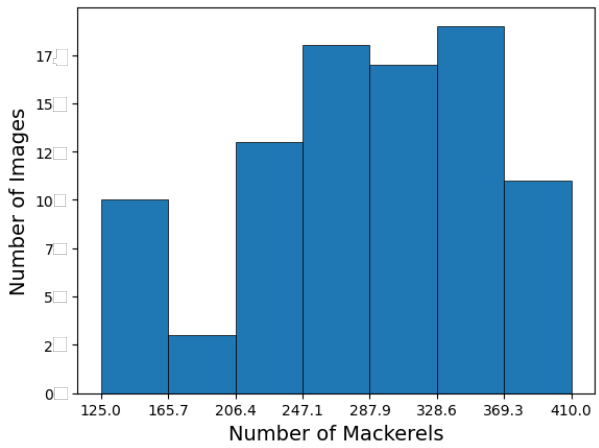


Figure 3.3: Histogram of mackerel counts of our new dataset.

The apparent size and crowd density vary greatly between images, as can be observed in Figure 3.4. In image 3.4(b) a fish can be as small as roughly 10 pixels, while in image 3.4(a) one can be as large as 275 pixels, making accurate estimations challenging, but hopefully making our final system more robust to these variations.

Figure 3.4: Comparison between apparent size and density distributions across different images. Left: Image with lowest total count - 125. Right: Image with highest total count - 410.

3.3 Annotation Process

As mentioned in the overview of Data Labelling, section 2.3, most current crowd-counting methods rely on dot annotations. In addition to these, we also perform what we denote as orientation annotations, seen in Figure 3.5. As was first described in chapter 1 schools of fish behave as if sharing a single collective will. A characteristic that emerges from this behavior is that neighboring fish share orientations, a characteristic we sought to exploit when designing our approach in chapter 4.

Orientation annotations intend to characterize a region of the school as to what is the orientation of its fish. The process of imparting this information was achieved by annotating several straight lines at different locations within the school, with the same orientation as the fish surrounding them. Of all the lines that are annotated on an image the one closest to a given fish should in theory have the same orientation, i.e. angle/slope, as that individual. This information can then be leveraged in the generation of ground-truth density maps following the steps laid out in further detail in Chapter 4 to adjust the Gaussian kernel so it better matches each individual.

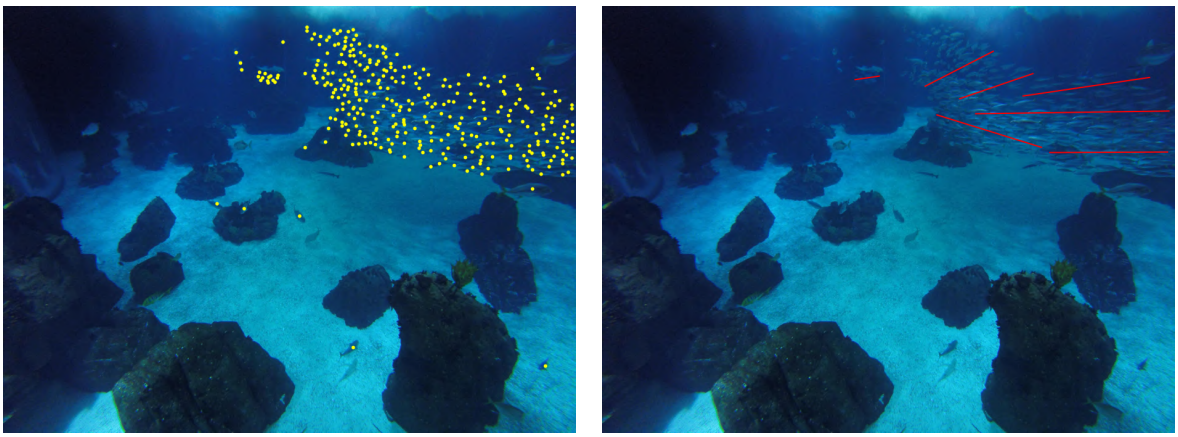


Figure 3.5: Annotations: Left - Typical dot-annotations.
Right - Novel orientation-annotation.

As mentioned in previous chapters, this is, to the best of our knowledge, the sole dataset consisting of photographs of schooling fish targeting the task of crowd counting. This we believe is a contribution, that is best made publicly available and the full dataset can be consulted and downloaded on Hugging Face.

https://huggingface.co/datasets/ravasco/mackerel-schools

Chapter 4

Methodology

Contents

In section 1.2 we presented an overview of our problem and listed some of the characteristics of our domain that make its setting particularly challenging. Both as a general crowd-counting task and more specifically those challenges, which given our marine setting, emerge and are not present in other domains. Such as the typical anthropological one. Following this, Chapter 2 allowed a deep dive into the current crowd-counting literature. Then in Chapter 3 we present our very own dataset alongside our new annotation type, orientation annotations, which encode fish direction. In this chapter, we coalesce all the previously presented information and put forth several approaches we believe are best suited to our task, given its constraints. Constraints which, again, are very similar to the ones observed in typical human crowd counting settings, but with the addition of two others.

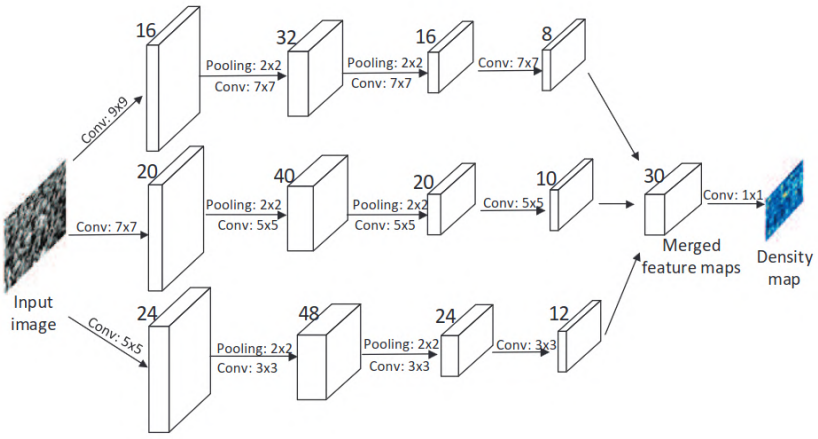
1. Distance between targets in the image plane bares no weight on the actual distance between objects.

2. Shape of fish is not circle-like.

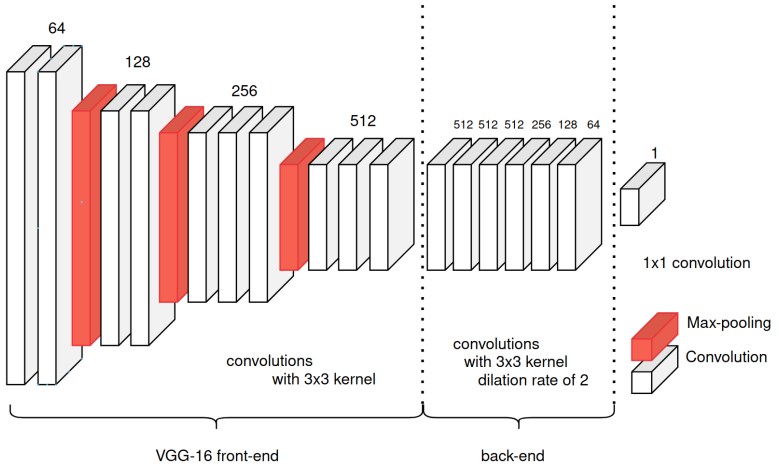
In subsequent sections we present approaches we hope can mitigate these issues.

4.1 Size Variation and Geometric Constraints in Marine Domain

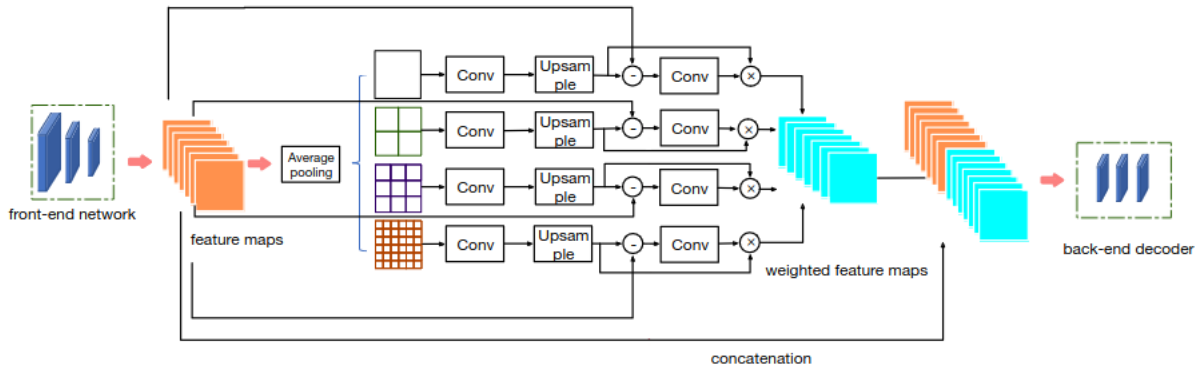
Much like all crowd-counting problems. Fish's apparent size decreases as they move further away from the observer. In our dataset, this ratio between the largest and smallest individuals can be in the order of $27\times$, a variation that poses a major conditioner to the architectures we choose to employ. Some of the solutions we presented in Chapter 2 already made this their main concern and area of focus. However, most if not all of the methods that were reviewed had a solely anthropological view, meaning the application of the best performant models to our task will not inherently result in the best possible solution. As such we saw fit to select a variety of different models. The models selected were the Multi-Column Convolutional Neural Network (MCNN), Congested Scene Recognition Network (CSRNet) and the Context Aware Network (CAN). These present different paradigms of taking scale into account. Namely through the use of a branching network structure; dilated convolutions and finally the use of Spatial Pyramid Pooling with a non-linear concatenation. These can be seen in figure 4.1. The modification of the CAN approach, namely the Extended Context Aware Network (ECAN) [36], was not considered. Much like the first ever deep learning crowd counting solution, CrowdCNN [57], this solution requires a perspective map to exploit scene geometry information, guiding the network to better adjust to scene context and mitigating the problem of scene geometry distortion.



(a)



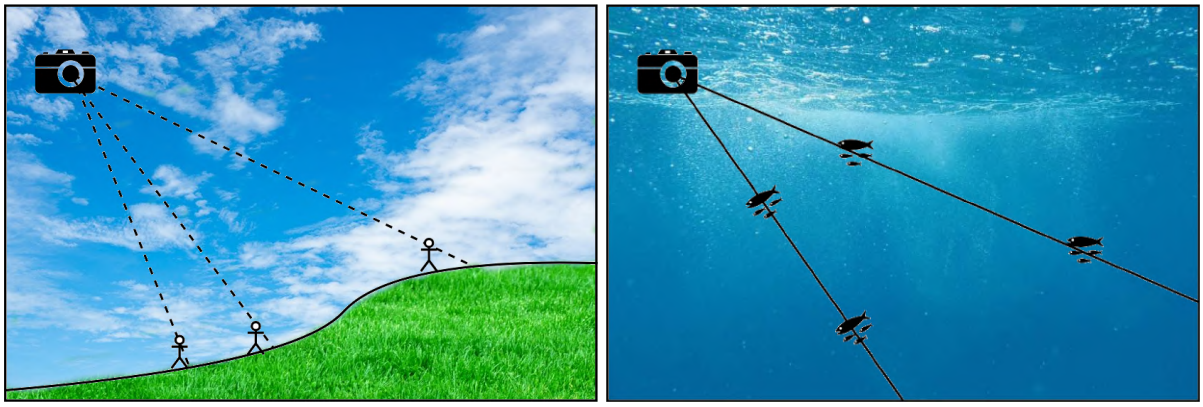
(b)



(c)

Figure 4.1: (a) Branching structure of the Multi-Column Convolutional Neural Network [59]. (b) Dilated kernels, employed in the CSRNet [32]. (c) Architecture of the CAN model [36].

The reason we did not consider the ECAN extension is that no such perspective maps can be produced. Our setting has a rather prohibitive constraint, information about apparent size is impossible to encode. While people are located in the plane that represents the floor of the scene, fish are not bound to this same constraint, see Figure 4.2. Their position within the scene relays no information as to how close or far from the camera they are and therefore what their relative size should be. A consequence of this major difference between domains is that the scene has no inherent structure and therefore no perspective map can be produced.



(a) **(b)**

Figure 4.2: (a) Humans are bound to the ground. Their position will always be at the intersection between the ground plane and the camera rays. (b) Fish can be anywhere along a camera ray. Two fish can occupy the same pixels on the final image while being at wildly different distances.

Another constraint we have already introduced is that the distance between targets in the image plane bears no weight on the distance between objects in the scene space. Since position within the image encodes no information has to where in the 3D geometry of the scene the fish are located, the relative position between fish in the image will also carry no information about relative position within the actual scene. This presents a major setback. Relative position among targets within the scene is used in human settings to infer the apparent size, done in the ground-truth generation process, by performing a Gaussian smoothing with an adaptive sigma, a process that was previously covered more in-depth in Chapter 2. Not being able to employ an adaptive sigma that uses pixel distance between targets as a metric for actual distance between individuals, means that we lose a very powerful method of approximating mackerel size at ground-truth generation. This limitation, though of sound reasoning is an intuition made by us that may prove false. The schooling behavior of fish may in fact impart some geometric information, whose basis, though different in logic from that applicable to humans, is still valid when applying an adaptive sigma to mackerels. If we consider the overall mean density of a school to be constant, a sigma based on the distance to the k nearest neighbors could produce in the fish setting the

same result as if applied to a human setting. Schools further away would show more compact annotations, and schools closer would display a more sparse dot-annotation map. This would, however, introduce the constraints that schools do not overlap each other or themselves and that pictures must be taken broad side, capturing the mackerel flanks, this should be done so as to not use photographs with too wide a school which would implicate a more dense annotation map.

4.2 Domain-specific Ground-Truth Generation

Another aspect specific to our domain we must consider is that fish are not people-like, this obvious fact does come with some implications when generating ground truth data. In typical crowd-counting applications, the kernels are round to approximate the shape of the human head. Fish however are not round but rather oblong, if we apply typical kernels to the generation of ground-truth density maps in this setting, we can wind up with two different results. Either our kernel is smaller than the fish, in which case we do not capture complete semantic information, or it is larger, meaning we capture the background, or other targets within our region of interest when performing the smoothing. This produces contradictory information during the training phase, therefore degrading the performance of the network.

To address this issue head-on, we propose a novel ground-truth density map generation process that exploits characteristics of the holistic fish school to better approximate each individual's shape, maximizing its area under the Gaussian kernel. If we consider that all fish share a similar shape we can adjust the Gaussian's covariance matrix, Σ, altering the kernel so it better approximates it. However, this approach neglects the fact that fish display different orientations. As such we need to be able to alter the covariance matrix to produce kernels that vary depending on the orientation of fish at different positions. As stated in Chapter 1 and Chapter 3, fish within a school behave as if belonging to a superorganism, with their orientation of being equal to that of its neighbors. This information, as first introduced in Chapter 3, can be encoded through the use of orientation annotations, a set of straight line segments with the same orientation as the fish located closest to it. From the association of a target point to the nearest orientation-annotation, we can infer an approximation of each fish's angle, as shown in Figure 4.3. This inference is sensitive to erratic school structure, seen in feeding and mating times, which luckily is not the case in our dataset as the photos were captured when these behaviors were not displayed.

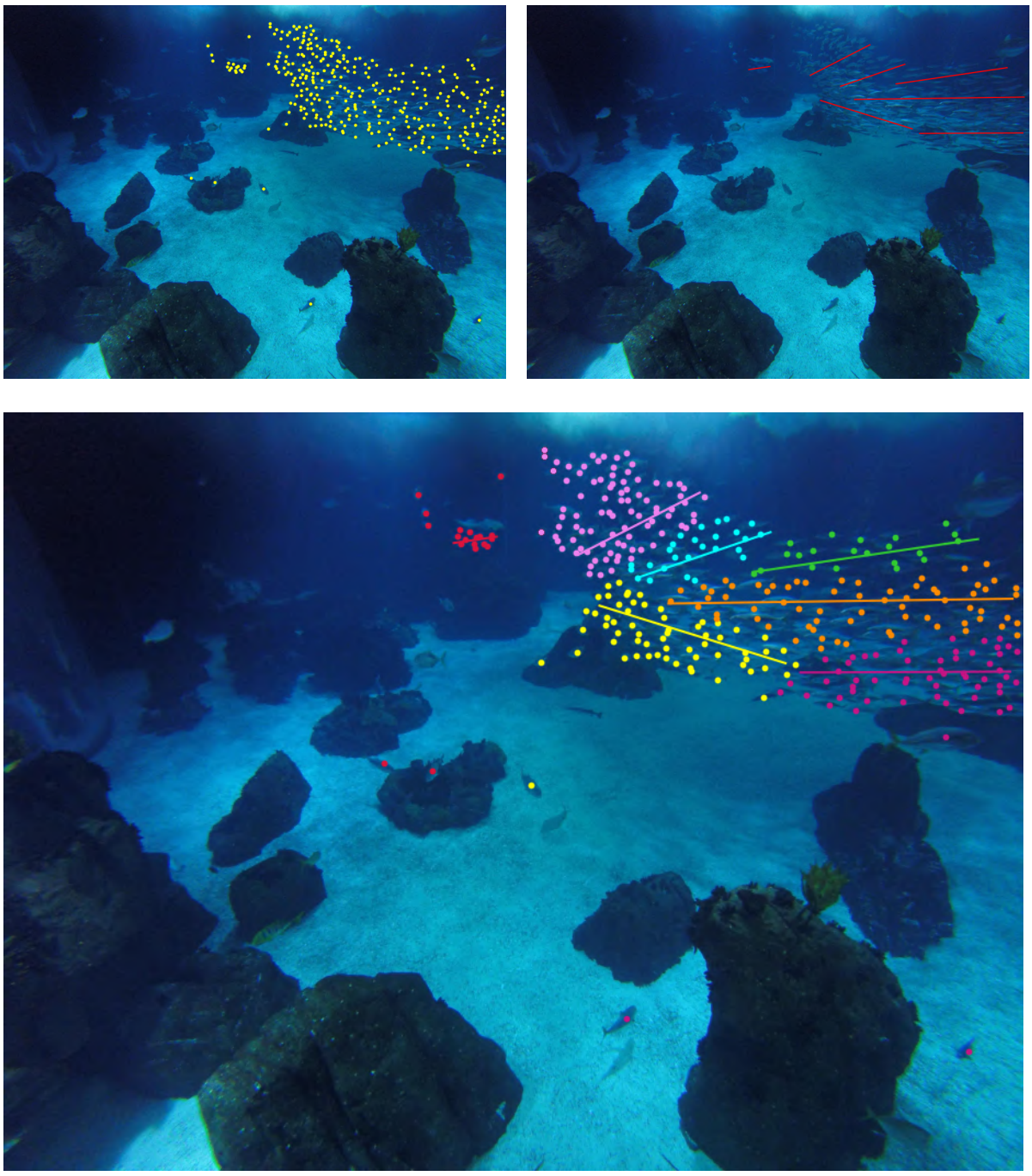


Figure 4.3: Association of fish to their respective orientation. Top-left: Typical dot annotations. Top-right: Our novel oriented annotations. Bottom: Association between each individual and the orientation-annotation closest to it.

The result of this assignment is a mapping, $y : \Theta \longrightarrow \mathcal{F}$. Such that each oriented-annotation's angle, $\theta_i \in \Theta = \{\theta_1, ..., \theta_N\}$, is matched to the respective set of fish, $\mathcal{F}_i = \{f \mid orientation\ of\ f\ is\ \theta_i\}$, where $\mathcal{F} = \bigcup_i \mathcal{F}_i$ and $\mathcal{F}_i \cap \mathcal{F}_j = \{\}$, $i \neq j$. Then for each angle, θ_i, we calculate the rotation of the chosen covariance matrix, as detailed in equation 4.1, thus obtaining the kernel that must be applied for each target in $\mathcal{F}_i$, seen in Figure 4.4.

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$\Sigma_{i_{rot}} = R(\theta_i) \cdot \Sigma_i \cdot R(\theta_i)^\intercal \tag{4.1}$$

A multiplication by the transpose of the rotation matrix, $R^\intercal$, is performed to ensure the resulting matrix maintains its positive definiteness, a characteristic of covariance matrices in our domain. Typically covariance matrices are positive semi-definite, whose geometric interpretation is that of a centered ellipse. However since fish are two dimensional, we know apriori that our ellipses will not have their sides touching the coordinate axis, the geometric representation of positive definite matrices.

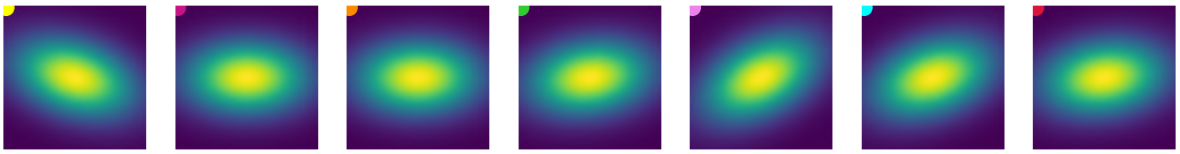


Figure 4.4: Kernels generated for image shown in Figure 4.3. Kernels are color-mapped to the school groups.

Density map generation in our approach follows the same logic as the one laid out in equation 2.1, with the sole difference that now the covariance is not the identity matrix, $\mathbf{I}$, but rather one that is dependent on the position of annotation $\mathbf{x}$. The full algorithm for constructing the ground-truth maps is presented in 4.1. And the resulting map can be observed in Figure 4.5.

Algorithm 4.1: Generation of oriented density map

Data: y, Θ, Σ, img

$w \leftarrow \text{img}.W$

$h \leftarrow \text{img}.H$

$d \leftarrow \mathbf{0}_{w \times h}$

for θ **in** Θ **do**

  $\Sigma_{rot} \leftarrow R(\theta) \cdot \Sigma \cdot R(\theta)^\intercal$

  $d_{tmp} \leftarrow \sum_{f \in y(\theta)} \mathcal{N}(\mathbf{x} | \mathbf{D}_f, \Sigma_{rot})$

  $d \leftarrow d + d_{tmp}$

Seeing as mackerel are obligate shoalers, meaning they spend all their time schooling, this exploit can be used in any captured image, not only on special occasions such as feeding time or breeding season. However, an aspect that needs to be taken under consideration is that schools cannot overlap each other or themselves. This may produce mixed orientation signals, introducing further noise.

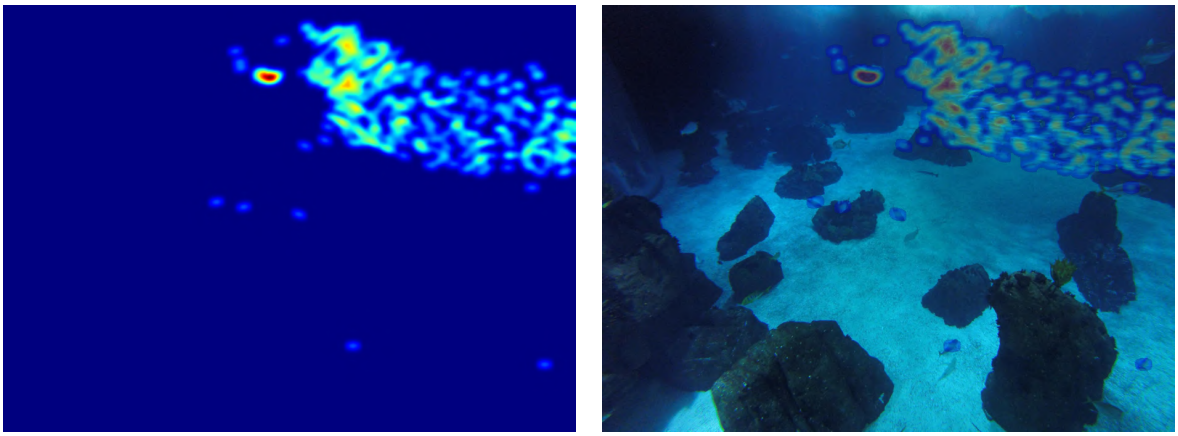


Figure 4.5: The novel oriented density map generated for the image in Figure 4.3. Standalone density map (left). Density map overlaid atop the original photograph (right).

With the intended purpose of conducting a comparative analysis between our kernel and the standard crowd-counting ground-truth generation processes, i.e. fixed and adaptive kernels. We propose training the MCNN, CAN, and CSRNet models with each of the differing ground-truth density maps. These models proposed previously in section 4.1, are based on distinct paradigms, and as such we believe them to be an apt basis upon which to test our novel domain-specific approach. The final proposed regression-based architecture overview can be seen in Figure 4.7.

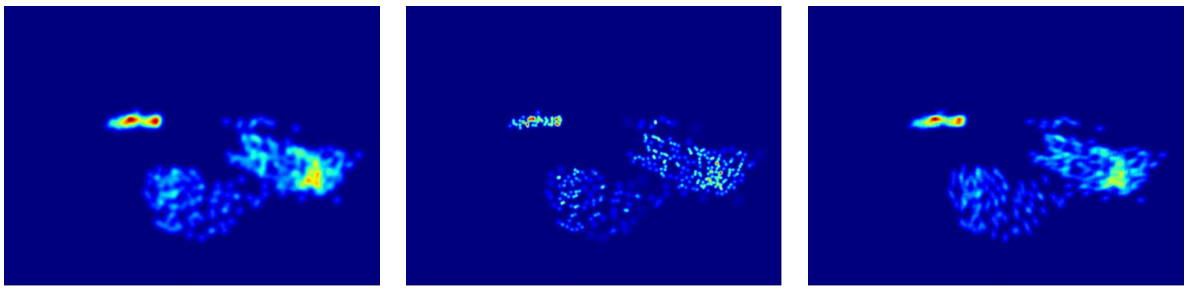


Figure 4.6: Density maps that results from the application of a fixed, adaptive and oriented kernel respectively.

This analysis will also allow us to determine the applicability of distance based heuristics as a means of determining an individual's apparent size. As already discussed, adaptive kernels though fitting in human settings, build upon false premises in our domain. Therefore their employment can introduce further

noise, deteriorating the training performance of the models. See Figure 4.6 to observe the differences from applying different kernels when generating density maps.

Ground-truth generation is a noisy prospect in crowd-counting tasks, and much research is done in the direction of mitigating it, both through modeling it in proposed loss functions [51], in architecting novel layers [13] and in frameworks that distance themselves from pixel-level regression [37, 46]. We propose the oriented kernel as another method of reducing noise in domains whose targets are not circular, improving the characterization of individuals and reducing the backpropagation of mixed learning signals during training time.

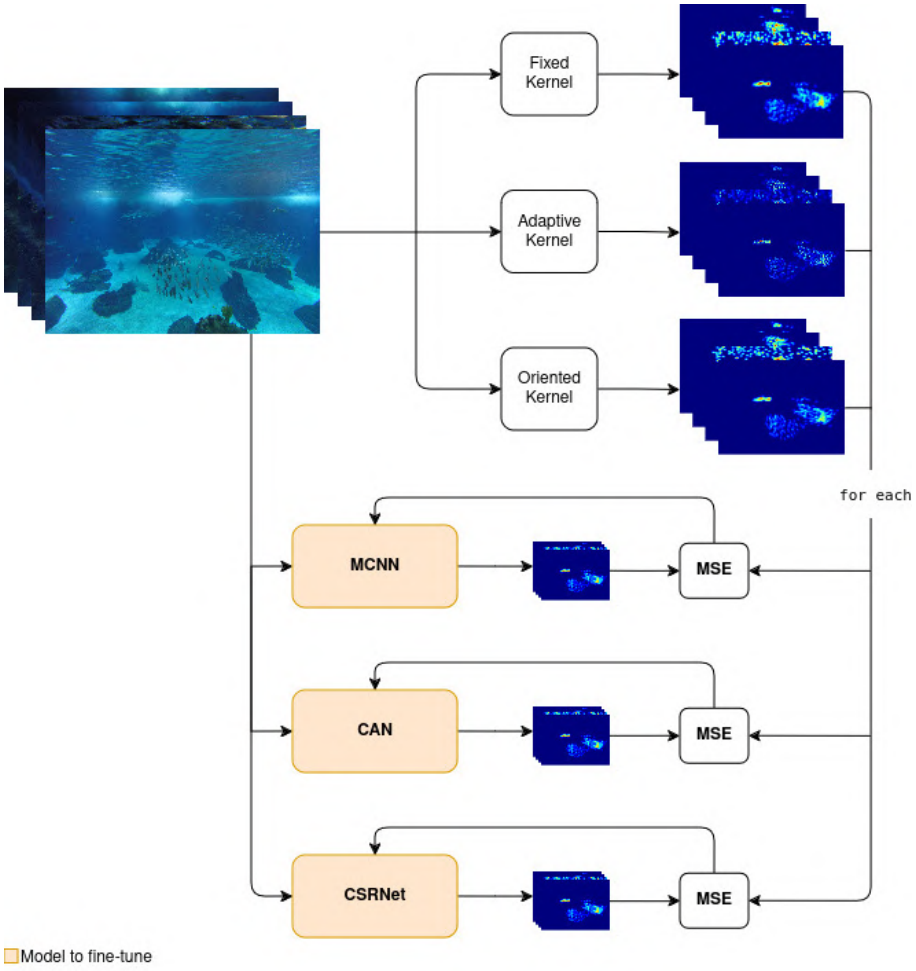


Figure 4.7: Regression-based solution.

4.3 Steering Clear of Pixel-level Regression

Though the previous ground-truth generation process intends to better approximate the shape of individual targets it still has its pitfalls, which are observable even before model training and evaluation.

First, mackerel are not perfectly oriented, therefore our kernels will only be an approximation and not the de facto orientation. In the case of very erratic school behavior, our method may even prove to be intractable, requiring orientation annotations for very small school subsets.

Second, even the best-designed kernel would not solve the problems of apparent size variance, nor would it perfectly fit each individual. This problem though displayed in human datasets is more prevalent in our domain.

Another characteristic of our dataset, observed in typical crowd-counting datasets as described by Liu et al. [35], is that of a skewed distribution, as seen in Figure 4.8. Though not as extreme as in human datasets, we still observe that high-density areas are undersampled.

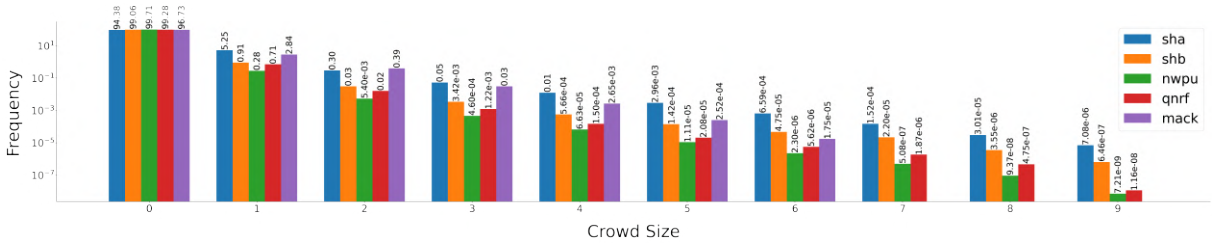


Figure 4.8: The distribution of the number of points in a 8×8 window across benchmark datasets (sha, shb, nwpu) and the mackerel one (mack). Showcases, that like many of the human crowd datasets, the mackerel dataset displays a long-tail count distribution, with zero being the most prevalent and high count values being underrepresented.

Motivated to rectify this long-tail distribution of count values we chose to employ CLIP-EBC, proposed by Ma et al. [37], in parallel with regression-based approaches. This framework's ground-truth density maps are kernel-free, using the binary count map directly while at the same time, its reframing of the task into a classification problem takes the under-sampling of high-density areas into account, as is shown in equation 2.15.

The author's proposed loss formalized in equation 2.15 does not constrain $\mathcal{L}_{count}$ to any particular loss. This as mentioned in Chapter 2 can be any method of differentiating a pair of density maps. We chose to employ a loss introduced by Wang et al. [52]. The DMCount Loss. Formalized in equation 4.2.

$$\mathcal{L}_{count} = \mathcal{L}_{DM} = \mathcal{L}_C(Y^*, Y) + \lambda_1 \mathcal{L}_{OT}(Y^*, Y) + \lambda_2 \mathcal{L}_{TV}(Y^*, Y) \tag{4.2}$$

where Y^* is the predicted density map, and Y is the binary dot-annotation map.

The overall loss is the combination of three other loss functions. The counting loss, $\mathcal{L}_C$, represents the absolute error between maps and whose objective is to get the total count of the predicted map, Y^*,

to approach that of the ground-truth map, Y.

$$\mathcal{L}_C = |\|Y\|_1 - \|Y^*\|_1|$$

The optimal transport loss, $\mathcal{L}_{OT}$, proposed by [48]. This refers to the optimal cost of transforming one probability distribution, Y^*, to another, Y, meaning it represents the dissimilarity between both matrices.

$$\mathcal{L}_{OT} = \mathcal{W}\left(\frac{Y}{\|Y\|_1}, \frac{Y^*}{\|Y^*\|_1}\right) = \min_{\gamma \in \Gamma} \langle \mathbf{C}, \gamma \rangle$$

where $\mathbf{C}$ represents the cost matrix. Each entry, $\mathbf{C}_{ij}$, is the cost of moving from a point in Y to one in Y^*. And Γ all possible ways of transporting the probability mass.

Finally the total variation loss, $\mathcal{L}_{TV}$, that computes the absolute difference between the predicted and ground-truth density maps, normalized by their total counts. The authors chose to employ this loss in conjunction with the optimal transport loss due to both the latter's poorer approximation for low-density areas and what they pose as an increase in the stability of the training process.

$$\mathcal{L}_{TV} = \frac{1}{2}\left\|\frac{Y}{\|Y\|_1} - \frac{Y^*}{\|Y^*\|_1}\right\|_1$$

Typical loss formulations which are employed require a ground-truth density map. DMCount loss was chosen since ground truth is not smoothed, hence our model frees itself from signals resulting from this noisy preprocessing. This in conjunction with the architecture's borrowing of YOLO anchor points, ensures that the chosen framework reduces as much ground-truth noise as possible. Not only is no smoothing noise introduced, but some leeway is also given to dot annotations, as the position of the annotation does not affect objects that are not truncated by the patching.

To adapt the model to our particular dataset. We first conduct a review of the distribution of counts. A consequence of the model's need for an apriori definition of bin classes. The authors propose three different bin granularities.

- Fine-grained bins, where each contains only one integer.
- Coarse-grained bins, where each, with the exception of the empty bin, contains two integers.
- Dynamic-grained bins, a mixed-strategy where until a certain count fine-grained bins are employed and after which they are substituted by coarse-grained bins.

Coarse-grained bins aim at increasing the sample size of each bin, through the merger of the less

represented ones, tackling the issue of a skewed count distribution. Dynamic bins, however, broach the same issue but do so only for the more critical classes, meaning not only are undersampled classes more represented, but the overall class counts become more balanced, as lower, and more prevalent, counts remain separate.

Our dataset, as seen in Figure 4.8, presents a distribution comparable to other datasets, as such we chose to employ the binning strategy the authors purport to obtain the best results, that of dynamic bins. And chose the mean as the strategy for calculating each class's representative value, chosen as opposed to the midpoint, due to the aforementioned non-uniform count distribution within each bin.

Lastly, pertaining to the class decision, we had to alter the inputs for the CLIP text encoder, replacing human/person in the template prompts with fish. These should be closer matches for the input images and therefore a better starting point.

The final CLIP-EBC solution is depicted in Figure 4.9.

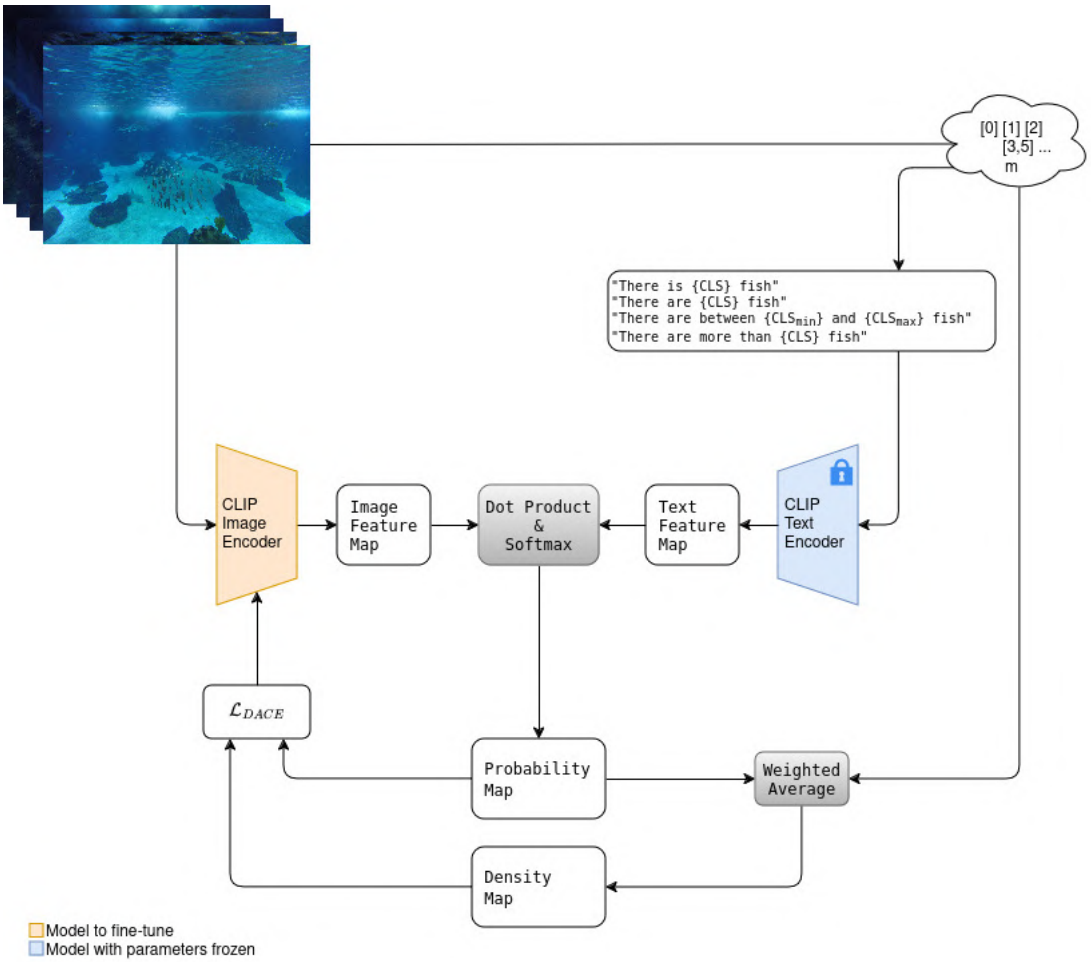


Figure 4.9: CLIP-EBC's architecture.

Chapter 5

Evaluation

Contents

In this chapter, we go over the setup used for training and evaluating the chosen models and density map generation methods. Followed, by the presentation of the evaluation metrics and a comparative analysis of each distinct solution. Finally, we make use of the end of this chapter as an opportunity to go over some of the limitations of the chosen models.

5.1 Training

Before diving into the training process of each solution, we present in Table 5.1 the hardware of the used setup.

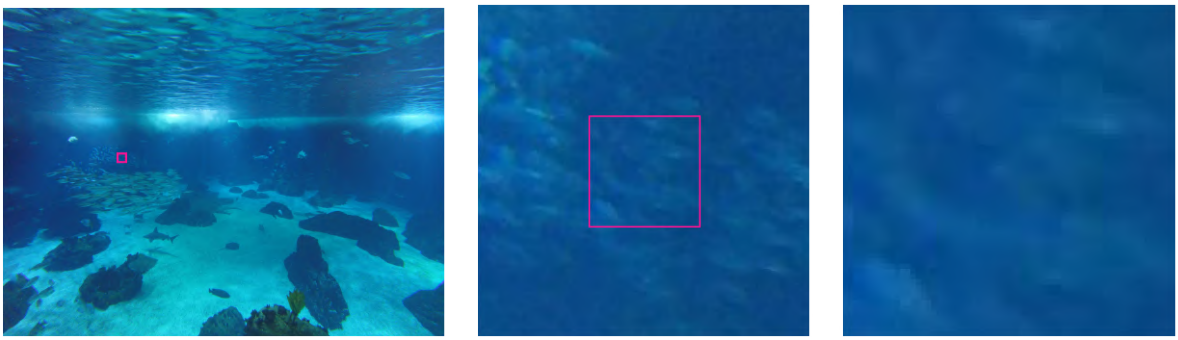
CPU	AMD Ryzen 5 3600 6-Core Processor 3.6GHz
GPU	NVIDIA GeForce RTX 3060
RAM	2×8 DDR4 2400MT/s

Table 5.1: System hardware specifications

As presented in Chapter 4 we chose to perform two distinct solutions, one regression-based, which allows for the comparison of different ground-truth generation processes, namely which kernels are best suited to our domain and another which reframes the problem into blockwise classification, representing a paradigm shift that though the state-of-the-art in the human domain is yet to be employed in other settings.

To standardize the training process between all regression-based solutions, we utilized the same loss function for each model, namely the Mean Squared Error, the same learning rate, optimizer, and batch size. Each model was trained until convergence, defined as a patience of 50 epochs. Our data points suffered geometric transformations for data augmentation, namely random-size cropping, rotations, and mirroring, noise was also introduced in the form of salt and pepper, jittering, and grid masking. The training for the CLIP-EBC made use of the same data augmentation techniques, except for grid masking. In models that use training signals from a ground-truth density map, this augmentation can be employed by simply masking the same area in the density map as in the input image. For blockwise classification, however, masking an area of the ground-truth is doing so to the dot-annotation map, meaning the partial occlusion of a small portion of an individual could mean the count being reduced by a whole fish, which differs from regression approaches where occlusions are equivalent between input and ground-truth.

As for architectural choices, the ViT B32 [19] was chosen as the image encoder with an input size of 1184 and a patch size of 32. As mentioned in chapter 4 we refer to the original authors' strategy of dynamic bins and set the truncation to six, this means that any patch with a count of six or over would fall in the same class. This truncation choice was made as a way, in conjunction with dynamic bins, of addressing class imbalance, though patches exist with a count of values higher than six, these areas of extreme counts, as seen in Figure 5.1, are seldom.



(a) **(b)** **(c)**

Figure 5.1: Within this patch of 64 × 64 pixels, 14 mackerels are present. (b) Area surrounding the extreme patch. (c) Patch contents.

Apart from making use of the same data augmentation techniques, the CLIP-EBC's stop condition was shared with the regression-based solutions. Though the latter converged between three and four hundred epochs, the former needed three thousand. This difference results in the training for the MCNN, CAN, and CSRNet requiring only two days each, as opposed to the full week the blockwise classification solution needed.

5.2 Evaluation Metrics

The metrics we chose, as first mentioned in chapter 1, are divided into count and localization-focused. The count metrics are the ones most customarily seen in benchmark datasets. Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left| {C_i}^{pred} - {C_i}^{ground} \right| \tag{5.1}$$

The Mean Absolute Error is the Manhattan distance between the actual and predicted counts. Though the most commonly used metric in crowd counting, it is also robust against outliers. The other metric usually paired with the MAE is the Mean Squared Error (MSE), this metric, though more punishing of large errors, is not in the same unit as the MAE and makes its interpretation more difficult, hence our choice to also calculate the RMSE.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left({C_i}^{pred} - {C_i}^{ground}\right)^2} \tag{5.2}$$

Though a good measure for comparing different models on the same dataset, MAE and RMSE, lack expressiveness when evaluating a single model. As such we also chose to calculate the Relative Mean Absolute Error (rMAE) and the Relative Root Mean Squared Error (rRMSE), as the names imply, the relative metrics are proportional to the counts, and as such, give more insight into the solution's performance on our dataset.

$$rMAE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\ C_i^{\ pred} - C_i^{\ ground}\ \right|}{C_i^{ground}} \tag{5.3}$$

$$rRMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{C_i^{\ pred} - C_i^{\ ground}}{C_i^{ground}}\right)^2} \tag{5.4}$$

Lastly, the metric we chose to employ for measuring the localization capacity of our trained models, is the Grid Average Mean Error (GAME). Calculated by computing the MAE separately over 4^L non-overlapping patches, it is the only measure we have for quantifying the variation of distribution between ground-truth and predicted density maps.

$$GAME = \frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{4^L}\left|C_{(i,l)}^{\ pred} - C_{(i,l)}^{\ ground}\right| \tag{5.5}$$

5.3 Results

After training the proposed models we calculated the metrics stated in the previous section, and their results can be seen in table 5.2.

Model	Kernel	MAE	rMAE	RMSE	rRMSE	GAME
MCNN [59]	fixed	28.8	10.5	39.3	13.7	130
	adaptive	34.2	12.3	45.3	15.6	161.2
	oriented	31.3	12.2	38	14.4	185.8
CSRNet [32]	fixed	18.2	6.6	23.5	8.8	81.4
	adaptive	**15.6**	**6**	**21.1**	**8.5**	81.6
	oriented	16.6	6.4	22.8	9.4	82.3
CAN [36]	fixed	24.7	9.6	30.7	12	72.1
	adaptive	52.1	19	56	20.1	85.3
	oriented	18.7	7.1	24.4	9	**70.6**

Table 5.2: Evaluation of regression-based approaches.

As can be seen, the model that obtains the best results in the typical benchmark metrics (MAE and RMSE) is the CSRNet model employing the adaptive kernel in its ground-truth generation process. Interestingly, though best in terms of counting errors in absolute terms and its sensitivity to outliers, this

model is outperformed by the CAN with our oriented kernel in terms of localization. These results go against our initial intuitions about the adaptive kernel's impact. Throughout this document, we referred to the fact that closeness amongst targets, does not correlate to closeness to the camera, best illustrated in Figure 4.2. However, as stated before, this is the ground-truth generation process that yielded the best results. In chapter 4 we alluded as to how a constant school density could encode the same information, so long as schools do not overlap, which coincidentally they do not in our dataset. Though we did not conduct any empirical method to determine our assumption, Pitcher et al. state that though density is not a constant throughout a school, it is typically that of one per cube of body length [41], supporting our proposition. However, it does not provide an explanation of why its use had such a negative impact on the CAN model. The choice of the kernel had seemingly little to no impact on the performance of other models, except for the case of the CAN network where it resulted in more than double the error.

Figure 5.2 displays the predicted versus actual values for the CSRNet, these reflect the metrics stated in table 5.2. The overarching trend is identical to the identity line, with a very small distance to erroneous data points. Both these facts, though better visualized in this chart, are not surprising considering the evaluated metrics, seeing as the CSRNet is consistently the model with the lowest MAE and RMSE. Much as, in the case, of the MCNN network, discussed further ahead, the kernel choice seems to have little impact on the model's predictions.

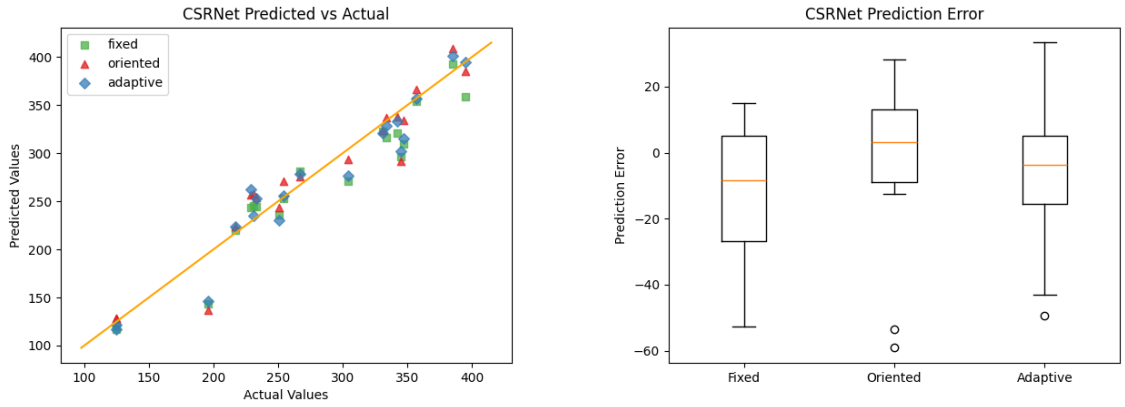


Figure 5.2: Predicted vs Actual values for model CSRNet's inference, and error distribution according to the kernel choice.

The CAN model's chart however, see figure 5.3, displays something unexpected. Though its trend is slope consistent with that of the identity line we can observe a systematic error where the model seems to consistently underpredict the school's cardinality, this is only observed in this particular model, but interestingly not across kernel choices. The kernel devised by us seems to, somewhat, correct this constant difference. This is also portrayed in the boxplot chart, where we observe that the oriented kernel error

distribution is more centered around zero and displays lower variability within its quantiles one and three.

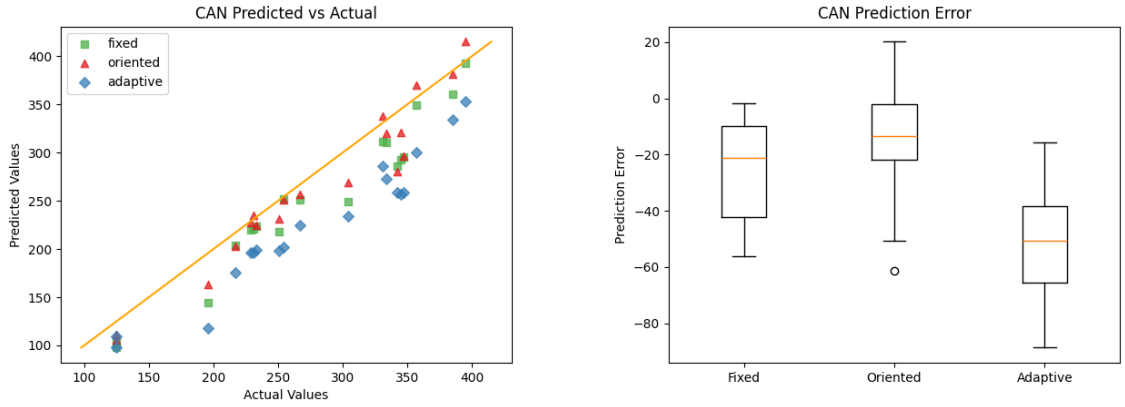


Figure 5.3: Predicted vs Actual values for each model's inference, and error distribution according to the kernel choice.

Much like in the case of the CSRNet, the MCNN model perfectly mirrors the values in table 5.2, it is the model that displays the data points farthest from the identity line, and most consistently so. It is also the model where the kernel choice shows the lowest impact., having an identical prediction error distribution across all kernels.

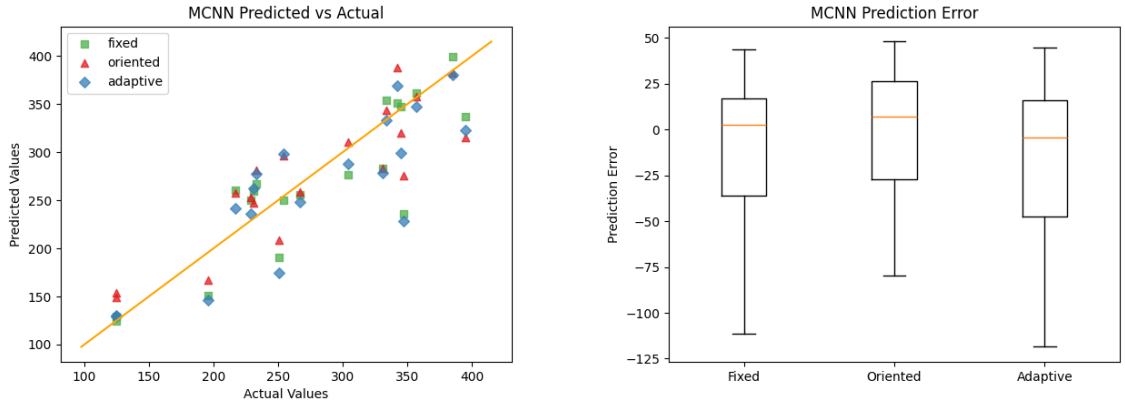


Figure 5.4: Predicted vs Actual values for model MCNN's inference, and error distribution according to the kernel choice.

Something that does distinguish the oriented kernel from the others is that it leads to overall higher estimations, regardless of the model.

As for the matter of each model's ability to accurately localize individuals, the CAN Network consistently obtains better GAME scores relative to its MAE. This may be a consequence of its inherently localizing architecture. The spatial pyramid pooling it performs is very similar to the metric's gridding

formula, where the space is iteratively subdivided into fourths. The MCNN and CSRNet however do not employ any strategy of utilizing spatial information in its training and inference steps and are therefore less performant. We believe this effect is less prevalent in the CSRNet due to its dilated convolutions, which implicitly impart some spatial information into the training process. In summary, it appears as though models that impose some sort of geometric restriction result in superior localization prowess.

Both the adaptive kernel's impact and each model's potential localization capability were expounded upon by us without any concrete and empirical proof. However, something that can be observed directly in figures 5.6 through 5.11 is the impact of the kernel choice on the inference of each model. Between the fixed and adaptive kernel, there is a clear distinction, fixed kernel results in the inferred "Gaussians" being very similar. Contrast this with the case of the adaptive kernel where there is a variance, through the usage of this kernel, the models are able in this scenario to learn the non-uniform density distribution of the school. Observed also, is the fact that all models were able to learn the shape of the oriented kernel, not only that, but as seen best in the CAN's output the models were also able to approximate their orientation.

Having observed the aforementioned figures, three major issues become apparent. The first and most obvious one is that of over-estimation, primarily and contrary to our will, over-estimation in areas where there are no individuals, mackerel or otherwise. This issue is most predominant in model MCNN where even rocks or the ripples of water on the surface of the tank are signaled as containing mackerel, even in its best predictions. Both other models present this issue, but more vestigially. This limitation of the MCNN model is mirrored in its poorer localization capacities, measured with the GAME metric.

Limitations that are observed across all observed models are those of confusion between species, not species that only share shape, such as the yellow jack which is never detected as being a mackerel, but those that also share color. We also observe a difficulty in inferring the presence of lone-wolf mackerel. Much as there is an underrepresentation of denser areas of the school, so do we observe the same characteristic when it comes to single, isolated mackerels. This can also come down to the fact that the convolutional layers learn what are the features present in a school section and not what are the features of individual fish.

Regression models as a whole performed better than what our initial intuition proposed, given the tail distribution and what are the exacerbated challenges that come from our domain setting. In table 5.3 we can see a comparison of the metrics obtained from the top models evaluated thus far and the blockwise classification model.

Model	MAE	rMAE	RMSE	rRMSE	GAME
CSRNet (adaptive)	15.6	**6**	21.1	8.5	81.6
CAN (oriented)	18.7	7.1	24.4	9	70.6
CLIP-EBC	**15.3**	6.1	**20.2**	**8.3**	**67.7**

Table 5.3: Comparison between regression-based approaches and blockwise classification's metric results.

As expected the latter is better at both counting and showcasing better localization prowess, having achieved better scores across the board, even if marginally. Notably, though the absolute metrics are marginally better, the relative metrics display the opposite relationship, with the classification model displaying a slightly worse relative score. This leads to the conclusion that this model tends to falter with a higher error in more populous scenarios. The CLIP-EBC model does seem to display a more robust inference capacity as can be observed from the fact that both squared error metrics, be it relative or absolute, are the better of the selected models and at the same time a better ability to localize the mackerel schools. This means that though perhaps not the best at estimating counts of large mackerel schools its reliability and aptness to pinpoint individuals makes it a better-suited choice for the problem we are tasked with.

If we observe Figure 5.5 we can note that the degree of error does not seem to increase in relationship with the actual school size. So the narrowly higher relative error may be a consequence of this particular train test split, with different sets purporting a slight difference in the other direction regarding the rMAE.

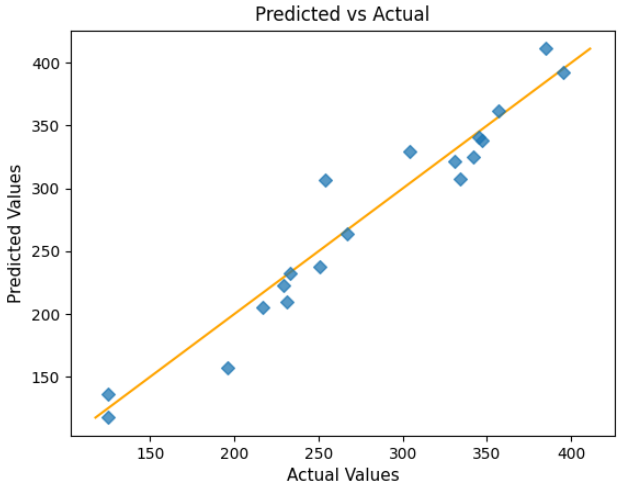


Figure 5.5: CLIP-EBC model's Predicted vs Actual values.

From the small set of CLIP-EBC's worst and best predictions, see figures 5.12 and 5.12 respectively, we can gather more evidence that the model does estimate large schools with surprising accuracy when taken into account the quality of the photographs and the amount of partial occlusions and shadowing amongst fish. Also gathered is the fact that of the limitations we observed in regression models, some seem to have been mitigated, while others are still very much present. Fish wish share a similarity with mackerels seem to be better distinguished by this method. Lone fish, however, still seem to be very hard to detect. The reason for this has to do with the process through which we arrived at the dataset statistic and how the learning signals of "single" fish may also suffer from some form of imbalance. The counts for

the dataset were obtained with a sliding window counting that has a stride of one, meaning we perform the counts over $(W_{image} - W_{sliding\_window} + 1) \times (H_{image} - H_{sliding\_window} + 1)$ windows, insert them into bins and normalize the bins. A large proportion of single-count windows are, as a result of schooling behavior, patches of the school border that only capture a single annotation. Actual isolated lone fish, which are found separate from the rest of the population, will account for a small set of the "single" fish counts. This will inherently introduce a form of error. Most training signals that refer to one fish will commonly contain features from multiple fish, this supports the results where the more isolated the fish the more likely it is that it goes uncounted.

5.4 Summary

When we take the evaluation process holistically we conclude the paradigm of performing blockwise classification with aggregation of bins, as opposed to patch-level regression, is the solution that best tackles our problem.

In every aspect, the CLIP-EBC model outperforms other compared solutions. Though having the second-best relative mean absolute error, even if marginally, it obtained the best results concerning counting and localization while being the most robust solution.

As for our kernel choice, it seems as though it has no impact on the model performance when applied to the marine domain. To summarize the conclusions drawn in previous sections, the error distribution that results from the application of the oriented kernel shares a very similar curve with other methods. The sole difference is a higher median which is closer to zero. In extreme cases the kernel proves itself to be detrimental, reason why we believe the choice of its employment is not recommended.

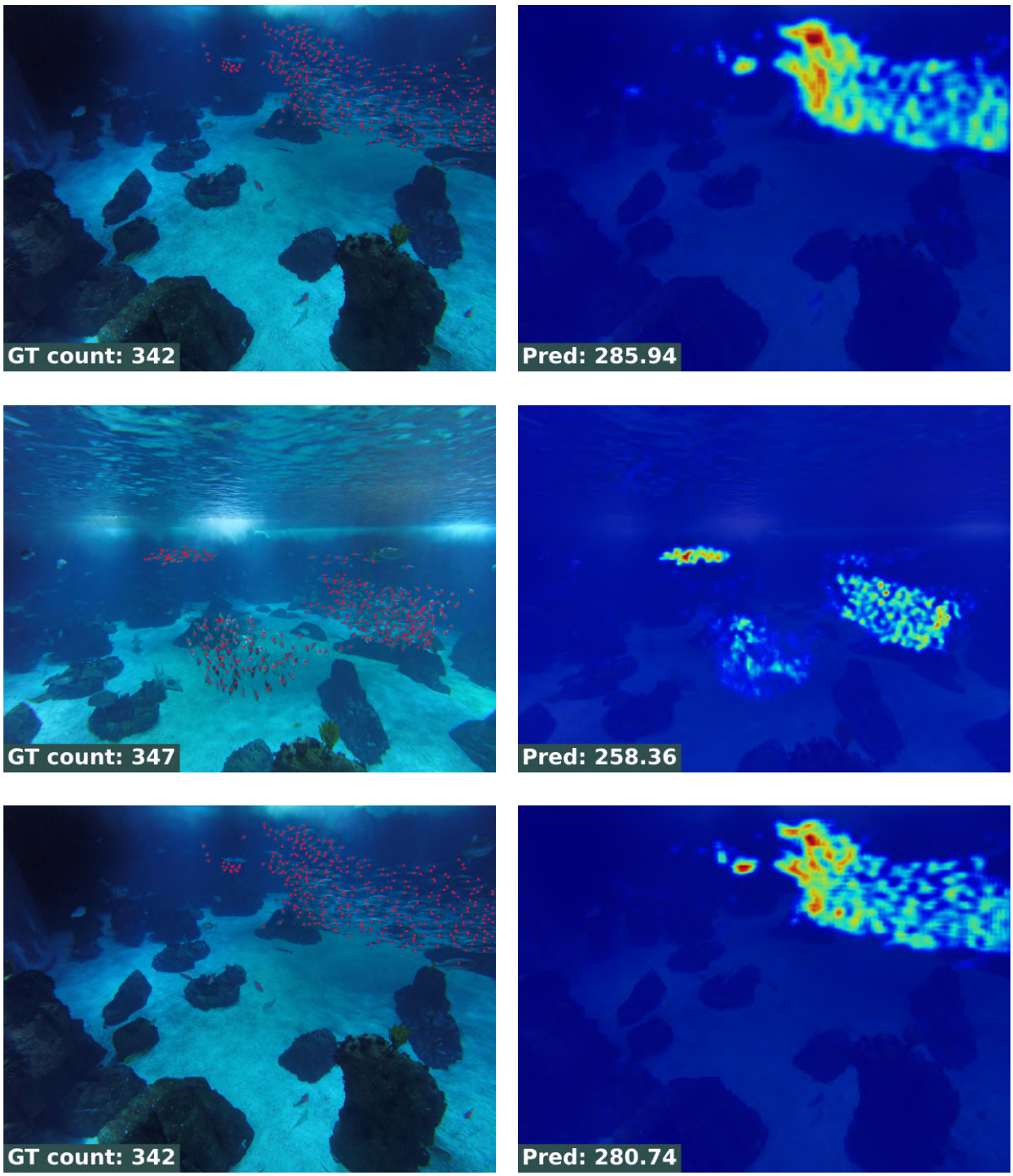


Figure 5.6: CAN model's worst predictions. Top - Fixed kernel. Middle - Adaptive kernel. Bottom - Oriented kernel.

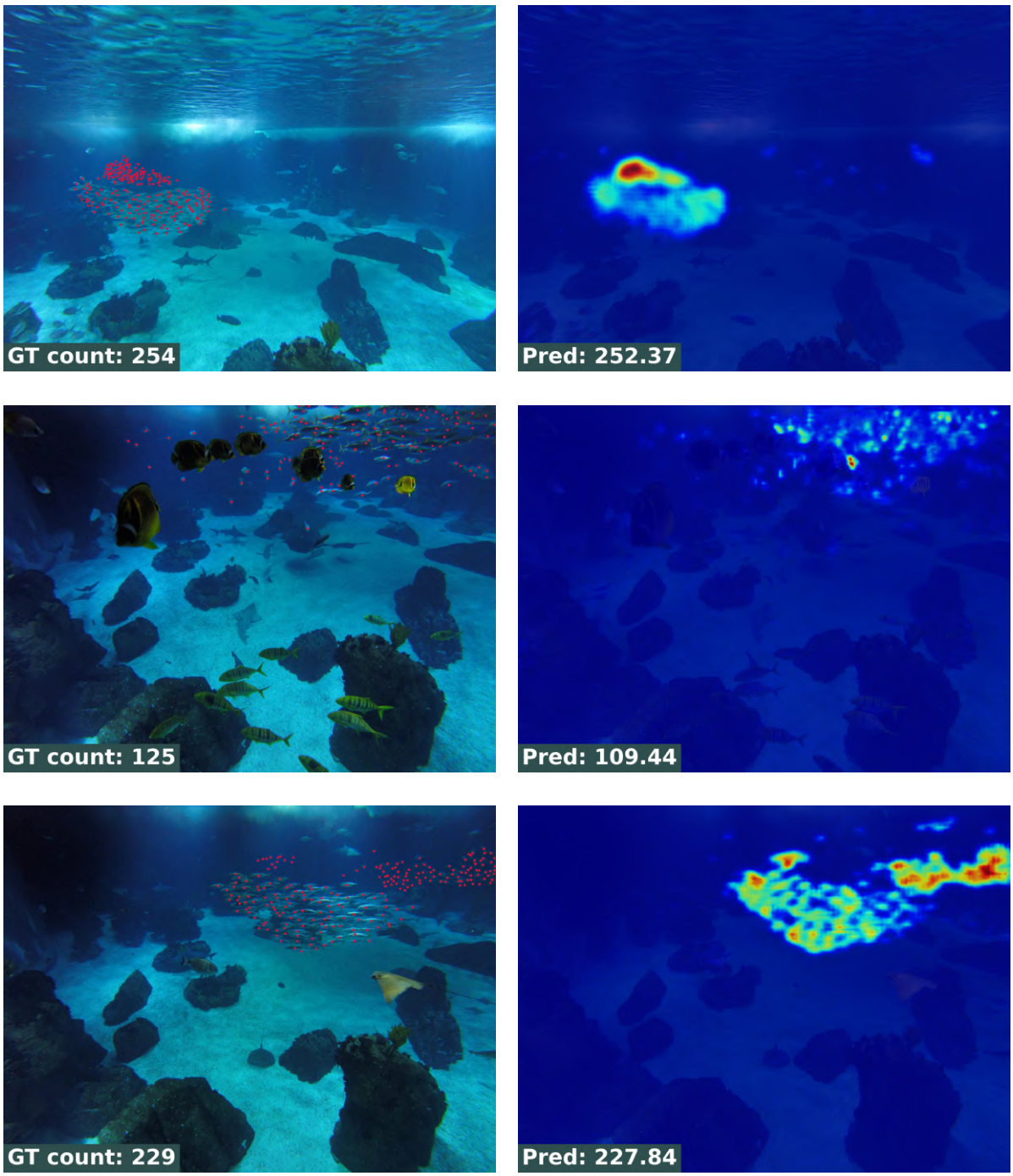


Figure 5.7: CAN model's best predictions. Top - Fixed kernel. Middle - Adaptive kernel. Bottom - Oriented kernel.

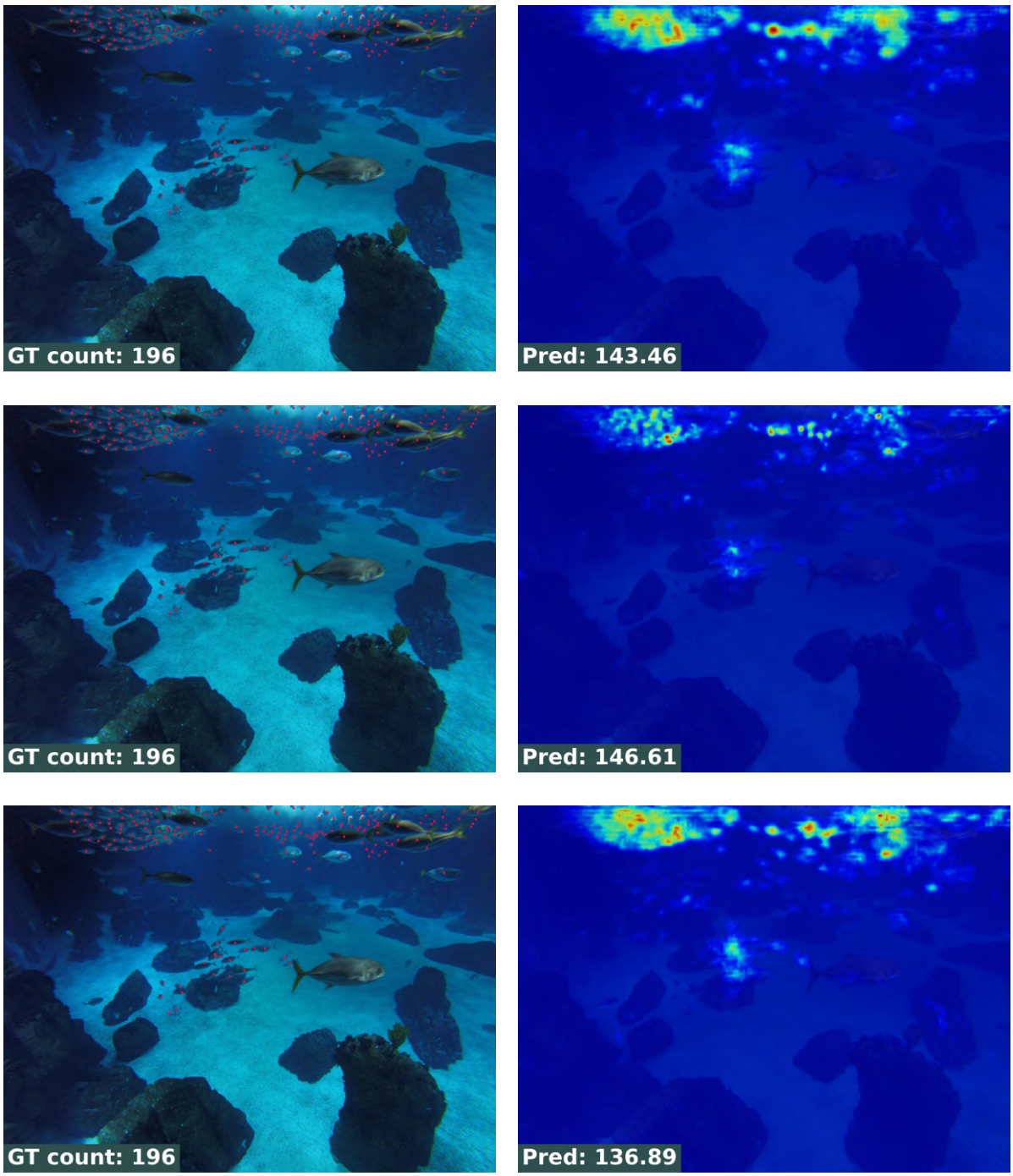


Figure 5.8: CSRNet model's worst predictions. Top - Fixed kernel. Middle - Adaptive kernel. Bottom - Oriented kernel.

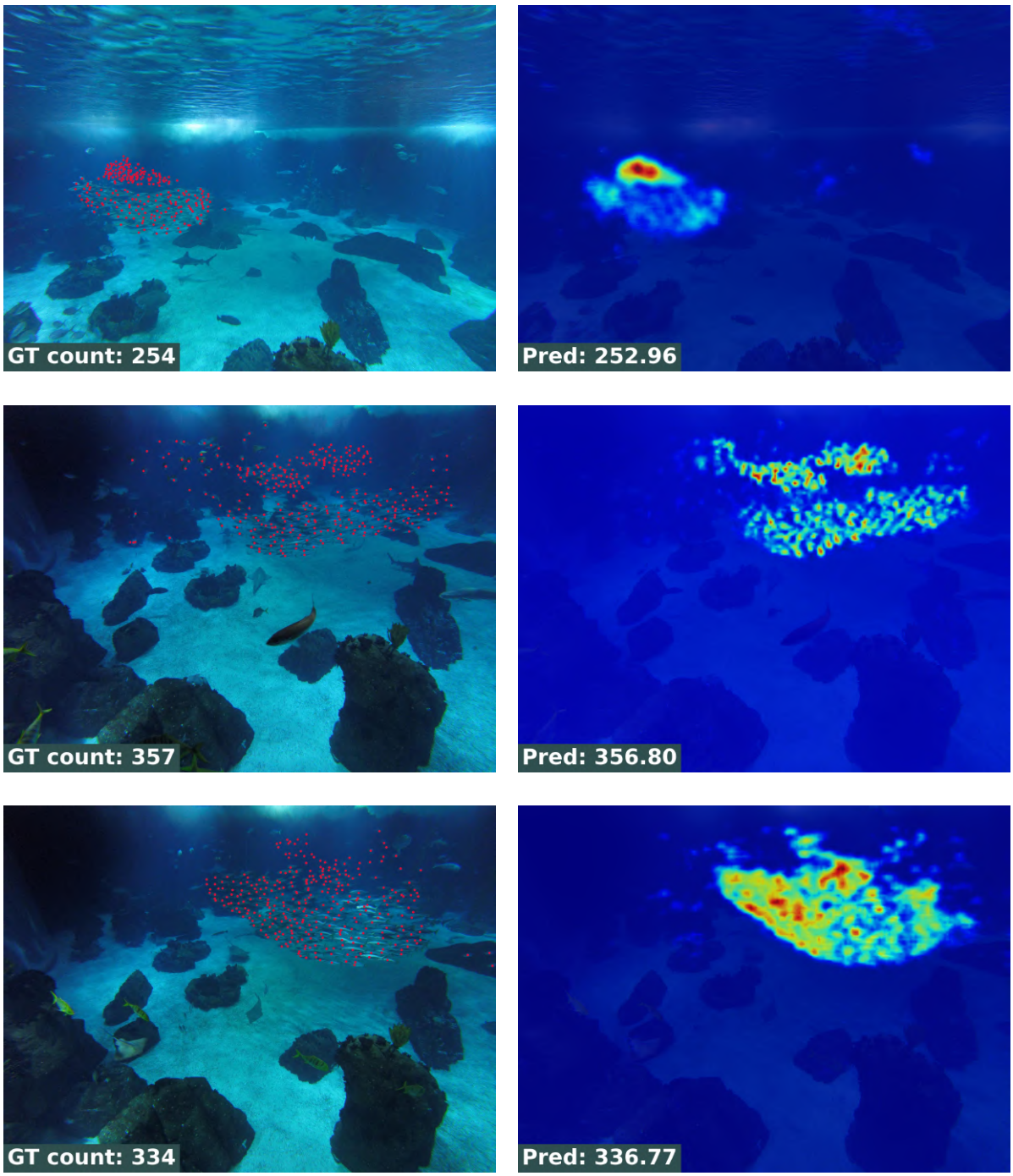


Figure 5.9: CSRNet model's best predictions. Top - Fixed kernel. Middle - Adaptive kernel. Bottom - Oriented kernel.

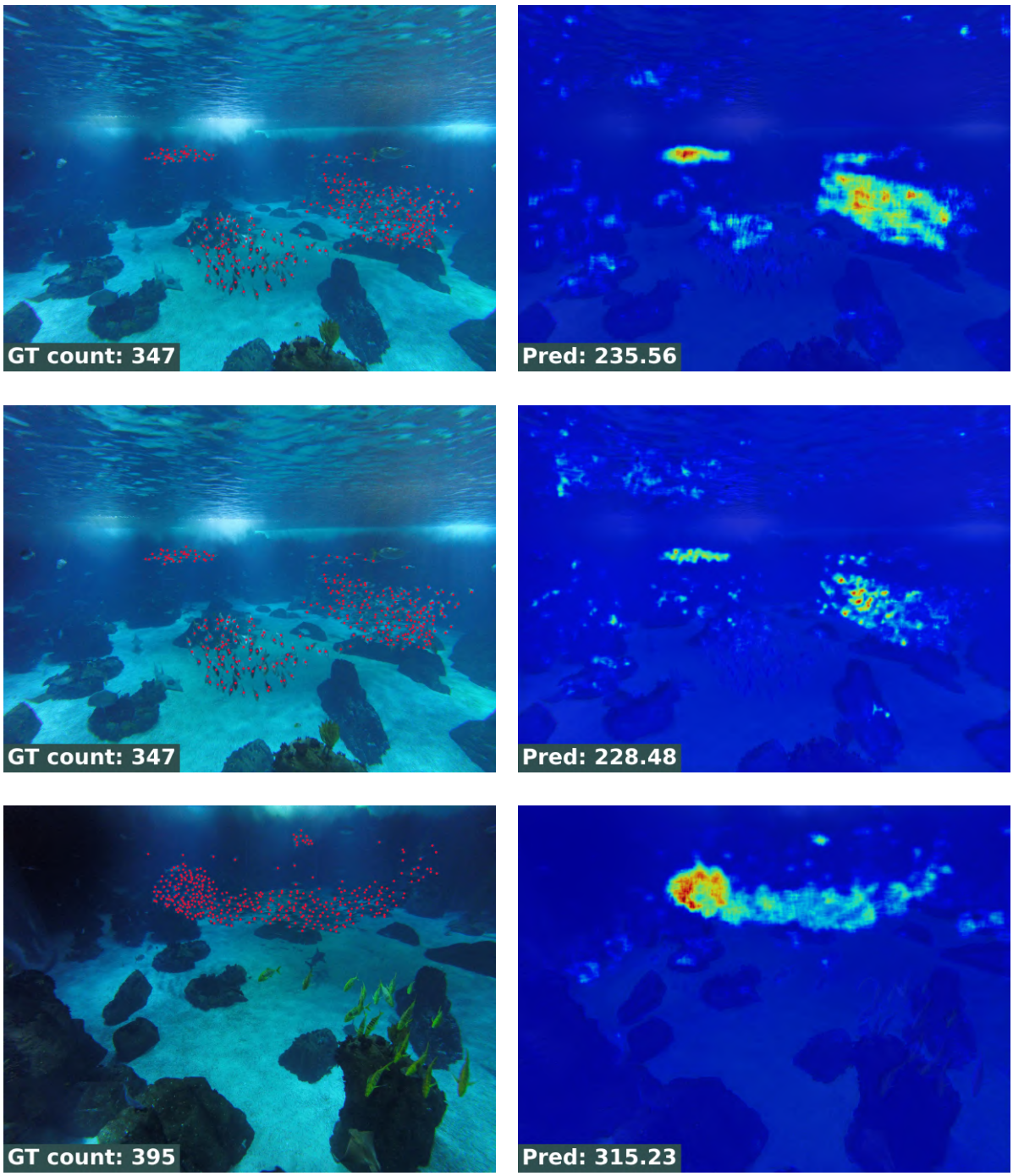


Figure 5.10: MCNN model's worst predictions. Top - Fixed kernel. Middle - Adaptive kernel. Bottom - Oriented kernel.

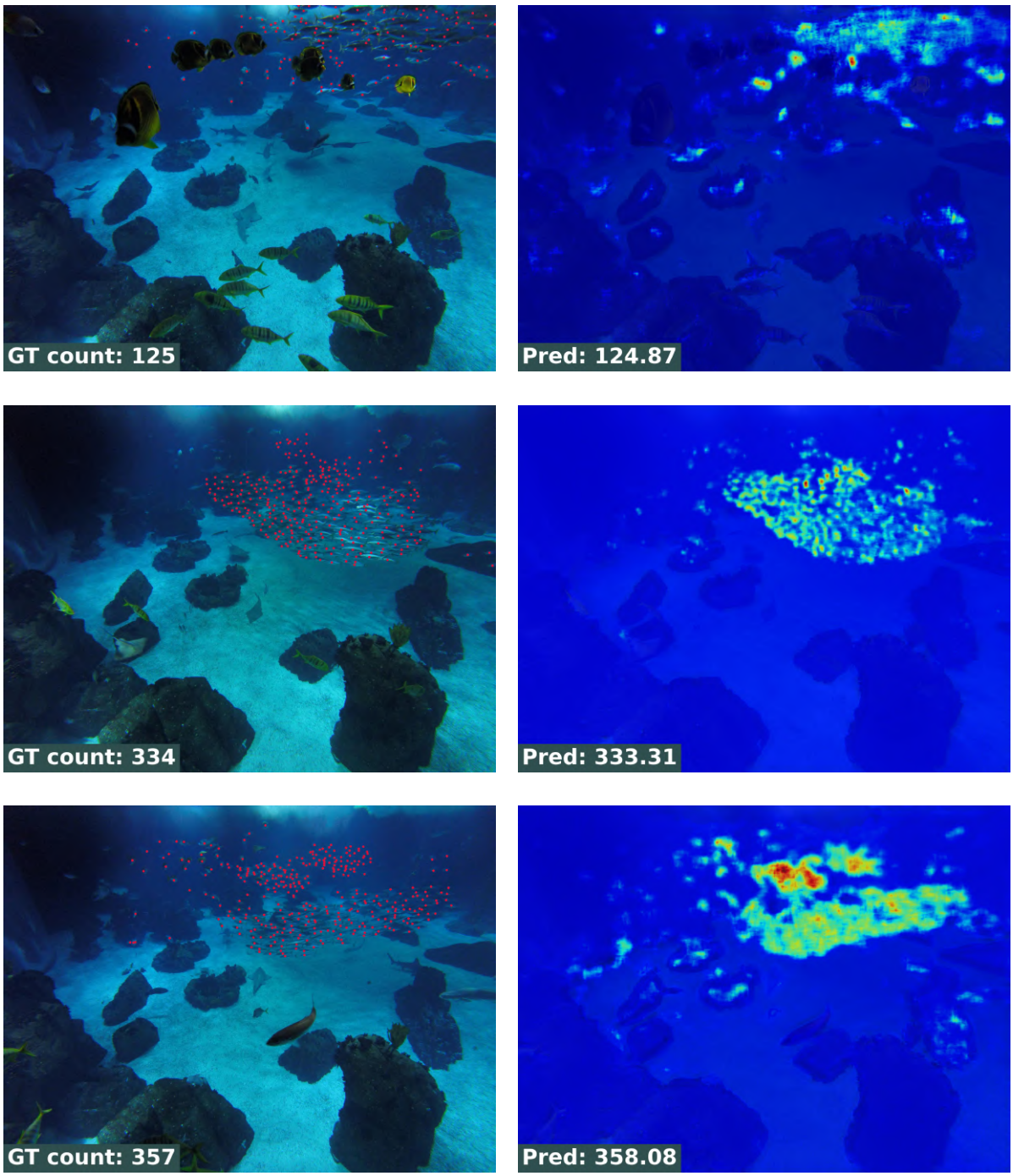


Figure 5.11: MCNN model's best predictions. Top - Fixed kernel. Middle - Adaptive kernel. Bottom - Oriented kernel.

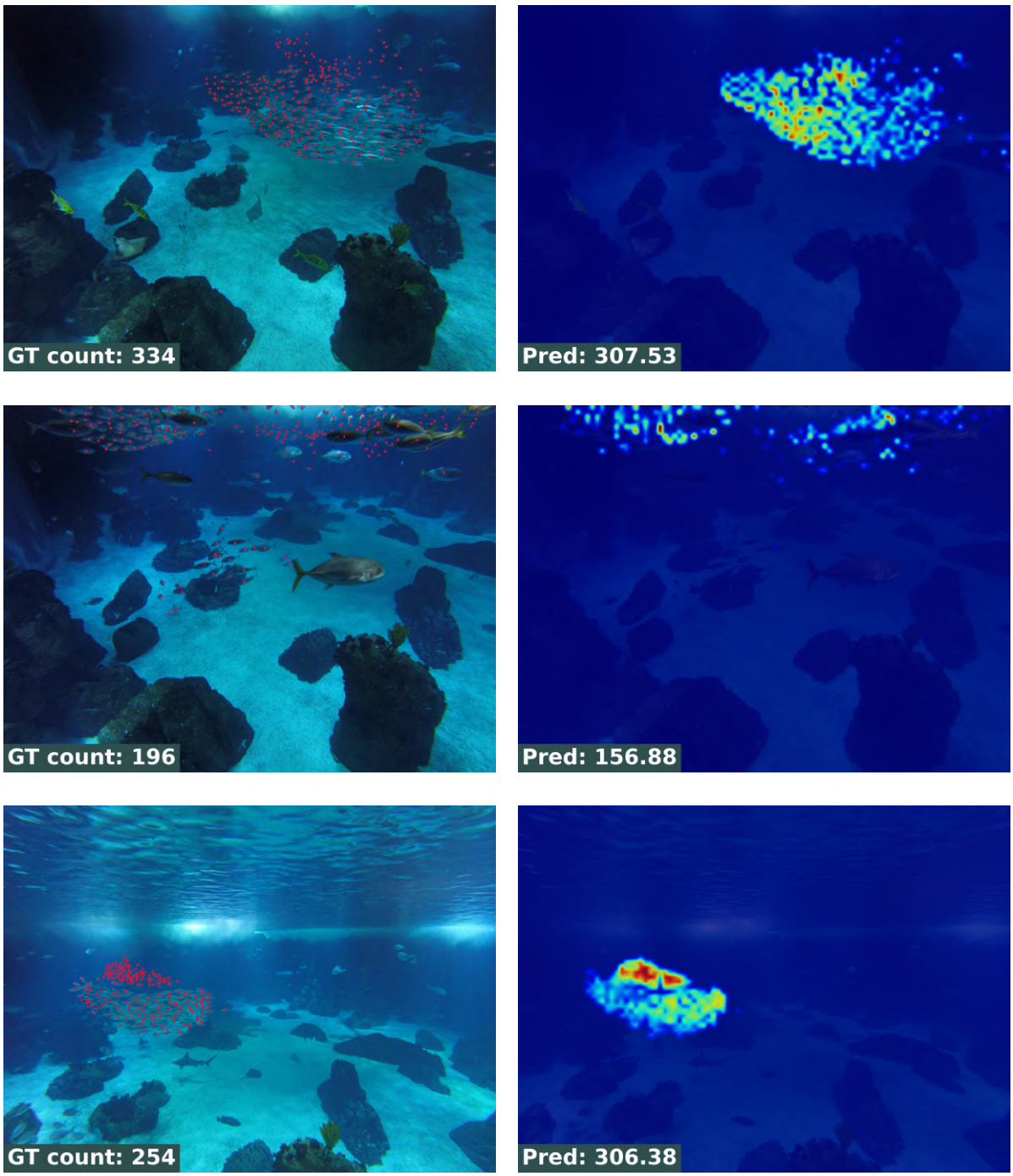


Figure 5.12: CLIP-EBC model's worst predictions.

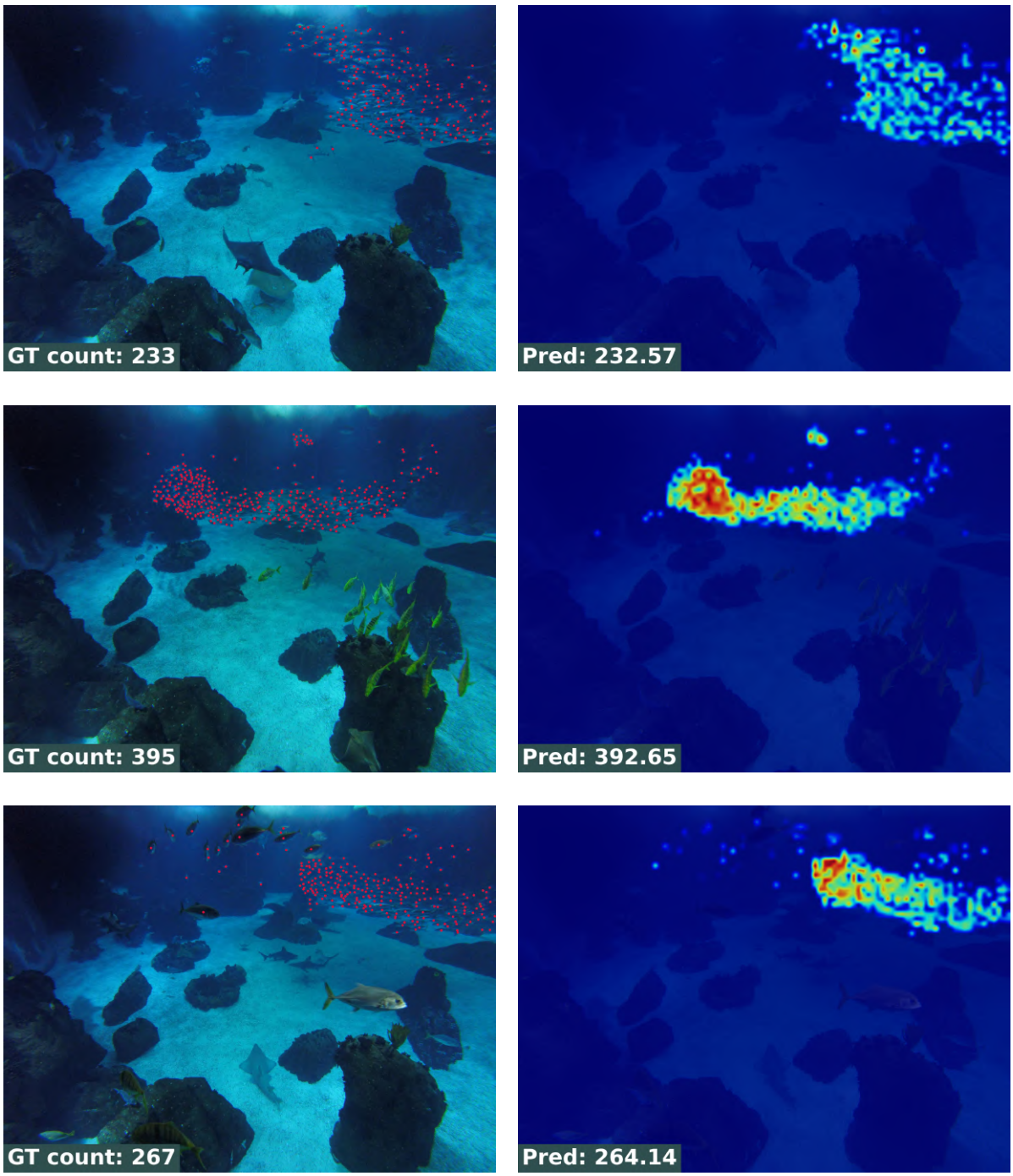


Figure 5.13: CLIP-EBC model's best predictions.

Chapter 6

Conclusion

Contents

6.1 Conclusions

In this document, we presented a multitude of different crowd-counting systems for mackerel in a public aquarium. The setting of our domain created multiple challenges to computer vision tasks in general and crowd-counting tasks in particular. From occlusions, partial and total alike, through size variance; to more domain-specific challenges such as light attenuation and blocking, color shift, oblong individuals, and similarity to other objects which should remain untallied.

The problem of providing a reliable count for a given population is of relevance to staff. It is a metric, amongst others, that allows for the proper care and maintenance of the aquarium ecosystem. It gives biologists invaluable insights into the health, sustainability, and evolution of the population.

To tackle this problem, first, we reviewed the current crowd-counting literature. The distinct shape of our targets and how they occupy the space stood out as distinguishing factors. Two different approaches were then conducted in parallel.

The first is a regression-based approach where density maps are used as ground truth. This served as the basis for testing our proposed oriented kernel. The school is divided into non-intersecting "sub-schools" and classed with a given orientation. The density map generation process leverages this information to produce a more informative learning target, where each individual is better approximated. Alongside this ground-truth generation process, we also generated density maps through the use of an adaptive kernel to test the proposition that fish position in the image plane unlike humans is not indicative of their relative position in the scene.

The other approach we used was that of a blockwise classification framework. This gave us the freedom to neglect the ground-truth generation process, as density maps are not utilized. We elected to follow this solution as it tackles the problems of erroneous annotations and skewed density distribution in the scene.

To test these approaches, a dataset of mackerel schools was required. A search for this type of data yielded no results and as such led to a collection and annotation process conducted by us. The orientation annotations we performed, for reasons discussed in this and the Evaluation chapters, though creating no real added complexity to the annotation process were not required, as the kernel did not produce meaningful results, except for the case of the CAN network where the best GAME score was obtained.

Ultimately after a comparative analysis, we selected the model we believe, and the evaluation purports, as being the best suited given our domain. We concluded that our newly devised strategy of using rotated oblong kernels for ground-truth generation showed no significant improvement on any of the calculated metrics. The use of a blockwise classification framework also showed no measurable improvement when compared with regression-based approaches, except for its localization capacity. This was surprising as we expected a greater improvement across all metrics

6.2 Future Work

After an overview of the models' outputted density maps, some limitation were made apparent. First the problems of confusion between similar species and other features in the images such as ripples on the water surface or rocks on the aquarium floor. So too surfaced the problem of lone-wolf, singular fish going uncounted. Both these issues could be solved by extending the current dataset. Deep learning models are famous for requiring larger amounts of truth data, therefore, an increase in data points should produce more capable models. Not only would a larger dataset allow for further training of the models, but it would also allow for a better analysis. In Chapter 5 we present the figures with the residual errors alongside their distribution, a larger dataset would allow us to get a better more descriptive, and most importantly more realistic view of each model's performance.

Still, regarding conducting a fairer comparative analysis we propose that in the future, a larger time frame would allow for more training runs of each model. In this document, our evaluation, due to computational and time constraints, boiled down to training each model until the best hyperparameters were found and then presenting their metrics. This methodology though commonly used in the crowd counting task, is not without flaws. The information gained is that each model performed best given that data split. When metric values are close to each other it may come down to the randomness of training as opposed to actual model performance. We propose the use of a more sound evaluation methodology where several runs of k-fold cross-validation are conducted for each model and then follow a Bayesian Analysis proposed by Benavoli et al. [5].

We close this document with fair results in crowd counting and localization, however, we do identify two further experiments that could potentially further improve results.

First, is the application of another crowd-counting paradigm, discussed in Chapter 2, that of a purely point-based approach. Much like the CLIP-EBC model, this approach is not constrained to the noisy intermediary representation of a density map, and neither does it suffer from problems of blockwise classification where classes are undersampled and items may be spliced.

The second. would be to modify the CLIP-EBC so that the image encoder is trained through Visual Prompt Tuning (VPT), an alternative to full tuning proposed by Jia et al. [27], where the model backbone is kept frozen and the parameters in the input space are trained, overall only one percent of the parameters are trained in many cases VPT outperforms full fine-tuning.

Bibliography

[1] Gonçalo Adolfo. Fish behavior detection through video frames and trajectories. Master's thesis, Universidade de Lisboa, Instituto Superior Técnico, Lisbon, Portugal, October 2021.

[2] Robert Azencott, Jia-Ping Wang, and Laurent Younes. Texture classification using windowed fourier filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):148–153, 1997.

[3] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5744–5752, 2017.

[4] Sergey Bartunov, Adam Santoro, Blake Richards, Luke Marris, Geoffrey E Hinton, and Timothy Lillicrap. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. *Advances in neural information processing systems*, 31, 2018.

[5] Alessio Benavoli, Giorgio Corani, Janez Demšar, and Marco Zaffalon. Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. *Journal of Machine Learning Research*, 18(77):1–36, 2017.

[6] Lokesh Boominathan, Srinivas SS Kruthiventi, and R Venkatesh Babu. Crowdnet: A deep convolutional network for dense crowd counting. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 640–644, 2016.

[7] José Castelo. Video based live tracking of fishes in tanks. Master's thesis, Universidade de Lisboa, Instituto Superior Técnico, Lisbon, Portugal, November 2019.

[8] Antoni B Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–7. IEEE, 2008.

[9] Antoni B Chan and Nuno Vasconcelos. Bayesian poisson regression for crowd counting. In *2009 IEEE 12th international conference on computer vision*, pages 545–551. IEEE, 2009.

[10] I Chen, Wei-Ting Chen, Yu-Wei Liu, Ming-Hsuan Yang, Sy-Yen Kuo, et al. Improving point-based crowd counting and localization based on auxiliary point guidance. *arXiv preprint arXiv:2405.10589*, 2024.

[11] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. Feature mining for localised crowd counting. In *Bmvc*, volume 1, page 3, 2012.

[12] Xinya Chen, Yanrui Bin, Nong Sang, and Changxin Gao. Scale pyramid network for crowd counting. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1941–1950. IEEE, 2019.

[13] Zhi-Qi Cheng, Qi Dai, Hong Li, Jingkuan Song, Xiao Wu, and Alexander G Hauptmann. Rethinking spatial invariance of convolutional networks for object counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19638–19648, 2022.

[14] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3642–3649. IEEE, 2012.

[15] Rod M Connolly, David V Fairclough, Eric L Jinks, Ellen M Ditria, Gary Jackson, Sebastian Lopez-Marcano, Andrew D Olds, and Kristin I Jinks. Improved accuracy for automated counting of a fish in baited underwater videos for stock assessment. *Frontiers in Marine Science*, 8:658135, 2021.

[16] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] Ellen M Ditria, Sebastian Lopez-Marcano, Michael Sievers, Eric L Jinks, Christopher J Brown, and Rod M Connolly. Automating the analysis of fish abundance using object detection: optimizing animal ecology with deep learning. *Frontiers in Marine Science*, page 429, 2020.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[20] Gamaleldin Elsayed, Prajit Ramachandran, Jonathon Shlens, and Simon Kornblith. Revisiting spatial invariance with low-rank local connectivity. In *International Conference on Machine Learning*, pages 2868–2879. PMLR, 2020.

[21] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[22] Junyu Gao, Qi Wang, and Xuelong Li. Pcc net: Perspective crowd counting via spatial convolutional network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10):3486–3498, 2019.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[24] CK Hemelrijk, DAP Reid, H Hildenbrandt, and JT Padding. The increased efficiency of fish swimming in a school. *Fish and Fisheries*, 16(3):511–521, 2015.

[25] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2547–2554, 2013.

[26] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–546, 2018.

[27] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022.

[28] Xiaolong Jiang, Zehao Xiao, Baochang Zhang, Xiantong Zhen, Xianbin Cao, David Doermann, and Ling Shao. Crowd counting and density estimation by trellis encoder-decoder networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6133–6142, 2019.

[29] Yinjie Lei, Yan Liu, Pingping Zhang, and Lingqiao Liu. Towards using count-level weak supervision for crowd counting. *Pattern Recognition*, 109:107616, 2021.

[30] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 878–885. IEEE, 2005.

[31] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. *Advances in neural information processing systems*, 23, 2010.

[32] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1091–1100, 2018.

[33] Dingkang Liang, Xiwu Chen, Wei Xu, Yu Zhou, and Xiang Bai. Transcrowd: weakly-supervised crowd counting with transformers. *Science China Information Sciences*, 65(6):160104, 2022.

[34] Zhe Lin and Larry S Davis. Shape-based human detection and segmentation via hierarchical part-template matching. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):604–618, 2010.

[35] Liang Liu, Hao Lu, Haipeng Xiong, Ke Xian, Zhiguo Cao, and Chunhua Shen. Counting objects by blockwise classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10):3513–3527, 2019.

[36] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5099–5108, 2019.

[37] Yiming Ma, Victor Sanchez, and Tanaya Guha. Clip-ebc: Clip can count accurately through enhanced blockwise classification. *arXiv preprint arXiv:2403.09281*, 2024.

[38] A NSAV Marana, Sergio A Velastin, L da F Costa, and RA Lotufo. Automatic estimation of crowd density using texture. *Safety Science*, 28(3):165–175, 1998.

[39] Krithika M Pai, KB Ajitha Shenoy, and MM Manohara Pai. A computer vision based behavioral study and fish counting in a controlled environment. *IEEE Access*, 10:87778–87786, 2022.

[40] Viet-Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3253–3261, 2015.

[41] TJ Pitcher and BL Partridge. Fish school density and volume. *Marine Biology*, 54:383–394, 1979.

[42] Narinder Singh Punn, Sanjay Kumar Sonbhadra, Sonali Agarwal, and Gaurav Rai. Monitoring covid-19 social distancing with person detection and tracking via fine-tuned yolo v3 and deepsort techniques. *arXiv preprint arXiv:2005.01385*, 2020.

[43] Payam Sabzmeydani and Greg Mori. Detecting pedestrians by learning shapelet features. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

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

[45] Vishwanath A Sindagi, Rajeev Yasarla, and Vishal M Patel. Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2594–2609, 2020.

[46] Qingyu Song, Changan Wang, Zhengkai Jiang, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yang Wu. Rethinking counting and localization in crowds: A purely point-based framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3365–3374, 2021.

[47] João Teixeira. Tracking animals in underwater videos. Master's thesis, Universidade de Lisboa, Instituto Superior Técnico, Lisbon, Portugal, December 2020.

[48] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[49] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57:137–154, 2004.

[50] Elad Walach and Lior Wolf. Learning to count with cnn boosting. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 660–676. Springer, 2016.

[51] Jia Wan and Antoni Chan. Modeling noisy annotations for crowd counting. *Advances in Neural Information Processing Systems*, 33:3386–3396, 2020.

[52] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai Nguyen. Distribution matching for crowd counting. *Advances in neural information processing systems*, 33:1595–1607, 2020.

[53] Yi Wang and Yuexian Zou. Fast visual object counting via example-based density estimation. In *2016 IEEE international conference on image processing (ICIP)*, pages 3653–3657. IEEE, 2016.

[54] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247, 2007.

[55] Bolei Xu and Guoping Qiu. Crowd density estimation based on rich features and random projection forest. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016.

[56] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[57] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 833–841, 2015.

[58] Song Zhang, Xinting Yang, Yizhong Wang, Zhenxi Zhao, Jintao Liu, Yang Liu, Chuanheng Sun, and Chao Zhou. Automatic fish population counting by machine vision and a hybrid deep neural network model. *Animals*, 10(2):364, 2020.

[59] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016.

[60] Tao Zhao, Ram Nevatia, and Bo Wu. Segmentation and tracking of multiple humans in crowded environments. *IEEE transactions on pattern analysis and machine intelligence*, 30(7):1198–1211, 2008.