Statistical Inference of Fish Populations from Deep-Learning Data

VII Iberian Modelling Week

November 26 - December 1, 2021

Authors:

Raúl Almuzara
Pablo Femenía

Supervisors:

José Luis Lisani
Bartomeu Coll

Contents

1 Introduction

The *Deep-Ecomar* project, supported by the *Govern de les Illes Balears*, seeks to estimate fish populations with the help of deep learning techniques.

An instance segmentation convolutional neural network (Mask R-CNN) has been trained to detect different fish species in sub-aquatic images. From the output of this network, the number of specimens TD of a given species can be obtained. However, this value may not match the actual number of fish A in the scene, due to: a) missed or mislabeled detections (false negatives); b) wrongly detected fish (false positives).

In order to assess the performance of a neural network, the ratio of true positives to the total number of detections (Precision, P) and the ratio of true positives to the total number of fish (Recall, R) are computed for a set of images S. Ideally, if this set of images is representative of the actual data on which the network is applied, these values can be used to estimate the number of false positives and false negatives in a scene.

The goal of this problem consists of giving an estimate of A (the actual number of fish of a given species in a scene), knowing the output TD of the network (number of fish of this species detected by the network) and the precision (P) and recall (R) ratios.

We will present some methods to obtain realistic estimations of A in an interval by studying the variables from a probabilistic and statistical point of view. We will study the properties of the variables and propose models that could be applied to obtain useful information. We study the relationship between the variables, methods of nonparametric inference and propose distributions for the variables in order to apply Bayesian modelling and carry out simulations.

This report has been written during the *VII Iberian Modelling Week*, which has been held from November 26 to December 1, 2021.

2 Problem statement

The problem is the estimation of fish populations. For that purpose, we obtain hundreds of underwater images where we can see multiple fish. For a specific photo, anyone can manually count the number of fish. This would be a first simple solution to the problem.

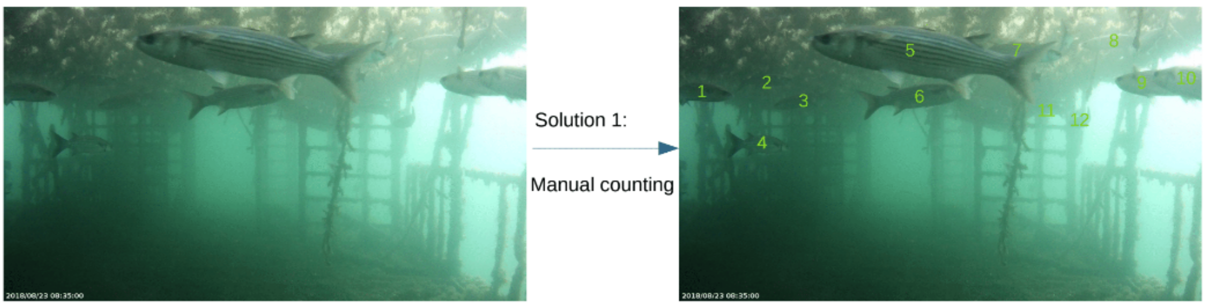

Figure 1: Given an underwater image, a person can count the fish to obtain the population.

Nevertheless, this is a time-consuming task and it is subject to human error.

A Mask R-CNN has been trained with labeled underwater images of fish. Obtaining the training data still requires that someone labels the photos manually by marking all the fish. However, after the neural network has been properly trained, it should automatically predict the number of fish in every new image that it has not seen before.

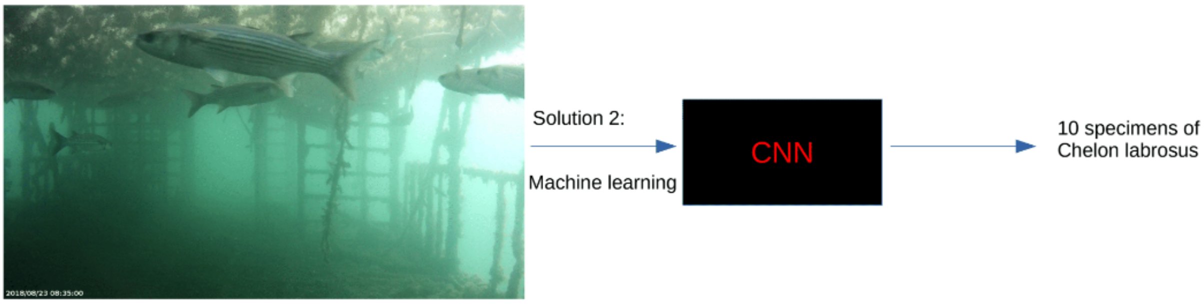

Figure 2: The trained convolutional neural network takes a new underwater image and tries to detect all the fish.

During the process of training, the CNN gets the value of its parameters by learning the information of the training dataset. This set contains images and labels.

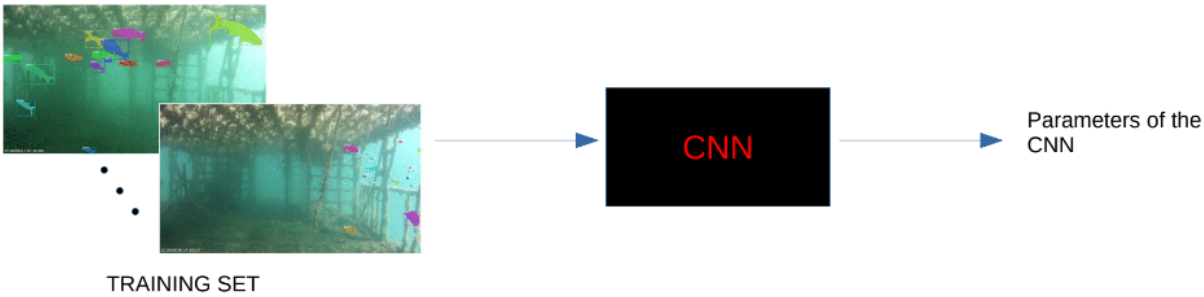

Figure 3: The training set contains labeled images, so the CNN can learn and adjust its parameters.

Then, its performance is measured with a validation set to see how well it generalizes to other images.

Figure 4: The performance of the CNN is tested with the validation set.

After this process, the CNN should be automatic and fast. However, it can still make mistakes. The objective of this small project is to study ways to estimate and bound the error of the predictions of the CNN. After a prediction, we would want to give a range of values where the correct answer should be, associate a credible interval to the result and take into account labeling errors. Undoubtedly, having wrong labels affects the quality of the predictions. Nevertheless, in this project, we will assume that the amount of wrong labels is negligible and we will concentrate on suggesting models and methods to quantify prediction errors and estimate the reliability of the information given by the CNN as an output.

3 Description of variables

We present some classic definitions in machine learning which are interesting from the point of view of statistics.

- TP (True Positives): *Number of correct detections.*
- FP (False Positives): *Number of wrong detections.*
- FN (False Negatives): *Number of missed detections.*
- TD: *Total number of detections given as an output of the CNN.* Note that

$$TD = TP + FP \tag{1}$$

.

- A: *Actual number of fish considered as ground truth.* Note that

$$A = TP + FN \tag{2}$$

.

- P (Precision):

$$P = \frac{TP}{TD} = \frac{TP}{TP + FP} \tag{3}$$

- R (Recall):

$$R = \frac{TP}{A} = \frac{TP}{TP + FN} \tag{4}$$

3.1 Example

The CNN works with images of fish. For instance, specimens of *Chelon labrosus* (thicklip grey mullet). The images could look like the following schematic illustration:

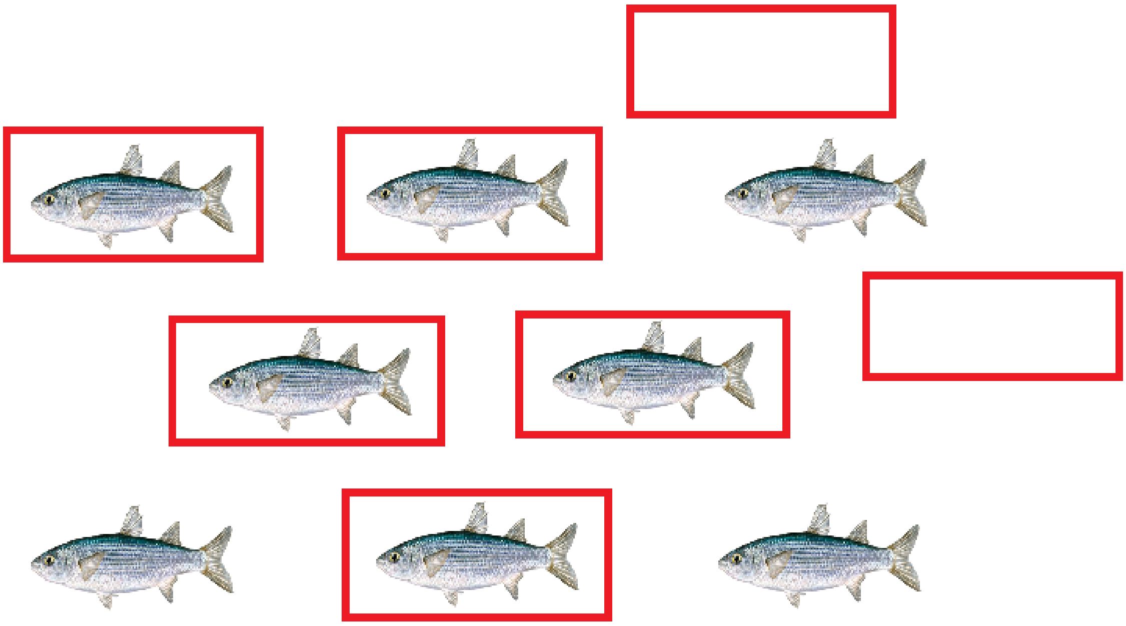

Figure 5: Example of detections of the CNN.

The red rectangles represent the places where the CNN thinks there is a fish. In this case, the value for the aforementioned variables would be the following:

- $TP = 5$ (there are 5 fish that have been correctly detected)
- $FP = 2$ (there are 2 places where the CNN thought there were fish, but there were not)
- $FN = 3$ (there are 3 fish that have not been detected)
- $TD = TP + FP = 5 + 2 = 7$ (in total, there have been 7 detections)
- $A = TP + FN = 5 + 3 = 8$ (in total, there are 8 fish)
- $P = TP/TD = 5/7 = 0.714$
- $R = TP/A = 5/8 = 0.625$

4 First approach to the value of A

4.1 *Naïve* model

The CNN gives the total number of detections TD as an output. Suppose we know the exact values of the precision P and the recall R. Then,

$$\frac{P}{R} = \frac{\frac{TP}{TD}}{\frac{TP}{A}} = \frac{A}{TD} \implies A = TD \cdot \frac{P}{R} \tag{5}$$

We want to compute A from the value of TD, P and R. If we were interested in a naive approximation of A for a certain image, it would be enough to plug in the three values in the formula above.

4.2 Extension of the model

Obviously, we can do better because we have not taken into account errors. Even if we assume that the CNN is fed with images similar to the ones that were used for training, it will not always make perfect predictions. We may consider, at least, two sources of error: mislabeled data and prediction errors.

Assuming that P and R have some uncertainty (δP and δR, respectively), we perform an error propagation to build a slightly more advanced model:

$$A = TD \cdot \frac{P}{R} \implies \delta A = TD \cdot \delta \left(\frac{P}{R} \right) \tag{6}$$

Calling $f = \frac{P}{R}$, we need to compute δf:

$$\delta f = \sqrt{\left(\frac{\partial f}{\partial P} \cdot \delta P \right)^2 + \left(\frac{\partial f}{\partial R} \cdot \delta R \right)^2} = \sqrt{\left(\frac{1}{R} \cdot \delta P \right)^2 + \left(-\frac{P}{R^2} \cdot \delta R \right)^2} = \tag{7}$$

$$= \frac{1}{R^2} \sqrt{(R \cdot \delta P)^2 + (P \cdot \delta R)^2} \tag{8}$$

Thus, the uncertainty associated to A is

$$\delta A = \frac{TD}{R^2} \sqrt{(R \cdot \delta P)^2 + (P \cdot \delta R)^2} \tag{9}$$

Now, the problem would be the calculation of the errors δP and δR. We should be able to estimate them or, at least, bound them with the necessary assumptions.

5 Nonparametric inference

One approach to this problem could make use of the methods of nonparametric inference. We do not make any assumptions about the random variable with which we are working. Let

$X_1, \ldots, X_n$ be random variables with cumulative distribution function F and we define the empirical distribution as

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \chi_{[X_i,\infty)}(x) \tag{10}$$

The Dvoretzky-Kiefer-Wolfowitz (DKW) inequality states that, for any $\epsilon > 0$,

$$\mathbb{P}\left(\sup_x |F(x) - F_n(x)| > \epsilon\right) \leq 2e^{-2n\epsilon^2} \tag{11}$$

The DKW inequality can be used to obtain a confidence band. If the DKW inequality holds, then

$$\mathbb{P}\left(|F(x) - F_n(x)| \leq \epsilon\right) = 1 - \mathbb{P}\left(\sup_x |F(x) - F_n(x)| > \epsilon\right) \geq 1 - 2e^{-2n\epsilon^2} \tag{12}$$

Let

$$L(x) = \max\{F_n(x) - \epsilon, 0\} \tag{13}$$

$$U(x) = \min\{F_n(x) + \epsilon, 1\} \tag{14}$$

$$\epsilon = \sqrt{\frac{1}{2n}\log\left(\frac{2}{\alpha}\right)} \tag{15}$$

Then, $\mathbb{P}(L(x) \leq F(x) \leq U(x) \text{ for all } x) \geq 1 - \alpha$. This inequality justifies the use of the empirical distribution to approximate the real distribution.

In order to give an interval for A given a certain TD, we can use a probability interval for P/R, R/P or P and R using the relationship $A = TD \cdot \frac{P}{R}$. In what follows, X can be P/R, R/P, P or R, considering them as random variables that take a random possible image with fish and give the corresponding output for that particular image. We can then use the error formula given in the Equation (9) to obtain the desired probability interval for A.

To construct such an interval, we would like to find C and D such that $\mathbb{P}(X \in (C, D)) \geq 1-\beta$ for an arbitrary β. We know that $\mathbb{P}(X \in (C, D)) = \mathbb{P}(F(X) \in (F(C), F(D)))$. It is clear that, if F is continuous and strictly increasing, $F(X) \sim U(0, 1)$. But, as we do not know F, we cannot use this information yet. Let us apply the confidence band constructed before and the definition of conditional probability:

$$\mathbb{P}\Big(F(X) \in (F(C), F(D))\Big) \geq \\ \mathbb{P}\Big(F(X) \in (F(C), F(D)) | \forall x, L(x) \leq F(x) \leq U(x)\Big)\mathbb{P}\Big(\forall x, L(x) \leq F(x) \leq U(x)\Big). \tag{16}$$

But

$$\mathbb{P}\Big(F(X) \in (F(C), F(D)) | \forall x, L(x) \leq F(x) \leq U(x)\Big) \geq \mathbb{P}\Big(F(X) \in (U(C), L(D))\Big) \tag{17}$$

We have computed L and U so, because $F(X)$ is a $U(0, 1)$ distribution,

$$\mathbb{P}\Big(F(X) \in (F(C), F(D)) | \forall x, L(x) \leq F(x) \leq U(x)\Big) \geq \max\{L(D) - U(C), 0\} \tag{18}$$

Merging all this information, we obtain

$$\mathbb{P}\Big(X \in (C, D)\Big) \geq \max\{L(D) - U(C), 0\}(1 - \alpha). \tag{19}$$

If we find α, C, D such that

$$\max\{L(D) - U(C), 0\}(1 - \alpha) \geq 1 - \beta, \tag{20}$$

we have found a confidence interval for X. We must avoid the situation in which $L(D) - U(C)$ is negative.

One criteria for finding D and C could be to obtain an interval "centered" at the median (we do not know it exactly). This could be attained by imposing $L(D) = 1 - U(C)$. The problem then is to find D with $(2L(D) - 1)(1 - \alpha) \geq 1 - \beta$ for the significance level desired. (This inequality could be developed as $L(D)$ depends on α). $L(D) = F_n(D) - \epsilon$ so

$$(2L(D) - 1)(1 - \alpha) \geq 1 - \beta \tag{21}$$

$$(2F_n(D) - 2\epsilon - 1)(1 - \alpha) \geq 1 - \beta \tag{22}$$

$$2F_n(D)(1 - \alpha) + (2\epsilon + 1)(\alpha - 1) \geq 1 - \beta \tag{23}$$

$$F_n(D) \geq \frac{1 - \beta}{2(1 - \alpha)} + \frac{1}{2} + \epsilon \tag{24}$$

6 Bayesian methods

Consider that we have a set of data D and a certain parameter α that we would like to study. Bayes' Theorem gives us a way to calculate conditional probabilities. In this case, knowing that the data D takes certain values, we would like to know the distribution of α:

$$\mathbb{P}(\alpha|D) = \frac{\mathbb{P}(D|\alpha) \cdot \mathbb{P}(\alpha)}{\mathbb{P}(D)} \tag{25}$$

Although the α is a fixed parameter and thus, according to the frequentist paradigm, the probability that α coincides with its real value is 1 and the probability that α is not α is 0, the Bayesian approach proposes to use the probability $\mathbb{P}(\alpha)$ to express our beliefs about the parameter α. This probability distribution is called the **prior** distribution. Using Bayes' Theorem, after obtaining the data D, we can update our beliefs about α and, in this way, make a more informed inference about α. The distribution of probability given by $\mathbb{P}(\alpha|D)$ shall be called the **posterior** distribution. Once we have obtained a posterior distribution, we could give a Bayesian estimator of α using the mean of α.
More information about Bayesian Inference can be consulted on [2].

Also, we could create a credible interval. For the case of continuous random variables, a $\gamma \cdot 100\%$ credible interval is an interval I such that

$$\int_I f(\alpha|D) \; \mathrm{d}\alpha = \gamma \tag{26}$$

where γ is a credibility level and $f(\alpha|D)$ is the posterior probability density function. It will also be useful to define the concept of Highest Density Interval (HDI), which is an interval C of the form

$$C = \{\alpha : f(\alpha|D) \geq k\} \tag{27}$$

where k is the largest constant such that

$$\int_C f(\alpha|D)\ \mathrm{d}\alpha \geq \gamma \tag{28}$$

We are interested in the fact that we have the following proportional relationship:

$$\mathbb{P}(\alpha|D) \propto \mathbb{P}(D|\alpha) \cdot \mathbb{P}(\alpha) \tag{29}$$

This allows us to avoid the computation of $\mathbb{P}(D)$ that might be difficult. All the machinery exposed above can be realized whether the "probability" adds up to 1 or not.

In our particular case, the known data is the total number of detections TD that the CNN gives as an output and we would like to infer the actual number of fish A. Therefore,

$$\mathbb{P}(A|TD) \propto \mathbb{P}(TD|A) \cdot \mathbb{P}(A) \tag{30}$$

One first way to tackle this problem could be to use the empirical distribution of A and $TD|A$ using the methods developed in the last section. A more sophisticated one would be to make assumptions about the random variables TP, FN, FP, TD and A.

7 Probabilistic modelling of the variables

From now on, the notation will be $Bin = Binomial$, $Po = Poisson$ and $N = Normal$ assuming its two parameters are the mean and the variance (i.e. $N(mean, variance)$).

It seems reasonable to model each variable according to the following distributions:

$$TP \sim \mathrm{Bin}(A, p) \tag{31}$$
$$FN \sim \mathrm{Bin}(A, 1-p) \tag{32}$$
$$FP \sim \mathrm{Po}(\lambda) \tag{33}$$
$$TD \sim \mathrm{Bin}(A, p) + \mathrm{Po}(\lambda) \tag{34}$$
$$A \sim \mathrm{Po}(\mu) \tag{35}$$

The reason behind each distribution is intuitive. For TP (correct detections) and FN (missed detections), we have that both depend on the number of actual fish A. We can see this as if we checked every actual fish one by one (a sequence of A independent experiments) so there is a probability p of success (correct detection) and a probability $1-p$ of failure (missed detection). This is the principle of binomial distributions.

Nevertheless, for FP and A, it seems more reasonable to propose Poisson distributions. We assume that we have events occurring in a fixed interval of time or space and these events occur with a constant mean rate and independently of the time since the last event. The actual number of fish and the number of wrong detections on every photo in a certain location and lapse may fit in this definition.

This is just an intuitive model that will let us obtain some hopefully realistic information from specific calculations that we can perform from these assumptions. Any further claim about the actual nature of the variables should be based on actual data and the actual results that the CNN and the dataset provide.

We would use the data from the validation set to estimate the parameters. The statistic that we will use is the Maximum Likelihood Estimator (MLE). Let us suppose that $x_1, \dots, x_n$ is a random sample from a population with probability density function (PDF) or probability mass function (PMF) $f(x;\theta)$. Then we define the Likelihood function as

$$L(\theta;\mathbf{x}) = L(\theta; x_1, \dots, x_n) = \prod_{i=1}^{n} f(x_i;\theta). \tag{36}$$

We use the Likelihood function assuming that there is a set of data. In this case, L represents the Likelihood that the data comes from a population with parameter θ. We also define $\hat{\theta}(\mathbf{x})$ as a maximum for the Likelihood function given $x_1, \dots, x_n$. The random variable $\hat{\theta}(\mathbf{X})$ is a Maximum Likelihood Estimator (MLE) of θ.

We are going to find the MLE for Poisson and Binomial distributions.

7.1 MLE for Poisson random variable

The PMF of a $\mathrm{Po}(\lambda)$ is $f(k;\lambda) = e^{-\lambda}\frac{\lambda^k}{k!}$, so the Likelihood function is

$$L(\lambda; k_1, \dots, k_n) = \prod_{i=1}^{n} e^{-\lambda}\frac{\lambda^{k_i}}{k_i!} = e^{-n\lambda}\frac{\lambda^{\sum_{i=1}^{n} k_i}}{\prod_{i=1}^{n} k_i!}. \tag{37}$$

In order to maximize this function, we take its derivative and see where it vanishes

$$\frac{\partial L}{\partial \lambda}(\lambda; k_1, \dots, k_n) = \frac{e^{-n\lambda}\lambda^{\sum_{i=1}^{n} k_i - 1}\left(\sum_{i=1}^{n} k_i - n\lambda\right)}{\prod_{i=1}^{n} k_i!} = 0. \tag{38}$$

It is readily seen that $\sum_{i=1}^{n} k_i - n\lambda = 0$ and the MLE for λ is $\hat{\lambda}(k_1, \dots, k_n) = \bar{k}$, that is, the sample mean.

7.2 MLE for Binomial random variable

In this case, we wish to estimate the parameter p from a random sample $X_1, \dots, X_n \sim \mathrm{Bi}(A, p)$ supposing that $A \sim \mathrm{Po}(\mu)$. We can use the Law of Total Probability

$$\mathbb{P}(X_i = k) = \sum_{a=0}^{\infty} \mathbb{P}(X_i = k | A = a)\mathbb{P}(A = a) \tag{39}$$

Thus, in this case, the Likelihood function is

$$L(p; k_1, \dots, k_n) = \prod_{i=1}^{n} \sum_{a=0}^{\infty} \binom{a}{k_i} p^{k_i}(1-p)^{a-k_i} e^{-\mu}\frac{\mu^a}{a!} \tag{40}$$

$$= e^{-n\mu} p^{\sum_{i=1}^n k_i} \prod_{i=1}^n \frac{1}{k_i!} \sum_{a=k_i}^{\infty} \mu^{k_i} \frac{(\mu(1-p))^{a-k_i}}{(a-k_i)!} \tag{41}$$

$$= e^{-n\mu} p^{\sum_{i=1}^n k_i} \prod_{i=1}^n \frac{1}{k_i!} \mu^{k_i} e^{\mu} e^{-p\mu} \tag{42}$$

$$= p^{\sum_{i=1}^n k_i} e^{-np\mu} \prod_{i=1}^n \frac{\mu^{k_i}}{k_i!}. \tag{43}$$

It will be easier to find the maximum of its logarithm, so

$$\log L(p; k_1, \dots, k_n) = \sum_{i=1}^n k_i \log p - np\mu + \sum_{i=1}^n \log \frac{\mu^{k_i}}{k_i!}, \tag{44}$$

$$\frac{\partial \log L}{\partial p}(p; k_1, \dots, k_n) = \frac{\sum_{i=1}^n k_i}{p} - n\mu = 0. \tag{45}$$

Thus, we deduce that $\hat{p}(k_1, \dots, k_n) = \frac{\bar{k}}{\mu}$

The dataset contains all the particular TP, FN, FP, TD and A for a number n of images. The mean of the sample of FP and A gives us the MLE estimators of λ and μ, respectively. In order to estimate the parameter p, we will use the sample mean of TP. $\hat{p} = \frac{\overline{TP}}{\hat{\mu}}$.

We can use this modelling of the random variables to do Bayesian Inference. Our prior distribution will be A that is distributed as a Poisson variable. The distribution of $TD|A$ is given by their convolution, that is,

$$\mathbb{P}(TD = td | A = a) = \sum_{j=0}^{\min(a,td)} e^{-\lambda} \frac{\lambda^{td-j}}{(td-j)!} \binom{a}{j} p^j (1-p)^{a-j}. \tag{46}$$

It is clear that the posterior distribution we obtain is

$$\mathbb{P}(A = a | TD = td) \propto e^{-\lambda-\mu} \frac{\mu^a}{a!} \Big(\sum_{j=0}^{\min(\mathrm{a,td})} \frac{\lambda^{td-j}}{(td-j)!} \binom{a}{j} p^j (1-p)^{a-j} \Big) \tag{47}$$

This modelling uses discrete variables. A usual approach to simulate the results would be to approximate them by a continuous variable. In particular, we can assume that A is approximated by a normal distribution $N(\mu, \mu)$. We are going to approximate TD given A by another normal distribution, which is $N(Ap+\lambda, Ap(1-p)+\lambda)$. Its mean is the sum of the means of TP and FP and its variance is the sum of the variances of TP and FP. Note that, in this case, the variance of $TP + FP$ is the sum of their individual variances because TP and FP are independent and they have zero covariance.

8 Simulations

We will perform some simulations with different parameters to understand the results that can be achieved through Bayesian analysis. This will be done with the help of a Python package (`PyMC3`) which is particularly convenient for Bayesian statistical modelling. We will simulate 4 scenarios, each representing a different performance of the CNN. In each case, we will suppose that the CNN has given the value of TD (total number of detections) as an output and we know the average actual number of fish that usually appear in the photos (μ) and the average number of wrong detections (λ). We will also consider different values for p, the probability parameter that appears in the binomial distribution of TP (number of correct detections). This should have been appropriately measured by observing the typical performance of the CNN.

Simulations have been carried out by sampling 2 chains for 1,000 tune and 100,000 draw iterations ($2 \cdot 1{,}000 + 2 \cdot 100{,}000 = 202{,}000$ draws in total). The sequence of random samples is obtained with the Metropolis algorithm, which is a Markov chain Monte Carlo method. The model consists in the consideration of a prior distribution for A such that

$$A \sim \mathrm{Po}(\mu) \tag{48}$$

and a distribution TD such that

$$TD \sim N(Ap + \lambda, Ap(1-p) + \lambda) \tag{49}$$

The following code has been used:

```python
import numpy as np
import matplotlib.pyplot as plt
import pymc3 as pm

#Choose parameters
obs= #observed (TD)
fish= #mu
wrong= #lambda
prob= #p

#Define Bayesian model
with pm.Model() as model:
    A = pm.Poisson("A", fish)
    TD = pm.Normal("TD",
                   mu = A*prob+wrong,
                   sigma = np.sqrt(A*prob*(1-prob)+wrong),
                   observed = obs)

#Sample
with model:
    trace = pm.sample(100000)

#Display results
pm.plot_posterior(trace[5000:])
plt.show()

pm.summary(trace[5000:])
```

The parameters chosen for each simulation are showed below for each case:

8.1 1^{st} simulation

$\boxed{TD = 10,\ \mu = 12,\ \lambda = 2,\ p = 0.8}$

That is, assume the CNN has detected 10 fish. Suppose we estimate that a typical photo contains 12 actual fish and the CNN tends to detect 2 fish where there are none. Also, $p = 0.8$ in the binomial distribution that models TP. Then, the distribution of A ends up taking the following *a posteriori* form

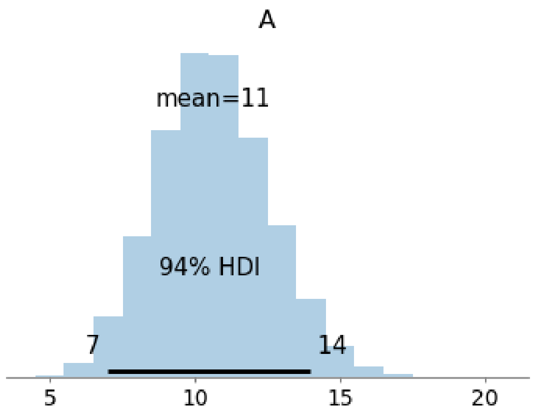

Figure 6: Posterior distribution of A after the first simulation.

More precisely, its mean is 10.642 and its standard deviation is 1.981. This implies that the distribution of A has adapted its mean and standard deviation (they have been reduced) in accordance with the observation TD. Also, we can observe the Highest Density Interval (in this case, it is $[7, 14]$) which could give a certain range of values around the mean where the actual number of fish A could be with a 94% credibility.

8.2 2^{nd} simulation

$\boxed{TD = 80,\ \mu = 12,\ \lambda = 2,\ p = 0.8}$

That is, assume the CNN has detected 80 fish. Suppose we estimate that a typical photo contains 12 actual fish and the CNN tends to detect 2 fish where there are none. Also, $p = 0.8$ in the binomial distribution that models TP. Then, the distribution of A ends up taking the following *a posteriori* form

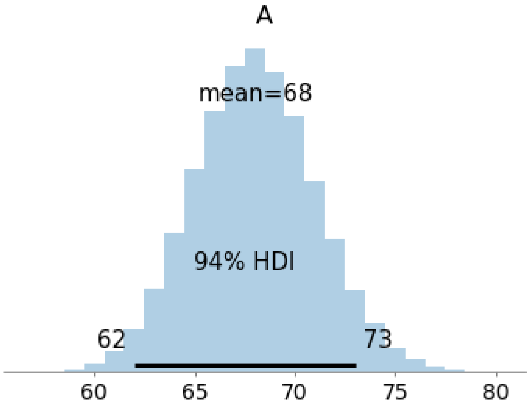

Figure 7: Posterior distribution of A after the second simulation.

More precisely, its mean is 68.033 and its standard deviation is 3.048. Now, we are considering a case where the CNN has happened to give a surprisingly large number of detections. Obviously, knowing the prior distribution of A, the mean of the posterior distribution will be below this large value, but still large because of the high reliability of the network. Also, we can observe the Highest Density Interval (in this case, it is $[62, 73]$) which could give a certain range of values around the mean where the actual number of fish A could be with a 94% credibility.

8.3 3^{rd} simulation

$\boxed{TD = 10,\ \mu = 12,\ \lambda = 20,\ p = 0.1}$

That is, assume the CNN has detected 10 fish. Suppose we estimate that a typical photo contains 12 actual fish and the CNN tends to detect 20 fish where there are none. Also, $p = 0.1$ in the binomial distribution that models TP. Then, the distribution of A ends up taking the following *a posteriori* form

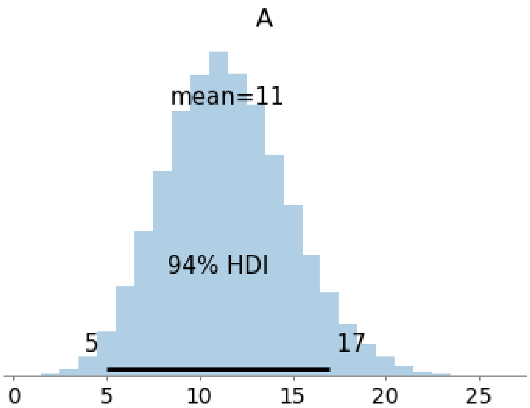

Figure 8: Posterior distribution of A after the third simulation.

More precisely, its mean is 11.5 and its standard deviation is 3.379. This has a very similar mean to the one of the first simulation, but the standard deviation is now considerably higher because the CNN would be known to be less reliable. Also, we can observe the Highest Density Interval (in this case, it is $[5, 17]$ and wider than in the first simulation) which could give a certain range of values around the mean where the actual number of fish A could be with a 94% credibility.

8.4 4^{th} simulation

$$\boxed{TD = 80,\ \mu = 12,\ \lambda = 20,\ p = 0.1}$$

That is, assume the CNN has detected 80 fish. Suppose we estimate that a typical photo contains 12 actual fish and the CNN tends to detect 20 fish where there are none. Also, $p = 0.1$ in the binomial distribution that models TP. Then, the distribution of A ends up taking the following *a posteriori* form

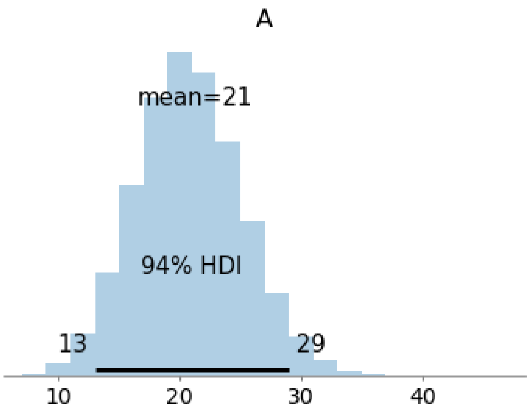

Figure 9: Posterior distribution of A after the fourth simulation.

More precisely, its mean is 21.282 and its standard deviation is 4.396. The mean of the distribution of A is now less affected by the 80 observations than in the second simulation because we have assumed that the network is less reliable. Also, we can observe the Highest Density Interval (in this case, it is $[13, 29]$) which could give a certain range of values around the mean where the actual number of fish A could be with a 94% credibility.

9 Conclusion

We have proposed a way to use the results of a Convolutional Neural Network to bound the error in the outputs. In our case, the objective was to bound the error in the detection of fish, but this approach could be generalized to other problems where a CNN counts the number of elements in an image. The nonparametric framework gives a very general approach that could be useful when we do not want to make any assumptions. Also, Bayesian analysis is particularly convenient when we want to update our probability distributions after knowing some data. This is our case because we would like to estimate the actual number of fish A after getting the output of the CNN, which is the total number of detections TD. The Python package `PyMC3` that uses Markov chain Monte Carlo methods lets us carry out simulations for this and compute a Highest Density Interval.

In this particular problem, we have presented a reasonable model for the random variables involved so we can obtain an accurate expression of the posterior distribution of the actual number of fish. In principle, this is a theoretical framework which has been designed to match reality with a series of realistic assumptions. A future line of research would be the comparison of the nonparametric and parametric approaches and testing the model with actual data from the *Deep-Ecomar* project.

References

[1] Abdulla, W. (2017). *Mask R-CNN for Object Detection and Segmentation.* `https://github.com/matterport/Mask_RCNN`

[2] Casella, G., Berger, R. (2002). *Statistical Inference.* Duxbury.

[3] Davidson-Pilon, C. (2013). *Probabilistic Programming & Bayesian Methods for Hackers.* `https://camdavidsonpilon.github.io/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/`

[4] Itzikovitch, R. (2019). *Are We Confident Our Model's Recall is Precise?.* `https://towardsdatascience.com/are-we-confident-our-models-recall-is-precise-133112a6c407`

[5] Khan, S. S. (2020). *An Introduction to Classification Using Mislabeled Data.* `https://towardsdatascience.com/an-introduction-to-classification-using-mislabeled-data-581a6c09f9f5`

[6] *VII Iberian Modelling Week website.* `https://www.spm.pt/PT-MATHS-IN/7imw/english/index.html`