

Identifying Characteristics of False Positives from a Flower Detection Network

Isa Lykke Hansen

Data Science Exam Paper
MSc. Cognitive Science - AU

May 28th 2020

Abstract

This will soon be a paper about image classification ... I hope.

1 Introduction

The method of using neural networks (NNs) to classify images have grown steadily in popularity over the last couple of years. These days image classification algorithms help us solve diverse tasks such as detecting oil spills, improving early cancer detection, predicting the weather and even exploring the distant corners of our universe (**vestfalenAarhusHeleByen2019**),e

Since 2018 researches at Universitetet i Tromsø (UiT) and Aarhus University (AU) have collaborated on a project to track the plant-pollinator interaction in selected species of Arctic flowers. This is done to investigate whether climate change is shifting the active seasons of the flowers and pollinating insects. Such a shift could potentially have very serious consequences, as it is ultimately the overlap between the pollinators and flowers that determine the outcome of food crops (**danmarksfrieforskningsfondAretsOriginaleIde2019**).

In short, the project implements large scale surveillance of the plant-pollinator interaction by way of time-lapse images. For two years, data was gathered from several time-lapse cameras set up at various locations in the Arctic, collecting thousands of images for analysis. The idea then is to train a NN to automatically detect flowers and insects in the images. This creates the possibility for investigating changes in large areas of over long time periods with minimal equipment and manual labour.

1.1 Image Classification

Broadly speaking, image recognition tasks can be divided into three sub-categories (**C4W3L01ObjectLocalization**):

1. Classification

2. Classification with localization

3. Detection

When searching an image for an object, X, Classification answers the simple question "is X present in the image?". Classification with localization in addition answers the question "where in the image is X located?". The exact location of X is specified by a so-called *bounding box*, which is parameterized by a vector containing the x,y coordinates for the midpoint of the box, as well as the height and width of the box. Finally, Detection allows for the localization of multiple objects from several classes within a single image. The output will be an array of bounding boxes each associated with a class and a probability for that class.

The images used in this paper were detections from an R-CNN trained on data from two of the locations, Narsarsuaq in southern Greenland and Thule in Northern Greenland.

2 Methods

All scripts used for analysis can be found on the Github Repository [https://github.com/IsaLykkeHansen/XXXX](#). X images were included in the analysis

2.1 Sorting Images

From the original XXXX images, XXXX were sorted into two groups of true- and false positives. X images were included in the analysis

2.2 Extracting Metadata

Next, metadata were extracted for all images using the script "`extract_metadata.py`". The script runs the following command:

In the following subsections we will go more in depth with a few of these measures.

2.2.1 The Q Complexity Measure

For the calculation of the Q complexity we attempted to implement the method described in (zanetteQuantifyingComplexityBlackandwhite2018). !!!However, due to time constraints we were not able to completely recreate the measures described in the paper. !!!

$$V_b = \frac{1}{L^2 - 1} * \text{sum}_{i \in b}(g_i - \bar{g}_b)^2 \quad (1)$$

$$\bar{g}_b = \frac{1}{L^2} * \text{sum}_{i \in b}(g_i) \quad (2)$$

$$V = \frac{L^2}{N'M'} * \text{sum}_b(V_b) \quad (3)$$

$$S = \frac{LN}{N'} \quad (4)$$

In addition a few changes were made in order to generalize the method. Most importantly, the scaling of the resized images was changed First, instead of using a predetermined array of scaling heights, as described on p. 10 we insted implemented a method that downsized the images according to a scaling factor of the original image. This method employed in the paper nessecitates(?) that all images be of the same dimensions. By using a scaling factor of the image size instead, the method becomes more broadly applicable, while at the same time still tilgodesse the requirements of keeping the image ratio constant across various resizing ranges.

In addition, as we are running over the image with a kernel of size 2x2, and our images are of varying sizes, we decided to crop the images prior to the calculation of V. In practice this consisted of cutting off one row or column of pixels on the instances where the height or width of the image did not satisfy $X \% 2 = 0$.

Secondly, since we are integrating over discrete values of S we require an interpolation for the computed values of V(S). Here we have used Bspline

$$Q = \frac{1}{s_{max} - s_{min}} * \int_{s_{min}}^{s_{max}} 1 - \text{frac14}(\text{fracd}vds)^2 ds \quad (5)$$

Q complexity:

I cropped the images if they were not I calculated the scale differently (more directly?)

2.2.2 The IC_{LS} Complexity Measure

The IC_{LS} was another measure for complexity implemented from (**yuImageComplexitySpatial2013**). The IC_{LS} measures the relative sizes of compressed to uncompressed size of images and employs it as a measure of complexity.

Since we didn't know the original compression of the images, but were told they had been saved with standard jpg formatting we assumed, this was assumed to be roughly equivalent to jpg 95 formatting. We therefore chose

2.3 Statistical Analysis in R

2.3.1 Cross Validation

In order to find out which model best predicted the data, we ran a series of cross validations using the R package "cvms" (CITE LUDVIG).

2.3.2 CNN

As a proof of concept a simple CNN was implemented to distinguish between positives and false positives

3 Results

We found that positives could best be distinguished from false positives with the linear model

4 Discussion

Due to time constraints we were not able to... In future analysis of the data one might include more measures of ... Another thing that might be interesting to

investigate is the possibility to add the NN to the original R-CNN to see if it leads to fewer false positives. However, layering lksdlk