Active Learning for Fish Detection

Simon Lebeaud

GreenAI U.P.P.A

17-10-2022

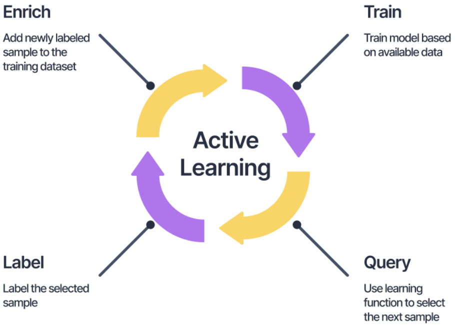
1. Motivations
2. Litterature
3. Results

What is Active Learning ?

Active learning is the subset of machine learning in which a learning algorithm can query a user interactively to label data with the desired outputs. The algorithm proactively selects the subset of examples to be labeled next from the pool of unlabeled data.

Goal

- Reach higher accuracy with less data
- Less time consuming labeling of data



Limitations/Challenges

- Human labeling error
- Selection bias
- Computational cost
- Complexity of the model (query by committe)
- Limited diversity

Methods

ncertainty Select the examples for which the model is the least sure of its prediction.

Diversity Select the examples that best represent the diversity of the data set.

Density Select the examples which are close to the decision frontier of the model.

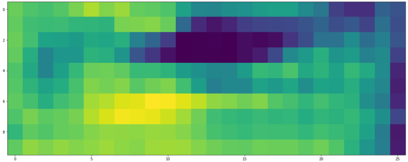
Embedding

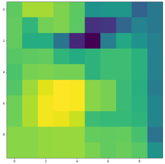
To describe each images I chose the output features of the head's convolution layer of size $128\times?\times?$

One problem : We can't compare boxes of different sizes

ROI pooling

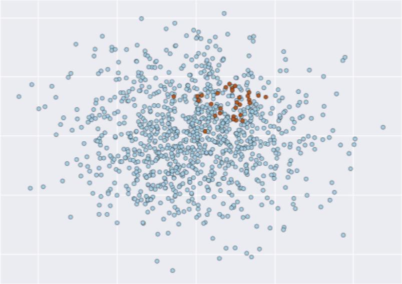
Resize each boxe's feature with ROI pooling, to get 10×10 images.





So for each images we get a $n \times 128 \times 10 \times 10$ with n the number of detections

Representativity problem



Big difficulties for a model to learn under-represented classes.

Core-Set

With Core-set we want to select representative and diverse subset of data the classification task

Advantages : avoids biases in the data set and allows a better generalization of the model

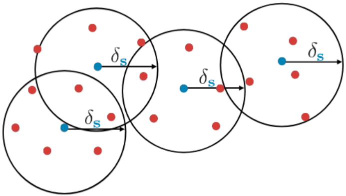
Core-Set

Step 1 Train with random ideal data,

Step 2 Select highest-scoring unlabeled samples (lots of variability depending on the metric),

Step 3 Compute the distances between each point of the labeled and unlabled data,

Step 4 Find the k-centers such that we have the least amount of unlabeld point coverings our data pool

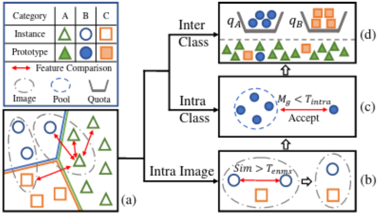


DivProto

Entropy-based Active Learning for Object Detection with Progressive Diversity Contraint

Contraints

- Intra-Image Diversity
- Intra-class Diversity
- Inter-class Diversity



Intra-Image Diversity with ENMS

Entropy-based Non Maximum suppression

Algo

1. Compute similarity between each box embedding
2. Remove one box if similarity smaller than T_{enms}
3. Compute image entropy from remaining detections:

$$\mathbb{H}(I_i|D_S) = \sum_{k\in[t]} -p_{i,k}log(p_{i,k}) - (1-p_{i,k})log(1-p_{i,k})$$

Diverse Prototype for Inter-Image Diversity

For each class in image we compute his prototype:

$$proto_{i,c} = \frac{\sum_{k\in[t]} 1(c, c_{i,k}) \cdot \mathbb{H}(I_i, k) \cdot f_{i,k}}{\sum_{k\in[t]} 1(c, c_{i,k}) \cdot \mathbb{H}(I_i, k)}$$

Than compute the intra-class diversity between images having the same classes:

$$M_g(I_i, [C]) = \min_{c\in[C]} \max_{j\in|\Delta S|} Sim(proto_{j,c}, proto_{i,c})$$

Inter-Image Redundancy Rejection

Intra-class rejection tends to favor majority classes, leading to class imbalance.

Inter-class balancing

We make a selection of the C_{minor} classes that have the least occurences

Fill a quota for each class, counter proportionnal to occurences $=$ lableling budget

DivProto

Algorithm 2 Diverse Prototype

Input: the labeled images $\mathcal{S}$
the unlabeled images $\{I_i\}_{i\in[n]} - \mathcal{S}$
the budget b and the thresholds T_{intra} and T_{inter}

Output: the selected image set $\Delta\mathcal{S}$ to be labeled

Initialize: $\Delta\mathcal{S} := \varnothing$

Calculate the entropy $\{E_i\}$ as well as the prototypes $\{\{\boldsymbol{proto}_{i,c}\}_{c\in[C]}\}$ for the set of the unlabeled images $\{I_i\}_{i\in[n]} - \mathcal{S}$ by ENMS and Eq. (3), respectively.
Calculate the quotas $\{q_c\}_{c\in[C_{minor}]}$ based on $\mathcal{S}$
Sort $\{I_i\}_{i\in[n]} - \mathcal{S}$ in descending order according to $\{E_i\}$
for i in $[|\{I_i\}_{i\in[n]} - \mathcal{S}|]$ **do**
  if $M_g(I_i, [C]) < T_{intra}$ and $M_p(I_i, [C_{minor}]) > T_{inter}$ **then**
    Select I_i and update $\Delta\mathcal{S} := \Delta\mathcal{S} \cup \{I_i\}$
    for c in $[C_{minor}]$ **do**
      Update $q_c := q_c - 1$ if $p(i, c) > T_{inter}$
      Update $C_{minor} := C_{minor} - 1$ if $q_c = 0$
    end for
  end if
end for
Fill up $\Delta\mathcal{S}$ with the rest images from the sorted set $\{I_i\}_{i\in[n]} - \mathcal{S}$ until $|\Delta\mathcal{S}| = b$

1. Motivations
2. Litterature
3. Results

Results

Random weight initialization

Base

Trained on 5% of the Dataset

Random

Base $+$ random 20% of dataset

Entropy

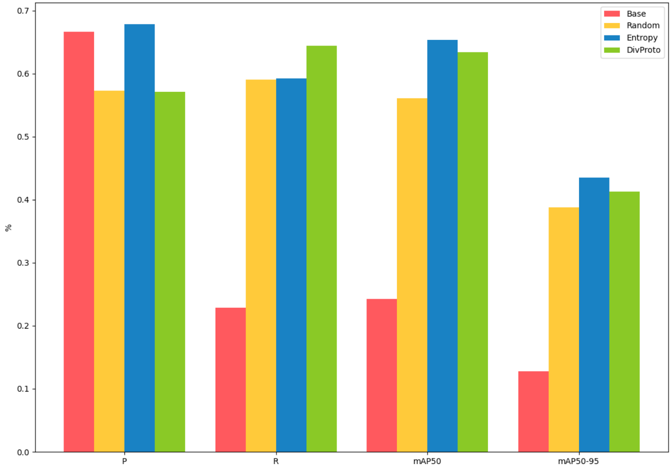
Base $+$ 20% best intra-image entropy

DivProto

Base $+$ 20% best based on entropy and diversity

For each : 10% Validation - 20% Test

Results


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
%
P
R
mAP50
mAP50-95
Base
Random
Entropy
DivProto

Thanks!