Annotating Digital Images of Insects

Automatic Generation of RDF files using Computer Vision Methods

vorgelegt von

Leander Neiß
Friedrich Horschig
Clemens Frahnow
Sten Ächtner

31th August 2015

Abstract

1

Inhaltsverzeichnis

1 Introduction

2   Related Work

2.1   Semi-automated Insect Analyzer

The Museum of Natural History Berlin (presenting us with our task) launched an own project to achieve the goal of digitalizing their insect photographs. The resulting tool is called "Inselect" [Gro14] and provides assistance for semi-automatic annotation of insects. During the time of this writing, their approach uses a simple edge detection from the library OpenCV (details about this library follow in the next subsection). The main part of annotation and segmentation is done by the user. They provide a possibility to scan the present QR codes for annotation proposals. There is a good coverage of tests but their test set consist of one image. Apparently, an evaluation of efficiency of their actual algorithm was not suitable so the projects are very hard to compare.

2.2   Approaches based on Machine-learning

During the last years, there was great process in image analysis. A lot of them are based on clustering algorithms which are very well suitable for either known numbers or sizes/shapes of objects [Pap92]. Others use learning algorithms which are usually trained with a large set of human-reviewed data to extract the most probable segmentations. One recent example for a rising technology is the use of neural networks [TMJ$^+$10].
During the chapter 4, we will explain why it would be very hard for our specific use case to gather a large training set.

2.3   Current state of Computer Vision

Luckily, there are very sophisticated approaches of static image analysis based on different features of a single image. The most famous, free and open-source collection of such computer vision algorithm is OpenCV [Bra]. This library contains very advances methods like the "SURF" algorithm as well as basic steps for image processing for threshold-based contour detection algorithms.
One particularily interesting part of this library implements a template matching algorithm. It uses a predefined template that is applied on different positions to an image to find similar regions. The comparison of those regions to the template image can base on different factors like oriented gradients or distribution in luminosity/color.

3   Requirements

4 Concept

TODO: Explain why machine learning is a hard-to-implement approach. Huge set of human-controlled training data and stuff.

5   Implementation

5.1   Integrating Benchmarking

FMeasure Precision Recall Methods test set (set maker)

6   Conclusion and Discussion

6.1   Benchmarks

FMeasure -> value, optimization Discussion of test set

Literatur

[Bra]   Bradski, G.: In: *Dr. Dobb's Journal of Software Tools*

[Gro14]   Group, Natural History Museum's Biodiversity I.: *Inselect Automated seg-
            mentation of scanned specimen images.* `http://naturalhistorymuseum.`
            `github.io/inselect/`. Version: 2014

[Pap92]   Pappas, T.N.: An adaptive clustering algorithm for image segmentation. In:
            Signal Processing, IEEE Transactions on 40 (1992), Apr, Nr. 4, S. 901–914.
            `http://dx.doi.org/10.1109/78.127962`. – DOI 10.1109/78.127962. – ISSN
            1053–587X

[TMJ+10]   Turaga, Srinivas C. ; Murray, Joseph F. ; Jain, Viren ; Roth, Fabian
            ; Helmstaedter, Moritz ; Briggman, Kevin ; Denk, Winfried ; Seung,
            H S.: Convolutional networks can learn to generate affinity graphs for image
            segmentation. In: *Neural Computation* 22 (2010), Nr. 2, S. 511–538