OOD and OMS evaluation paradigms for Object Detection

February 25, 2025

Quick Reminder: the case of image classification

In this section, we recall the definitions of the OOD and OMS detection concepts, as they were introduced in (Guérin et al., 2023). Note that those definition mainly apply to the monitoring of image classification, yet they can be adapted to the object detection task field.

Notation : Let T be our image classification task, defined by the oracle function Ω on an operational domain X, i.e. $\forall x \in X$, $\Omega(x)$ is the grouth-truth of T. Let f be our classification model and m_f the monitor used to reject the **unsafe** predictions of f.

The OMS paradigm

The out-of-model scope (OMS) detection setting evaluates the monitor on its capacity to detect sample from X that will lead to failures of the model f. In other words, it evaluates the capacity of the monitor to spot sample that are "out of the model's scope".

The scope of the model can thus be defined as:

$$D_f = \{x \in X | f(x) = \Omega(x)\}$$

Then, an ideal monitor m^*_f (according to the OMS paradigm) would be:

$$\forall x \in X, m^*_f(x) = \begin{cases} 0 & \text{if } x \in D_f \\ 1 & \text{otherwise} \end{cases}$$

The OOD paradigm

Another paradigm for evaluating a monitor is the out-of-distribution (OOD) detection setting. A deep neural network (DNN) is, in practice, trained upon a *training* dataset for which the ground truth is known, that we define as:

$$D_{train} = \{(x_i, y_i) | \forall i \in 1, \cdot, n x_i \in X, y_i = \Omega(x_i)\}$$

Then, we can define an in-distribution (ID) domain D_{DI} that contains all samples from X that are drawn from the same distribution as D_{train}. We can thus define the OOD evaluation as the evaluation of the capacity of the monitor to detect sample that are OOD, i.e. that do not belong to D_{ID}. An ideal monitor for this paradigm would be:

$$\forall x \in X, m^*(x) = \begin{cases} 0 & \text{if } x \in D_{ID} \\ 1 & \text{otherwise} \end{cases}$$

The idea behind the OOD paradigm is that a DNN trained on D_{train} must be good for samples from D_{ID} (similar) and less performant for sample too different (not from D_{ID}). It presents some advantages, as the evaluation is independant from f (only depends on the data), however it is only a proxy for the monitoring task that is the OMS detection and evaluating performances under OOD might give a biased indication on the real performances of the monitor ((Guérin et al., 2023)).

Generalizing OOD to Object Detection ?

TODO : Explain how ambiguous it is to extend the definition to OD, while the definition for image classification is not extremely clear itself…

Generalizing OMS to Object Detection

One can consider the task of object detection (OD) in compute vision as a combination of both a classification and a regression task. The model indeed tries to predict a bounding box around an object (localisation) and predict the class of the object inside that box (classification).

The continuous domain on which the localisation takes place prevents from using the exact same definition as explicited in the previous section. Indeed, for a ML regression task, the correctness level of a prediction cannot be assessed by checking an equality, which happens with very low probability.

Defining the scope of the model

As a first step, it is thus necessary to redefine the scope of the model and up to which point a prediction of this model is accepted as correct. Let $x \in X$ a sample and N_x the number of detections (predicted objects) to be found in this sample. We can then define the oracle function as:

$$\forall x \in X, \Omega(x) = \{(c_i, b_i), i \in \{1, \cdot, N_x\}\}$$

with c_i the predicted class of the i-th detection and b_i the predicted location of this detection.

We can then define the scope of f (i.e. the domain on which its predictions of good / safe enough) as the set of samples for which its prediction is close enough to ground-truth, which is the output of the oracle function. to measure how close the prediction is to the oracle, let's introduce an evaluation function (score function) s and a threshold τ, such that:

$$\forall x \in X, f(x) \text{ is correct enough iff } s(x, f) \geq \tau$$

With the above definition, we can derive the expression of an ideal monitor for the OMS paradigm for OD tasks, which goal would be to reject the **unsafe** predictions of f.

$$\begin{aligned} \forall x \in X, m^*_f(x) &= \begin{cases} 0 & \text{if } s(x, f) \geq \tau \\ 1 & \text{otherwise} \end{cases} \\ &= \begin{cases} 0 & \text{if } x \in D_f(s, \tau) \\ 1 & \text{otherwise} \end{cases} \end{aligned}$$

with $D_f(s, \tau) = \{x \in X | s(x, f) \geq \tau\}$

Once the scope of the model is defined, and an expression for an ideal monitor is derived from it, only remains the need to define a proper score function and associated threshold.

Defining the scoring function

In order to define the scoring function and evaluate the monitor on its capacity to learn the scope of the model, we first need to recall a few basics. To quantify the errors of the model, in the case of OD task, we can use the **IoU** (intersection-over-union) for the localisation as well as extensions of the confusion matrix coefficients (True Positives **TP**, False Positives **FP**, False Negatives **FN**, . . .).

To settle the context, let's consider a sample $x \in X$ and the set of detections $f(x)$ made by the model on x. Each detection $d \in f(x)$ corresponds to a tuple (c^d, b^d) (the predicted class and the predicted location). On the other hand, let's consider the N_x ground-truth (GT) detections, i.e. the expected objects in the sample. A grount-truth label l also corresponds to a tuple (c^l, b^l).

For ease of notation, we define $IoU(d, l)$ the intersection-over-union of the detection $d \in f(x)$ and the GT l. For a detection d and a GT l, we define two thresholds on the values of $IoU(d,l)$: the *background threshold* τ_b (below which a detection is considered detecting the background) and the *foreground threshold* τ_f (above which a detection matches the location of the GT).

The label assignement step

The first step in the evaluation of a prediction (set of detection) of the model f consists of matching the detections with the potential ground-truth they detected. In other words, every detection is matched with the closest GT label or left unmatched is no GT is close enough. Following the work of (Bolya et al., 2020), we qualify a positive match if the predicted class of the detection is the same than the GT's and the IoU between the detection and the GT is greater than τ_f (set to 0.5 by default). In case of multiple eligible detections, the one with greater IoU is kept, the remaining being left unmatched.

$$\forall d, \forall l, d \text{ is matched with } l \text{ iff } \begin{cases} c^d = c^l \\ IoU(d,l) \geq \tau_f \\ IoU(d,l) = \max_{p|c^p=c^l} IoU(p,l) \end{cases}$$

The error classification step

The second step allows to determine the TP, FP and FN coefficients from the previous label assignement step. Having those coefficients then allows to derive some of the commonly used metrics in classification: precision (**P**), recall (**R**), etc.

Let $N_d = |f(x)|$ the number of detections of the model on x. Let C be the number of possible classes the GT objects can be assigned to. Then, for each class $i \in [1, \cdots, C]$, we define N_d^i the number of detections f made of the class i and N_x^i the number of GT objects that belong to the class i.

We then derive the following definition for the confusion coefficients:

- Each detection that was positively matched to a GT label is a **TP**
- Each detection that was left unmatched is a **FP**
- Each GT label that has no detection matched to it is a **FN** (missed GT).

So, for each class $i \in [1, \cdot, C]$, we can count the numbers of True Positives TP^i, the number of False Positives FP^i and the number of False Negatives FN^i. Note that in the case of object detection, counting the number of True Negatives (i.e. objects that were correctly ignored by the model) does not make sense, as there would be an infinite (or quite) number of them.

The scoring function

From the extended confusion matrix coefficients, we can then try and define metrics such as the precision or the recall of f, for each class $i \in [1, \cdot, C]$ that the objects can belong to.

$$\forall i \in [1, \cdots, C], P^i = \frac{TP^i}{TP^i + FP^i} = \frac{TP^i}{N_d^i}$$

$$\forall i \in [1, \cdots, C], R^i = \frac{TP^i}{TP^i + FN^i} = \frac{TP^i}{N_x^i}$$

Note: The above expressions are ill-defined. The definition of R^i does not hold when there are not GT labels belongin to the class i in the sample (i.e. $N_x^i = 0$) and the definition of P^i does not hold when there are no detections belonging to the class i in $f(x)$, (i.e. $N_d^i = 0$).

To combine both the precision (which accounts for how many detections were correct among the model detections) and the recall (which accounts for how many objects to find were actually detected by the model) metrics, we can derive the expression of an extended F1-score upon which to apply to rejection threshold τ.

$$\forall i \in [1, \cdots, C], s^i(x, f) = \begin{cases} 1 & \text{if } N_d^i = 0 \text{ and } N_x^i = 0 \\ 0 & \text{if } N_d^i = 0 \\ 0 & \text{if } N_x^i = 0 \\ 2\frac{P^i R^i}{P^i + R^i} & \text{otherwise} \end{cases}$$

Finally, the overall score function s can be derived as the average score over all classes.

$$\forall x \in X, s(x, f) = \frac{1}{C} \sum_{i=1}^{C} s^i(x, f) = m\text{F1-score}$$

Specific usecase : LARD

In the usecase of LARD (Landing Approach Runway Detection) (Ducoffe et al., 2023), we deal with data samples that contain only and exactly one object belonging to only one possible class that is “runway”. Hence, we can simplify the definitions of the model’s scope and the score function that allows to define it.

Having only one class, no checking is needing on the matching of the predicted and ground truth classes in the label assignment step, leading to the following conditions for a positive match.

$$\forall d, \forall l, d \text{ is matched with } l \text{ iff } \begin{cases} IoU(d, l) \geq \tau_f \\ IoU(d, l) = \max_{p|c^p = c^l} IoU(p, l) \end{cases}$$

Having exactly one object to detect in the image assures that $N_x = 1 \neq 0$, leaving only one ill-defined case: when $N_d = 0$, i.e. when the model does not detect the object. The new and simplified score function is as follows, with only one class considered (exponent notation is removed for lisibility).

$$\forall x \in X, s(x, f) = \begin{cases} 0 & \text{if } N_d = 0 \\ 2\frac{PR}{P+R} & \text{otherwise} \end{cases}$$

References

Bolya, D., Foley, S., Hays, J. and Hoffman, J., 2020. *TIDE: A General Toolbox for Identifying Object Detection Errors.* https://doi.org/10.48550/arXiv.2008.08115.

Ducoffe, M., Carrere, M., Féliers, L., Gauffriau, A., Mussot, V., Pagetti, C. and Sammour, T., 2023. *LARD – Landing Approach Runway Detection – Dataset for Vision Based Landing.* https://doi.org/10.48550/arXiv.2304.09938.

Guérin, J., Delmas, K., Ferreira, R.S. and Guiochet, J., 2023. *Out-Of-Distribution Detection Is Not All You Need.* https://doi.org/10.48550/arXiv.2211.16158.