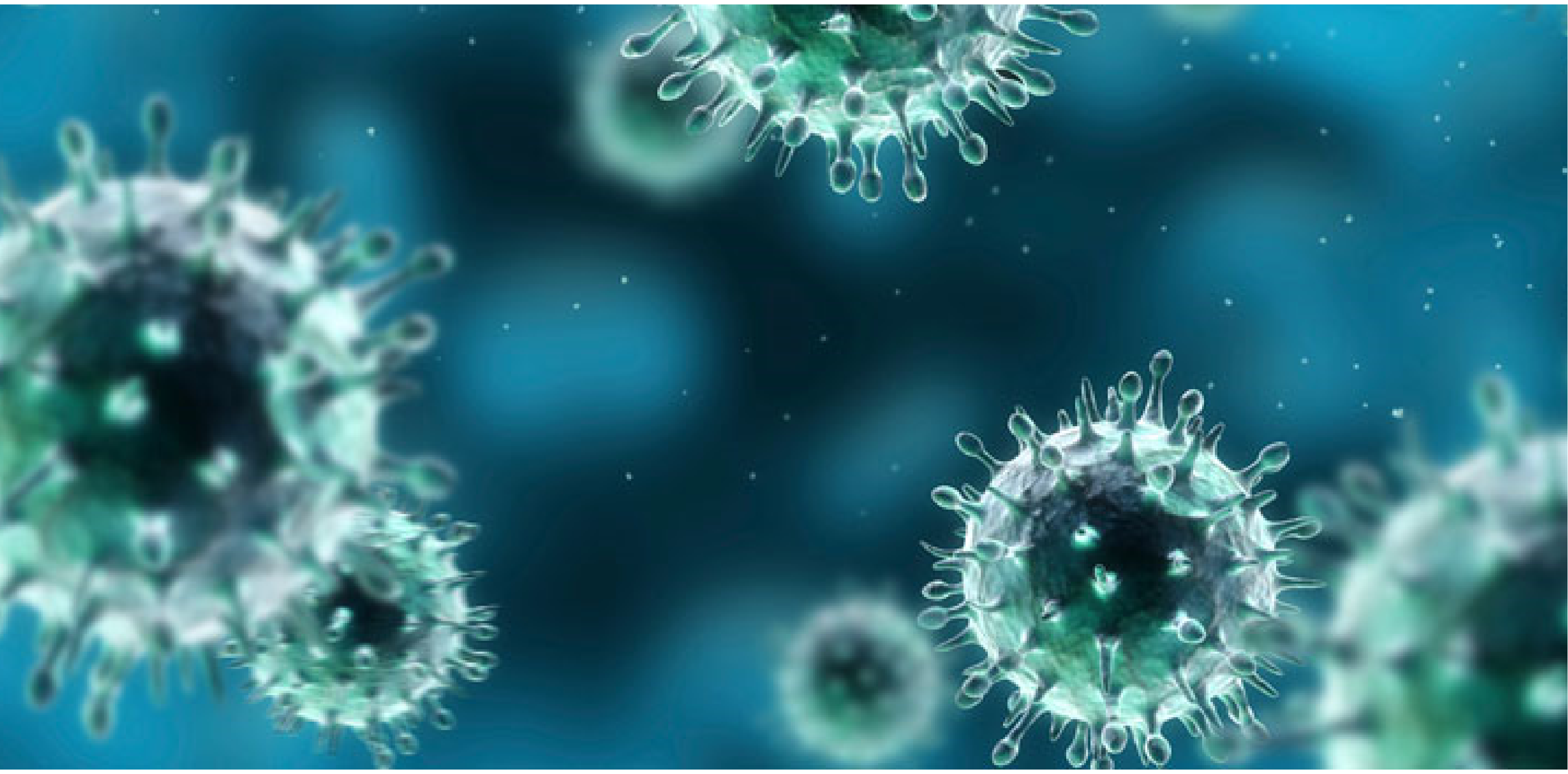
Universität Bremen

IMSAS
Institut für Mikrosensoren, -aktoren und -systeme

**Final Project Report**

Information & Automation Engineering



# Object Detection Proposals

*Evaluation of Image Processing Algorithms for Detecting Airborne Fungal Spores in Microscopic Images*

*Mahmood H. Shubbak*

## Object Detection Proposals*

The present research project studies the application of software algorithms for detecting object proposals in digital images based on their graphical characteristics. The fast generation of accurate object proposals is widely seen as a promising solution to the efficiency problems of heavy classification and object recognition algorithms, which require long time and huge computational power when applied on entire images. Instead, object detection proposals (ODPs) with smaller sizes and focused content are an optimum alternative for such classification algorithms.

This report provides a systematic review and benchmarking of the state-of-the-art ODP algorithms. It further introduces a novel algorithm "Smart-Superpixels", specifically developed for the purpose of airborne fungal spores' detection in microscopic images. The benchmarking considers several qualitative and quantitative criteria regarding the algorithms speed, spatial efficiency, recall accuracy, localization precision and redundancy.

The benchmarking results show that the introduced algorithm: Smart-Superpixels has the highest overall performance with its fast operation, relatively high spatial efficiency, high recall accuracy and localization precision, as well as low redundancy.

* This work is submitted as a final project report at the University of Bremen, Faculty of Physics and Electrical Engineering. Bremen, Germany, November 2018.

### Project Aim:

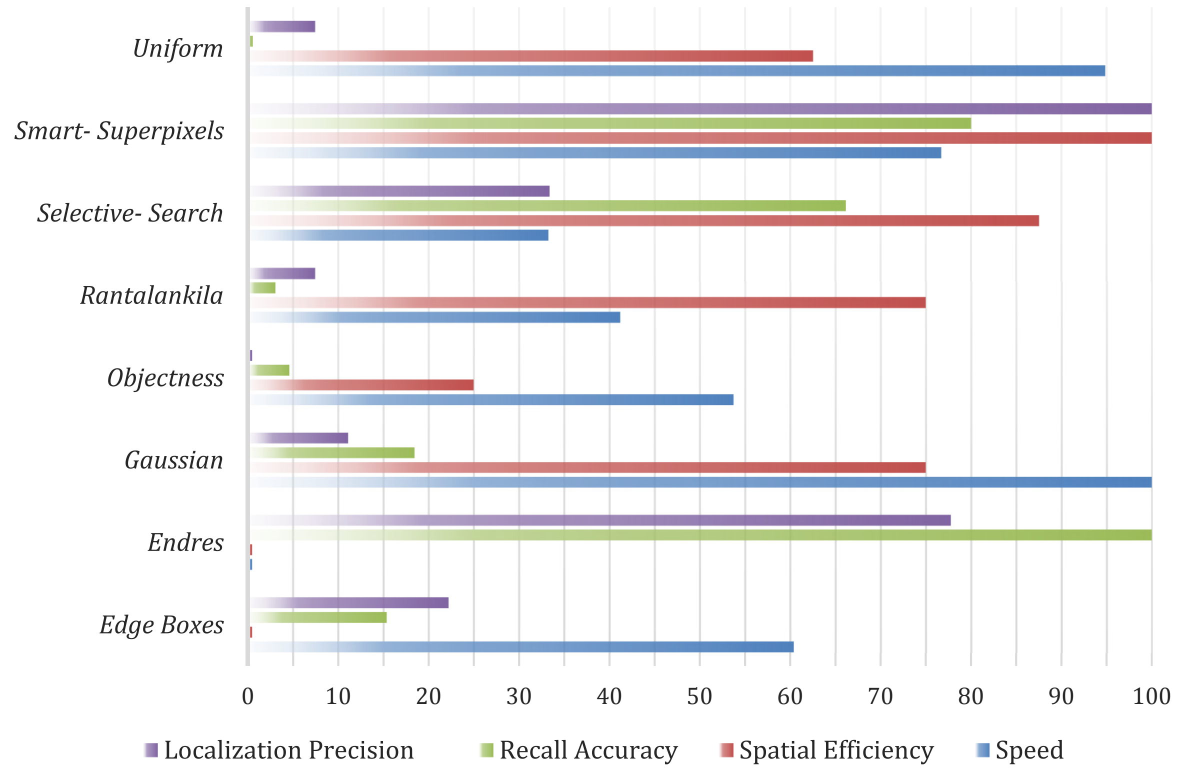
To provides a systematic benchmarking of state-of-the-art object detection proposal (ODP) algorithms and find the best method for airborne fungal spores' detection.

### Research Questions:

- Which are the main ODP algorithms?
- How do they differ in terms of speed, efficiency and accuracy?
- Which is the most convenient ODP algorithm for fungal spores' detection?

### Objectives:

1. To review the state-of-the-art ODP algorithms.
2. To experimentally implement and test them in MATLAB.
3. To benchmark them against several qualitative and quantitative criteria.
4. To find the best method for airborne fungal spores' detection.




Institut für Mikrosensoren, -aktoren und -systeme

# Object Detection Proposals:

## *Evaluation of Image Processing Algorithms for Detecting Airborne Fungal Spores in Microscopic Images*

November 2018

Author: Dr. Mahmood H. Shubbak

Universität Bremen

**Object Detection Proposals: Evaluation of Image Processing Algorithms for Detecting Airborne Fungal Spores in Microscopic Images**



Printed in Germany

For the source codes and further information regarding this work, please consider contacting the author using the prescribed form at the website: https://www.shubbak.de/

This work is submitted as a final project report at the University of Bremen, Faculty of Physics and Electrical Engineering, Institut für Mikrosensoren, -aktoren und –systeme (IMSAS), Otto-Hahn-Allee, NW1, 28359 Bremen, Germany

**Cover page illustration**: front cover image - Microbiology: a scanning electron microscopic image of the human immunodeficiency virus (HIV). Image source: [online] accessed on 15 Nov. 2018, available at: http://blogs.lshtm.ac.uk/students/files/2013/04/virus-730.jpg

# Table of Contents

1

CHAPTER ONE

# 1. INTRODUCTION

## 1.1. Research Motivation

Besides the main gas components of the Earth's atmosphere, fine solid and liquid components (aerosols) comprise an important part especially in urban ecosystems [1, 2]. Aerosols can be found in various concentrations and forms. Accordingly, they can be classified into (i) inorganic materials such as fine dust, sea salt and water droplets; (ii) organic materials such as pollen, bacteria, fungi and spores; as well as (iii) anthropogenic products of combustion processes such as smoke and ashes. [3, p. 54]. While such aerosols can play vital roles in organic decomposition, weather conditions, condensation and precipitation processes, they nonetheless pose serious economic and health challenges.

Organic aerosols and airborne particles are widely considered as the major causes of respiratory ailments of humans, such as asthma, allergies, and respiratory tract's pathogenic infections [4, 5]. Additionally, the direct exposure of open wounds to pathogenic microbial pollution can cause severe health consequences. Accordingly, surgery rooms are designed to eliminate such microbial aerosols as far as possible. Besides its role in the dissemination of many common fungi, airborne fungal spores are also considered main causes and agents of plant disease [4], food spoilage and archives damaging [5].

Furthermore, the existence of aerosols and tiny solid particles ($<$10 μm) in air can impose serious burdens on micro-technology fabrication processes of microprocessors, integrated circuits, electronic components, sensors as well as pharmaceuticals. Such substances are thus manufactured in cleanrooms, where the concentrations and sizes of aerosol particulates are controlled to the minimum levels according to international standards.[1]

Against this background, the effective detection of aerosol particulates, measurement of their concentration in air, and accurate identification of their types are of significant importance for examining air quality and evaluating

[1] For instance, the ISO 14644 class-1 standard for cleanrooms sets a limit of maximum ten particles (size $\geq$0.1 $\mu$m) for each cubic meter of air.

filtration processes in surgery- and cleanrooms, as well as for facing the previously mentioned challenges of such particles.

Among the various methods used for atmospheric aerosol detection and measurement [2] [6], the optical detection method seems promising due to its fast and cost-effective application characteristics [7]. In such method, collected air samples are examined using microscopic system to produce digital images. The images are then analyzed via complicated image processing algorithms for detecting and classifying their content of aerosol particles.

However, the application of such image processing and artificial intelligence algorithms for the identification and classification purposes is considered very time consuming as they require a lot of computational power [8]. In order to overcome this limitation, some pre-identification image processing methods are increasingly being used to automatically suggest proposal regions within each image, where there is high probability for detecting objects. So that, instead of applying the heavy identification algorithms on the whole image, they can be applied just on the proposed regions, which are called "Object Detection Proposals (ODP)". Accordingly, the classification process can be done much faster, enabling for efficient and real-time identification.

In this research project, seven state-of-art ODP algorithms are reviewed, implemented and used for detecting airborne fungal spores in microscopic images. Additionally, an additional ODP method tailored for the particular purpose of this research is developed and introduced by the author.

The research project hence provides a systematic comparison and benchmarking of the eight methods against several evaluation criteria.

This report aims at reviewing the main ODP algorithms, experimentally implementing and testing them, as well as benchmarking them against several qualitative and quantitative criteria, in order to find the best method among them for detecting airborne fungal spores in microscopic images.

## 1.2. Research Questions

Accordingly, the present research project attempts to answer the following research questions:

[2] Such methods are chemical, physical and optical detection techniques [6].

1. Which are the main Object Detection Proposal (ODP) algorithms?
2. How do they differ in terms of speed, efficiency and accuracy?
3. Which is the most convenient ODP algorithm for the purpose of fungal spores' detection in microscopic images?

### 1.3. Report Structure

The present report is organized in six chapters. The next chapter (Ch.2) provides an overview and literature review on the available detection proposal algorithms in terms of their developers, main principles and features. Additionally, a novel algorithm developed by the author is also introduced and explained. Chapter 3 introduces the research methodology and data sources. It further defines the benchmarking criteria as both qualitative and quantitative indicators. In chapter 4, the results of applying the eight ODP algorithms are presented on individual basis, followed by combining them together for the benchmark and discussion of the results in chapter 5. Finally, the main conclusions are synthesized in chapter 6.

2
Chapter Two

# 2. LITERATURE REVIEW (DETECTION PROPOSAL ALGORITHMS)

In image processing paradigm, object detection has long been mostly dependent on employing the “sliding window” algorithm [9, 8]. Sliding Window algorithms are mainly based on generating a huge number of bounding boxes with different sizes and aspect ratios to be used for object classification through assigning a K label to every pixel in the image in correspondence to detected objects [10]. Such classification assignment employs sophisticated and powerful algorithms such as Deep Convolutional Neural Networks [11] and Histogram of Oriented Gradients (HOG) [5] achieving high object detection accuracy.

However, given the high processing time per window, applying such classifiers along with the sliding window scanning is considered inefficient due to its high time and computational power consuming [8, 12]. Consequently, the alternative object detection proposals (ODP) approach has been recently introduced as compromise solution that guarantee both the time efficiency of the detection process, as well as the high classification accuracy [12].

Object proposals are defined as “*a set of candidate regions or bounding boxes in an image that may potentially contain an object*” [13, p. 835]. The generation of such proposals is done via relatively fast algorithms that use several characteristics of the input image in order to suggest specific boxes therein. Accordingly, the sophisticated classifiers are then applied only for the proposals instead of the entire image. Such approach saves time and computational power through the early filtration of false positives prior to classification [12].

Reviewing the literature on object proposal algorithms, five open-source algorithms were identified in this report as the state-of-art algorithms. These are the ***Edge Boxes*** algorithm, the Category-Independent Object Proposals with Diverse Ranking (throughout this report, it will be called “***Endres*** Algorithm” referring to the name of its developer), the ***Objectness*** algorithm, ***Rantalankila***’s algorithm for generating object segmentation proposals using global and local search, as well as the ***Selective Search*** algorithm.

Additionally, two stochastic baseline algorithms are used in this research for benchmarking purposes, these are the ***Gaussian*** and the ***Uniform*** proposal algorithms in accordance with [8].

Finally, a novel algorithm is developed by the author and is introduced in this research project for the purpose of generating fungal spores' detection proposals, namely the ***Smart Superpixels*** algorithm. In the following sections within this chapter, a brief review of each algorithm is introduced.[3]

## 2.1. EdgeBoxes

Developed by a group from Microsoft research in 2014, the EdgeBoxes algorithm generates object proposal bounding box using edges [14]. The basic idea of this method is to use the number of contours that are completely enclosed within a bounding box as an indicator for its probability of being contained a true object.

Accordingly, the algorithm assigns objectness scores to proposal boxes based on the number of contours inside the box minus the number of contours crossing its boundary. [14, 10]

The algorithm developers shown that the EdgeBoxes algorithm is capable of evaluating huge number of candidate boxes in less than a second. They further claimed their method to be not only significantly more accurate but also faster than the prior state-of-the-art. For instance, their results shown object recall scores between 75% and 96% for 1000 proposals. [14]

As a window scoring proposal method, EdgeBoxes algorithm uses efficient data structures and search strategies starting from a sliding window and applying structured decision forests [8, 10]. In their evaluation survey of object proposal methods, Hosang at al. [8, p. 828] concluded that the EdgeBoxes algorithm "*provides a good compromise between speed and quality*".

## 2.2. Endres

The second ODP algorithm to be considered in this review is called the Category-Independent Object Proposals with Diverse Ranking, shortly 'Endres' [15, 16]. Being developed by researchers at the University of Illinois in 2010, the Endres algorithm applies a category-independent method to produce a set of ranked

[3] The appearance of the algorithms in the sections of this chapter is based on alphabetical order of their names.

bounding boxes, so that the higher the rank a box has, the more likely it represents an accurate segmentations of an object.

The algorithm is thus based on generating a diverse set of hierarchical segmentations through graph cuts. Such process depends on both a seed region and a learned affinity function. The resulted regions are then ranked using structured learning (Figure 2.1). Accordingly, the developers claimed their method to achieve good object recall scores with much less proposals than the state-of-the-art segmentation methods [16].

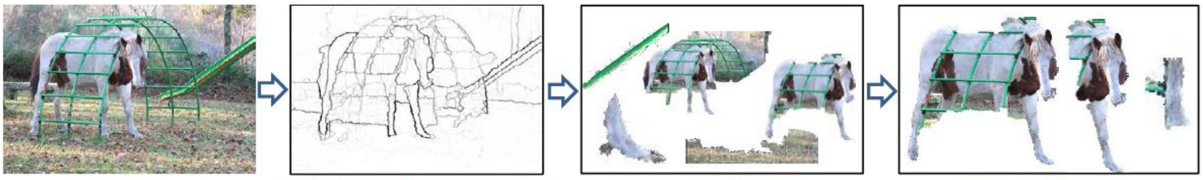



**Figure 2.1: Endres object detection proposal algorithm** [15, p. 223]

Additionally, applying a ranking procedure, the Endres algorithm can maintain high level of diversity even with its small number of proposals. For example, the authors shown a recall scores between 74.9% and 83.4% for only 100 proposals per image [15]. The Endres proposal algorithm can further be used with uncategorized models and applications as well as for automatic object discovery due to its active learning framework.

Despite its general good performance, the Endres algorithm has some limitations especially when considering small objects or very detailed and fine grained images. For such cases additional domain knowledge, e.g. shape models, are needed for the algorithm to successfully generate accurate object proposals.

## 2.3. Gaussian

This object proposal algorithm has been put forward by Hosang at al. [8] as a stochastic baseline to benchmark several ODP methods. The algorithm estimates a multivariate Gaussian distribution for its bounding box center positions, square root areas, and log aspect ratios based on PASCAL VOC 2007 training set [17, 18]. It further calculates mean and covariance on the training set sampling its proposals from that distribution.

## 2.4. Objectness

In this algorithm, a generic objectness measure were put forward by [19], such measure is considered as an indicator of the probability of a bounding box to contain a true object. The measure combines several measuring characteristics of objects. It hence presumes image objects to have a closed boundary and to be visually different from their surrounding background (figure 2.2) [20].

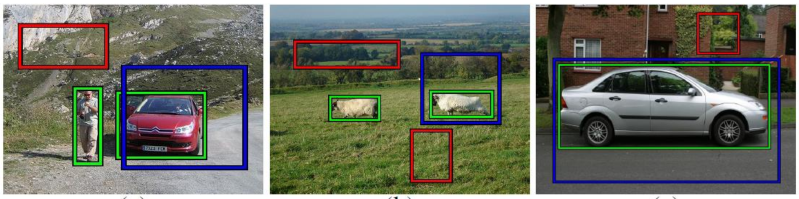



Fig. : **Desired behavior of an objectness measure.** *The desired objectness measure should score the blue windows, partially covering the objects, lower than the ground truth windows (green), and score even lower the red windows containing only stuff or small parts of objects.*

**Figure 2.2: Objectness detection proposals algorithm** [19, p. 1]

Testing the Objectness algorithm, the developers claim a significant better performance than the state-of-the-art methods, such as traditional saliency, interest point detectors, Semantic Labeling, and the HOG detector, both in terms of speed and efficiency [20]. The algorithm shows recall scores of 71% and 91% for 100 and 1000 proposals respectively.

## 2.5. Rantalankila

Being developed by a group of researchers at University of Oulu (Finland) in 2014, the Rantalankila algorithm (named to its developer) aims at generating object segmentation proposals for color images using global and local search techniques [21]. The basic idea of this algorithm is to group superpixels together in order to build object detection proposals.

As the segmentations of Rantalankila's algorithm are class-independent, its computational cost is independent of the number of object classes. The algorithm combines global search and local search of superpixels. While the local search is used to merge adjacent pairs of superpixels in a bottom-up segmentation hierarchy, the global search performs several graph-cut segmentations on the overall superpixel graph (figure 2.3) [21].

Figure Illustration of different phases in the proposed approach. From left to right: The dense initial superpixelization created using SLIC; the proposed refined superpixelization; and illustration of the superpixel pair merging in the local search; a proposal obtained as a result of one global search branch.

**Figure 2.3: Rantalankila object proposals algorithm** [21, p. 4]

The developers claim that applying the Rantalnkila algorithm, similar recall accuracy of the state-of-the-art can be achieved within significantly less computational cost and a relatively low number of regions. Their experimental results shows recall scores between 79% - 91% for around 1200 proposals [21].

## 2.6. Selective Search

The selective search algorithm for object recognition (figure 2.4) was developed by a group of researchers at the University of Trento (Italy) and the University of Amsterdam (Netherlands) in 2011.

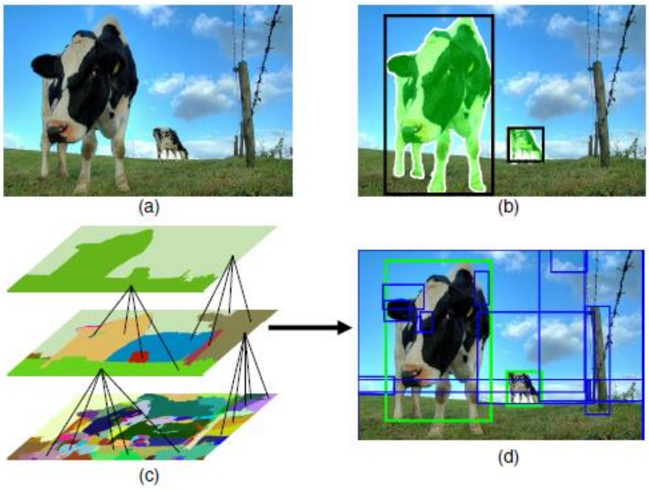



Figure . Given an image (a) our aim is to find its objects for which the ground truth is shown in (b). To achieve this, we adapt segmentation as a selective search strategy: We aim for high recall by generating locations at all scales and account for many different scene conditions by employing multiple invariant colour spaces. Example object hypotheses are visualised in (d).

**Figure 2.4: Selective search ODP algorithm** [22, p. 1]

The main aim of the algorithm is to accurately highlight possible locations for object recognition. To do so, the selective search algorithm combines both a structure-guided sampling process for segmentation and comprehensive search to capture all possible object locations.

The algorithm further implements several complementary and hierarchical partitioning and grouping techniques of superpixels [22]. Within an iterative process, visually similar neighboring regions are grouped together. Similarity is thus calculated based on regions size and texturing characteristics.

The developers claim their algorithm to have superior performance in compare to the state-of-the-art ODP methods. Their results show that the selective search algorithm is fast, robust, stable, independent of object-classes, and can achieve very high recall accuracy scores (96.7% for around 1500 proposals per image) [23].

## 2.7. Smart Superpixels

The smart superpixels algorithm is developed by the author explicitly for the purpose of this research project. It mainly aims at suggesting accurate object proposals for fungal spores in microscopic images of air samples.

Being developed with MATLAB, the smart superpixels algorithm applies comparative analysis of images' superpixels in order to filter out the background pixels and to keep the other pixels as seeds for object proposals.

Consequently, the introduced algorithm starts by mapping all the image's superpixels. It uses the visual characteristics of the entire image as well as those of each superpixel's neighborhood in order to decide whether the pixel belongs to an object (active) or to the image background (passive). Finally, active superpixels are expanded into bounding boxes given the prior knowledge of image topography. Such bounding boxes are considered detection proposals that are most likely contain airborne fungal spore objects.

In chapters 4 and 5 of this report, the results of applying this algorithm on the microscopic images dataset are presented. Theoretically, the smart superpixels algorithm is expected to have high recall accuracy and efficiency scores as it is specifically designed for identifying microbiological objects in line with the purpose of this research project.

Furthermore, due to its simple and straight forward procedures, the method is expected to be relatively faster than the state-of-the-art methods and to have a much higher spatial efficiency.

### 2.8. Uniform

Similar to the Gaussian algorithm (section 2.3), a second stochastic baseline ODP algorithm was introduced by [8] for methods evaluation purposes, namely the Uniform algorithm.

In this method, proposals are generated by uniformly sampling the bounding box centroids, square root areas, and log aspect ratios [8, p. 817]. The ranges of these parameters were also estimated based on the PASCAL VOC 2007 training set [17].

3
CHAPTER
THREE

# 3. DATA & METHODOLOGY

In this chapter, the microscopic dataset of the project is briefly described. Additionally, the chapter presents an overview of the research plan for implementing, applying, and benchmarking the various ODP algorithms that have been reviewed in chapter 2.

Finally, this chapter explains in high detail the evaluation criteria and indicators that will be used in the results and discussion chapters (chapters 4 and 5) for ranking and benchmarking the various algorithms. Such indicators are classified into five main dimensions and contain both qualitative and quantitative measures.

## 3.1. Data

As the main aim of this research project is to systematically benchmark several open-source algorithms for object proposals detection, the data used throughout our analysis is digital microscopic images of air samples. These microscopic images are taken using the optical sensor system developed by [7] for the purpose of detecting airborne fungal spores. Such system represents a fast fully automated mold sensor system.

The working principle of the sensor system is explained in detail in [7, 24]. It can be summarized as follows:

- The sensor first collects air aerosol samples using an adhesive tape. The use of the adhesive tape can guarantee a fast, flexible and cost efficient sensing process.
- In the second step, several colored images of the samples are captured using a digital camera equipped with a microscope objective lens. The microscopic lens has a magnification power of 50x.
- Finally, the high resolution microscopic digital images of air samples are ready to be analyzed by computer software tools using image processes algorithms.

Figure 3.1 below shows the main components of the optical sensor.

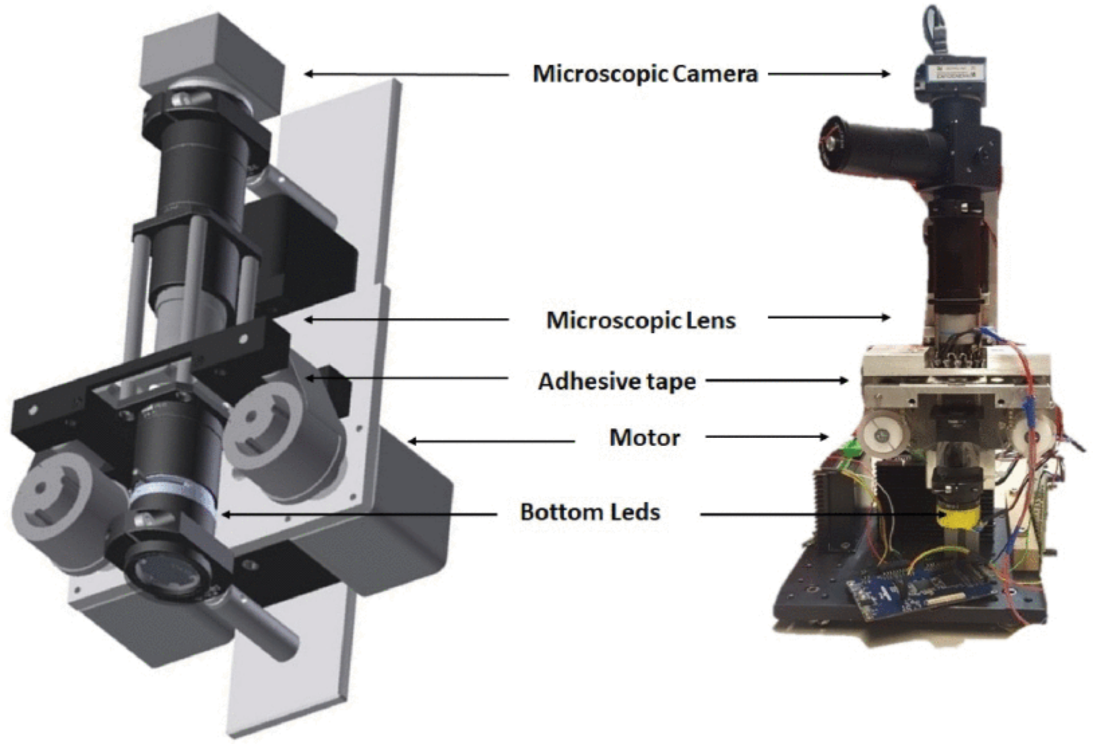



**Figure 3.1: The microscopic optical sensor system** [24, p. 283]

Accordingly, the generated digital images from the previously explained optical sensor represent the input data for my research project. They are hence used for testing and evaluating the object detection proposal algorithms. The sensor system developers provided the author with eleven image samples to be used in the algorithms benchmarking process.

The digital microscopic images are RGB 24-bit depth of 96 dpi resolution. The dimension of each image is 2560 pixels by 1920 pixels. Figure 3.2 shows a sample of such images [24].

In the figure (3.2), fungus spores can be recognized in the foreground through their dark colors and circular structured shapes. Additionally, occasional black dirt spots can be found in some images. The light colored background represents clean air that is fungus-spore free.

In additional to the microscopic digital images, the dataset [24] contains accurate annotation of fungus spore that has been done manually by a group of trained engineering students. The annotations are in the form of x and y coordinates of bounding boxes' edges that surrounds the actual fungus spores in each image. Such annotation is very useful in our benchmarking analysis here, as they provide us with the reality reference for calculating the accuracy and efficiency indicators for each ODP algorithm. The detailed definition of such indicators along with the use of the annotations therein are explained in section 3.3.

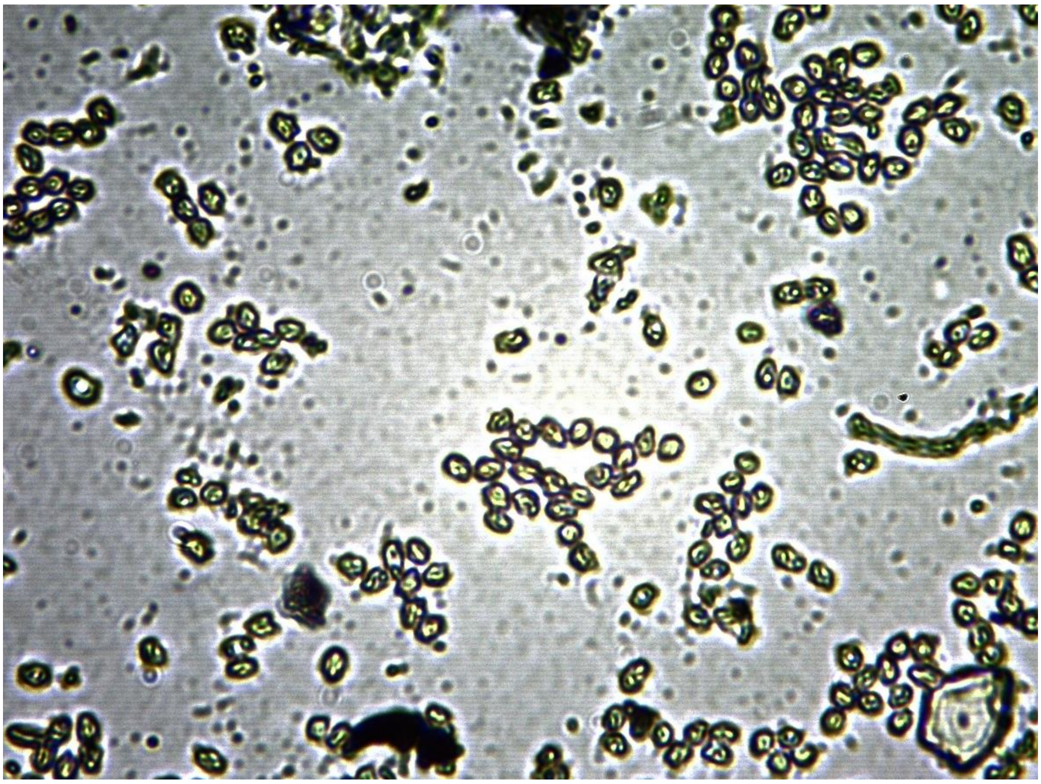


**Figure 3.2: Sample of the optical sensor output images**

## 3.2. Research Plan

To achieve the main aim of this research project concerning the benchmarking of ODP algorithms, a computer program code is developed by the author using MATLAB. The program procedures are designed to be in line with the project plan and its functional use cases.

The research plan of this project consists of eight successive stages of reading digital images, preparing and preprocessing them, running proposal algorithms, calculating evaluation criteria, and finally comparing and benchmarking the algorithms in order to find the optimal method for the purpose of this research project (Figure 3.3).

The overall process implements two FOR loops: the first is over the microscopic digital images, while the second is over the ODP algorithms.

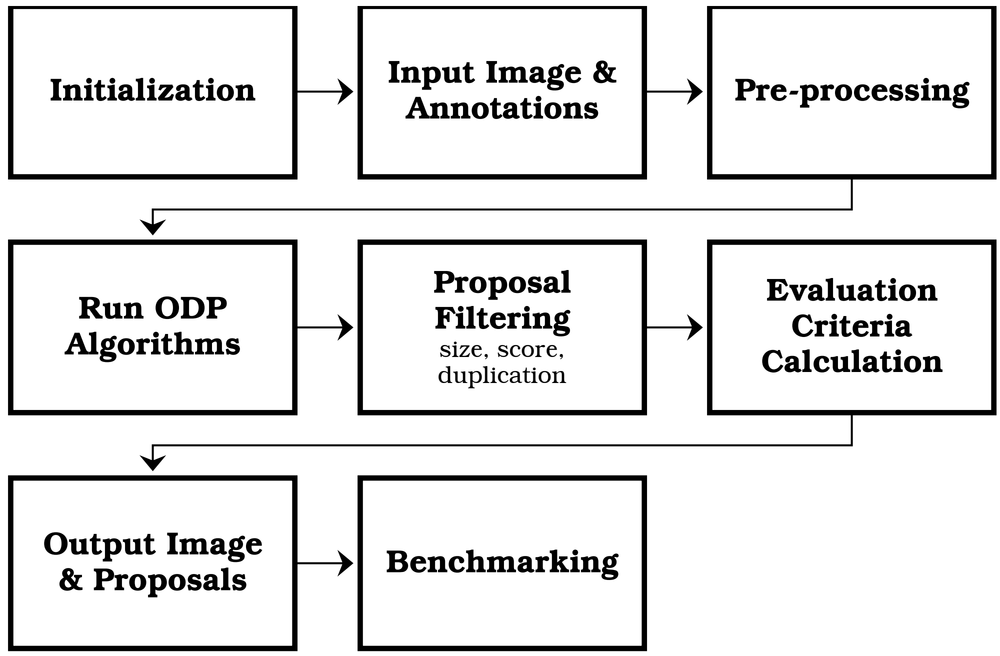



**Figure 3.3: Flowchart of the research project plan**

### 3.2.1. Initialization:

The initialization stage comprises the first steps of preparing the memory and MATLAB environment as well as acquiring software codes in a compatible manner. Accordingly, this stage contains the following processes:

- Clearing the MATLAB working space and its reserved memory,
- Loading default settings and initial values, such as the suggested number of proposals for some ODP algorithms.
- Adding correct directory paths for both software functions and input data of images and annotations.
- Running basic add-ons that are necessary for the smooth execution of some ODP algorithms.
- Reserving memory addresses for general purpose variables, arrays and matrices.
- Harmonizing the considered ODP algorithms through the controlling and uniformity of their input and output data types.
- Converting ODP source codes to MATLAB, in case they were not already written in MATLAB. Such process includes calling their C and C++

subroutines as built-in functions through generating corresponding binary MEX files. These binary files can then be dynamically interpreted and executed from MATLAB.

### 3.2.2. Data input:

In this stage, microscopic digital images and their corresponding annotation files are loaded into the program to be considered for the image processing and further analysis steps.

### 3.2.3. Image Pre-processing:

The pre-processing of images stage includes re-sizing of the input images and re-allocation of annotations accordingly. It thus mainly comprises a down sampling process of the digital images while preserving their graphical details.

The objective of this process is to reduce the needed time and computational power when applying the ODP algorithms in the next stage. Consequently, it can highly increase the overall temporal efficiency and speed of the testing, comparison and benchmarking processes.

### 3.2.4. Running ODP algorithms:

This stage is considered the core and most important step in the developed program code of the research project, as it contains the real-time implementation of the ODP methods.

Accordingly, in this stage, the ODP algorithms discussed in chapter 2 are run on each input image. The project program code is developed in a way, where each algorithm has a specific function subroutine taking the image and the suggested number of proposals as parameters and returning back proposal positions and scores. In the running meanwhile, the program stores the raw data of evaluation indicators in previously declared variables and matrices designed for this purpose (see section 3.2.1).

### 3.2.5. Proposal Post-processing:

The post-processing stage of object proposals comprises filtering of the generated proposals in order to reduce their number by removing unnecessary and inaccurate object proposals. This includes three types of filters:

- **Size filter,** where outlier proposals with very big or very small sizes are filtered out.
- **Duplication filter,** by which any redundant and duplicated proposals are detected and removed.
- **Score filter,** where proposals of very low scores are also filtered out from the final set of object proposals.

### 3.2.6. Calculation of evaluation criteria:

In this stage, several evaluation criteria indicators are calculated using the data collected throughout the previous stages. Detailed explanations of these indicators, their mathematical formulas and physical meanings are presented in section 3.3.

The calculated indicators in this stage are of high importance for performing the benchmarking of ODP algorithms in the final stage of the research plan.

### 3.2.7. Data output:

The data output of digital images marked with object detection proposals and annotations as well as final values of evaluation indicators is prepared in this stage. It is designed to have the form of data packets. Each packet represents the pair combination of image-algorithm. Accordingly, a total number of 88 data packets are generated.

After that, the evaluation indicators of the data packets relevant to each algorithm are aggregated through summation, averaging or maximization (as explained in section 3.2). Consequently, the resulted data output contains the final values of evaluation criteria indicators as well.

### 3.2.8. Benchmarking of ODP algorithms:

The final stage of the research plan is the use of the resulting indicators from the previous stages for comparing and benchmarking the eight ODP algorithms. This is done to figure out the strengths and weaknesses of each algorithm as well as to find the most suitable algorithm for the purpose of fungal spore identification.

Accordingly, this stage contains the generation of comparison tables and charts as well as composite indicators for the several aspects of the benchmark. The benchmarking results are considered in detail in chapter 5.

## 3.3. Evaluation Criteria

To systematically differentiate between the eight ODP algorithms that have been introduced in chapter 2, several qualitative and quantitative evaluation criteria are used. This section presents detailed definition of each indicator along with the characteristic aspects, to which it is used as a proxy. Finally, a weighted average of all indicators is calculated as a composite indicator for the purpose of benchmarking and selection of the best algorithm for the research purpose as stated in the project aim statement (chapter 1).

### 3.3.1. Algorithm Features

The first group of the evaluation criteria comprises qualitative indicators regarding the main features of each algorithm. Such features are:

a. The **programming language**, in which the algorithm's source code is originally written. Such as C, C++, C#, Visual Basic, Fortran, Java, Python, and MATLAB.

b. **Suggestable number of Proposals**: this feature concerns with whether the algorithm has a control variable for suggesting the seed number of proposals to start with or not.

c. **Proposal scores**: this feature concerns with whether the algorithm assign scores to its generated proposals or not.

d. **Image dependency**: this feature concerns with whether the proposals generation process of the algorithm depends on the graphical characteristics of the image's content or not.

e. **Stochastic**: this feature concerns with whether the algorithm process of generating object proposals is stochastic, i.e. depends on randomly determined values.

### 3.3.2. Temporal Efficiency

The temporal efficiency indicator is used to measure the speed of ODP algorithms. It represents the average time per image needed by the algorithm to process input microscopic images and to generate object detection proposals for them. The temporal efficiency is hence measured in seconds. The larger its value, the slower the algorithm is. The speed of the ODP algorithm is an important feature especially when considering real-time applications.

### 3.3.3. Spatial Utilization

This indicator considers the effectiveness of the ODP algorithm in reducing the computational power of object classification methods by reducing the searching area to a small portion of the image size. In other words, it measures how much of the image's area is still considered in the detection proposals.

It is thus calculated as the ratio of union area of all proposals over the original image area (Equation 1).

$$Spatial\ util. = \left| \frac{\cup \{Proposal\}}{image\ area} \right| \qquad \text{(Equation 1)}$$

The higher the spatial coverage utilization, the weaker the algorithm is in reducing the classification computational power.

### 3.3.4. Accuracy

Two accuracy indicators are considered in this research project (figure 3.4):

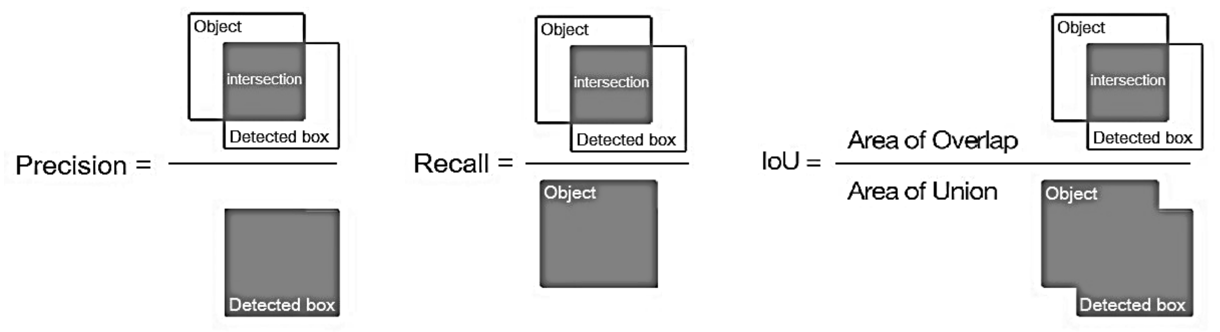



**Figure 3.4: Detection proposals' accuracy indicators** [25]

a. **Recall**, considering the question of: How much relevant area is detected? Accordingly, it is calculated as the ratio of proposal-annotation intersection area over the annotation area (Equation 2). The higher the recall score, the better the algorithm is in detecting true objects.

$$Recall = \frac{Proposal\ \cap Annotation}{Annotation} \qquad \text{(Equation 2)}$$

For each microscopic image, the average recall accuracy score is calculated for the benchmarking purpose.

b. **Localization Precision**, considering the question of: How much of the detected area is relevant? Accordingly, it is calculated as the ratio of proposal-annotation intersection area over the proposal area (Equation 3). The higher the precision score, the more accurate is the algorithm in terms of the positions of it detection proposals.

$$Localization\ Precision = \frac{Proposal\ \cap Annotation}{Proposal} \qquad \text{(Equation 3)}$$

For each microscopic image, the average localization precision score is calculated for the benchmarking purpose.

### 3.3.5. Redundancy

This indicators group consider extent to which the ODP algorithm generates duplicated, highly overlapped, or insignificant (useless) proposals. Accordingly, two measures are used to capture the redundancy:

a. **The number of filtered proposals**: which is equal to the number of removed proposals due to their low scores, outlier sizes, or duplications (see section 3.2.5).

b. **The Jaccard-index for proposals** (intersection over union IoU): which is calculated as the ratio of the proposals' intersection area over their union area, as illustrated in Equation 4.

$$Jaccard\ Index\ (IoU) = \frac{\cap \{Proposals\}}{\cup \{Proposal\}} \qquad \text{(Equation 4)}$$

The larger the Jaccard-index of an ODP algorithm, the more redundant proposals it generates.

A final remark regarding the union operation used in calculating the spatial utilization (section 3.3.3) and the Jaccard-index (section 3.3.5), it is done via the summation of each two proposal area then subtracting their intersection from the result. Occasionally, when several proposal bounding boxes are available inside each other, a negative union value can appear. Therefore, any negative value in both indicators will be interpreted as a high redundancy in the generated proposals by the ODP algorithm.

4
CHAPTER
FOUR

# 4. RESULTS

In this chapter, the main results at the individual level of each ODP algorithm are presented. This includes the experimental outcome of each algorithm regarding the qualitative and quantitative evaluation criteria of the algorithm's main features, speed, spatial utilization, accuracy and redundancy.

## 4.1. EdgeBoxes

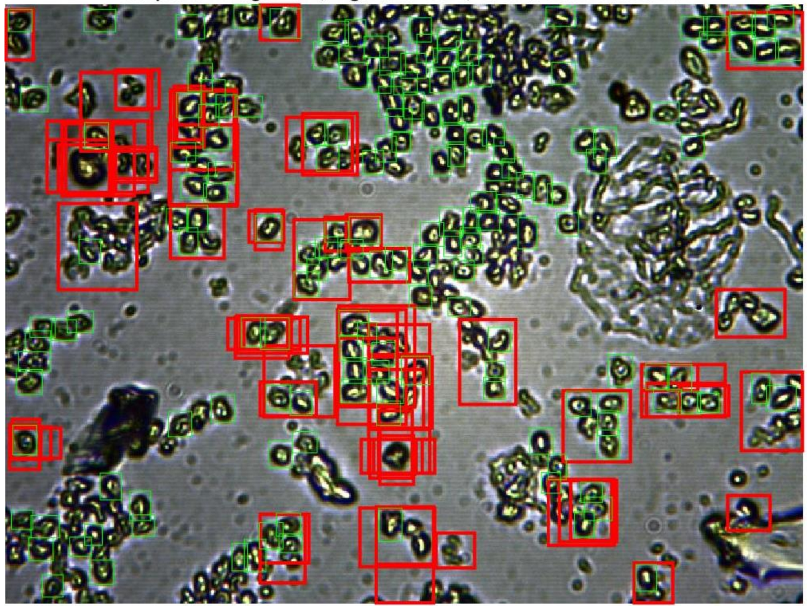
The Edge-Boxes algorithm is originally developed using C++ programming language. It supports the feature of controlling for suggestable number of proposals. Accordingly, the suggested number of proposals used in the experimental testing of this algorithm was 700 proposals. The algorithm is non-stochastic and it is image dependent. However, it does not assign scores to its resulting proposals.

Out of the 700 initial proposals, the final resulted number of proposals per image was 99 on average. Regarding the quantitative criteria, the running time of the algorithm was 1.5 seconds on average. It resulted in negative spatial utilization and Jaccard-index, reflecting a high redundancy of multiple suggested proposals inside each other. The algorithm scored a recall accuracy of 37% and a localization precision of around 9%.

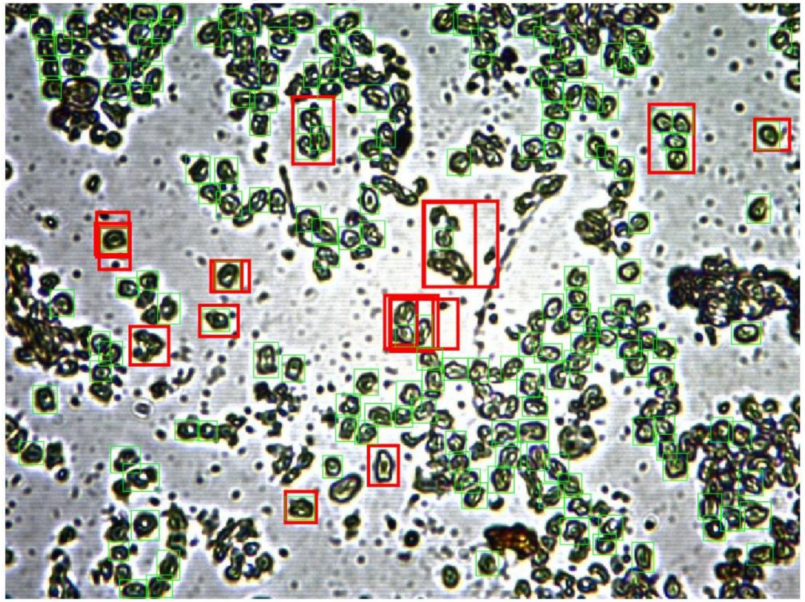
Table 4.1 below lists the evaluation criteria results of the algorithm. Additionally, figure 4.1 shows selected samples of the results when applying the algorithm on air samples' microscopic digital images. In the images, the green boxes represent the annotations, while the red bounding boxes represents the resulting object proposals by the algorithm.

**Figure 4.1: Algorithm #1: EdgeBoxes**

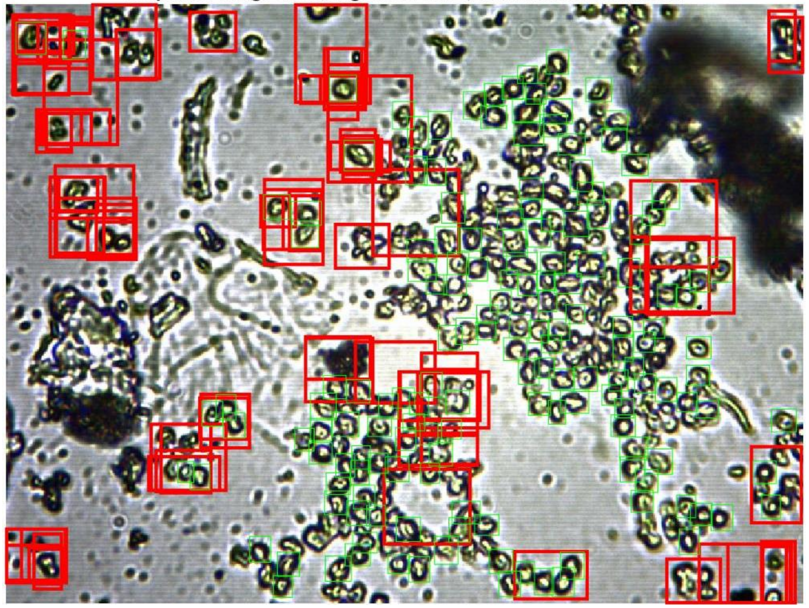
Output for image no.:1, Algorithm no.:1, Resulted Candidates:78



Output for image no.:4, Algorithm no.:1, Resulted Candidates:18



Output for image no.:6, Algorithm no.:1, Resulted Candidates:90



Output for image no.:10, Algorithm no.:1, Resulted Candidates:152

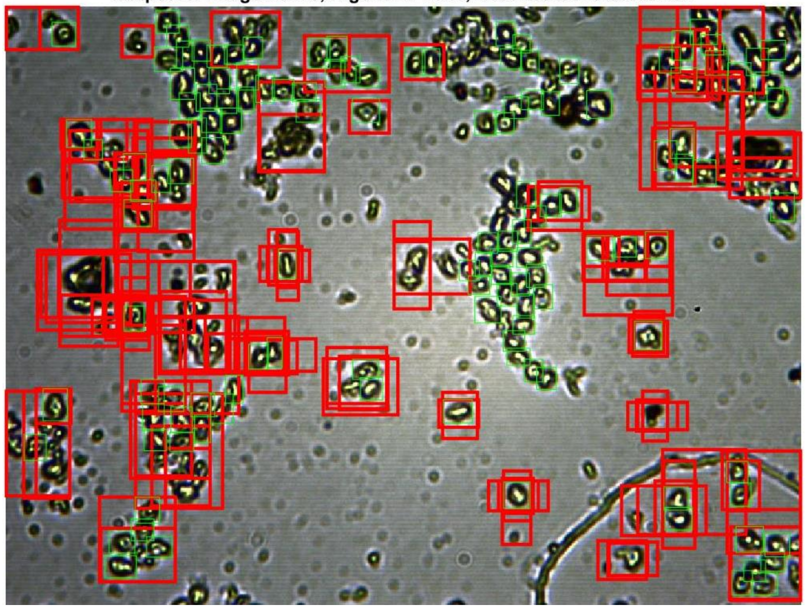


**Figure 4.1: Evaluation criteria results of EdgeBoxes ODP algorithm**

| **Algorithm Name:** EdgeBoxes | | **Programming Lang.:** C++ | | | |
|---|---|---|---|---|---|
| **Qualitative Criteria** | | ***Proposals Suggestable*** | ***Proposal Score*** | ***Image Dependent*** | ***Stochastic*** |
| *1. **Algorithm Features:*** | | Yes (700 used) | No | Yes | No |
| **Quantitative Criteria:** | | **Avg.** | **Std.** | **Min.** | **Max.** |
| *2. **Temporal Efficiency*** (Algorithm Speed) [running time in sec.] | | 1.5 | 0.11 | 1.3 | 1.7 |
| *3. **Spatial Utilization*** | | -0.3 | 0.27 | -0.9 | 0.04 |
| *4. **Accuracy:*** | | | | | |
| *Recall* | | 0.37 | 0.18 | 0.09 | 0.62 |
| *Localization Precision* | | 0.088 | 0.04 | 0.03 | 0.15 |
| *5. **Redundancy:*** | | | | | |
| *Jaccard-index* | | -3.96 | 2.93 | -9.2 | 0.92 |
| ***Reductions*** | *Score-filter* | 0 | 0 | 0 | 0 |
| | *Size-filter* | 600.9 | 38.8 | 542 | 682 |
| | *Duplications-filter* | 0 | 0 | 0 | 0 |
| *Resulted Proposals (per image)* | | 99 | 39 | 18 | 158 |

## 4.2. Endres

The Endres algorithm is originally developed in MATLAB. It does not support the feature of suggestable number of proposals. The initial number of proposals found by the algorithm was 8500 per image on average. The algorithm is non-stochastic image-dependent. However, it does not assign scores to proposals.

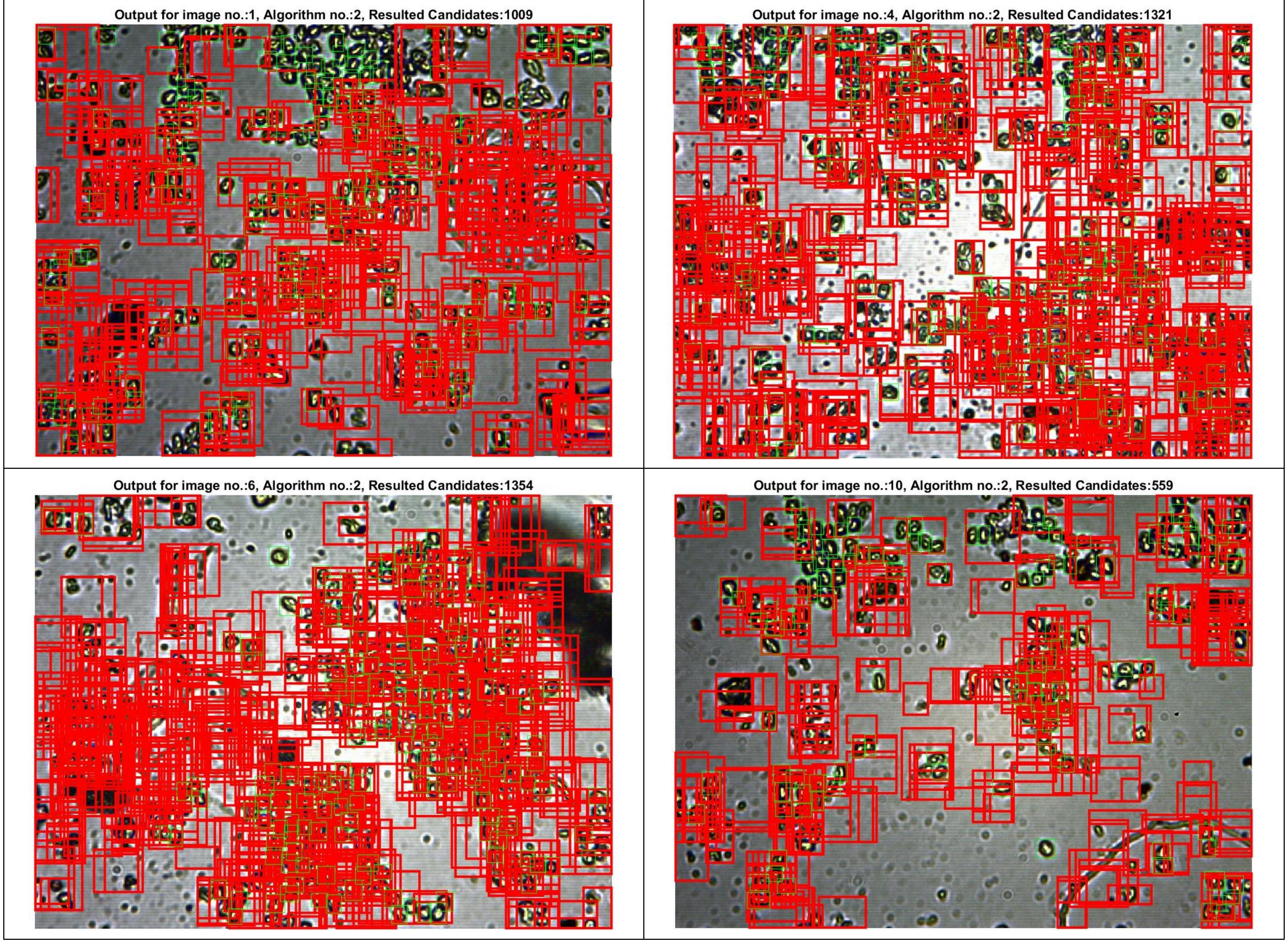
The final number of proposals per image after applying the filters was 903 on average. Regarding the quantitative criteria, the average running time of the algorithm was 586.8 seconds (approx. 10 minutes). It resulted in negative spatial utilization and Jaccard-index, reflecting a very high redundancy. The algorithm scored a recall accuracy of 92% and a localization precision of 24%.

Table 4.2 lists the evaluation criteria results of the algorithm. Additionally, figure 4.2 shows results of applying the algorithm on air sample images. Annotations are in green, while proposals are in red bounding boxes.

**Figure 4.2: Evaluation criteria results of Endres ODP algorithm**

| **Algorithm Name:** Endres | | **Programming Lang.:** MATLAB | | | |
|---|---|---|---|---|---|
| **Qualitative Criteria** | | ***Proposals Suggestable*** | ***Proposal Score*** | ***Image Dependent*** | ***Stochastic*** |
| 1. ***Algorithm Features:*** | | No (avg. 8500) | No | Yes | No |
| **Quantitative Criteria:** | | **Avg.** | **Std.** | **Min.** | **Max.** |
| 2. ***Temporal Efficiency*** (Algorithm Speed) [running time in sec.] | | 586.8 (9.8 minutes) | 245.1 | 302.2 (5 minutes) | 1046.9 (17.5 min) |
| 3. ***Spatial Utilization*** | | -56 | 22.4 | -85.3 | -23.3 |
| 4. ***Accuracy:*** | | | | | |
| *Recall* | | 0.92 | 0.06 | 0.82 | 0.99 |
| *Localization Precision* | | 0.24 | 0.02 | 0.2 | 0.27 |
| 5. ***Redundancy:*** | | | | | |
| *Jaccard-index* | | -1.14 | 0.02 | -1.18 | -1.11 |
| ***Reductions*** | *Score-filter* | 0 | 0 | 0 | 0 |
| | *Size-filter* | 7234 | 2791.2 | 3935 | 12843 |
| | *Duplications-filter* | 440.2 | 107.2 | 243 | 573 |
| *Resulted Proposals (per image)* | | 903 | 233 | 571 | 1196 |

Figure 4.2: Algorithm #2: Endres



### 4.3. Gaussian

The Gaussian baseline algorithm is originally developed in MATLAB. It supports the feature of suggestable number of proposals (the suggested number in this experiment was 700, out of them avg. 84 proposals per image resulted after applying the post-processing filters). The algorithm is stochastic image-independent. However, it does not assign scores to proposals.

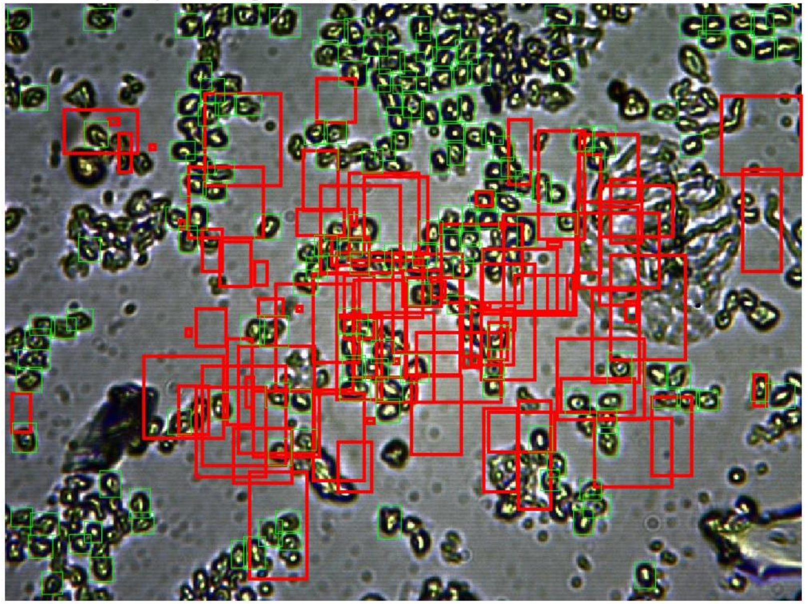
Regarding the quantitative criteria, the average running time of the algorithm was 0.033 seconds. It resulted in 27% spatial utilization and an average Jaccard-index of 1. The algorithm scored a recall accuracy of 39% but only 6% localization precision. Table 4.3 lists the evaluation criteria results. Additionally, figure 4.3 shows samples of the results of applying the algorithm on microscopic images.

**Figure 4.3: Evaluation criteria results of Gaussian ODP algorithm**

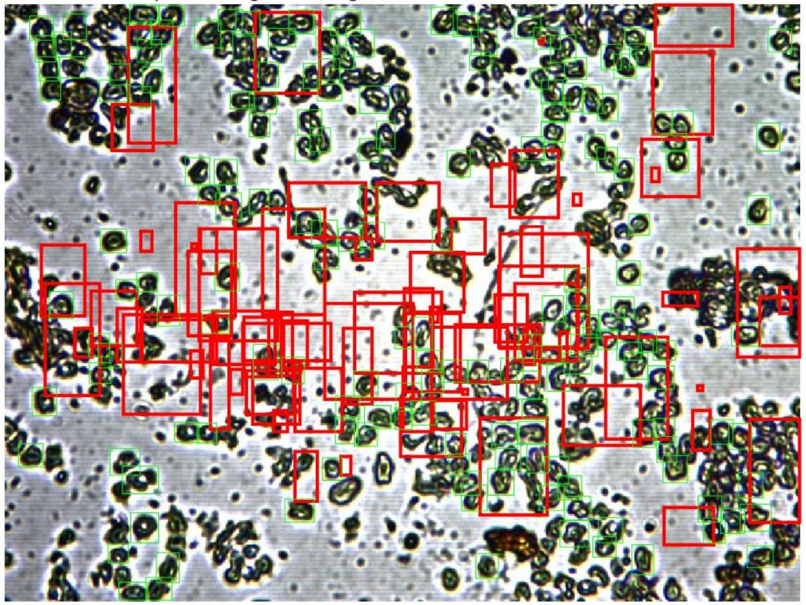
| **Algorithm Name:** Gaussian | | **Programming Lang.:** MATLAB | | | |
|---|---|---|---|---|---|
| **Qualitative Criteria** | | ***Proposals Suggestable*** | ***Proposal Score*** | ***Image Dependent*** | ***Stochastic*** |
| 1. ***Algorithm Features:*** | | Yes (700 used) | No | No | Yes |
| **Quantitative Criteria:** | | **Avg.** | **Std.** | **Min.** | **Max.** |
| 2. ***Temporal Efficiency*** (Algorithm Speed) [running time in sec.] | | 0.033 | 0.098 | 0.002 | 0.33 |
| 3. ***Spatial Utilization*** | | 0.27 | 0.05 | 0.19 | 0.35 |
| 4. ***Accuracy:*** | | | | | |
| *Recall* | | 0.39 | 0.07 | 0.3 | 0.54 |
| *Localization Precision* | | 0.06 | 0.01 | 0.04 | 0.08 |
| 5. ***Redundancy:*** | | | | | |
| *Jaccard-index* | | 1 | 0.48 | 0.36 | 1.74 |
| ***Reductions*** | *Score-filter* | 0 | 0 | 0 | 0 |
| | *Size-filter* | 615.5 | 6.2 | 605 | 624 |
| | *Duplications-filter* | 0 | 0 | 0 | 0 |
| *Resulted Proposals (per image)* | | 84 | 6 | 76 | 95 |

Figure 4.3: Algorithm #3: Gaussian

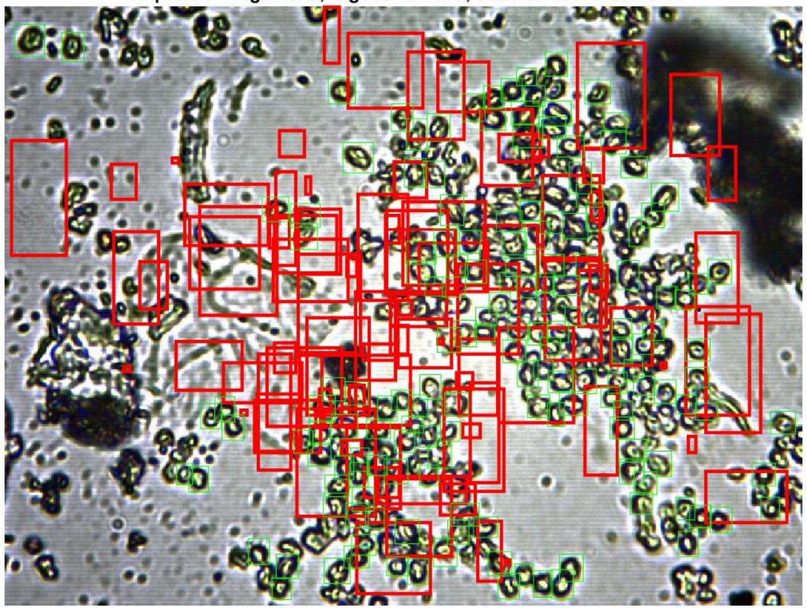
Output for image no.:1, Algorithm no.:3, Resulted Candidates:86



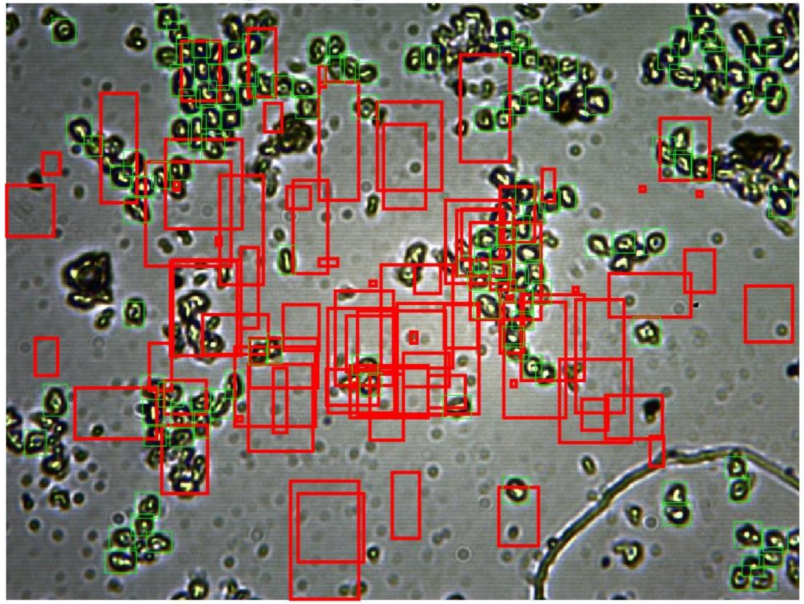
Output for image no.:4, Algorithm no.:3, Resulted Candidates:74



Output for image no.:6, Algorithm no.:3, Resulted Candidates:89



Output for image no.:10, Algorithm no.:3, Resulted Candidates:78



### 4.4. Objectness

The Objectness ODP algorithm is originally developed in C++ language having some add-ons in C. It supports suggestable number of proposals (the suggested number here was 700, out of them only 19 proposals per image on average resulted after applying the post-processing filters). The algorithm is non-stochastic image-dependent, and it assigns scores to its resulting proposals.

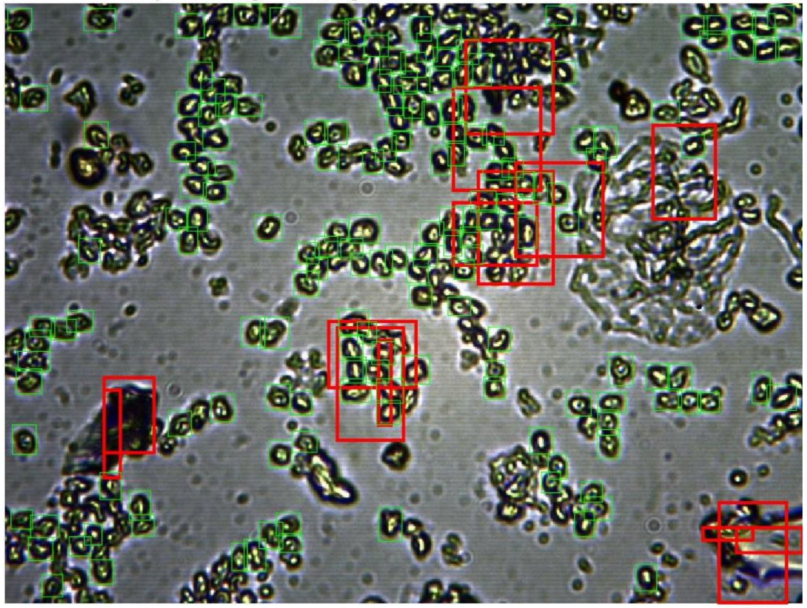
Regarding the quantitative criteria, the average running time of the algorithm was 2.85 seconds. It resulted in 11% spatial utilization and an average Jaccard-index of 0.91. The algorithm scored a recall accuracy of 30% and only 3% localization precision. Table 4.4 lists the evaluation criteria results. Additionally, figure 4.4 shows some results of applying the algorithm on air sample images.

**Figure 4.4: Evaluation criteria results of Objectness ODP algorithm**

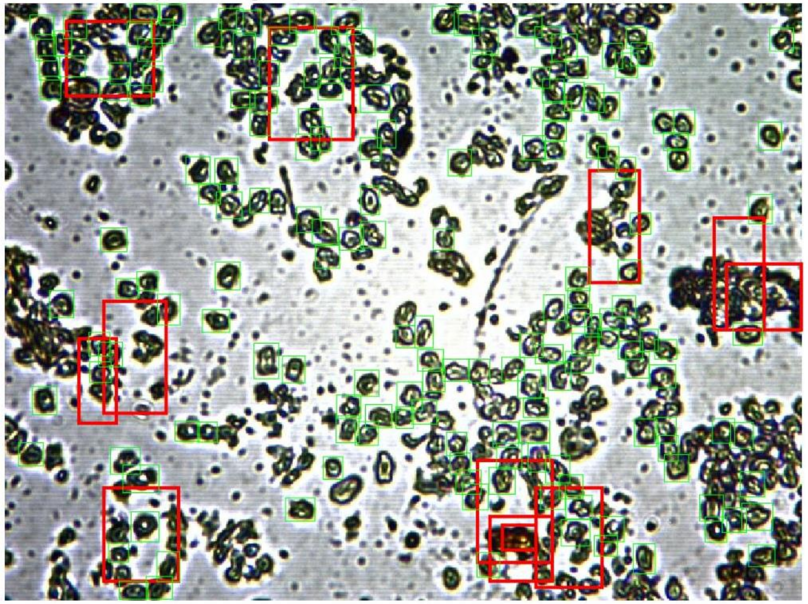
| **Algorithm Name:** Objectness | | **Programming Lang.:** C, C++ | | |
|---|---|---|---|---|
| **Qualitative Criteria** | ***Proposals Suggestable*** | ***Proposal Score*** | ***Image Dependent*** | ***Stochastic*** |
| 1. ***Algorithm Features:*** | Yes (700) | Yes | Yes | No |
| **Quantitative Criteria:** | **Avg.** | **Std.** | **Min.** | **Max.** |
| 2. ***Temporal Efficiency*** (Algorithm Speed) [running time in sec.] | 2.85 | 0.63 | 2.56 | 4.75 |
| 3. ***Spatial Utilization*** | 0.11 | 0.03 | 0.05 | 0.17 |
| 4. ***Accuracy:*** | | | | |
| *Recall* | 0.3 | 0.11 | 0.14 | 0.44 |
| *Localization Precision* | 0.029 | 0.014 | 0.011 | 0.049 |
| 5. ***Redundancy:*** | | | | |
| *Jaccard-index* | 0.91 | 1.48 | 0.05 | 5.17 |
| ***Reductions*** *Score-filter* | 84.4 | 64.8 | 3 | 182 |
| *Size-filter* | 596.6 | 69.3 | 486 | 685 |
| *Duplications-filter* | 0 | 0 | 0 | 0 |
| *Resulted Proposals (per image)* | 19 | 10 | 9 | 35 |

**Figure 4.4: Algorithm #4: Objectness**

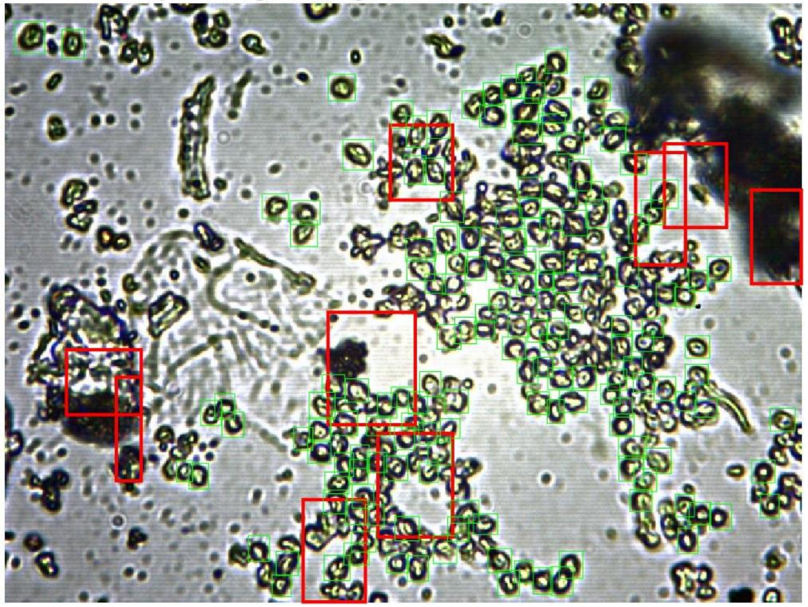
Output for image no.:1, Algorithm no.:4, Resulted Candidates:14



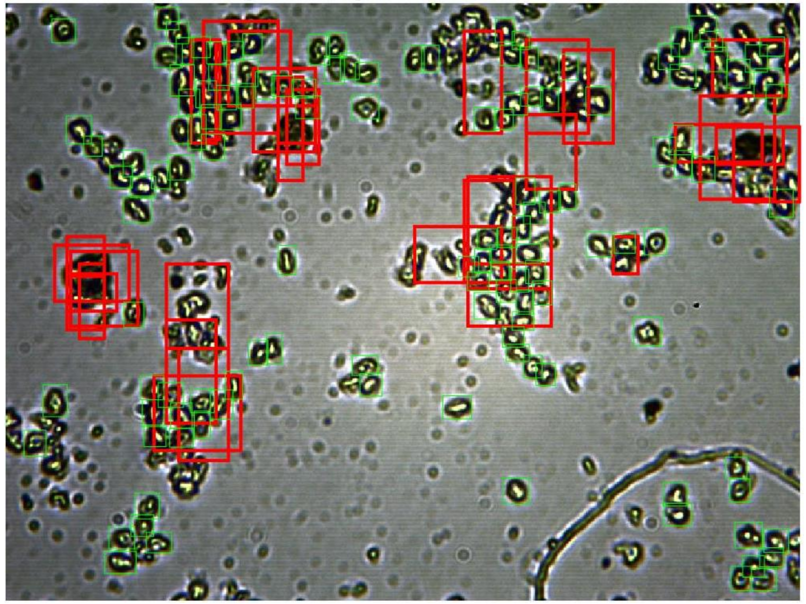
Output for image no.:4, Algorithm no.:4, Resulted Candidates:12



Output for image no.:6, Algorithm no.:4, Resulted Candidates:9



Output for image no.:10, Algorithm no.:4, Resulted Candidates:32



## 4.5. Rantalankila

The Rantalankila algorithm is originally developed in C++ language. It does not support suggestable number of proposals (The initial number of proposals found by the algorithm was 194 per image on average, out of them avg. 47 proposals per image remain after applying the filters). The algorithm is non-stochastic image-dependent. However, it does not assign scores to its resulting proposals.

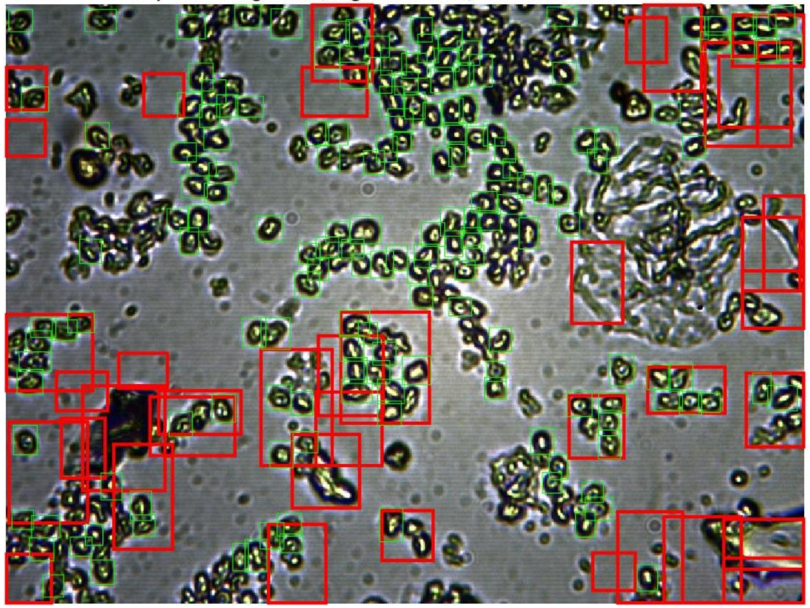
Regarding the quantitative criteria, the average running time of the algorithm was 10 seconds. It resulted in 26% spatial utilization and an average Jaccard-index of 0.41. The algorithm scored a recall accuracy of 29% but only 5% localization precision. Table 4.5 lists the evaluation criteria results. Additionally, figure 4.5 shows samples of applying the algorithm on microscopic images.

**Figure 4.5: Evaluation criteria results of Rantalankila ODP algorithm**

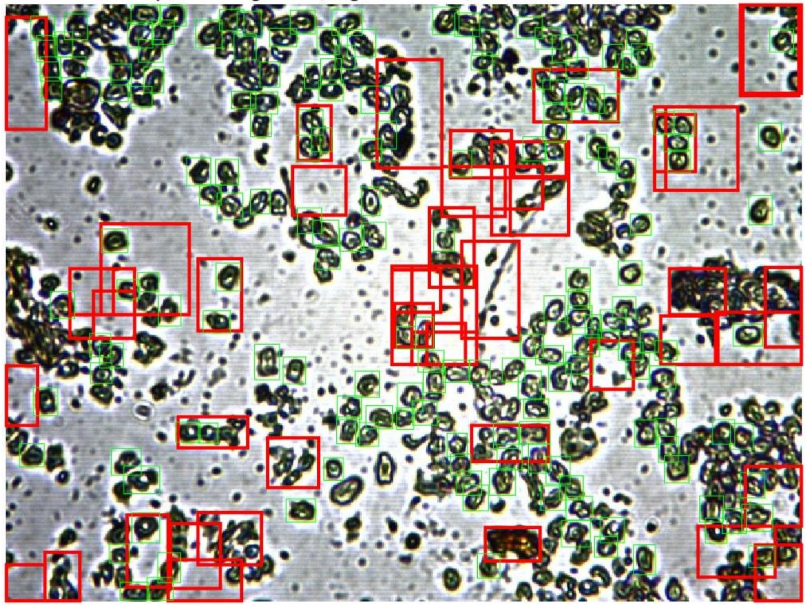
| | | | | | |
|---|---|---|---|---|---|
| **Algorithm Name:** | Rantalankila | | **Programming Lang.:** | C++ | |
| **Qualitative Criteria** | | ***Proposals Suggestable*** | ***Proposal Score*** | ***Image Dependent*** | ***Stochastic*** |
| 1. ***Algorithm Features:*** | | No (avg. 194) | No | Yes | No |
| **Quantitative Criteria:** | | **Avg.** | **Std.** | **Min.** | **Max.** |
| 2. ***Temporal Efficiency*** (Algorithm Speed) [running time in sec.] | | 10 | 0.44 | 9.35 | 10.85 |
| 3. ***Spatial Utilization*** | | 0.26 | 0.05 | 0.19 | 0.32 |
| 4. ***Accuracy:*** | | | | | |
| *Recall* | | 0.29 | 0.04 | 0.22 | 0.38 |
| *Localization Precision* | | 0.045 | 0.008 | 0.036 | 0.061 |
| 5. ***Redundancy:*** | | | | | |
| *Jaccard-index* | | 0.41 | 0.25 | 0.04 | 0.92 |
| ***Reductions*** | *Score-filter* | 0 | 0 | 0 | 0 |
| | *Size-filter* | 145.9 | 18 | 126 | 192 |
| | *Duplications-filter* | 0.8 | 1.3 | 0 | 4 |
| *Resulted Proposals (per image)* | | 47 | 12 | 27 | 64 |

**Figure 4.5: Algorithm #5: Rantalankila**

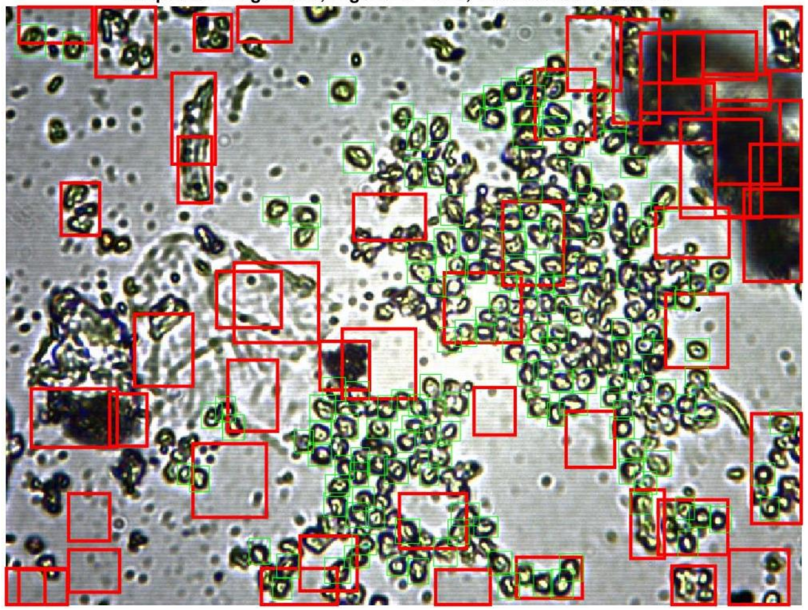
Output for image no.:1, Algorithm no.:5, Resulted Candidates:41



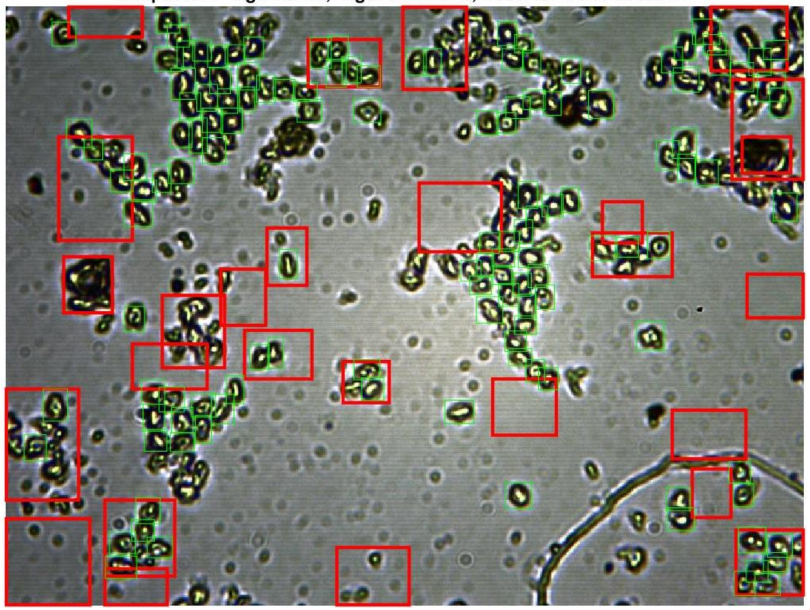
Output for image no.:4, Algorithm no.:5, Resulted Candidates:49



Output for image no.:6, Algorithm no.:5, Resulted Candidates:51



Output for image no.:10, Algorithm no.:5, Resulted Candidates:27



### 4.6. Selective Search

The Selective-Search algorithm is originally developed in MATLAB. It supports suggestable number of proposals (out of suggested 700 proposals, avg. 118 proposals per image remain after applying the filters). The algorithm is non-stochastic image-dependent, and it assigns scores to its resulting proposals.

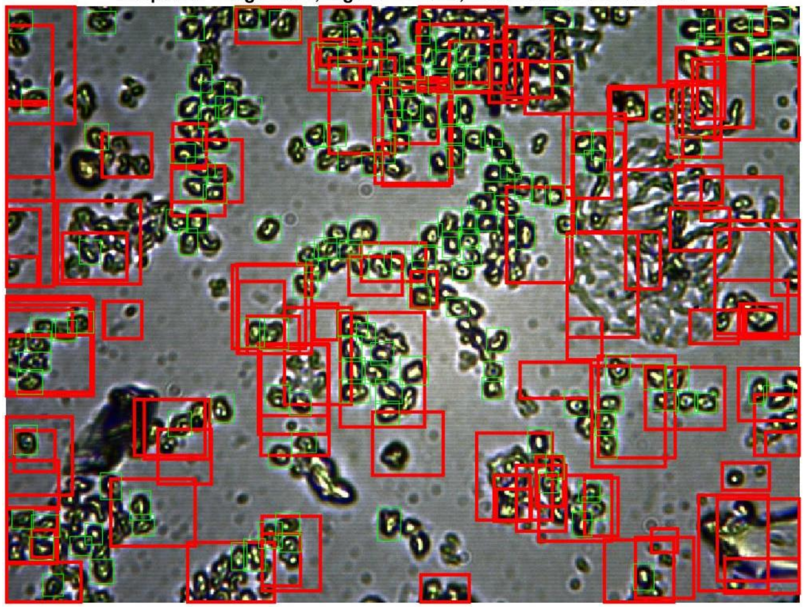
Regarding the quantitative criteria, the average running time of the algorithm was 22.3 seconds. It resulted in 35% spatial utilization and an average Jaccard-index of 1.76. The algorithm scored a recall accuracy of 70% and a localization precision of 12%. Table 4.6 lists the evaluation criteria results. Additionally, figure 4.6 shows samples of applying the algorithm on microscopic images.

**Figure 4.6: Evaluation criteria results of Selective-Search ODP algorithm**

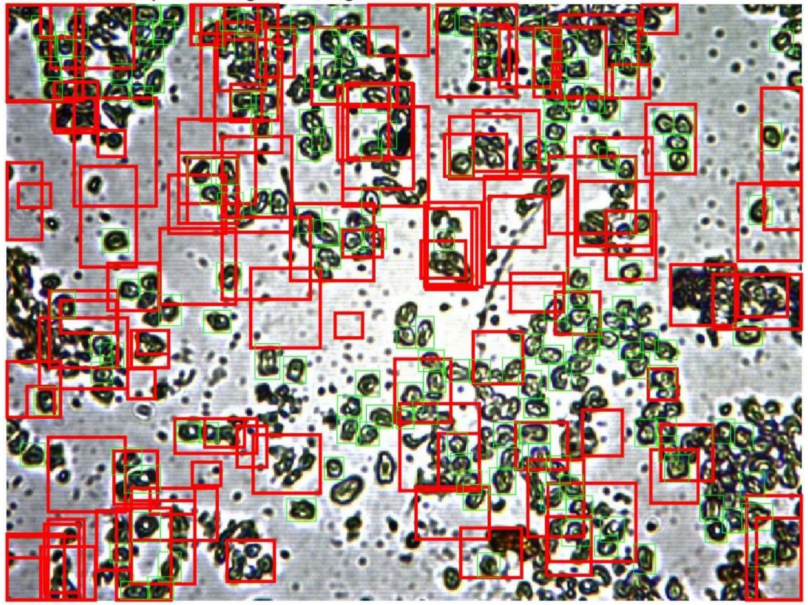
<table>
<tr><td colspan="2">Algorithm Name: Selective Search</td><td colspan="4">Programming Lang.: MATLAB</td></tr>
<tr><td colspan="2">Qualitative Criteria</td><td>Proposals Suggestable</td><td>Proposal Score</td><td>Image Dependent</td><td>Stochastic</td></tr>
<tr><td colspan="2">1. Algorithm Features:</td><td>Yes (700)</td><td>Yes</td><td>Yes</td><td>No</td></tr>
<tr><td colspan="2">Quantitative Criteria:</td><td>Avg.</td><td>Std.</td><td>Min.</td><td>Max.</td></tr>
<tr><td colspan="2">2. Temporal Efficiency (Algorithm Speed) [running time in sec.]</td><td>22.3</td><td>1.3</td><td>20.3</td><td>24.3</td></tr>
<tr><td colspan="2">3. Spatial Utilization</td><td>0.35</td><td>0.1</td><td>0.16</td><td>0.46</td></tr>
<tr><td colspan="2">4. Accuracy:</td><td colspan="4"></td></tr>
<tr><td colspan="2">Recall</td><td>0.7</td><td>0.1</td><td>0.56</td><td>0.9</td></tr>
<tr><td colspan="2">Localization Precision</td><td>0.124</td><td>0.02</td><td>0.088</td><td>0.15</td></tr>
<tr><td colspan="2">5. Redundancy:</td><td colspan="4"></td></tr>
<tr><td colspan="2">Jaccard-index</td><td>1.76</td><td>1.2</td><td>0.64</td><td>4.52</td></tr>
<tr><td rowspan="3">Reductions</td><td>Score-filter</td><td>182.4</td><td>10.7</td><td>167</td><td>199</td></tr>
<tr><td>Size-filter</td><td>399.2</td><td>19.8</td><td>367</td><td>441</td></tr>
<tr><td>Duplications-filter</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="2">Resulted Proposals (per image)</td><td>118</td><td>16</td><td>92</td><td>152</td></tr>
</table>

**Figure 4.6: Algorithm #6: SelectiveSearch**

Output for image no.:1, Algorithm no.:6, Resulted Candidates:109

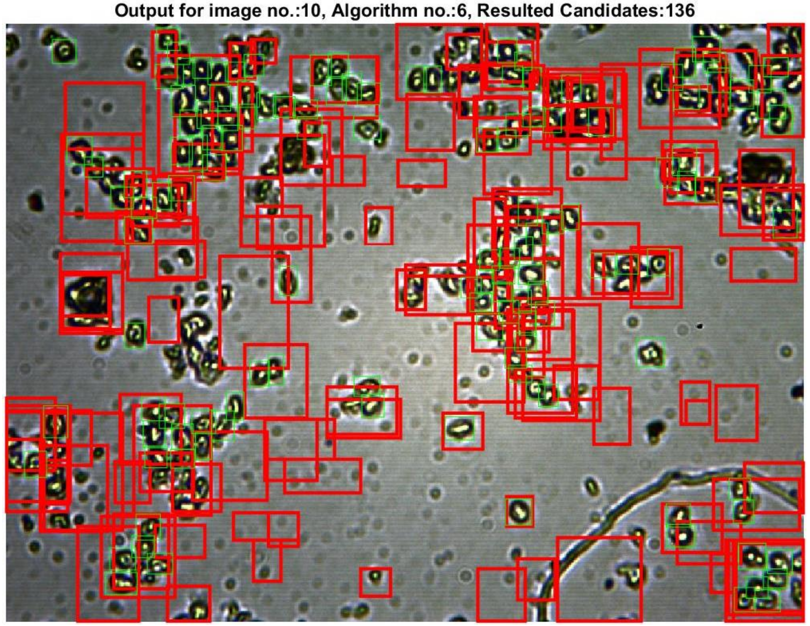


Output for image no.:4, Algorithm no.:6, Resulted Candidates:110



Output for image no.:6, Algorithm no.:6, Resulted Candidates:104

Output for image no.:10, Algorithm no.:6, Resulted Candidates:136



### 4.7. Smart Superpixels

The Smart-Superpixels algorithm is originally developed in MATLAB. It supports suggestable number of proposals (out of suggested 700 proposals, avg. 166 proposals per image remain after applying the filters). The algorithm is non-stochastic image-dependent, but it does not assign scores to its proposals.

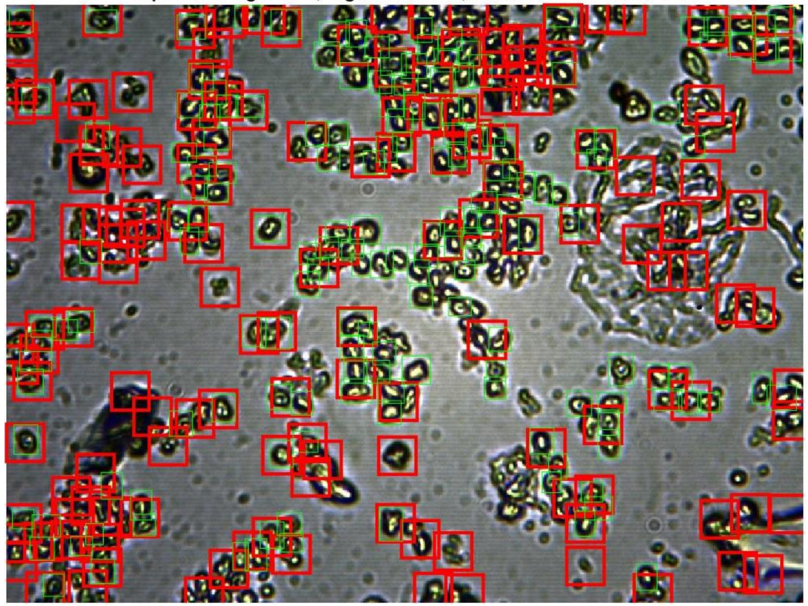
Regarding the quantitative criteria, the average running time of the algorithm was 0.31 seconds. It resulted in 38% spatial utilization and an average Jaccard-index of 0.26. The algorithm scored a recall accuracy of 79% and a localization precision of 30%. Table 4.7 lists the evaluation criteria results. Additionally, figure 4.7 shows samples of applying the algorithm on microscopic images.

**Figure 4.7: Evaluation criteria results of Smart-Superpixels ODP algorithm**

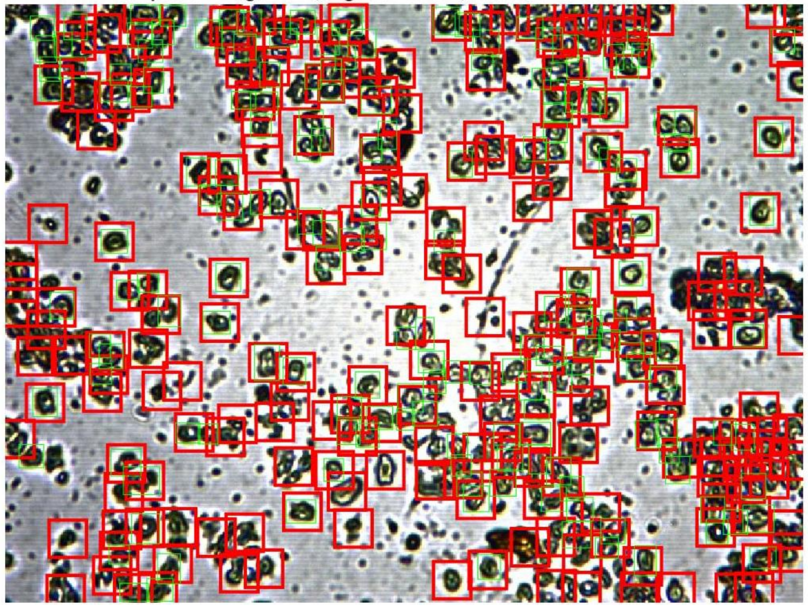
| **Algorithm Name:** Smart Super-Pixels | | **Programming Lang.:** MATLAB | | |
|---|---|---|---|---|
| **Qualitative Criteria** | ***Proposals Suggestable*** | ***Proposal Score*** | ***Image Dependent*** | ***Stochastic*** |
| 1. ***Algorithm Features:*** | Yes (700) | No | Yes | No |
| **Quantitative Criteria:** | **Avg.** | **Std.** | **Min.** | **Max.** |
| 2. ***Temporal Efficiency*** (Algorithm Speed) [running time in sec.] | 0.31 | 0.06 | 0.28 | 0.48 |
| 3. ***Spatial Utilization*** | 0.38 | 0.06 | 0.29 | 0.49 |
| 4. ***Accuracy:*** | | | | |
| *Recall* | 0.79 | 0.07 | 0.67 | 0.88 |
| *Localization Precision* | 0.3 | 0.04 | 0.25 | 0.35 |
| 5. ***Redundancy:*** | | | | |
| *Jaccard-index* | 0.26 | 0.05 | 0.19 | 0.37 |
| ***Reductions*** *Score-filter* | 0 | 0 | 0 | 0 |
| ***Reductions*** *Size-filter* | 0 | 0 | 0 | 0 |
| ***Reductions*** *Duplications-filter* | 163.5 | 30 | 123 | 218 |
| *Resulted Proposals (per image)* | 166 | 30 | 125 | 220 |

Figure 4.7: Algorithm #7: Smart Superpixels

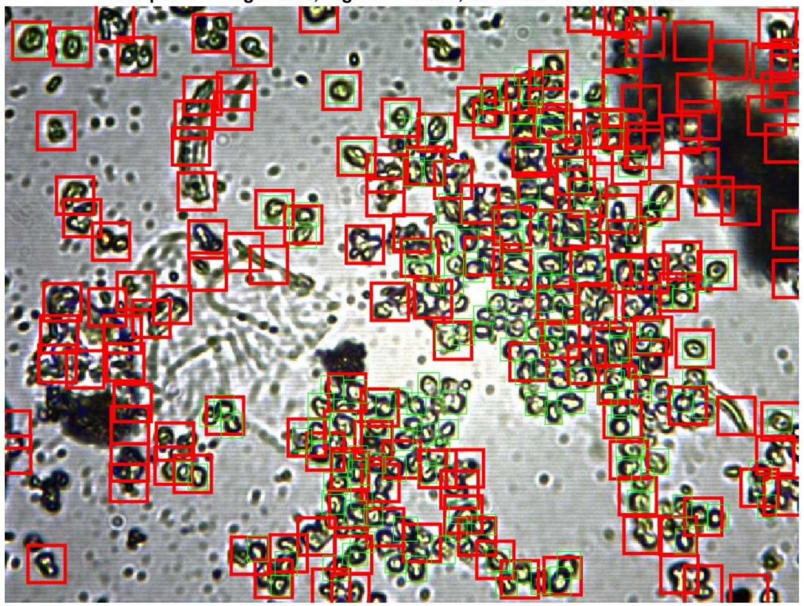
Output for image no.:1, Algorithm no.:7, Resulted Candidates:153



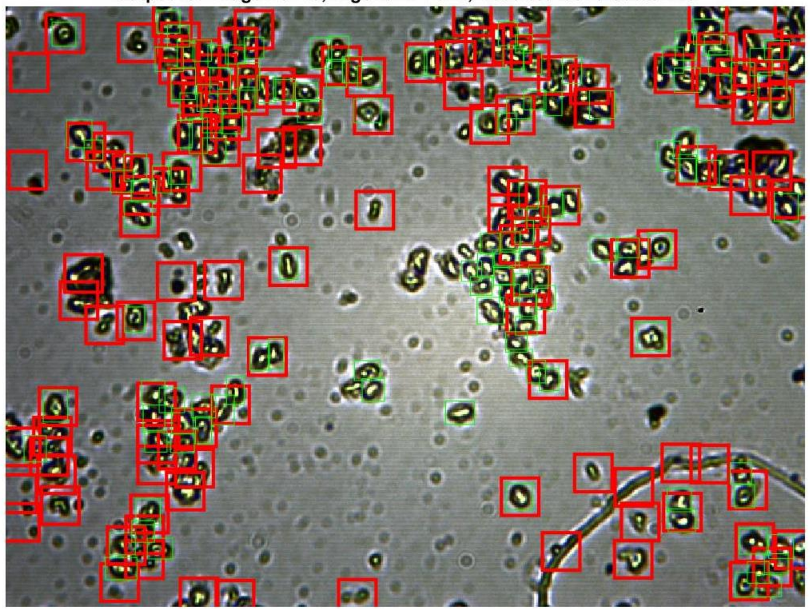
Output for image no.:4, Algorithm no.:7, Resulted Candidates:220



Output for image no.:6, Algorithm no.:7, Resulted Candidates:169



Output for image no.:10, Algorithm no.:7, Resulted Candidates:125



### 4.8. Uniform

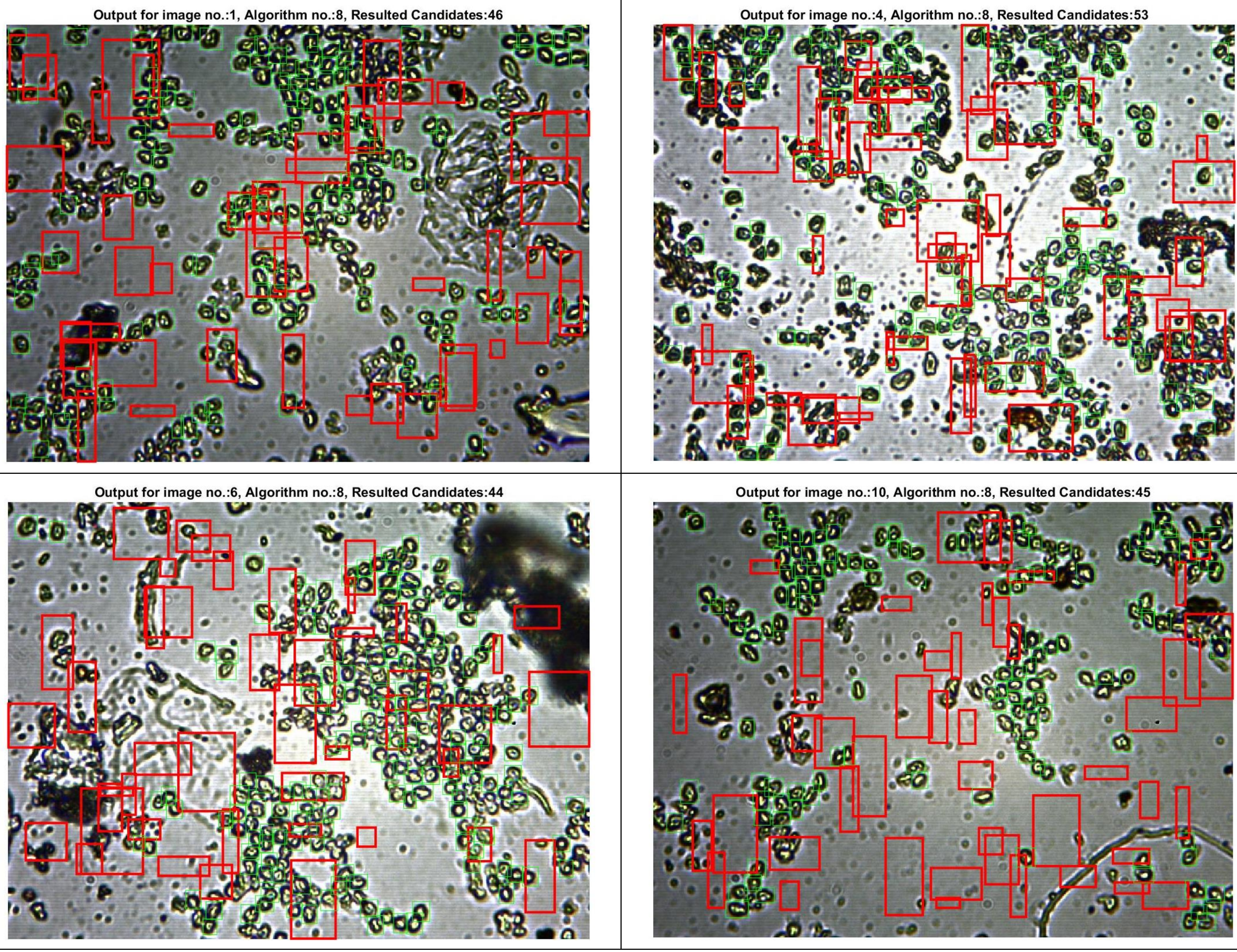
The Uniform baseline algorithm is originally developed in MATLAB. It supports suggestable number of proposals (the suggested number here was 700, out of them avg. 53 proposals per image resulted after applying filters). The algorithm is stochastic image-independent, and it does not assign scores to proposals.

Regarding the quantitative criteria, the average running time of the algorithm was 0.05 seconds. It resulted in 25% spatial utilization and an average Jaccard-index of 0.23. The algorithm scored a recall accuracy of 27% but only 5% localization precision. Table 4.8 lists the evaluation criteria results. Additionally, figure 4.8 shows samples of the results of applying the algorithm on microscopic images.

**Figure 4.8: Evaluation criteria results of Uniform ODP algorithm**

| **Algorithm Name:** Uniform | | **Programming Lang.:** Matlab | | |
|---|---|---|---|---|
| **Qualitative Criteria** | ***Proposals Suggestable*** | ***Proposal Score*** | ***Image Dependent*** | ***Stochastic*** |
| 1. ***Algorithm Features:*** | Yes (700) | No | No | No |
| **Quantitative Criteria:** | **Avg.** | **Std.** | **Min.** | **Max.** |
| 2. ***Temporal Efficiency*** (Algorithm Speed) [running time in sec.] | 0.05 | 0.14 | 0.0016 | 0.48 |
| 3. ***Spatial Utilization*** | 0.25 | 0.03 | 0.21 | 0.3 |
| 4. ***Accuracy:*** | | | | |
| *Recall* | 0.27 | 0.05 | 0.2 | 0.35 |
| *Localization Precision* | 0.05 | 0.01 | 0.034 | 0.062 |
| 5. ***Redundancy:*** | | | | |
| *Jaccard-index* | 0.228 | 0.09 | 0.143 | 0.432 |
| ***Reductions*** *Score-filter* | 0 | 0 | 0 | 0 |
| ***Reductions*** *Size-filter* | 646.5 | 6.7 | 637 | 657 |
| ***Reductions*** *Duplications-filter* | 0 | 0 | 0 | 0 |
| *Resulted Proposals (per image)* | 53 | 7 | 43 | 63 |

Figure 4.8: Algorithm #8: Uniform



5
CHAPTER
FIVE

# 5. DISCUSSION

In this chapter, the experimental results of testing the eight ODP algorithms are discussed and considered all together in a systematic benchmarking process. While the previous chapter concerns with the characteristics and evaluation of each algorithm individually, this chapter will compare between these algorithms in a collaborative manner.

Consequently, the present chapter aims at highlighting the relative strengths and weaknesses of each ODP method as well as selecting the best algorithm among them for the purposes of our research application. That is the generation of efficient object proposals for detecting airborne fungal spores in microscopic digital images, both temporally and spatially.

## 5.1. Recall Accuracy versus Speed

Profiling the recall accuracy scores of each algorithm against its processing speed, Figure 5.1 shows a quantitative comparison between the eight ODP algorithms considered in this research project.

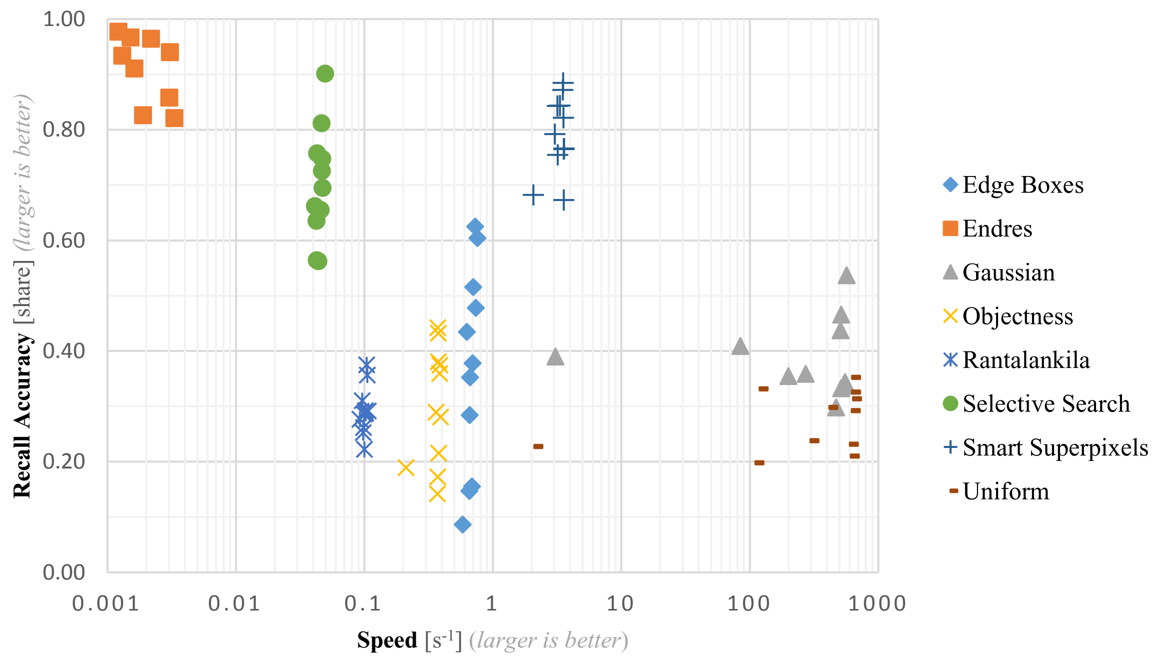



**Figure 5.1: Recall Accuracy vs Speed of ODP Algorithms**

It is highlighted in the figure 5.1 that for both indicators on the x and y axes, the larger the algorithm's score, the better performance it has. While the recall accuracy score is calculated using Equation 2 (section 3.3.4), the speed of each algorithm is calculated as the multiplicative inverse of the algorithm running time $Speed = 1 / RunningTime$.

Endres algorithm has scored the highest recall accuracy, however, it has shown to be the slowest algorithm. On the other hand, the baseline algorithms (Gaussian and Uniform) were very fast but resulting in low accuracy scores (Figure 5.1). Rantalankila had moderate speed along with low recall accuracy, while the Objectness and Edge-Boxes algorithms were a little bit faster having a large standard deviation of recall accuracy with abstract values varying between 10% and 60%.

With an average speed of 0.045 $s^{-1}$, the Selective-Search algorithms achieved high recall accuracy scores reaching 90%. The best trade-off performance in this two-dimentional comparison has been achieved by the Smart-Superpixels algorithm, for which a relative fast processing (average speed of 3 $s^{-1}$) was accompanied with high recall accuracy scores varying between 67% and 87% (Figure 5.1).

### 5.2. Recall Accuracy versus Spatial Utilization

The second aspect in comparing ODP algorithms is given to the consideration of spatial utilization versus the recall accuracy (Figure 5.2). It is worth noting here that the spatial utilization score is calculated using Equation 1 (section 3.3.3), so that the higher its value, the worse performance its corresponding algorithm has in reducing the computational power of object recognition and classification algorithms.

Similar to the previous section, the Endres algorithms scored had high contrast in its performance: with its high recall accuracy and very low spatial performance. As shown in Figure 5.2, the total area of the proposals suggested by the Endres algorithm are (on average) fifty times the image size. This makes it the worst option when considering computational power.

The Objectness algorithm had the best spatial efficiency score followed by the Edge-Boxes algorithm. However, given their low recall accuracy, they are not the optimal methods when considering the combination of the two criteria.

On the other hand, the Selective-Search and Smart-Superpixels algorithms had the best trade-off performance. Whereas their recall accuracy scores were relatively high, their resulting proposals covered 35% of the original image size on average. Finally, the remaining ODP and baseline algorithms have scored low performance in recall accuracy, despite their high spatial efficiency (Figure 5.2).

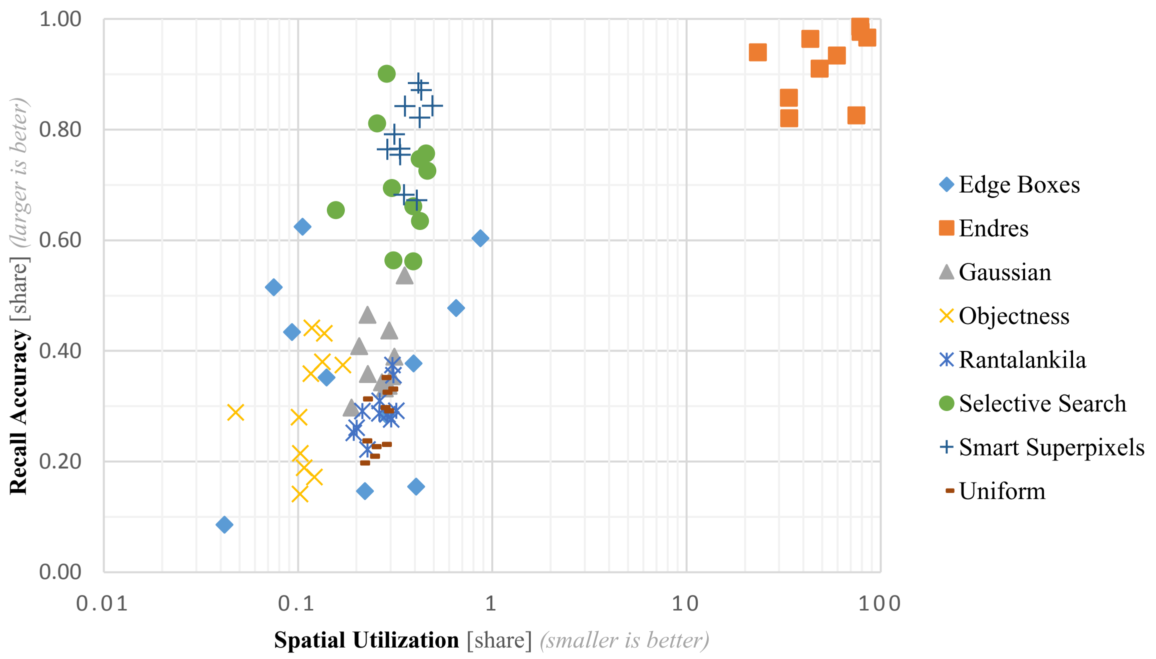



**Figure 5.2: Recall Accuracy vs Spatial Utilization of ODP Algorithms**

## 5.3. ODP Algorithms Benchmarking

To combine all the evaluation criteria for the benchmarking purpose, I introduce four composite indicators based on weighted average. These are: The qualitative features score; The speed score mainly based on temporal efficiency; The accuracy score based on recall, localization precision and the number of off-size filtered proposals; And finally, the efficiency score based on the spatial utilization and redundancy Jaccard-index.

Table 5.1 shows the scores of all ODP algorithms in all evaluation criteria indicators. Furthermore, it shows their scores in the four composite indicators along with a final benchmarking score.

As shown in the table, the Smart-Superpixels algorithm has the highest benchmarking score of 88% followed by the Selective-Search (66%). In third place comes the baseline algorithms: Uniform and Gaussian (with the overall scores of 56% and 51% respectively). The Objectness and Rantalankila algorithms follow with overall scores of 45% and 42% respectively.

Finally, the Endres and Edge-Boxes algorithms had the lowest overall scores in this benchmark with nominal values of 36% and 38% respectively.

**Table 5.1: Overall Benchmarking of ODP Algorithms**

| | | *Edge Boxes* | *Endres* | *Gaussian* | *Objectness* | *Rantalankila* | *Selective-Search* | *Smart-Superpixels* | *Uniform* |
|---|---|---|---|---|---|---|---|---|---|
| **Comparison Criteria** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| ***Proposals Suggestable*** | | ✓ | × | ✓ | ✓ | × | ✓ | ✓ | ✓ |
| ***Proposal Score*** | | × | × | × | ✓ | × | ✓ | × | × |
| ***Image Dependent*** | | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | × |
| ***Stochastic*** | | × | × | ✓ | × | × | × | × | × |
| ***Temporal Efficiency*** | | 1.5 | 587 | 0.03 | 2.9 | 10 | 22 | 0.3 | 0.05 |
| ***Spatial Utilization*** | | -0.30 | -56 | 0.27 | 0.11 | 0.26 | 0.35 | 0.38 | 0.25 |
| ***% Recall Accuracy*** | | 37% | 92% | 39% | 30% | 29% | 70% | 79% | 27% |
| ***Localization Precision*** | | 0.09 | 0.24 | 0.06 | 0.03 | 0.05 | 0.12 | 0.30 | 0.05 |
| ***Redundancy J-index*** | | -4 | -1 | 1 | 0.9 | 0.4 | 1.8 | 0.26 | 0.23 |
| ***% Duplications*** | | 0% | 5% | 0% | 0% | 0.4% | 0% | 0% | 0% |
| ***% Off-size*** | | 86% | 85% | 88% | 85% | 75% | 57% | 23% | 92% |
| ***Proposals per image*** | | 99 | 903 | 84 | 19 | 47 | 118 | 166 | 53 |
| ***Normalized Score*** | **Features Score** | 75% | 50% | 25% | 100% | 50% | 100% | 75% | 50% |
| | **Speed Score** | 60% | 0% | 100% | 54% | 41% | 33% | 77% | 95% |
| | **Accuracy Score** | 17% | 93% | 14% | 1% | 5% | 58% | 100% | 0% |
| | **Efficiency Score** | 0% | 0% | 66% | 26% | 72% | 74% | 100% | 78% |
| **OVERALL SCORE** | | **38%** | **36%** | **51%** | **45%** | **42%** | **66%** | **88%** | **56%** |

To sum up, figure 5.3 below shows the algorithms score in the main four comparative aspects: that are the localization precision, the recall accuracy, the spatial efficiency, and the speed.

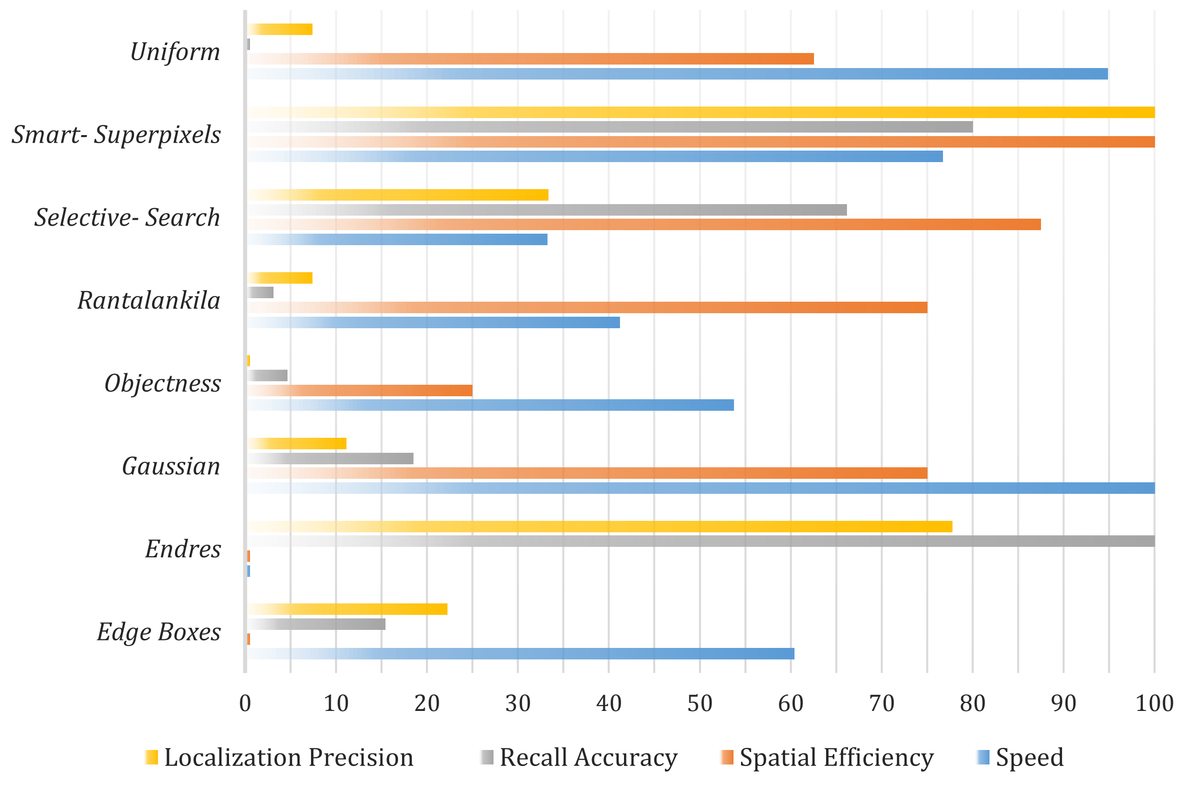
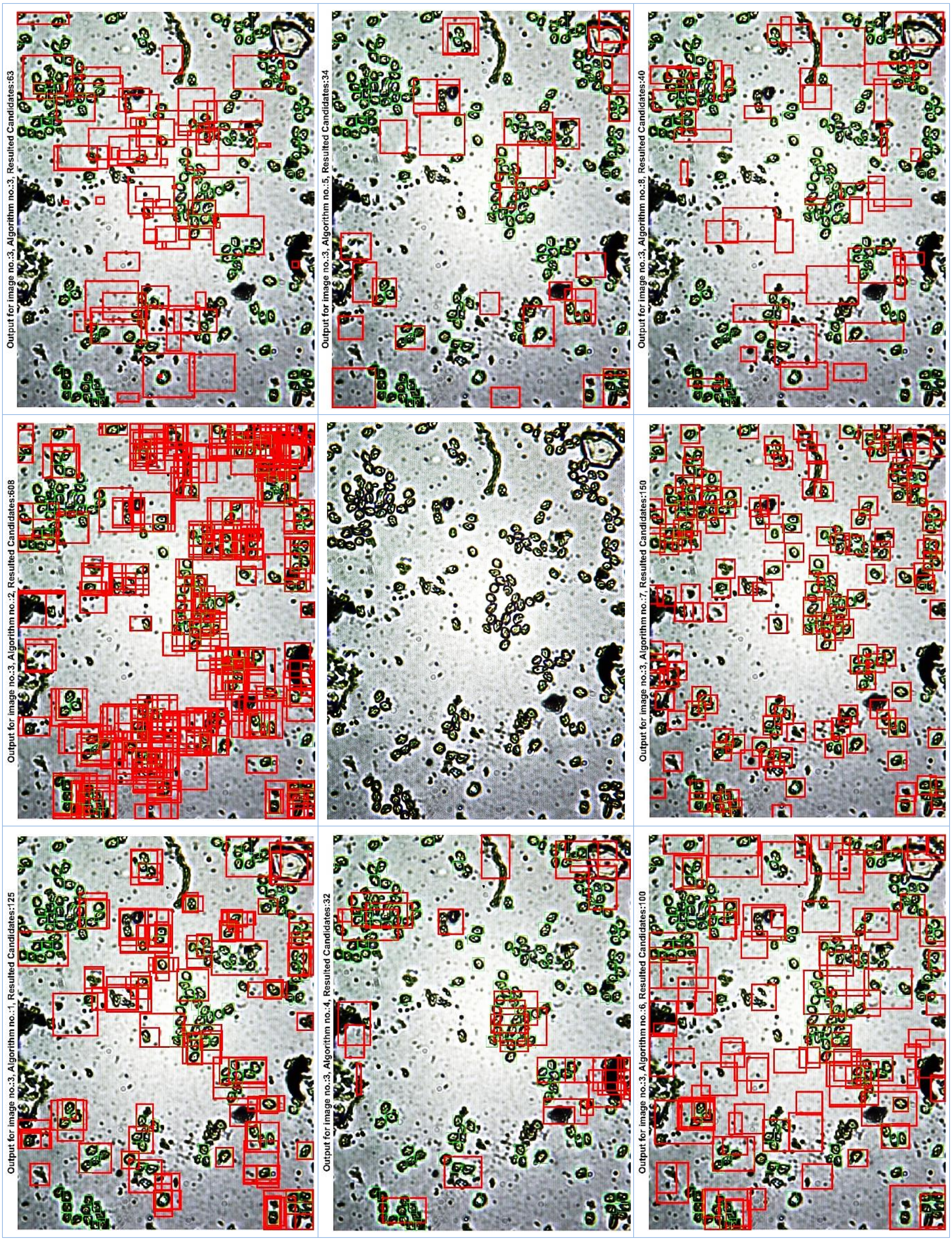



**Figure 5.3: Overall Benchmarking of ODP Algorithms**

Finally, figure 5.4 (on the next page) shows the actual results of all algorithms regarding the same input image, with the red bounding boxes represent the resulting proposals by the algorithm, and the green boxes represents the annotations.


Output for image no.:3, Algorithm no.:3, Resulted Candidates:63
Output for image no.:3, Algorithm no.:5, Resulted Candidates:34
Output for image no.:3, Algorithm no.:8, Resulted Candidates:40
Output for image no.:3, Algorithm no.:2, Resulted Candidates:608
Output for image no.:3, Algorithm no.:7, Resulted Candidates:150
Output for image no.:3, Algorithm no.:1, Resulted Candidates:125
Output for image no.:3, Algorithm no.:4, Resulted Candidates:32
Output for image no.:3, Algorithm no.:6, Resulted Candidates:100

6
CHAPTER
SIX

# 6. CONCLUSION

The present research project has studied the application of software algorithms for detecting object proposals in digital images based on their contents and graphical characteristics. The fast generation of accurate object proposals is widely seen as a promising solution to the efficiency problems of heavy classification and object recognition algorithms, which require long time and huge computational power to be applied on entire large digital images. Instead, object detection proposals (ODPs) with smaller sizes and focused content are an optimum alternative for such classification algorithms.

This report documents a research project of reviewing and benchmarking the state-of-the-art object detection proposal algorithms. Throughout the project, seven different algorithms from the computer science literature were reviewed, implemented and tested. Additionally, a novel algorithm was developed by the author, specifically for the purpose of generating object proposals of airborne fungal spores in microscopic images. The introduced algorithm is called the "Smart-Superpixels" algorithm.

Creating a MATLAB program code, the project comprises the computational steps of harmonizing the eight ODP algorithms, applying them on several digital images for air samples, and calculating a set of quantitative and qualitative evaluation indicators for benchmarking. The results have highlighted the main strengths and weaknesses of each algorithm in respect of their temporal and spatial efficiency, recall accuracy, localization precision and redundancy.

The ODP algorithms considered in this research project are: Edge-Boxes, Endres, Gaussian baseline, Objectness, Rantalankila, Selective-Search, Smart-Superpixles, and Uniform baseline.

The benchmarking results show that the introduced algorithm: Smart-Superpixels and the Selective-Search algorithm have the highest overall performance with fast operation, relatively high spatial efficiency, high recall accuracy and localization precision, as well as low redundancy.

Despite its very high recall accuracy, the Endres algorithm is very time consuming and can result in high redundant proposals with low spatial efficiency.

On the other hand, the Objectness, Rantalankila and baseline algorithms are fast and spatial efficient, however, their recall accuracy and localization precision are relatively low. Despite its good qualitative features and fast operation, the low accuracy, precision and spatial efficiency of Edge-Boxes algorithm has made it unsuitable for the purpose of generating fungal spores object proposals.

The best trade-off performance of speed, accuracy and spatial efficiency were found in the Selective-Search algorithm as well as in the introduced ODP method of Smart-Superpixels. Accordingly, these two algorithms have proved to be the most convenient algorithms for purpose of airborne fungal spores' detection in microscopic images.

While the Smart-Superpixels algorithm can be considered as a promising and under development ODP method, further development of its operation is yet to be considered in future research. For instance, it could be beneficial to expand the algorithm's detection proposals through union combinations of continuous neighboring proposals to generate larger ones. Such process can result in more accurate proposals with various sizes based on the real objects, and thus could result in the generalization of the algorithms for further object detection applications beyond the microbial identification in microscopic samples. Additionally, it can reduce the total number of resulting proposals and reduce the redundancy indicators accordingly.

All in all, the project has highlighted the potential importance of object detection proposals in supporting real-time classification and pattern recognition algorithms.

# 7. REFERENCES

[1] L. I. Kleinman, P. H. Daum, Y.-N. Lee, E. R. Lewis, A. J. Sedlacek, G. I. Senum, S. R. Springston, J.Wang, J. Hubbe, J. Jayne, Q. Min, S. S. Yum und G. Allen, „Aerosol concentration and size distribution measured below, in, and above cloud from the DOE G-1 during VOCALS-REx,“ *Atmos. Chem. Phys.*, Bd. 12, Nr. doi:10.5194/acp-12-207-2012, p. 207–223, 2012.

[2] S. Kecorius, N. Kivekäs, A. Kristensson, T. Tuch, D. S. Covert, W. Birmili, H. Lihavainen, A.-P. Hyvärinen, J. Martinsson, M. K. Sporre, E. Swietlicki, A. Wiedensohler und V. Ulevicius, „Significant increase of aerosol number concentrations in air masses crossing a densely trafficked sea area,“ *Oceanologia*, Bd. 58, Nr. 1, pp. 1-12, 2016.

[3] R. Aditya, „Studies on the non-linear behavior of the monsoon rainfall of Tamil Nadu & prediction using statistical and artificial neural network techniques,“ University of Madras, July 2012. [Online]. Available: http://hdl.handle.net/10603/204458. [Zugriff am 15 May 2018].

[4] J. Deacon, „The Microbial World: Airborne Microorganisms,“ Institute of Cell and Molecular Biology, The University of Edinburgh, 2001. [Online]. Available: http://archive.bio.ed.ac.uk/jdeacon/microbes/airborne.htm. [Zugriff am 15 May 2018].

[5] M. W. Tahir, N. A. Zaidi, R. Blank, P. P. Vinayaka, M. J. Vellekoop und W. Lang, „Detection of Fungus through an Optical Sensor System using the Histogram of Oriented Gradients,“ *Proc. IEEE SENSORS*, Bd. Nov., pp. 418-420, 2016.

[6] P. H. McMurry, „A review of atmospheric aerosol measurements,“ *Atmospheric Environment*, Bd. 34, Nr. 12-14, pp. 1959-1999, 2000.

[7] R. Blank, P. P. Vinayaka, M. W. Tahir, M. J. Vellekoop und W. Lang, „Optical sensor system for the detection of mold: Concept for a fully automated sensor system for the detection of airborne fungal spores,“ in *Proc. IEEE SENSORS*, Busan, 2015.

[8] J. Hosang, R. Benenson, P. Dollár und B. Schiele, „What Makes for Effective Detection Proposals?," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Bd. 38, Nr. 4, pp. 814-830, 2016.

[9] N. Dalal und B. Triggs, „Histograms of oriented gradients for human detection," *Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR),* Bd. 1, p. 886–893, 2005.

[10] P. Dollár, „Generating Object Proposals," 22 Dec. 2013. [Online]. Available: https://pdollar.wordpress.com/2013/12/22/generating-object-proposals/. [Zugriff am 28 Jun. 2018].

[11] A. Krizhevsky, I. Sutskever und G. Hinton, „Imagenet Classification with Deep Convolutional Neural Networks," *Proceedings Adv. Neural Inf. Process. Syst. (NIPS),* 2012.

[12] C. C. Pham und J. W. Jeon, „Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks," *Signal Processing: Image Communication,* Bd. 53, Nr. April, pp. 110-122, 2017.

[13] N. Chavali, H. Agrawal, A. Mahendru und D. Batra, „Object-Proposal Evaluation Protocol is 'Gameable'," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 835-844, 2016.

[14] C. L. Zitnick und P. Dollár, „Edge Boxes: Locating Object Proposals from Edges," in *European Conference on Computer Vision (ECCV)*, Zurich, 2014.

[15] I. Endres und D. Hoiem, „Category-Independent Object Proposals With Diverse Ranking," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Bd. 36, Nr. 2, pp. 222-234, 2014.

[16] I. Endres und D. Hoiem, „Category Independent Object Proposals," in *The 11th European Conference on Computer Vision (ECCV)*, Crete, Greece, 2010.

[17] M. Everingham, L. Van Gool, C. Williams, J. Winn und A. Zisserman, „The PASCAL Visual Object Classes Challenge (VOC2007)," 2007. [Online].

Available: http://host.robots.ox.ac.uk/pascal/VOC/voc2007/. [Zugriff am 14 July 2018].

[18] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn und A. Zisserman, „The PASCAL Visual Object Classes Challenge: A Retrospective,“ *International Journal of Computer Vision,* Bd. 111, Nr. 1, pp. 98-136, 2015.

[19] B. Alexe, T. Deselares und V. Ferrari, „What is an object?,“ in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, 2010.

[20] B. Alexe, T. Deselares und V. Ferrari, „Measuring the objectness of image windows,“ *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Bd. 34, Nr. 11, pp. 2189 - 2202, 2012.

[21] P. Rantalankila, J. Kannala und E. Rahtu, „Generating Object Segmentation Proposals Using Global and Local Search,“ in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2417-2424*, Columbus, OH, 2014.

[22] K. van de Sande, J. Uijlingsy, T. Gevers und A. Smeulders, „Segmentation as Selective Search for Object Recognition,“ in *The 13th International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011.

[23] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers und A. W. M. Smeulders, „Selective Search for Object Recognition,“ *International Journal of Computer Vision,* Bd. 104, Nr. 2, pp. 154-171, 2013.

[24] M. W. Tahir, N. A. Zaidi, A. A. Rao, R. Blank, M. J. Vellekoop und W. Lang, „A Fungus Spores Dataset and a Convolutional Neural Network Based Approach for Fungus Detection,“ *IEEE Transactions on NanoBioscience, vol. ,* Bd. 17, Nr. 3, pp. 281-290, 2018.

[25] Alex, „How to calculate mAP for detection task for the PASCAL VOC Challenge?,“ Data Science - StackExchange, 26 November 2017. [Online]. Available: https://datascience.stackexchange.com/questions/25119/how-to-calculate-map-for-detection-task-for-the-pascal-voc-challenge. [Zugriff am 4 August 2018].