On the development of an unsupervised probabilistic algorithm for grayscale fluorescence image segmentation

Bachelor thesis - FYTK02

Magnus Brander

supervised by
Tobias Ambjörnsson

LUND
UNIVERSITY

Department of Astronomy and Theoretical Physics

Lund University

June 12, 2018

Abstract

In the field of computational biology, fluorescence microscopy images often constitute the input source of information. The process of binarization of raw images to delineate interesting objects requires image segmentation into signal and background pixels.

Several methods to perform image segmentation exist, the Otsu method being a popular unsupervised example. The Otsu method's lack of probabilistic predictions in terms of accuracy is a limiting factor when it comes to evaluation of the segmented images and their correctness.

Based on the assumption that the background intensity distribution is Gaussian we present a new unsupervised probabilistic segmentation algorithm. The new algorithm uses Bayesian decision theory to classify pixels as signal and background respectively, and provides a prior estimate for the fractions of correctly classified pixels.

Segmentation tests performed on artificial fluorescent images show that the new algorithm performs significantly better for high level of additive noise than the Otsu method. For a low level of additive noise, the new algorithm performs similarly to the Otsu method. Furthermore, the new algorithm provides a prior estimate for the fraction of correctly classified pixels close to the true values.

We hope the new algorithm will constitute a good alternative to already established methods, offering precise probabilistic image segmentation.

Popular science – Quantifying the seemingly obvious

With help of our eyes and brain we as humans excel when it comes to extracting meaningful information from our surrounding. The fact that you can read this text and hopefully understand its message, demonstrates great capabilities in handling complex systems and huge amounts of information.

The invention of the camera enabled decades of attempts and endeavors in perfecting the art of portraying reality to be out-performed; taking a picture can today be done with only a fraction of all the practice that earlier painters needed. When portraying is done by the help of the human hand, mind and sight, there will always be room for subjectivity, and the only certain thing is that two people most likely will portray the same scene differently. However, with a camera there is no room for statements like "I saw it that way", and humanity can enter a new era of objective book-keeping.

As our access to information through images increase with a modernized society, the era of internet and digitalization, the need of methods to efficiently and accurately extract meaningful information from images increases as well. *Image analysis*, which aims to extract meaningful information from images, is therefore present in more areas of society that many can imagine. Face recognition features in smartphones, goods inspection in industry and cutting edge research of DNA all rely on image analysis.

In science there is a need of quantifying results and ensuring their reproducibility. Subjective statements are of little value even though they are based on trivial observations. Therefore, one of many challenges in image analysis is to quantify the seemingly obvious.

In the field of computational biology, image analysis is used to extract the information embedded in fluorescence microscopy images. Fluorescent images are for example used to study cancer cells and DNA molecules in closer detail. An important tool in the analysis of fluorescent images consists of image segmentation, which aims to produce a binary image highlighting interesting objects. The work presented in this study approaches the task of achieving optimal binary images, and tries to improve the result as compared to already existing methods designed for this task.

We present a new method to convert fluorescent images into binary images that performs remarkably better than an existing popular method, and in addition to this can predict the accuracy of the achieved extraction of interesting objects. From the predicted accuracy we can, in addition to just reporting a result, also give valuable information to the researcher whether the results can be trusted or not. As knowledge about uncertainty in experiments is of great importance in science we see that the new method could potentially fill that gap found in the methods used today.

We hope that the new method with its benefits for fluorescent image segmentation will help biological and medical research to progress with the image segmentation tasks they are challenged with.

Acknowledgements

I would like to thank my supervisor Tobias Ambjörnsson for all guidance, help and inspiration throughout the entire process of writing this thesis. I would also like to thank the participant of the bio-math seminar group, Albertas Dvirnas and Julie Zhu for their contribution with ideas to solution of problems I faced during this project.

Table of contents

1 Introduction and background

In this section we give a brief introduction to *image analysis*. In addition, we give background information regarding the task of segmenting grayscale images and explain how this study intends to improve upon that task.

1.1 Image analysis broad perspective

Image analysis aims to extract meaningful information from images and it is to be found in many branches of science, industry and other parts of our society. The applications and impacts from image analysis range from fields of research such as medicine, astronomy, physics, biology and meteorology, to more industrial applications such as goods inspection, quality controls and machine perception, and finally societal areas such as surveillance, face recognition and simple barcode scanning [1]. It is of no doubt that the advances made in the field of image analysis will play, and has played, a role in the developments of our society. As a sub-branch of image analysis we find digital image processing, a branch which process an input image consisting of a finite number of so called pixels. There are basically three levels of digital image processing, low-, mid- and high-level processes. In the realm of low-level processes one finds rather simple tasks such as reduction of noise or image sharpening where both the input and output are images. The mid-level processing incorporates tasks such as segmentation of images, finding simple objects, edges and contours in images. When it comes to high-level processing we find tasks dealt with by fields such as machine learning and vision as well as artificial intelligence (AI). At this stage the output aims to make sense of the input in some preferable manner and the tasks as well as implementations vary [1, 2, 3].

In this study we will stay in the mid-level of image processing and we can further characterize this level into four different types: supervised and probabilistic, unsupervised and probabilistic, non-probabilistic and supervised as well as probabilistic and unsupervised. Unsupervised methods require no input parameters from the user, e.g., the Otsu method [4]. Probabilistic methods incorporate a probabilistic framework and have the benefits of being able to predict the accuracy of the output, as for example neural networks [5].

We will in this work mainly focus on segmentation of fluorescence images. Such images are common in, for example, different fields of cell biology [6, 7]. Image segmentation in general terms refers to the task of subdividing an image into its constituent objects or regions [1].

1.2 The Otsu method

The Otsu method is a popular and often used method for image segmentation of grayscale images. This method is rather simple and implemented in popular software and here serves as a benchmark to which we compare our new method. The segmentation in the Otsu method builds on the fact that there exists two classes of pixels: foreground and background. The method segments images by choosing a threshold in intensity such that the sum of their intra-class variance is minimized [4]. The Otsu method can further be generalized to include more than two classes of pixels. As an image processing algorithm the Otsu method sets itself in the mid-level of processing with the further characteristics of being unsupervised but not probabilistic. In other words, it is not requiring any user input but does not provide any prior estimation of how good one can expect the segmentation to be.

1.3 Our alternative algorithm

In this study we aim to develop a method that is both unsupervised and probabilistic. To the assumption that there exist two classes, we add that the intensity distribution of the background follows a known functional form. For this study we have limited ourselves to the intensity distribution of the background being normally distributed, also called Gaussian distributed. With this further assumption, that in some cases can be rather well motivated, especially for the fluorescence microscopy images, we will try to develop a segmentation technique that provides an estimation of how successful the segmentation is expected to be, and does not have any free parameters. The goal for the new algorithm is that it, in contrast to the Otsu method, can provide an estimate of

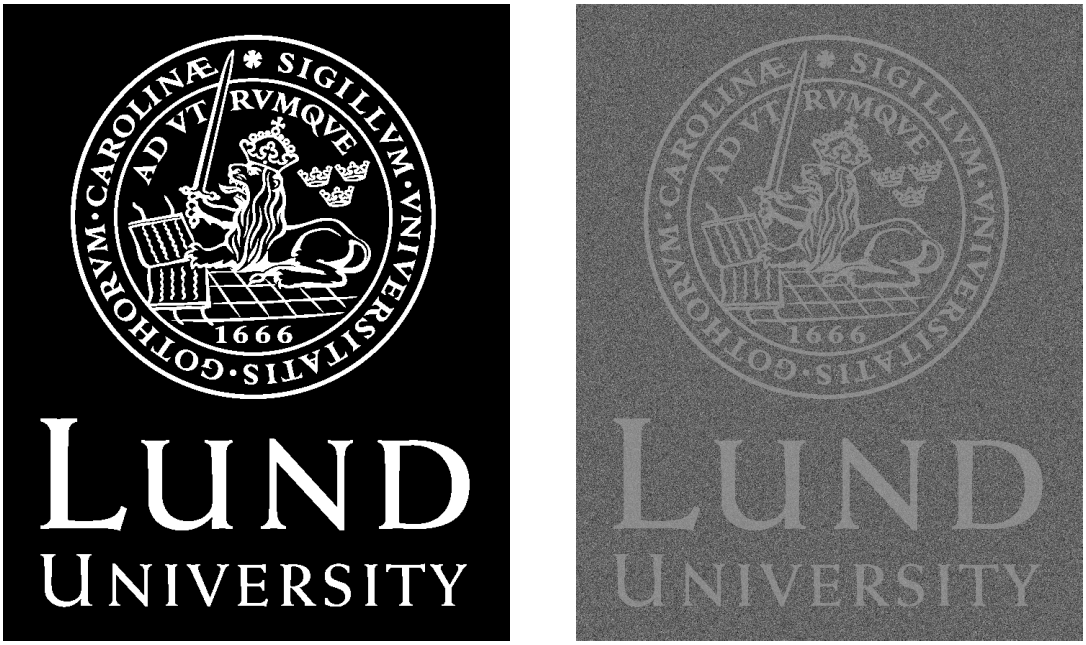
the number of correctly classified pixels as well as perform on a level equal or better than the Otsu method when it comes to the number of correctly classified pixels.

2 Problem statement

Let us here clarify what the problem at hand consist of and what we aim to solve throughout the rest of the report. From fluorescence microscopy imaging one obtains an image in grayscale, see Fig. 1b for an example of an artificially created version of such an image. Some spatial regions in these images are considered more interesting and appear to be higher in intensity. Let us denote the intensity with I, and call these spatial regions – *signal regions*. Other regions which are considered to be of less interest are generally lower in intensity – *background regions*. The challenge is to separate the signal regions from the background regions, to obtain a binary image with two categories of pixels.

The intensity background regions in this study will be taken as normally distributed with a mean intensity μ_{bg} and a standard deviation of σ_{bg}. The signal intensities, which in this study can be both normally and uniformly distributed, will have a mean μ_{sig} which is greater than μ_{bg}. The signal and background intensities may (however) overlap to some extent, but not completely.

Now consider the binary image seen in Fig. 1a and take this to be the true state of signal and background regions, indicated with white and black pixels. By applying some noise to the image, where the intensity for each individual black pixel is replaced by a random value drawn from the Gaussian distribution with mean μ_{bg} and standard deviation σ_{bg}, and every white signal pixel is given a new random intensity value drawn from another Gaussian distribution with mean μ_{sig} and standard deviation σ_{sig}, the noisy image seen in Fig. 1b is obtained. This kind of image aims to mimic real fluorescence images.



(a) Original image (b) Noisy image

Figure 1: **Example images (a):** The original logotype of Lund University, downloaded from www.medarbetarwebben.lu.se, with some slight modifications to exclude smooth edges. **(b):** An example of a blurred grayscale version of the original logotype shown in **(a)**. The white (signal) pixels in **(a)** have in the new noisy image randomly been given a new intensity which follows a Gaussian distribution with the parameters $\mu_{\text{sig}} = 0.55$ and $\sigma_{\text{sig}} = 0.035$, where μ_{sig} is the mean and σ_{sig} the standard deviation of the distribution. The black pixels (background) in **(a)** have in the same manner as the signal pixels been given new random intensities according to a different Gaussian distribution with the parameters $\mu_{\text{bg}} = 0.4$, $\sigma_{\text{bg}} = 0.1$, μ_{bg} being the mean and σ_{bg} the standard deviation of the distribution. In the Result section 4, the parameters for the intensity distributions are varied in order to get images which are of different difficulty to segment.

The problem at hand now is: Given only the image in Fig. 1b, recover the binary image seen in Fig. 1a to as high a degree as possible, by classifying each pixel as either signal or background, i.e. white or black. As with the Otsu method, we aim to perform this without knowing what the original image might look like and what shapes the signal or background regions might have.

In order to evaluate to which accuracy the recovered image represents the true image, we will in this report only use a known binary image as the one found in Fig. 1a to obtain noisy images as seen in Fig. 1b. This allows us to count how many pixels have been correctly classified, and constitutes throughout the report the actual number of correctly classified pixels. With real images, thus the answer, or rather correct signal and background pixel positions, is not known which is the reason why only artificial images will be used.

3 Methods

Let us start with some background to motivate the image segmentation procedure which shall follow in this section. An image is nothing else than a matrix, a matrix containing elements, elements here being simply numbers where the number of photons that hit a certain pixel is stored. The number of photons that arrived after a certain time to each pixel, assuming that they come from a stochastic process (i.e. noise), can in many cases be described by a Poisson distribution [8, 9]. In the limit when the number of photons increase, i.e., the mean grows larger, the Poisson distribution approaches a Gaussian distribution (see appendix A). It is therefore reasonable to assume the background intensity distribution of images with high level of random noise in the form of photons to follow a Gaussian. In the cases when the background intensity distribution does not follow a Gaussian, there are in principle no problem changing the functional form to something more suitable, as for example the Poisson distribution. Henceforth, we consider the cases when the background intensity distribution is indeed Gaussian and work from that fact.

3.1 The new algorithm – overview

So, assuming the background intensity distribution to be Gaussian, a new algorithm to classify pixels as signal or background will be proposed. The algorithm consists of a few elementary steps which will now be stated.

1. **Fitting a truncated Gaussian**
   (a) We automatically find an optimal intensity value $I_{\text{cut-off}}$ for which all pixels lower than this in intensity can be considered as true background pixels. With this intensity value at hand we fit a truncated Gaussian to all intensities up to $I_{\text{cut-off}}$ and extract the mean μ_{bg} and standard deviation σ_{bg} of the background intensity distribution.
   (b) We now use μ_{bg}, σ_{bg} and $I_{\text{cut-off}}$ to calculate an estimate for the number of background pixels, N_{bg}.
2. **Bayesian statistics to classify pixels as signal and background**
   (a) We here start by estimating the prior probability in Bayes theorem, see equation (15), for all pixels with an intensity higher than $I_{\text{cut-off}}$ with the help of the fitted Gaussian and the total intensity histogram for the image.
   (b) Using the true background pixels as training data, a conditional probability density function, which depends on a feature value extracted from the characteristics of a pixel's local neighborhood, for both signal and background pixels, is obtained.
   (c) With the prior and conditional probability we use Bayes' theorem to calculate the posterior probabilities for all pixels which are higher than $I_{\text{cut-off}}$ in intensity. With the posterior probabilities we classify all pixels above $I_{\text{cut-off}}$ to obtain a binary image as well as estimate the number of misclassified pixels.

3.2 The new algorithm – details

The steps presented in the following parts build upon the assumption that the intensities of the background pixels follow a Gaussian distribution, see section 2. Furthermore, it assumes that the regime of background and signal intensities do not overlap completely, i.e. that there exist an intensity regime of non-negligible size consisting of true background pixels. If this is not the case, the algorithm will not be applicable.

Step 1 - Fitting a truncated Gaussian

We will here present strategies for how to estimate the background distribution parameters μ and σ. Furthermore, we provide a method to automatically estimate the cut-off intensity $I_{\text{cut-off}}$ and calculate the estimated number of background pixels N_{bg}.

Step 1a – Estimation of μ, σ and $I_{\text{cut-off}}$

To begin with, we need to cover the procedures inherited in step 1a, starting with the procedure of how to fit a truncated Gaussian and extract μ and σ. So to illustrate, let us consider a grayscale image as the one seen in Fig. 1b with the intensity values $I(x, y)$, where x and y labels all the pixels in the image. x and y only take on integer values as $1 \leq x \leq N_x$ and $1 \leq y \leq N_y$. N_x and N_y here denote the number of rows and columns of the image. In the procedure of fitting a truncated Gaussian we start by creating a sorted 1-dimensional array, let us here denote it by $\vec{I}_{\text{sorted}}$, containing all the values in $I(x, y)$ and sorted in ascending order. The intensity cut-off value $I_{\text{cut-off}}$, bellow which all pixels are regarded as true background pixels, is here assumed to be known (see page 6 for auto-estimation of this quantity).

We now want to estimate μ and σ for the background intensities. Looking at the intensity histogram H, obtained from $\vec{I}_{\text{sorted}}$, see Fig. 2, we see that the intensities for the signal and background overlap.

In order to get around this problem we fit a *truncated* Gaussian probability density function, here called a Gaussian PDF, to the intensities below $I_{\text{cut-off}}$ [10]. We start with an untruncated Gaussian PDF, g,

$$g(I; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(I-\mu)^2}{2\sigma^2}}, \quad -\infty \leq I \leq \infty \tag{1}$$

where μ is the mean and σ the standard variation of the distribution. From this the truncated Gaussian PDF p becomes

$$p(I; \mu, \sigma) = \begin{cases} Ag(I; \mu, \sigma) & -\infty \leq I \leq I_{\text{cut-off}}, \\ 0 & I_{\text{cut-off}} \leq I \leq \infty \end{cases} \tag{2}$$

where A is a normalization constant and $I_{\text{cut-off}}$ is the upper intensity value at which the PDF is truncated. Since it is required that

$$\int_{\mathbb{R}} p(I; \mu, \sigma)\, dI = 1, \tag{3}$$

A becomes

$$A = \frac{1}{\Phi\left(\frac{I_{\text{cut-off}}-\mu}{\sqrt{2}\sigma}\right)}, \tag{4}$$

with

$$\Phi\left(\frac{I'-\mu}{\sqrt{2}\sigma}\right) = \int_{-\infty}^{I'} g(I; \mu, \sigma)\, dI = \frac{1}{2}\left[1 + \text{erf}\left(\frac{I'-\mu}{\sqrt{2}\sigma}\right)\right]. \tag{5}$$

To extract the unknown parameters μ and σ with a given value of $I_{\text{cut-off}}$ we here use the maximum likelihood method [11]. The idea behind the maximum likelihood estimation is to

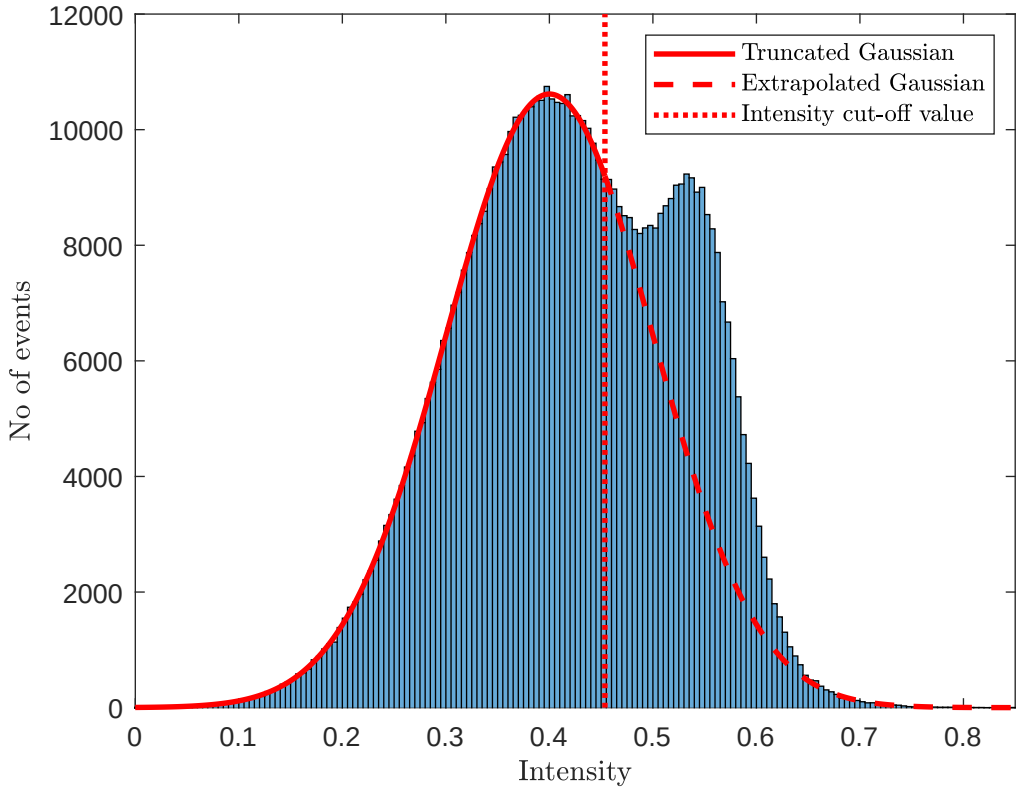


Figure 2: **Final truncated Gaussian fit.** In the figure the intensity histogram is shown for the image in Fig. 1b. The corresponding obtained truncated Gaussian fit is indicated with a solid line up to the intensity cut-off value $I_{\text{cut-off}} = 0.454$, which was estimated using the procedure of auto-estimation of $I_{\text{cut-off}}$ (see page 6). From the procedure of fitting a truncated Gaussian, the parameters for the background distribution was found to be $\mu_{\text{bg}} = 0.40007 \pm 0.00045$ and $\sigma_{\text{bg}} = 0.09999 \pm 0.00025$, which agrees well with the true values $\mu_{\text{bg}} = 0.4$ and $\sigma_{\text{bg}} = 0.1$. The Gaussian fit is then extrapolated into the region where background and signal intensities are mixed and is there indicated with a long-dashed line. From the fit, the pre-factor of the distribution, or equivalently the number of background pixels N_{bg} in the image (see (14)), was found to be $N_{\text{bg}} = 532184 \pm 2970$ which agrees well with the true value $N_{\text{bg-true}} = 532038$.

maximize the probability of observing the data at hand for a set of parameters. The likelihood of a set of n i.i.d. random variables can with our definition of the Gaussian PDF in (2) be expressed

$$L = \prod_{i=1}^{n} p(I_i|\mu, \sigma), \tag{6}$$

where L is the likelihood function and $p(I|\mu, \sigma)$ is the probability of observing I given the parameters μ and σ [11]. In general, it is common to, instead of maximizing the likelihood function, maximize the log-likelihood function. This is the same since the natural logarithm is a continuously strictly increasing function. The procedure of minimizing a function with respect to two parameters simultaneously can be rather challenging. To cope with this an inbuilt function (`mle()` in Matlab) has been used, which is an iterative maximization algorithm.

The estimate of the parameters also provides us with an estimated variance-covariance matrix, describing the error in μ and σ as well as their correlation. The variance-covariance matrix for the estimated parameters is [12]

$$\boldsymbol{C} = \begin{bmatrix} \Gamma_{\mu\mu} & \Gamma_{\mu\sigma} \\ \Gamma_{\mu\sigma} & \Gamma_{\sigma\sigma} \end{bmatrix}, \tag{7}$$

for two correlated parameters. Above, $\Gamma_{\mu\mu}$ gives the estimated variance in μ and $\Gamma_{\sigma\sigma}$ gives the estimated variance in σ, while the off-diagonal terms give the covariance of the two parameters.

The matrix in (7) can be used to estimate the propagation of error for expressions containing the parameters μ and σ.

At this point we have laid out methods to fit a truncated Gaussian, and extract its mean μ and standard deviation σ. So, let us move on and try to deal with the problem of automatically estimating $I_{\text{cut-off}}$ as needed for step 1a in section 3.1.

We so far assumed $I_{\text{cut-off}}$ to be known but presented no way to obtain this value. Let us now try to change that and here present a way to automatically find an optimal value, minimizing the number of signal pixels while maximizing the number of background pixels included in the truncated Gaussian fit.

Consider the sorted intensity vector $\vec{I}_{\text{sorted}}$ containing N elements where $N = N_x N_y$. Split the components of the vector $\vec{I}_{\text{sorted}}$ into s equally sized bins, labeled by $j = 1, ..., s$ (in this work $s = 10$), such that there exist s bins with N/s elements in each bin; if $N \mod s \neq 0$ they have to be distributed among the bins as equally as possible. For each bin j we denote the lower index limit in I_{sorted} with k_j, which gives an upper index limit $k_{j+1} - 1$. Let us further associate the index k_j with a corresponding intensity value I_{k_j} in $\vec{I}_{\text{sorted}}$ by picking the intensity of the element k_j in $\vec{I}_{\text{sorted}}$. The intensity value corresponding to element k_{s+1}, namely $I_{k_{s+1}}$, is here taken to be intensity of the $k_{s+1} - 1$ element in I_{sorted}. The true number of pixels in each bin j is now known to be $n(j) = k_{j+1} - k_j$. To compute the expected number of elements in each bin, we use the extracted full Gaussian PDF $h(I; \mu, \sigma)$, given in (11), and it is now just a matter of performing s integrals from I_{k_j} to $I_{k_{j+1}}$; that is

$$\int_{I_{k_j}}^{I_{k_{j+1}}} h(I; \mu, \sigma)\, dI, \tag{8}$$

which will give s new values representing the expected numbers of background pixels between the intensities I_{k_j} and $I_{k_{j+1}}$.

We now compare the expected number from the background distribution with the true number given by $n(j)$, and denote the deviation as

$$\varepsilon_j = n(j) - \int_{I_{k_j}}^{I_{k_{j+1}}} h(I; \mu, \sigma)\, dI. \tag{9}$$

To evaluate whether the value of ε_j is within what would be an acceptable magnitude of error, we assume that the events filling bin j is independent of each other such that the number of events in bin j is binomially distributed. The probability of a random event to land in bin j can be estimated by $P_j = \frac{n(j)}{N}$. The variance of a binomial process with the total number of events N and associated probabilities P_j, is equal to [13]

$$\sigma_j^2 = N P_j (1 - P_j). \tag{10}$$

To determine whether the number of events in bin j reasonably could come from the Gaussian background distribution $h(I; \mu, \sigma)$, we check if $\varepsilon_j \leq 3\sigma_j$ is satisfied. This condition thus serves as goodness of fit criterion.

In order to find the optimal $I_{\text{cut-off}}$ we approach the problem with a little bit of iteration. Now let for simplicity $I_{\text{cut-off}}$ be the intensity value corresponding to the intensity for k_2, i.e., $I_{\text{cut-off}} = I_{k_2}$. From the fitted truncated Gaussian using this value of $I_{\text{cut-off}}$ it is possible to find the lowest value of j that satisfies $\varepsilon_j \geq 3\sigma_j$, which we here denote by m. Let us now remember this value of m and move on fitting a new truncated Gaussian using $I_{\text{cut-off}} = I_{k_3}$; which will give a new value of m that satisfies $\varepsilon_j \geq 3\sigma_j$. In the end after repeatedly fitting truncated Gaussians using $I_{\text{cut-off}} = I_{k_j}$, where $j = 2, ..., s$, there are s number of m values, $\vec{m} = (m_1, m_2, ..., m_s)$. We now choose $I_{\text{cut-off}}$ to be the intensity value matching the highest number in $\vec{m}$ that appears after a series of non-decreasing numbers of m. We have at this point arrived at the optimal value of $I_{\text{cut-off}}$, which gives us the corresponding optimal values of μ and σ.

An example to illustrate this would be that we have a set of m-values as following: $\vec{m} = (3, 4, 4, 5, 6, 7, 5, 4)$. Since the values in $\vec{m}$ strictly increase up until the 6:th value where $m = 7$, we simply pick $I_{\text{cut-off}}$ to be 6:th intensity value that was used in order to fit the truncated Gaussian.

Let us here try to understand the logics of how the method of arriving at the optimal value of $I_{\text{cut-off}}$ works. In the cases where $I_{\text{cut-off}}$ is increased and the fit just includes more background pixels, we expected that the fit will yield a better estimate of the true background distribution and its parameters. So, the number of bins that fall within the tolerance is increasing, i.e., m is strictly increasing with an increased precision of the fit. On the other hand, as soon as the fit starts to include signal pixels, which does not give the optimal Gaussian distribution, the value m should decrease. In this manner one hopefully should be able to tell at which intensity value the fit seemed to be the best.

Step 1b – Estimating the number of background pixels

Let us now estimate the number of background pixels in the image using the extracted parameters μ and σ. The functional form of the PDF describing the full background distribution can, with the help of the number of background pixels N_{bg} and (1), be expressed as

$$h(I;\mu,\sigma) = N_{\text{bg}} g(I;\mu,\sigma). \tag{11}$$

Let us here integrate both sides of the equation from $-\infty$ to $I_{\text{cut-off}}$, and solve for N_{bg} to obtain

$$N_{\text{bg}} = A \int_{-\infty}^{I_{\text{cut-off}}} h(I;\mu,\sigma)\, dI, \tag{12}$$

where A is earlier defined in (4). We here recognize that the integral in (12) is nothing else than the number of elements in $\vec{I}_{\text{sorted}}$ up to the intensity $I_{\text{cut-off}}$, which we can denote by $n_{\text{cut-off}}$ to obtain a final expression for the number of background pixels as following

$$N_{\text{bg}} = A n_{\text{cut-off}}. \tag{13}$$

In appendix B we provide expressions for the variance of the estimated number of background pixels with help of the variance-covariance matrix in (7), such that the estimated number of background pixels can be written as

$$N_{\text{bg}} \pm \sigma_{N_{\text{bg}}}, \tag{14}$$

where $\sigma_{N_{\text{bg}}}$ is the standard deviation of N_{bg}.

At this point we have managed to overcome the challenges in step 1 (see section 3.1). We know how to fit a truncated Gaussian using the maximum likelihood estimate, extracting μ and σ. Furthermore, we have a strategy for automatically estimating $I_{\text{cut-off}}$ and (14) tells us how many background pixels we expect in the whole image.

An example of a final fit to the histogram obtained from the image in Fig. 1b invoking all the steps and strategies brought up so far is given in Fig. 2. It can be seen in Fig. 2 that the algorithm, in this case, seems to perform rather well, and correctly estimates the number of background pixels as well as the parameters for the background distribution μ_{bg} and σ_{bg}. Furthermore, it seems that the algorithm for automatically estimating $I_{\text{cut-off}}$, at least visually, arrives at a reasonable value.

The grayscale image seen in Fig. 1b has after all the steps under 1 (see section 3.1) been divided into two categories, true background pixels, and mixed signal and background pixels, which are still to be classified. The resulting image obtained from performing all the steps under 1 can be seen in Fig. 3. We notice that the final output image of the truncated Gaussian procedure seen in Fig. 3 have gray areas that include most of the true signal pixels seen in Fig. 1a, which is crucial since all pixels classified as background at this stage will remain as background until the end. The next task is to classify all the gray pixels in Fig. 3 as either signal or background, which is addressed in the next section.

Step 2 – Bayesian statistics

In this section we concern ourselves with the challenges of how to implement the Bayesian decision theory framework [14] and its components to classify the pixels which are considered uncertain (gray) in the image seen in Fig. 3, as either signal or background.

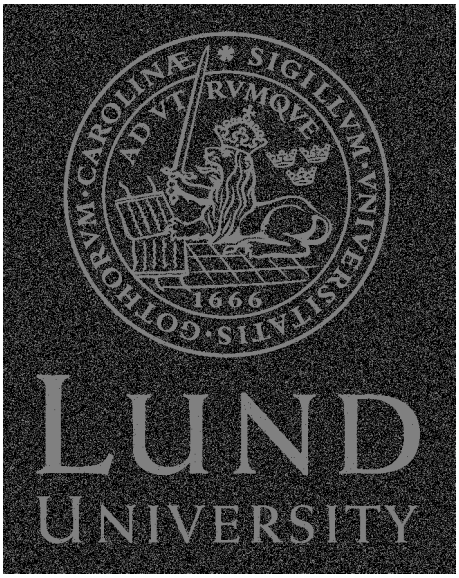


Figure 3: **Segmentation with $I_{\text{cut-off}}$.** The resulting image from the auto estimation of $I_{\text{cut-off}}$ and the truncated Gaussian fit. The black pixels represent the true background pixels, whereas the gray pixels constitute uncertain pixels in the upper mixed intensity region and are yet to be classified.

From now on it will be assumed that the steps found under 1 (section 3.1), leading up to this point were successfully performed, and the values of μ_{bg}, σ_{bg} and N_{bg} could be extracted.

Let us here, before moving on to step 2a found in section 3.1, introduce the Bayesian framework and how it is implemented for the specific task of classifying pixels. The idea behind the steps that follow in this section is to utilize Bayes' theorem to classify the gray pixels in Fig. 3 as either background or signal based on some conditional as well as prior probability. The conditional probability for the background pixels is obtained with the help of the pixels already classified as background, using them as training data. Furthermore, it is possible, with the training data, to obtain the conditional probability for the signal pixels by methods starting at page 11. The prior probability is obtained with help of the known Gaussian distribution of the background pixels. From the conditional and prior probability we calculate the posterior probability and classify each pixel according to a minimum risk criterion.

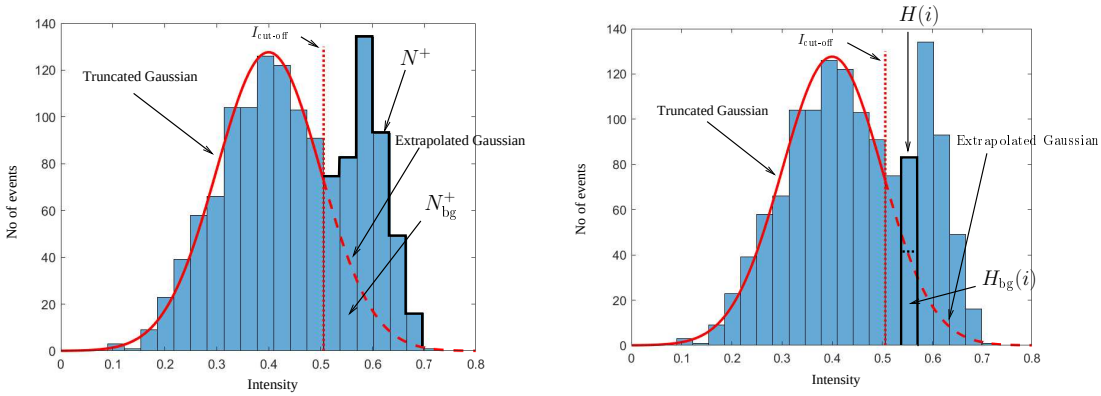
It is now time to formalize the framework to be adopted. We start by assigning the signal pixels with a state ω_1 and the background pixels with a state ω_2. Let us further assume that the conditional probability, which depends on some feature f for both the signal and background pixels $p(f|\omega_i)$, is known, where $i \in [1, 2]$. The feature value f is in this study taken to be a typical intensity value for a local neighborhood for the pixel of interest, which for example could be a median or average value. Our specific choice of feature value, which may very well not be the optimal one, will be dealt with in more detail starting from page 10. Furthermore, assume that the the prior probability of observing a signal or a background pixel is known to be $P(\omega_i)$, where $i \in [1, 2]$. From this it is possible to write the posterior probability of observing either a signal or background pixel given a feature value f as

$$P(\omega_i|f) = \frac{p(f|\omega_i)P(\omega_i)}{p(f)}, \tag{15}$$

where the normalization factor is $p(f) = \sum_i p(f|\omega_i)P(\omega_i)$ [14]. Equation (15) is usually referred to as Bayes' theorem.

To classify a pixel such that the probability of error is minimized, given a feature value f, is just a matter of finding the largest of $P(\omega_1|f)$ and $P(\omega_2|f)$. This yields the classification rule as follows:

$$\begin{cases} P(\omega_1|f) \geq P(\omega_2|f) & \Rightarrow \quad \text{signal pixel} \\ P(\omega_2|f) > P(\omega_1|f) & \Rightarrow \quad \text{background pixel} \end{cases} \tag{16}$$



(a) Intensity-independent prior

(b) Intensity-dependent prior

Figure 4: **Illustration of prior probability. (a) Intensity-independent prior**. In the figure an illustration of the two important ideas is shown to obtain an intensity-independent prior probability. The number of background pixels above the intensity cut-off value can be divided into two groups. The first group is the number of background pixels N^+_{bg}, corresponding to the area under the extrapolated Gaussian fit. The second group is all pixels N^+ above $I_{\text{cut-off}}$. The intensity-independent background prior probability estimate is given by the ratio of N^+_{bg} and N^+. The prior probability for the signal pixels follows from the law of total probability (18). The histogram, the corresponding truncated Gaussian, the extrapolated Gaussian, and the intensity cut-off value are based on synthetic data and merely intended to act in an illustrative purpose. **(b) Intensity-dependent prior**. The figure intends to illustrate, as opposed to Fig. 4a, a visual representation of how an intensity-dependent prior probability for the background pixels is obtained, by finding the number of background pixels $H_{\text{bg}}(i)$ in each bin and dividing this by the total number of pixels $H_{\text{all}}(i)$ in that bin. H_{bg} is here a histogram of only background pixels and H is the total histogram. By doing this for all bins higher than $I_{\text{cut-off}}$ in intensity a more precise intensity-dependent prior for the background pixels is obtained. In the same manner as for the intensity-independent prior probability, the law of total probability (18) gives the prior probability for the signal pixels, once the prior probability for the background pixels is determined. As in Fig. 4a, the data shown in the histogram is purely synthetic.

We have at this point established what the classification procedure is all about, and it is now a matter of finding what is needed to be able to compute the posterior probability in (15).

Step 2a – the prior probability

In this section we intend to find the prior probability $P(\omega_i)$ for both the signal and background pixels. In simple terms that is only a matter of achieving an estimate of the ratio of background pixels compared to all pixels in some intensity span (see Fig. 4).

Intensity-independent prior probability

Let us here deal with the task of finding the probability $P(\omega_i)$ along the lines of what was illustrated in Fig. 4a, by calculating the fraction of background pixels found among all pixels above the intensity $I_{\text{cut-off}}$. All the pixels up to $I_{\text{cut-off}}$ are already assumed to be background, so there is no need to consider them in the calculations for the moment. Let us recall from earlier that the full functional form for the background distribution could be derived from the fitting procedure and can be expressed as $h(I; \mu_{\text{bg}}, \sigma_{\text{bg}})$, where we denote the mean and standard deviation, respectively, obtained from the fit with μ_{bg} and σ_{bg}. Starting by calculating the number of background pixels N^+_{bg} above $I_{\text{cut-off}}$, it is found that

$$N^+_{\text{bg}} = \int_{I_{\text{cut-off}}}^{\infty} h(I; \mu_{\text{bg}}, \sigma_{\text{bg}})\, dI. \tag{17}$$

The total number of pixels above $I_{\text{cut-off}}$ can be estimated as $N^+ = N - n_{\text{cut-off}}$, where $n_{\text{cut-off}}$ is the corresponding index to $I_{\text{cut-off}}$ in $\vec{I}_{\text{sorted}}$. From this the prior probability for the background pixels in the non-classified regime can be written as $P(\omega_2) = N^+_{\text{bg}}/N^+$, and from the law of total probability

$$P(\omega_1) + P(\omega_2) = 1, \tag{18}$$

it follows that $P(\omega_1) = 1 - P(\omega_2)$.

Intensity-dependent prior probability

In the earlier estimate of the prior probability, only a single value of the ratio between the background and all pixels was considered. To provide a more precise estimate of $P(\omega_i)$ than the one obtained in the case for the intensity-independent prior above, we will here utilize the information extracted from the truncated Gaussian to a higher degree as visually presented in Fig. 4b.

So, to make a better estimate of the prior probability $P(\omega_2)$, we compute the ratio of background pixels $H_{\text{bg}}(i)$ and all pixels $H(i)$ in each bin above $I_{\text{cut-off}}$ (see Fig. 4b), where H_{bg} is the histogram containg all background pixels and H is the histogram containg all pixels, which is

$$P(\omega_2; i) = \frac{H_{\text{bg}}(i)}{H(i)}. \tag{19}$$

In (19) i only takes on values according to the bin indices for all bins above $I_{\text{cut-off}}$. The prior probability for the signal pixels $P(\omega_1; i)$ follows from the law of total probability (18). Furthermore, we have assumed that all pixels below $I_{\text{cut-off}}$ are background, so the prior probability for signal pixels $P(\omega_1; i) = 0$ for all bins lower than $I_{\text{cut-off}}$. So, given a pixel with intensity I it is now just a matter of finding the corresponding bin i in H which I falls within, and use (19) to obtain the prior probability. We use Scott's rule [15] to determine the bin width in H, the bin width in H_{bg} is the same as in H. Further discussion about the choice of bin width can be found in section 5.2.2. The bin width for all histograms in this work are obtained using Scott's rule when nothing else has been specified.

In principle we now just need the histogram H_{bg} which contains only background pixels, in order to compute a prior probability that depends on the intensity. In appendix C.1 we present a method to obtain H_{bg} in a statistically feasible way, with the restriction $H_{\text{bg}}(i) \leq H(i)$, to satisfy $P(\omega_2; i) \leq 1$.

So, using the method described in appendix C.1 to obtain H_{bg} we now have two methods to obtain the prior probability, one which is dependent on the intensity and another which is not. The intensity-dependent prior probability will have an intensity resolution equal to the bin-width in H, meaning it is not possible to obtain a different prior probability for two pixels with an intensity which matches the same bin.

Step 2b – the conditional probability

In this section we seek to find expressions for $p(f|\omega_i)$, see (15), the conditional probability density functions for the signal and background pixels as a function of the feature value f, using the true background pixels as training data.

To that end, let us now properly introduce the feature value f used in this section. As mentioned briefly at page 8, the feature value intends to categorize the intensity values of the neighborhood to a pixel of interest. In this study the feature value is obtained by finding the median value of the neighborhood of the pixel of interest, excluding the intensity value of the pixel to categorize. The procedure of obtaining f is illustrated in Fig. 5 and is inspired by the original median filtering technique [1].

As seen in Fig. 5, the procedure of obtaining f is rather trivial, though in some cases time consuming. Problem arises when pixels with a neighborhood stretching beyond the boundaries of the image shall be computed. In most cases when the window size is sufficiently small this is not a big issue, but to obtain reasonable values at the affected boundary pixels we here took the boundaries to be mirrored outside the actual image, to such extent that all pixels had a neighborhood completely filled with intensity values. The boundary at which the mirroring takes place is chosen such that the edge values of the image sees itself just outside the image. The reason why the pixel of interest is excluded when computing the modified median value builds on the fact that, for our new method, we require f to be independent of I. Why this is the case is discussed and motivated in more depth in appendix D.

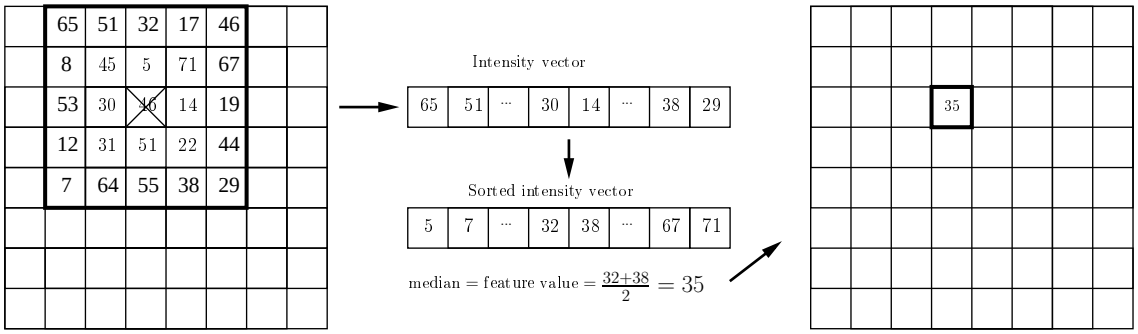


Figure 5: **Modified median filter.** The figure aims to schematically explain the modified median filter used to obtain the feature value f. The procedure of performing the modified median filtering technique is rather simple and has many similarities with the original median filtering technique. First the neighborhood of the pixels of interest is extracted to a vector containing all the intensity values. The next step is to sort all the intensities and pick the median value. In this case the median does not exist, so we simply replace this value by the average of the neighboring values to this missing median value. This modified median value of the neighboring intensity values for the pixel of interest is our choice of feature f, used for Bayesian pixel classification.

Conditional probability – background pixels

With the tool to extract a feature value, it is now possible to move on in the procedure of estimating $p(f|\omega_i)$. Let us here start with $p(f|\omega_2)$, constituting the probability density function for the background pixels in feature space.

We first of all have to perform the procedure of finding the feature value f (see Fig. 5) for all pixels in the original image, seen in Fig. 1b, and create a new image containing all the obtained values for f in each pixel in $I(x, y)$. Since we can find the location of all true background pixels in the original image $I(x, y)$, by simply requiring $I(x, y) \leq I_{\text{cut-off}}$, we can locate the position of the true background pixels as $(\vec{x}_{\text{bg}}, \vec{y}_{\text{bg}})$. In principle a normalized histogram of the feature values of the true background pixels – let us denote it with F_{bg}^{-} – gives a probability density function $p(f|\omega_2)$ for the background pixels in the feature space. This holds if the distribution of f for the true background pixels is representing the distribution of all background pixels in f-space, which is the case if the feature value in each pixel is independent of the intensity value in the corresponding pixel.

Applying the modified median filter to the image seen in Fig. 1b and obtaining the feature values for the true background pixels (black pixels in the image seen in Fig. 3) gives the histogram F_{bg}^{-} seen in Fig. 6. Furthermore, we also obtain the feature value histogram for all pixels, let us denote it by F, which is also shown in Fig. 6.

Conditional probability – signal pixels

The problem now arises whether it is possible to extract the corresponding probability density function $p(f|\omega_1)$ for the signal pixels, since no training data exists. In the same way as the background intensity histogram could successfully be estimated dealing with the prior probability (see page 10), it is now possible, with some modifications, to replicate that procedure to obtain the feature histogram for both the signal and background pixels. Notice that the functional form of the distribution in f for the background pixels is not known and most likely will deviate from the original Gaussian distribution. A method which despite this problem allows us to obtain the full feature background histogram F_{bg} from the reduced histogram at hand, F_{bg}^{-}, is presented in appendix C.2. This approach builds on the assumption that F_{bg}^{-} is representative in terms of its distribution for the full background histogram F_{bg}. The resulting values of F_{bg} for different values of f, obtained from the histograms F_{bg}^{-} and F in Fig. 6, represented as a dashed line, is given in Fig 6 using the method described in appendix C.2.

So with the full feature background histogram F_{bg}, obtained with the constraint that $F_{\text{bg}}(i) \leq F(i)$, it is now straight-forward to obtain the signal feature histogram F_{sig} from F, by $F_{\text{sig}}(i) = F(i) - F_{\text{bg}}(i)$, where i runs over all bin indices in F; meaning that each bin value in F_{bg} is subtracted from the corresponding bin value in F. Normalizing F_{sig} such that the area of the histogram

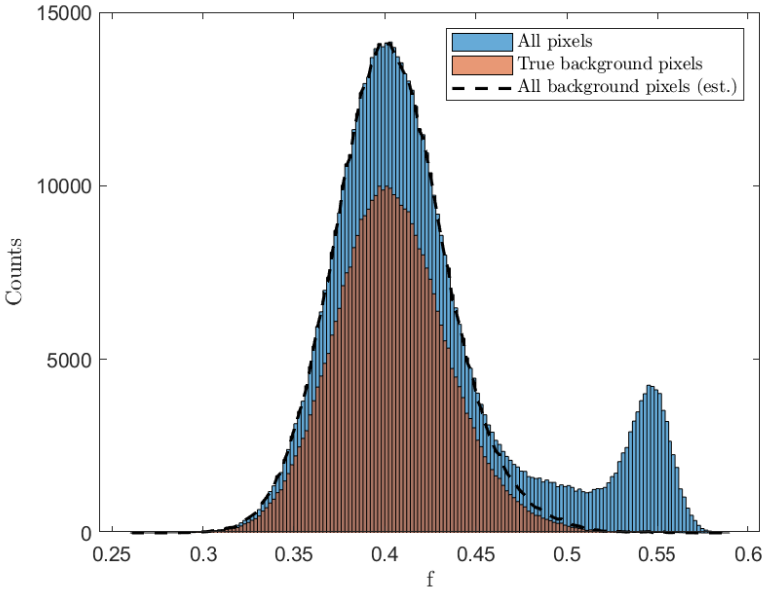


Figure 6: **Feature value histogram for training (true background) pixels.** In the figure the histogram for all feature values (F) obtained by performing the procedure illustrated in Fig. 5, as well as the histogram for all the feature values for the true background pixels (F_{bg}^{-}) is shown. Since only a fraction of all background pixels is considered to be true background pixels after step 1a is completed (see page 6), the number counts in the background histogram is lower or equal in all bins than in the histogram for all pixels. The distribution of the background histogram though matches the lower part of the total histogram rather well, mimicking the shape to a large extent for low values of f. It is therefore reasonable to assume that the true background pixels can be used as training data to obtain the full background distribution, and with this extract the signal feature value histogram. The dashed line indicates the estimated values for the full background histogram (F_{bg}), obtained with the method presented in appendix C.2, and satisfies the constraint of not exceeding the bin values for the histogram of all pixels.

equals to 1, we obtain the conditional probability density function for the signal pixels $p(f|\omega_1)$. The resulting conditional probability density functions $p(f|\omega_1)$ and $p(f|\omega_2)$ are shown in Fig. 7, obtained based on the two histograms seen in Fig. 6 and the knowledge about the total number of background pixels obtained from the procedures presented on page 7.

The probability density functions shown in Fig. 7 seems to verify the functionality of the proposed algorithm to a high extent; the difference between the optimal results and the actual ones is for many values of f negligibly small.

Step 2c – the posterior probability and final decision making

With the work done so far it is now possible, given a feature value f and an intensity value I, to calculate the posterior probabilities for signal and background according to (15). We now calculate the posterior probabilities for all the uncertain (gray) pixels in Fig. 3. With these posterior probabilities we use the decision rule stated in (16) to classify all uncertain pixels as either signal or background. Since we classify each pixel according to the state represented by $\max(P(\omega_1|f), P(\omega_2|f))$, there is a probability of classifying each pixel wrong which equals $\min(P(\omega_1|f), P(\omega_1|f))$. On the other hand, this also allows for estimation of the error in each decision, as well as estimation of the error for several decisions in a row. So for a set of decisions $j \in [1, n]$ it is possible to estimate the number of wrong decisions E as

$$E = \sum_{j}^{n} \min(P_j(\omega_1|f), P_j(\omega_2|f)). \tag{20}$$

This gives us an estimated number of wrongly classified pixels, which for simplicity can be expressed as the fraction of correctly classified pixels out of all pixels

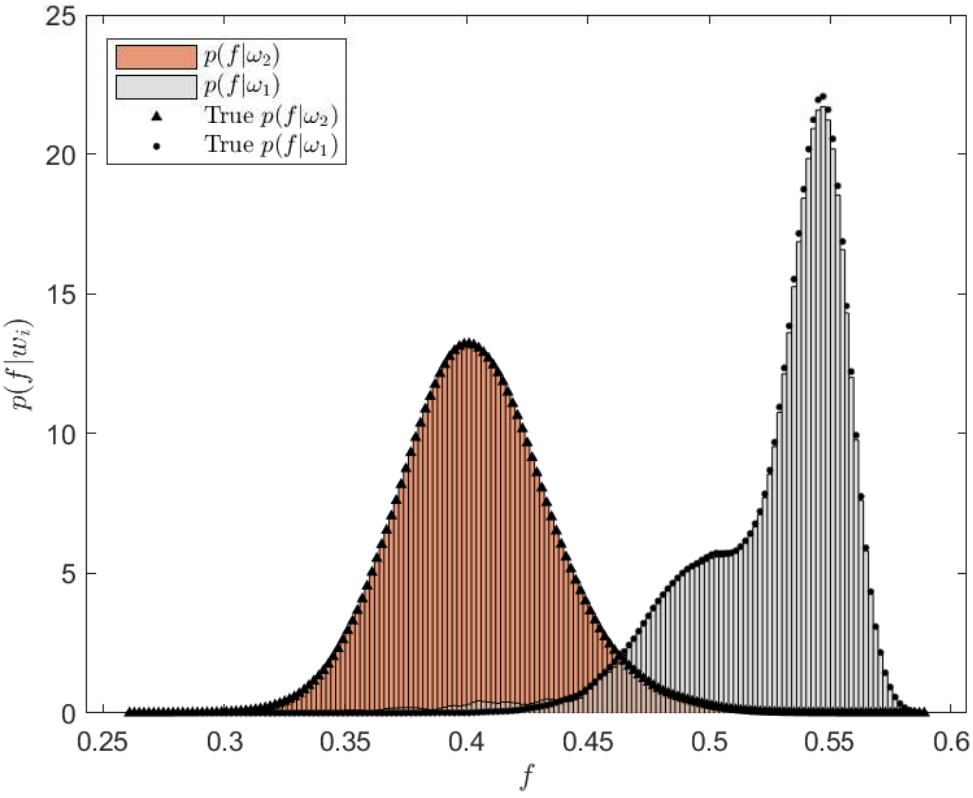


Figure 7: **Conditional probability density functions.** In the figure the conditional probability density functions $p(f|\omega_i)$ are shown. These are obtained using the procedures described at page 10 and 11, as well as the information from the fit shown in Fig. 2. For comparison, the conditional probabilities calculated using all the background and signal pixels are shown with triangles and dots representing the true value for each value of f. It is clear that the obtained forms of $p(f|\omega_i)$ match quite well with the true values. Furthermore, it can be seen that the peaks for background and signal are noticeably much more separated than in the intensity histogram seen in Fig. 2, where they overlap completely. The separation of the peaks indicates that a good feature value has been chosen.

$$\eta_{\text{est}} = 1 - \frac{E}{N}. \tag{21}$$

For the case when considering only the intensity-independent prior probability (see page 9), the posterior probability computed from (15) reduces to a function of only one variable and can therefore be easily illustrated, as seen in Fig. 8.

For the case when considering the intensity-dependent prior probability it is not possible to make the same simple plot as seen in Fig. 8 illustrating the posterior probabilities, since there is an additional intensity dependency added, requiring three dimensions to illustrate the complete probabilities. It can, however, be rather illustrative to plot all the pixels classified as signal respective background with their feature value f versus the intensity value I to obtain how they are distributed in feature-intensity (f-I) space as seen in section 4.1.

At this point we can consider ourselves to be done with the core methods of the new algorithm, after completing all the steps found in section 3.1. The binary versions of the image seen in Fig. 1b, segmented using the new algorithm, are shown in section 4.

4 Results

We will here present results which to a large extent is based on comparing the new algorithm with the Otsu method. First, we consider the special case used throughout the report with the

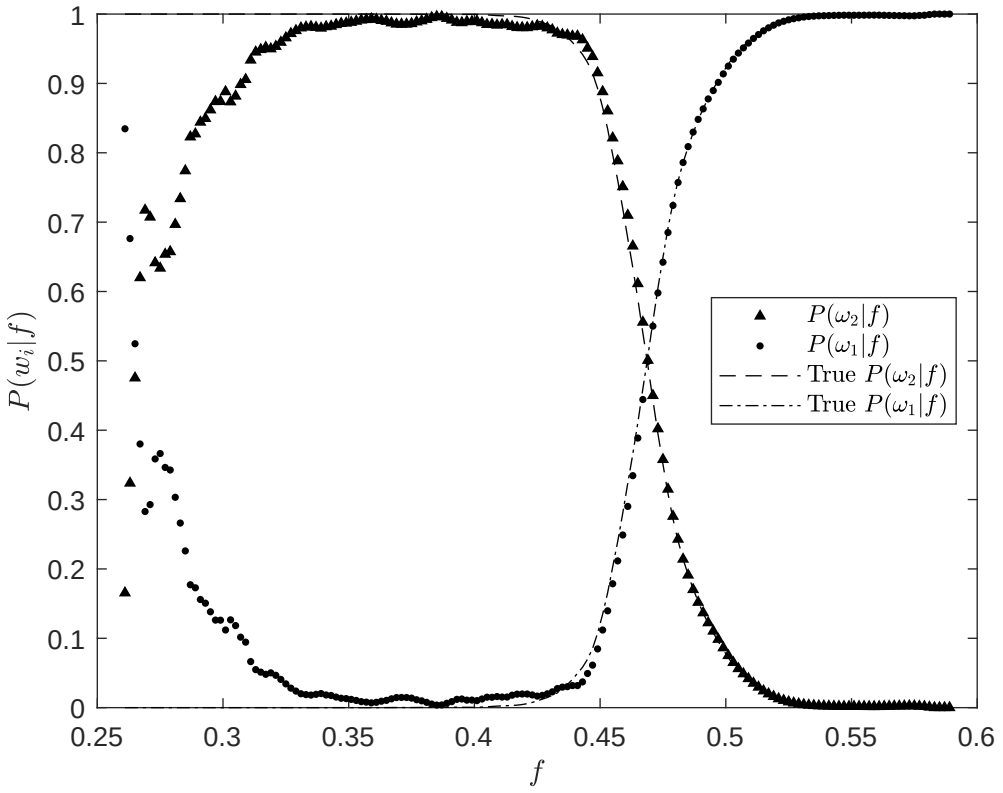


Figure 8: **Posterior probabilities.** In the graph the posterior probability as a function of the feature value f for the intensity-independent prior probability is plotted for both background and signal probabilities; that is simply $P(\omega_1|f)$ and $P(\omega_2|f)$. The dots and triangles indicate the signal and background posterior probability, respectively. The markers are accompanied by dashed lines indicating the posterior probabilities that would have been obtained if all signal and background pixels had been used in the creation of the conditional probability density functions $p(f|\omega_i)$. As can be seen, the true values for $P(\omega_i|f)$ do follow the obtained values rather good for high values of f; on the contrary, the deviation from the true values increases for low values of f. An important detail here is the accuracy of the intersection point for $P(\omega_1|f)$ and $P(\omega_2|f)$, since it strongly affects the final classification. Furthermore, we can here visually understand the meaning of the decision rule in (16) – if $P(\omega_1|f) \geq P(\omega_2|f)$, choose ω_1; otherwise choose ω_2.

image found in Fig. 1b as an example. Furthermore, we consider images with different difficulties (see section 2), which will be segmented both for the case when the signal intensity distribution is Gaussian and uniformly distributed. We use our new algorithm and the Otsu method, and compare the results obtained to each other. Finally, we will concern ourselves with computational time.

4.1 A first case study

As a first example, we here consider the image in Fig. 1b and present the segmented images obtained by the new algorithm as well as the Otsu method. Let us to that end introduce the symbols for the true fraction of correctly classified pixels η_{na} (na = new algorithm) that will be used in the sections to come. The resulting binary images obtained from the image seen in Fig. 3 using the new algorithm can be seen in Fig. 9a for the case of intensity-independent prior probability and in Fig. 9b for the case of intensity-dependent prior. The corresponding binary image obtained using the Otsu method can be seen in Fig. 9c.

One of the things that can be noticed about the segmented images in Fig. 9a and 9b, is the fact that they visually appear almost identical. Studying the actual number of correctly classified pixels and comparing this to the result obtained by the Otsu method, it becomes clear that the new algorithm performs significantly better than the Otsu method in this case. The binary image

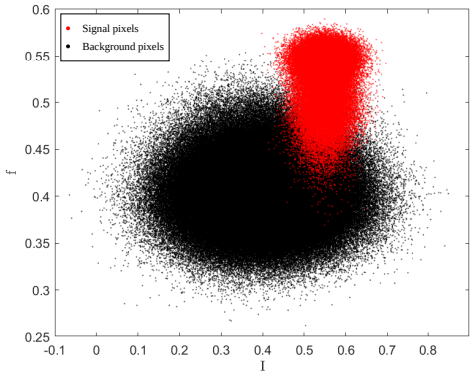
(a) Intensity-independent prior probability (b) Intensity-dependent prior probability (c) Otsu segmentation method

Figure 9: **Final binary images. (a):** The image shown is the result of the final classifications done using a fixed prior probability, i.e. using the posterior probabilities shown in Fig. 8. The fraction of correctly classified pixels compared to the original image seen in 1a is obtained to be $\eta_{\text{na}} = 0.9863$, which can be compared to the estimated fraction of correct pixels provided by the Bayesian decisions rule (see (16)) $\eta_{\text{est}} = 0.9855$. This can be considered as satisfactory both from the perspective of classifying as many pixels as possible correct, but also from estimating the error correctly. **(b):** Here the resulting image of the classification, done using the algorithm for an intensity-dependent prior together with the conditional probabilities in Fig. 7, is found. In the image the actual correct number of classified pixels is $\eta_{\text{na}} = 0.9883$, the estimated fraction of correctly classified pixels is $\eta_{\text{est}} = 0.9835$. As one can notice, the fraction of correctly classified pixels is slightly higher for the algorithm based on the intensity-dependent prior; on the other hand is the estimated fraction of correctly classified pixels closer to the true value for the algorithm using a fixed prior. **(c):** The resulting binary image obtained by segmenting the image in Fig. 1b using the Otsu method. The number of correctly classified pixels in the image is obtained to be $\eta_{\text{Otsu}} = 0.6393$, which is drastically lower than the result obtained from the new algorithm. The intensity cut-off value determined by the Otsu method was in this case found to be $I_{\text{cut-Otsu}} = 0.4196$. It appears that the segmentation managed to catch most of the true signal, while the number of signal pixels in total by far surpasses the correct number.

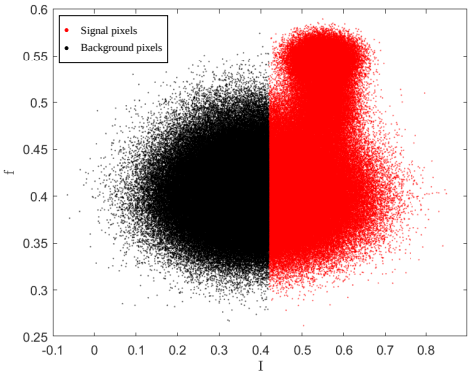
obtained by the Otsu method suffers to a large extent from pixels classified as signal when they in fact should have been background. It is though expected, looking at the intensity histogram in Fig. 2, that any method using a simple threshold in intensity would not do particularly good in this situation, due to the high degree of mixing among signal and background intensities.

We will here look closer on the obtained segmentation from the perspective of feature-intensity space, i.e. the results of the classification into signal and background pixels illustrated with f as a function of I. In Fig. 10a the perfect classification is plotted for the image seen in Fig. 1b, this is possible to give since we know the correct answer, given by the image in 1a. The result of segmenting the same image using the Otsu method can be seen in Fig. 10b. Furthermore, the obtained segmentation using the intensity-independent prior is shown in Fig. 10c and the obtained segmentation using the intensity-dependent prior is shown in Fig. 10d.

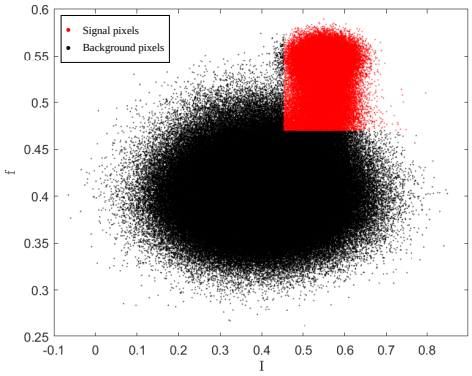
Looking at the plot in Fig. 10a we obtained that the signal and background pixels are highly mixed for certain values of f and I. After all, we see that inferring the feature value makes it easier, at least with higher certainty, to classify pixels as either signal or background compared to only using the intensity value. In Fig. 10b where the segmentation using the Otsu method is shown, we obtain that the result is far from perfect, and the problem with restricting the decision to just including the intensity value does not yield a satisfactory result. On the other hand, it becomes clear how the new algorithm, both with the intensity-independent prior probability (Fig. 10c), and the intensity-dependent prior probability (Fig. 10d), does a better job classifying the pixels correctly. In particular, we see that the algorithm using the intensity-dependent prior probability achieve a classification that mimics the perfect one to a rather high extent.



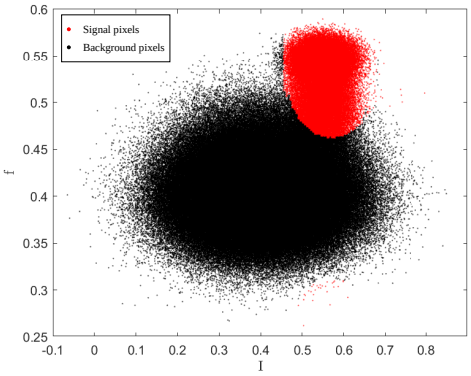
(a) Perfect classification



(b) Otsu classification



(c) Intensity-independent prior



(d) Intensity-dependent prior

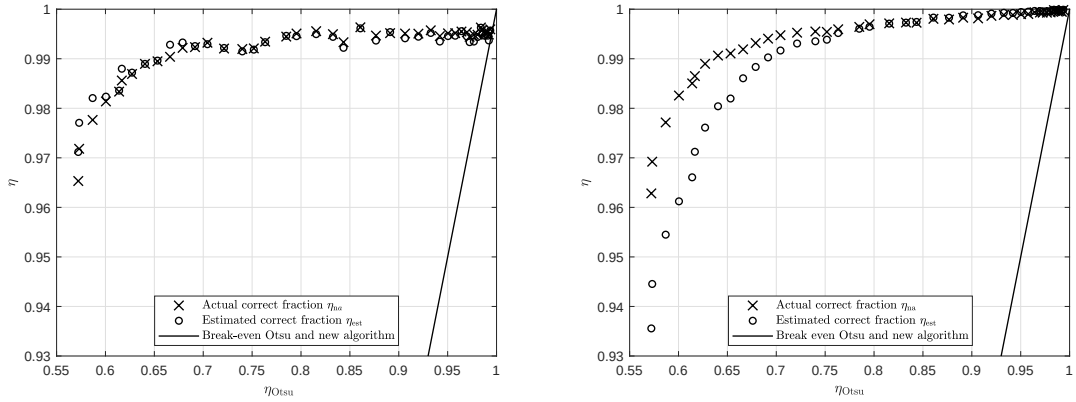
Figure 10: **Feature-intensity space plots. (a):** In the figure the feature value f is plotted for all pixels in the image seen in Fig. 1b against the corresponding intensity value I, where the true signal pixels are marked as red dots and the true background pixels are marked as black dots. The true signal and background pixels are known from the original image seen in Fig. 1a and reveal the challenges of segmenting this image. **(b):** In the figure the obtained segmentation using the Otsu method is shown, which reveals the problems that occur when one is forced to use only the intensity as a tool for classifying pixels. The intensity cut-off value of $I_{\text{Otsu-cut}} = 0.4196$ seems to include most of the signal pixels, but also classifies a lot of pixels that should have been background as signal. **(c):** In the figure, the classification obtained using the new algorithm with an intensity-independent prior probability is shown. The black dots are pixels classified as background and the red dots are pixels classified as signal. As expected the classified signal pixels now are determined by two straight lines, one along a certain intensity value ($I_{\text{cut-off}}$) and another one along the feature for which the probability for signal is higher than background (see Fig. 8). **(d):** Here is the achieved classification for the algorithm using the intensity-dependent prior probability shown, with pixels classified as background in black and signal in red. As opposed to the classification for the intensity-independent prior probability we here obtain that the region of pixels classified as signal does not follow any straight line and adopts, to a higher extent, the shape of the optimal classification seen in **(a)**.

4.2 Fraction of correctly classified pixels

We here perform further tests to evaluate the performance of the new algorithm over a span of images, a span in terms of the location and shape of the signal intensity distribution in the intensity histogram. The shapes of the intensity distributions considered here are Gaussian and uniform distributions. The following test aims to show how the new method handle the process of transforming a grayscale image to a binary image for various levels of difficulties, where the difficulty is taken to be the fraction of correctly classified pixels achieved by the Otsu method. The difficulty is here changed by changing the relative position of the signal and background intensity distribution with respect to each other; that is simply moving the signal peak further or closer away from the background peak seen in Fig. 2.

4.2.1 Fraction of correctly classified pixels – Gaussian signal

We here consider the result obtained by segmenting images with a Gaussian signal intensity distribution of varying difficulty. The images tested all had a background intensity distribution according to $\mu_{\text{bg}} = 0.4$ and $\sigma_{\text{bg}} = 0.1$. Furthermore, the variance of the signal distribution was kept fixed at $\sigma_{\text{sig}} = 0.035$, while the mean of the signal intensity distribution was varied between $\mu_{\text{sig}} = 0.48$ and $\mu_{\text{sig}} = 0.9$ in steps of 0.01 to change the difficulty. The result is measured in the fraction of correctly classified pixels η_{na} for the new algorithm and the corresponding estimated fraction of correctly classified pixels η_{est}. Final values of the results can be seen in Fig. 11a for the algorithm using an intensity-independent prior probability, and in Fig. 11b for the algorithm using an intensity-dependent prior probability. All the numerical values for the data points shown in Fig. 11 can be seen in Table 1 in appendix F.



(a) Intensity-independent prior probability

(b) Intensity-dependent prior probability

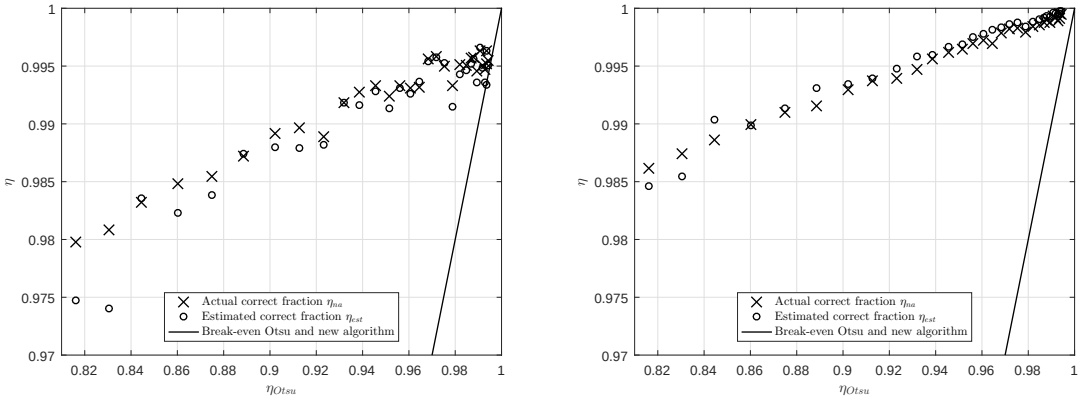
Figure 11: **Classification accuracy – Gaussian signal.** Plotted here is the fraction of correctly classified pixels for the new algorithm η_{na} and the estimated rate of correctly classified pixels provided from the new algorithm η_{est} as a function of the correct classifies pixels of the Otsu method η_{Otsu}, for both the intensity-independent prior probability algorithm in **(a)** and the intensity-dependent prior probability algorithm in **(b)**. A Gaussian signal intensity distribution has been used. To achieve different values of the classification rate, the mean of the signal distribution has been changed towards higher values starting from $\mu_{\text{sig}} = 0.48$ in steps of 0.01 until reaching $\mu_{\text{sig}} = 0.9$. The remaining parameters were kept fixed at $\mu_{\text{bg}} = 0.4$, $\sigma_{\text{bg}} = 0.1$ and $\sigma_{\text{sig}} = 0.035$. Each data point is computed from an average of 10 different values where all the input parameters for the signal and background distributions have been kept constant. Note that the overall rate of correctly classified pixels in both graphs appears to be much higher for the new algorithm compared to the Otsu methods. Furthermore, one can note that the fluctuations in η_{na} seem to be lower for the intensity-dependent algorithm than for the intensity-dependent algorithm. On the other hand, the intensity-independent algorithm tends to achieve a value of η_{est} closer to η_{na} for low values of η_{Otsu} compared to the intensity-dependent algorithm. The exact numerical results for all data points in both **(a)** and **(b)** and the corresponding input parameters for the intensity distributions can be seen in Table 1 in appendix F.

As seen in Fig. 11a and 11b, the value of η_{na} depending on η_{Otsu} remains more or less constant to a certain point, from where it drops significantly. On the other hand, this reduction of η_{na} appears over a total scale of a few percent, which can be considered rather small compared to the change in η_{Otsu}. When it comes to the estimated fraction of correctly classified pixels η_{est}, it can be seen that the correlation with η_{na} seems to be rather high, even though not completely perfect, and it gives good indications about the actual value of η_{na}. We finally conclude that both the algorithm using an intensity-independent as well as dependent prior probability performs much better, or on a level equal to, the Otsu method for a wide range of values for μ_{sig}, regarding the fraction of correctly classified pixels.

4.2.2 Fraction of correctly classified pixels – uniform signal

In relation to the results presented earlier in section 4.2.1 we will here consider the same type of test, but now for a uniform signal intensity distribution. The parameters for the background intensity distribution are, as before, given by $\mu_{\text{bg}} = 0.4$ and $\sigma_{\text{bg}} = 0.1$. Now however, the parameters for

the signal intensity distribution are given by the width of the uniform distribution, let us denote it with s_{sig}, and the mean μ_{sig}. In the test, the width of the signal intensity distribution has been kept constant at $s_{\mathrm{sig}} = 0.5$, while the mean μ_{sig} has been varied from an initial value of $\mu_{\mathrm{sig}} = 0.65$ to a final value of $\mu_{\mathrm{sig}} = 0.95$ in steps of 0.01. The result from the test using the new algorithm with an intensity-independent prior probability can be seen in Fig. 12a and with an intensity-dependent prior probability in Fig. 12b. Furthermore, are the numerical values for all obtained data points in Fig. 12 shown in Table 2 in appendix F.



(a) Intensity-independent prior probability

(b) Intensity-dependent prior probability

Figure 12: **Classification accuracy − uniform signal.** In the two figures the fraction of correctly classified pixels for the new algorithm η_{na} and the estimated fraction of correctly classified pixels resulting from the new algorithm η_{est} are plotted as a function of the fraction of correctly classified pixels of the Otsu method η_{Otsu}, for both the intensity-independent prior probability algorithm in **(a)** and the intensity-dependent prior probability algorithm in **(b)**, using a uniform signal intensity distribution. The starting point for the parameters are $\mu_{\mathrm{bg}} = 0.4$, $\sigma_{\mathrm{bg}} = 0.1$, $\mu_{\mathrm{sig}} = 0.65$ and $s_{\mathrm{sig}} = 0.5$, with s being the width of the uniform distribution. The mean of the signal has been changed in steps of 0.01 until reaching $\mu_{\mathrm{sig}} = 0.95$ to obtain various levels of difficulty for the classification. Each data point is computed from an average of 10 different values where the input parameters for the signal and background have been kept constant. The exact numerical results for all data points in both **(a)** and **(b)**, and the corresponding input parameters for the intensity distributions, can be seen in Table 2 in appendix F.

The result found in Figs. 12a and 12b indicates that the trend from the result seen in Fig. 11 is more or less consistent independently of the signal intensity distribution being Gaussian or uniform. In both plots seen in Fig. 12, a more linear increase of η_{na} for increasing values of η_{Otsu} is found than for the case of a Gaussian signal intensity distribution. When it comes to the values of η_{est} it now appears that they correspond better to the values of η_{na} considering the intensity-dependent prior probability, with a rather low difference in general. As seen before in Fig. 11, both the algorithm using the intensity-independent as well as dependent prior probability, in general performs better than or similar to the Otsu method.

4.3 Computational time scaling with number of pixels

The computational time (t) becomes critical if one wants to study a series of images, when the time scale can become significant in terms of convenience and productivity. Acceptable computational times and what is convenient will depend on the application. To verify the expected time scaling for the new algorithm, as discussed in section 5.1, the execution time, both for the new algorithm with intensity-independent and intensity-dependent prior, has been measured as a function of the number of pixels in the image, denoting the respective times with t_f and t_v. Furthermore, we will only consider the total execution time of the algorithm; which here is measured from the time the image is loaded into the program until the binary image is returned with the estimated error. Specific execution times for parts of the algorithm will not be considered, even though it can be of interest.

Different execution times require different number of pixels. To create a varying number of pixels, the image in 1b has been chopped up in 15 smaller, equally sized, images, from which bigger images gradually could be built. The starting point for the number of small images included in

the image to which the new algorithm later was applied to is 3. The choice is arbitrary, but a lower value might risk the absence of signal pixels which would lead to unpredictable behavior in the process of segmentation. Different conditions for each processed image can play a role here, partly in terms of the proportion of signal to background pixels, but also in terms of the computer performance. The result presented in Fig. 13 should therefore be viewed with the knowledge of unwanted effects that might have been present.

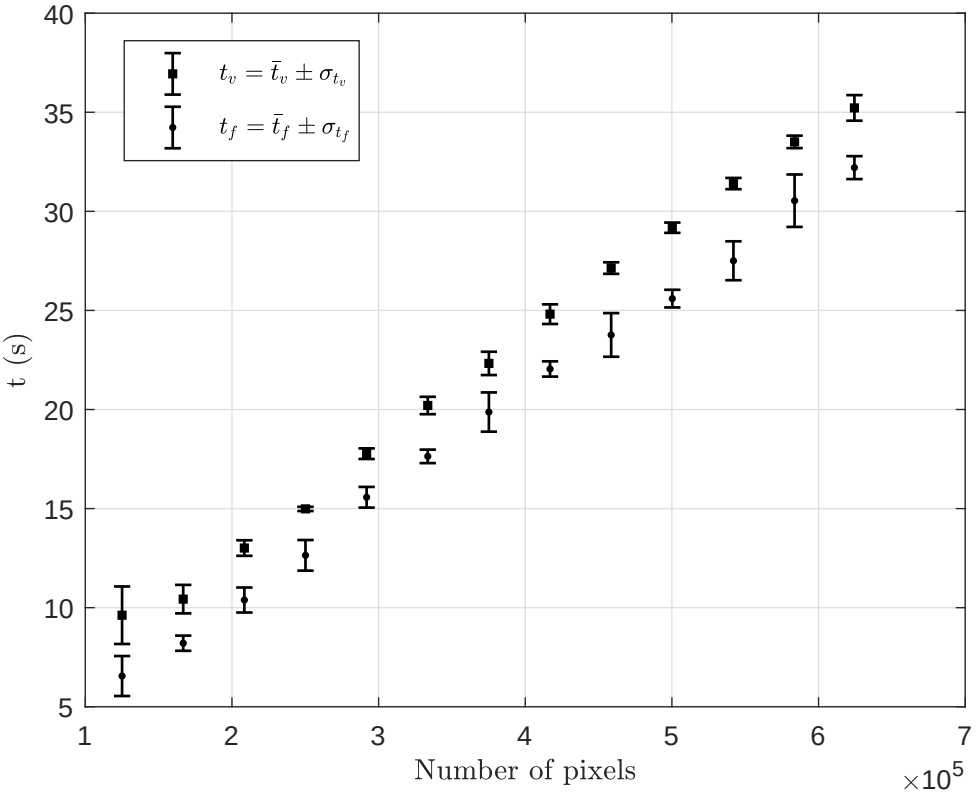


Figure 13: **Scaling of execution time with number of pixels.** The graph shows the execution time in seconds for the new algorithm as a function of the number of pixels. The image seen in Fig. 1b has been divided into smaller parts to which the complete algorithm, including all steps described to produce a final binary image, then has been applied. The times shown are the average total times for executing all necessary steps in the process of transforming a grayscale image into a binary image, including the error estimation over a set of 15 times for each image. The average time for the algorithm using an intensity-independent prior (t_f) is indicated with dots and the average time for the algorithm using an intensity-dependent prior (t_v) is indicated with squares. Times for plotting information of interest has been excluded in the total time. The algorithm is written in MATLAB and run on the R2018a version. Regarding the hardware: the test has been performed on a normal laptop (Lenovo) with a 2.2GHz Intel i7 processor and 8GB of RAM memory. It appears that the time scales roughly linearly, but from the short scale of pixel variation it is hard to determine.

From the result seen in Fig. 13 it becomes clear what time scales one should expect the algorithm to complete in. The number of pixels might very well for some images exceed the maximum number of pixels used in this test, which of course would result in longer execution times. The increase in execution time seems to be rather linear with increasing pixel number. From the discussion in section 4.3 the expected scaling is somewhat uncertain depending on what process might be dominant. The hope of exactly predicting the scaling is beyond the scope of this work, so for now we will settle with the fact that it appears to be acceptable in terms of the times presented in Fig. 5.1, as well as the fact that the time seems to scale approximately linear with the number of pixels.

5 Discussion

Let us first note that from the results presented in section 4, the new algorithm appears to outperform the Otsu method for almost all images that have been tested. This is promising in the sense that the new algorithm holds some benefits against the Otsu method for the gap it aims to fill, namely to achieve a higher fraction of correctly classified pixels as well as providing a prior estimate for the fraction of correctly classified pixels. Nevertheless, the tests that have been done merely show a fraction of all possible combinations that could appear in terms of different input parameters for the signal and background intensity distributions, which should be taken into concern.

We can further conclude that the benefits of including an intensity-dependent prior probability, compared to using an intensity-independent prior probability, for the images tested here, does not yield a significantly higher fraction of correctly classified pixels. On the other hand, it can be seen in the feature-intensity space plots shown in Fig. 10 that the classification using the intensity-dependent prior probability gives a more dynamic separation of signal and background pixels compared to both the algorithm using an intensity-independent prior as well as the Otsu method.

Let us now move on and discuss some more specific concerns of the new algorithm in the sections that follow.

5.1 Computational cost

We will in this section try to investigate how fast and efficient the new algorithm is and potentially can become. Let us start from the beginning with the truncated Gaussian fit and sorting of intensity values. The maximum likelihood fit requires a sorted vector of data, to deliver this the image $I(x, y)$ has to be converted to a one dimensional array which further has to be sorted. The transformation from a 2-d to 1-d array is a simple iteration through every element in the image and should scale linearly with an increasing number of pixels. The sorting on the other hand does not scale linear with an increasing number of elements needed to be sorted, the sorting algorithm used in this work is a so called quick sort algorithm [16] and is found to scale as $O(N \log N)$.

Regarding the computational cost for the following major step, namely the truncated Gaussian fit, things get more complex. The maximum likelihood fit performs a minimization of a function, which in this case depends on two variables μ_{bg} and σ_{bg}, and due to this the rate of convergence is hard to predict in general. What can influence the number of iterations needed to reach the minimum is the accuracy of the initial guesses, i.e. starting points for μ_{bg} and σ_{bg}. Related to this comes the auto estimation of $I_{\text{cut-off}}$ which in practice just consists of fitting a certain number of truncated Gaussians.

As for the next step when considering the extraction of the intensity-dependent prior (for the case when this is relevant), and the conditional probabilities, we expect a certain amount of the time to be consumed by the process of generating new histograms according to a given distribution, as presented in appendix C. This problem has been studied earlier [17], not in exact detail perhaps, but to a high enough degree, and it seems that the scaling for this method approximately is $O(N \log(N))$, which can be considered acceptable. On the other hand, this does not tell us anything about the magnitude of the pre-factor setting the overall scale. Some preliminary tests indicate a time scale of a few seconds (no longer than 10 seconds) to generate about 15 histograms of each $\sim 10^5$ pixels.

For the last step in the algorithm we consider the computation of posterior probabilities to generate the final binary image. To calculate the posterior probabilities given a feature value f and an intensity value I, it is necessary to check the difference between the current values and each of the central bin values in F_{bg}^{-}, F_{sig} and H, in order to decide which bins the current values best match. This procedure to calculate the posterior probability scales linearly with increasing number of pixels, assuming the number of bins does not drastically change due to an increase in the number of pixels.

We can conclude that the scaling of needed iterations for an increasing number of pixels seems to increase linearly is some cases, as $O(N \log(N))$ for other cases, and in a more or less unknown way for the maximum likelihood procedure.

In some applications the time it takes to process an image, and convert it to a binary one, might be too long and in need of being shortened down. In order to do so, and avoid too many iterations, one can consider sampling a subset of N' pixels out of all N pixels in the image, and to this set of pixels apply the truncated Gaussian fitting procedure, in order to reduce the number of required iterations. If the pixels are sampled randomly the underlying distribution should be preserved to some degree. It can now be of interest to investigate for which fraction N'/N fits to the N'-subset produces values of the parameters with an acceptable accuracy. There is of course a trade-off here between computational cost and accuracy in the fitting procedure, which one should consider. How this should be done most likely varies from case to case. Some additional material and minor results of how this procedure (of using only a fraction of all pixels to extract the parameters for the background distribution) tends to behave in terms of error can be found in appendix E.

5.2 Algorithm considerations

We will here discuss some of the concerns the new algorithm faces and has to overcome, in order to achieve good performance in terms of the fraction of correctly classified pixels and a good prior estimate of the fraction of correctly classified pixels.

5.2.1 Uncorrelated feature value

Let us start by mentioning the importance of the feature value f being uncorrelated with its corresponding intensity value I. All steps in the Bayesian approach in section 3.2 is dependent on the fact that f and I are uncorrelated, which makes it utterly important that this is indeed the case. Without the Bayesian approach, we can still fit a truncated Gaussian and determine the approximate number of background pixels in the image, but after that we still have to rely on the intensity itself to make decisions for classifying pixels as either background or signal. The performance of the new algorithm will most likely decrease in the presence of a significant correlation between f and I, so knowing if f and I are correlated is crucial to evaluate whether the segmentation can be trusted. In order to verify that f and I are uncorrelated for the set of pixels that are used as training data, the user should consider making a histogram like that seen in Fig. 15.

5.2.2 Preserving the underlying distribution

Since the new algorithm is dependent on the underlying distribution of both intensity and feature values, approximation of these are important. In this work we have chosen histograms to approximate the underlying distributions, but other ways of approximating the underlying distributions is possible, such as non-parametric estimations [18]. The complexity such methods would add to the procedures of performing all steps in the new algorithm resulted in the choice of histograms for convenience.

To generate a histogram in the best possible way, in terms of preserving the underlying distribution, the choice of bin size is crucial. No absolute rules for how this should be done, independent of the underlying distribution, are to be found. On the other hand, some ways to estimate the best number of bin do exist, though under some assumptions [19, 15, 20]. For the work done in this study, Scott's rule [15] has been used in order to estimate the optimal number of bins. This rule assumes that the underlying distribution follows a normal distribution, which will not be true in many cases. The rule has however proven to be rather effective for the cases studied so far. For an algorithm that assumes less about the underlying distribution, one might consider [21].

5.2.3 Padding of images

Let us here briefly touch upon the concerns of padding and its impact on the performance of the new algorithm. For pixels that are located sufficiently close to the edges of the image, the modified median filter will include regions where no pixels' intensity values are to be found. It is in this situation possible to make several different decisions and what should be considered best is not obvious. One could for example replace the missing pixels with zeros, or reflect parts of the image. Furthermore, it is possible to modify the technique to obtain the modified median filter by simply excluding the parts of the window that stretch outside the image. In addition to this, one should

consider excluding the pixels affected by padding from the set of pixels constituting the training data. The effects of this should after all have a rather small impact on the final result as long as the images are large, since the overall fraction of affected pixels decreases for increasing image size.

5.3 Real images

The whole study has been conducted on artificial images. It is utterly important to point this out and also mention something about why this should be taken under consideration before evaluating the newly presented algorithm and its potential functionality on real images. For most fluorescence images obtained there are mainly two concerns, which will be brought up in section 5.3.1 and 5.3.2. In the cases when studying fluorescence images obtained through a cellphone microscopy [22] there is an additional concern, which will be brought up in section 5.3.3.

5.3.1 Beyond Gaussian background

Maybe the most obvious thing that could cause problems for the new algorithm and its ability to satisfactorily segment real images is the case when the background intensity distribution does not follow a Gaussian distribution. Since most of the progress achieved from the new algorithm compared to the Otsu method, depends on the fact that a truncated Gaussian successfully, and with low errors in the estimated parameters μ_{bg} and σ_{bg}, could be fitted to the background intensity distribution, the absence of this possibility drastically reduces the benefits of the new algorithm.

On the other hand, we are not bound to only use a Gaussian distribution to model the background intensities. As mentioned in the very beginning of section 3, the background noise can in many cases be rather well described by a Poisson process. So, an alternative approach to the Gaussian PDF would simply be to exchange this functional form to something more suitable, as for example a Poisson distribution. Let us take one step further here and also mention the problem which can occur if we hope to model the background distribution with a Poisson distribution. Problems can occur since the camera with which the images are taken does not register one single photon as one count in that specific pixel. Incoming photons are, when detected, transformed to an electric current which then is amplified by a multiplicative factor in order for the electronics to detect the signal [23]. Thus, one photon does not necessary correspond to one count in intensity in the pixel in which it was detected. Fortunately, since the amplification is linear, it should be possible to obtain the number of photons in a pixel as a constant factor times the intensity value in that pixels, which not necessary has to be an integer value. Further concerns might appear due to the technicalities of avoiding dark currents [23] in the camera electronics, which is done to avoid registration of false signals not triggered by a photon, whereby all registered intensity values are shifted upwards in intensity. This causes a potential Poisson distribution not to start at zero intensity as expected, but at a higher value.

5.3.2 The point spread function

Of further concern when dealing with real images is the so-called point spread function (PSF) and its influence on capabilities of precise location of an emitting source. A photon-emitting fluorescence bead will be more extended in an image than its actual spatial size [24]. This is a problem for our segmentation method since it relies on the neighboring pixel intensities to be uncorrelated with the pixel of interest. In the case of having a point spread function the spatial region around the pixel will be affected by the intensity of the pixel itself. In addition to this the microscopy resolution is limited by optics to the order of $\sim \lambda/2NA$, where λ is the wavelength of the incoming light and NA is the numerical aperture which depends on the microscope objective. With λ in the range of visible light (400 - 700 nm) and common values of the numerical aperture $NA \sim 1$ [25], will give a limited resolution in the order of a few hundred nanometers.

The finite resolution requires precautions when choosing a feature value, in order to obtain the conditional probabilities. A neighborhood of a too local character for obtaining the feature value f could miss important characteristics of the neighborhood. However, avoiding this by excluding the local neighborhood and increasing the window size for the modified median filter (see section 3.2) could result in a lower accuracy in the final classification. A potential solution to this problem

could be to model the PSF and deduce the true intensity value in each pixel; further work to check the validity of such an approach is, however, needed.

5.3.3 Nonuniform illumination

The last main problem that the new algorithm could meet in the case it is applied to cellphone images is nonuniform illumination [22]. The nonuniform illumination is in general rather tricky to model and the illumination profile might vary from image to image. To avoid this problem, or rather correct for it after the image is taken, a number of possible approaches are possible.

One way of dealing with this would be, before the real image containing the sample of interest is produced, to take an image without the sample in place and simply subtract or divide this image from the final one. In practice though, several problems can occur if this is attempted. Firstly, the position of the cellphone might vary in between images, and secondly, the position of the sample itself might vary in between images.

Another way to overcome the nonuniform illumination issue would be to use a homomorphic filtering technique [1]. This technique is based on the illumination-reflectance model and utilizing filtering in the frequency domain instead of the spatial domain. The method is used in other application than fluorescent images [26]. A difficulty with this approach is choosing an appropriate *filter function* that allows for variations in intensity, such that signal regions can be preserved while variations in intensity due to nonuniform illumination will be filtered away.

A third way of taking into account the nonuniform illumination would be to model the illumination profile with a known functional form. This would depend on the experimental setup, and most likely vary slightly in between images. It is here preferable to fit a new functional form to each image that will be segmented to correct for displacement of components.

Which of the methods to correct for potential nonuniform illumination that will yield the best result, is from this discussion not possible to deduce. Implementation and test have to be done on real images in order to verify what works and what does not work. The optimal way may further depend on the specific equipment construction and design.

5.4 Optimal classifier

Let us here discuss the choice of the window size for the modified median filter. In this study the size of 5×5 pixels in window width has been used, but nothing has been put forward to motivate that choice of size and nothing tells us that is indeed the best value. Rigorous tests with different sizes and shapes of the modified median filter might bring clarity to this aspect, and is of course just possible as long as we know the correct answer. For the case where the answer is not known, we would require a different approach to tackle this problem. Luckily, we can from the estimated number of correctly classified pixels, or rather the estimated fraction of correctly classified pixels η_{est}, estimate how well the classification went rather accurately (see section 4.2). So, in principle what one could do is to try a lot of different sizes for the modified median filter and based on the values of η_{est} decide which one that gave the best classification.

We after all have to remember that the way we looked at the problem of optimizing the window size for the modified median filter up until now did not take in to account the fact that we want to keep the feature and intensity value uncorrelated. It must be mentioned that the estimated fraction of correctly classified pixels most likely will be rather inaccurate if the feature value and intensity are correlated. In line with this argumentation, minimization of the correlation of f and I might well be equally important as maximizing the estimated fraction of correctly classified pixels η_{est}.

It is further possible to consider f to be something else than the modified median filter. A simple average of the neighborhood for the pixel of interest or a more sophisticated weighted average could potentially represent good feature values. The advantage of looking at a local neighborhood is that, the variance around the sample mean of median (or mean) values drawn from a normal distribution with mean μ and variance σ^2, is asymptotic to $\sigma\sqrt{\frac{\pi}{2n}}$, where n is the number of elements in the set from which the median (or mean) is extracted [27, p.369]. This tells us that for large window sizes of the modified median filter, we expect the distribution of the feature value for the background pixels to have lower variance than the original intensity distribution for the

background pixels. Furthermore, one advantage of the median filter compared to a linear mean filter is that it causes significantly less blurring [1], but still reduces noise, and is therefore good at preserving edges in images.

Beyond this it would also be possible to add many features making f multi-dimensional, $\vec{f} = (f_1, f_2, ..., f_n)$. On top of this one might even consider weighting the importance of the different features as well, such that the decision is mostly based on some feature but still dependent on others.

6 Summary

We have in this study introduced a new algorithm for segmenting grayscale images with a Gaussian background intensity distribution and tested it on a set of synthetic images. The new algorithm appears to perform equally well or better than the Otsu method in all cases that has been tested. Furthermore, the new algorithm manages to, rather accurately, estimate the fraction of correctly classified pixels in the final binary image.

6.1 Missing pieces

There is a number of improvements that could be done if more time and effort were spent on further developing the new algorithm, such as improving the procedure of estimating the underlying distribution, finding an optimal feature value automatically, implementation of other functional forms for the intensity background distribution, and automatically correcting for nonuniform illumination.

In case most of the concerns about the implementation of the new algorithm on real images can be solved, it can be expected that it will handle the task of segmenting these well and in addition give the user an estimated fraction of correctly classified pixels close to the true value.

6.2 My contributions

The idea to start this new project was initialized and introduced to me by my supervisor Tobias Ambjörnsson. He handed over the theoretical foundation of fitting a truncated Gaussian as well as algorithms to perform this task. I continued on the project by implementing the Bayesian decision theory framework, and conducted tests evaluating the validity of the new algorithm.

References

[1] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Pearson Prentice Hall, 2008.

[2] C.-W. Wang and C.-P. Yu, "Automated morphological classification of lung cancer subtypes using h&e tissue images," *Machine Vision and Applications*, vol. 24, pp. 1383–1391, Oct 2013.

[3] L. Zhuo, L. Jiang, Z. Zhu, J. Li, J. Zhang, and H. Long, "Vehicle classification for large-scale traffic surveillance videos using convolutional neural networks," *Machine Vision and Applications*, vol. 28, pp. 793–802, Oct 2017.

[4] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[5] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networksa review," *Pattern Recognition*, vol. 35, no. 10, pp. 2279 – 2301, 2002.

[6] W. Quanli, N. Jarad, T. CheeMeng, Y. Lingchong, and W. Mike, "Image segmentation and dynamic lineage analysis in singlecell fluorescence microscopy," *Cytometry Part A*, vol. 77A, no. 1, pp. 101–110.

[7] O. Dzyubachyk, W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering, "Advanced level-set-based cell tracking in time-lapse fluorescence microscopy," *IEEE Transactions on Medical Imaging*, vol. 29, pp. 852–867, March 2010.

[8] H. Talbot, H. Phelippeau, M. Akil, and S. Bara, "Efficient poisson denoising for photography," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 3881–3884, Nov 2009.

[9] X. Jin and K. Hirakawa, "Approximations to camera sensor noise," in *Image Processing: Algorithms and Systems XI*, vol. 8655, p. 86550H, International Society for Optics and Photonics, 2013.

[10] J. del Castillo, "The singly truncated normal distribution: A non-steep exponential family," *Annals of the Institute of Statistical Mathematics*, vol. 46, pp. 57–66, Mar 1994.

[11] C. Blocker, "Maximum likelihood primer," *CDF-6507*, 2002.

[12] J. Tellinghuisen, "Statistical error propagation," *The Journal of Physical Chemistry A*, vol. 105, no. 15, pp. 3917–3921, 2001.

[13] L. Rade and B. Westergren, *Mathematics handbook for science and engineering*. Springer Science & Business Media, 2013.

[14] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*, vol. 3. CRC press Boca Raton, FL, 2014.

[15] D. W. Scott, "Scott's rule," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 497–502.

[16] C. A. R. Hoare, "Quicksort," *The Computer Journal*, vol. 5, no. 1, pp. 10–16, 1962.

[17] G. Blom, L. Holst, and D. Sandell, *Problems and Snapshots from the World of Probability*. Springer New York, 2012.

[18] B. W. Silverman, *Density estimation for statistics and data analysis*. Chapman and Hall/CRC, 1 ed., 1986.

[19] D. Freedman and P. Diaconis, "On the histogram as a density estimator:l2 theory," *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 57, pp. 453–476, Dec 1981.

[20] D. W. Scott, "Sturges' rule," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 3, pp. 303–306.

[21] K. H. Knuth, "Optimal data-based binning for histograms," *arXiv preprint physics/0605197*, 2006.

[22] A. Skandarajah, C. D. Reber, N. A. Switz, and D. A. Fletcher, "Quantitative imaging with a mobile phone microscope," *PLOS ONE*, vol. 9, pp. 1–12, 05 2014.

[23] P. H. David Dussault, "Noise performance comparison of iccd with ccd and emccd cameras," 2004.

[24] K. I. Mortensen, L. S. Churchman, J. A. Spudich, and H. Flyvbjerg, "Optimized localization analysis for single-molecule tracking and super-resolution microscopy," *Nature methods*, vol. 7, no. 5, p. 377, 2010.

[25] H. Gross, B. Achtner, and F. Blechinger, *Handbook of optical systems*. Wiley-VCH, 2007.

[26] C.-N. Fan and F.-Y. Zhang, "Homomorphic filtering based illumination normalization method for face recognition," *Pattern Recognition Letters*, vol. 32, no. 10, pp. 1468 – 1479, 2011.

[27] C. Harald, *Mathematical methods of statistics*. Princeton University Press, 1946.

[28] A. B. Sunter, "List sequential sampling with equal or unequal probabilities without replacement," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 26, no. 3, pp. 261–268, 1977.

[29] C.-K. Wong and M. C. Easton, "An efficient method for weighted sampling without replacement," *SIAM Journal on Computing*, vol. 9, no. 1, pp. 111–113, 1980.

[30] V. A. Epanechnikov, "Non-parametric estimation of a multivariate probability density," *Theory of Probability & Its Applications*, vol. 14, no. 1, pp. 153–158, 1969.

Appendices

A Poisson distribution in the limit of large mean

Here, we will try to illustrate that the Poisson distribution approaches the normal distribution when the mean λ is large. The Poisson distribution is given by the probability mass function

$$P(\lambda) = \frac{\lambda^x e^{-\lambda}}{x\,!} \quad x = 0, 1, 2, 3, \ldots, \tag{22}$$

where λ is the mean and variance of the distribution. We obtain the characteristic function $\varphi(t)$ of the Poisson distribution as the expected value of $e^{itx}P(\lambda)$ with respect to x, which is [13]

$$\varphi(t) = \sum_{x=0}^{\infty} e^{itx} \frac{\lambda^x e^{-\lambda}}{x\,!} = e^{-\lambda} \sum_{x=0}^{\infty} \frac{\left(\lambda e^{it}\right)^x}{x\,!} = e^{\lambda\left(e^{it}-1\right)}. \tag{23}$$

For large values of λ, we require small values of t to get $\varphi(t)$ significantly different from zero. Therefore, we Taylor expand the exponent in (23) for small values of t to second order as

$$\varphi(t) \approx e^{i\lambda t - \frac{1}{2}\lambda t^2}. \tag{24}$$

Now, we look at the normal distribution $\mathcal{N}(\mu, \sigma^2)$ with mean μ and variance σ^2. Its corresponding characteristic function $\Theta(t)$ is [13]

$$\Theta(t) = e^{i\mu t - \frac{1}{2}\sigma^2 t^2}, \tag{25}$$

and we notice that for large values of λ, the Poisson distribution approaches a normal distribution with mean $\mu = \lambda$ and variance $\sigma^2 = \lambda$, as illustrated in Fig. 14.

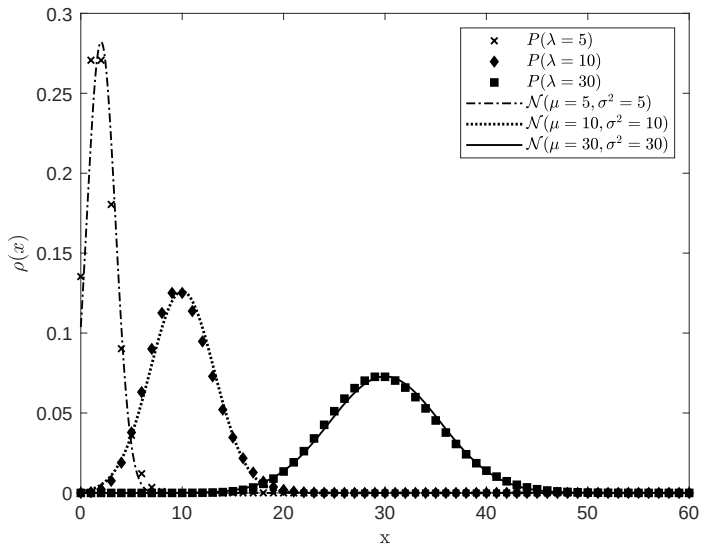


Figure 14: **The Poisson distribution for increasing mean.** In the figure, the Poisson distribution $P(\lambda)$ is plotted for three different values of the mean λ. In addition to that, three normal distributions are plotted with the mean $\mu = \lambda$ and variance $\sigma^2 = \lambda$, which is the expected form of $P(\lambda)$ for large values of λ. As seen in the figure, this is indeed the case and the Poisson distribution approximates well the normal distribution for $\lambda = 30$. The approximation is expected to be even better for even higher values of λ.

B Variance in the estimated number of background pixels

Let us not settle with an estimate of N_{bg} (see page 7) and here compute the error of this estimate. After all, it is interesting to have such an estimate of the error in N_{bg}, which arises due to the

error in the fitted parameters. In order to provide such estimate, let us recall that the maximum-likelihood fit provided us with a variance-covariance matrix, as found in (7). In general, with the help of this matrix it is possible to estimate the variance of a function u as [12]

$$\sigma_u^2 = \vec{g}^T \boldsymbol{C}\, \vec{g}. \tag{26}$$

Here $\vec{g}$ is a vector composed by the partial derivatives of $u(q_i)$, if q_i are parameters with an uncertainty, i.e. $\boldsymbol{g}^T = \left[\frac{\partial u}{\partial q_1}, .., \frac{\partial u}{\partial q_i}\right]$. To estimate the uncertainty of N_{bg}, the following expression has to be computed

$$\sigma_{N_{\text{bg}}}^2 = \begin{bmatrix} \frac{\partial N_{\text{bg}}}{\partial \mu} & \frac{\partial N_{\text{bg}}}{\partial \sigma} \end{bmatrix} \begin{bmatrix} \Gamma_{\mu\mu} & \Gamma_{\mu\sigma} \\ \Gamma_{\mu\sigma} & \Gamma_{\sigma\sigma} \end{bmatrix} \begin{bmatrix} \frac{\partial N_{\text{bg}}}{\partial \mu} \\ \frac{\partial N_{\text{bg}}}{\partial \sigma} \end{bmatrix}. \tag{27}$$

Let us start by computing the partial derivative of N_{bg}, given by (13), with respect to μ

$$\frac{\partial N_{\text{bg}}}{\partial \mu} = n_{\text{cut-off}} \frac{\partial A}{\partial \mu} = \frac{n_{\text{cut-off}} e^{-\frac{(I-\mu)^2}{2\sigma^2}}}{\frac{\sqrt{\pi}\sigma}{2\sqrt{2}} \left(1 + \text{erf}\left(\frac{I-\mu}{\sqrt{2}\sigma}\right)\right)^2}, \tag{28}$$

and completing the calculation by calculating the partial derivative of N_{bg} with respect to σ, we obtain

$$\frac{\partial N_{\text{bg}}}{\partial \sigma} = n_{\text{cut-off}} \frac{\partial A}{\partial \sigma} = \frac{n_{\text{cut-off}} e^{-\frac{(I-\mu)^2}{2\sigma^2}} (I-\mu)}{\frac{\sqrt{\pi}\sigma^2}{2\sqrt{2}} \left(1 + \text{erf}\left(\frac{I-\mu}{\sqrt{2}\sigma}\right)\right)^2}. \tag{29}$$

Since the propagation of error from μ and σ has been estimated, it is possible to, not only estimate the number of background pixels N_{bg}, but also provide the standard deviation of the estimate as following

$$N_{\text{bg}} \pm \sigma_{N_{\text{bg}}}, \tag{30}$$

where the expression for $\sigma_{N_{\text{bg}}}$ is found in (27), using the results in (28) and (29).

C Generate histogram with bin count constraint from known distribution

C.1 Gaussian distribution

Let us consider the task of generating a new, randomly sampled, histogram according to an analytic distribution (see page 10). A potential way to do this, is with a simple "brute-force" way as following:

1. Start by creating an empty new histogram H_{bg} with the same bin edges as H.
2. Determine the number of background pixels, $N_{\text{bg}}^{'}$, that should be included in H_{bg}, by generating one random number according to (14).
3. Generate a random number from the distribution with mean μ_{bg} and standard deviation σ_{bg}.
4. Match the random number to one individual bin in H_{bg}.
5. If the number of events in the matching bin in H_{bg} is less than the number of events in the corresponding bin in H, increase the number of events in this bin by one. If this is not the case, then reject the attempt and return to step 3.
6. Repeat step 3-5 until the number of distributed random numbers equals to $N_{\text{bg}}^{'}$.

To minimize the fluctuations in the number of events in H_{bg}, it might be favorable to repeat the entire process above (step 1-6) several times, in order to obtain an averaged histogram for H_{bg}. The final histogram for the signal pixels can now be obtained by subtracting the final background histogram from the total histogram as $H_{\text{sig}} = H - H_{\text{bg}}$, without obtaining any negative bin counts.

Regarding the scaling of computational time required for the algorithm described above, we conclude that it depends on how many random numbers that need to be distributed. Analysis of a different but still related problem, called the coupon collector, found the scaling to $O(N\log(N))$ [17]. In some cases, the algorithm has proven to be rather time-consuming, despite the tolerable scaling. For the purpose of optimizing computational time, one should consider another, more efficient, way to obtain H_{bg}. Some algorithms that could be adopted, to potentially achieve a better computational time, are those which build on the strategy of sampling without replacement [28, 29].

C.2 Distribution given by a histogram

Now, we are seeking to obtain the full background histogram for the feature values, F_{bg} (see page 11). The challenge is, only a fraction of the number of events in the full background histogram F_{bg} is given to us in the form of a reduced background histogram F_{bg}^{-}. Furthermore, none of the values in any bin in F_{bg}, that we aim to obtain, are allowed to exceed its corresponding bin value in the full feature histogram F. To progress, we assume that the F_{bg}^{-} is distributed approximately in the same way as F_{bg}, i.e., the data obtained in F_{bg}^{-} is not truncated with respect to F_{bg}. Furthermore, we expect the number of background pixels in F_{bg} to be given by (14).

Now, let us go through a possible algorithm to obtain the full background histogram F_{bg}. Start by creating a vector $\vec{v}$ which contains the same number of elements as the number of events in F_{bg}^{-}, that is $n = \sum_i F_{\text{bg}}^{-}(i)$ where i constitutes all bin indices, in both F_{bg}^{-}, and F_{bg}, since they have the same number of bins and same bin edges. Now, give the first $F_{\text{bg}}^{-}(1)$ elements in $\vec{v}$ a value according to the bin index to the first bin in F_{bg}^{-}, namely 1. After this, move on and give the following $F_{\text{bg}}^{-}(2)$ elements in $\vec{v}$ a value corresponding to the bin index of the second bin in F_{bg}^{-}, namely 2. Repeat this until $\vec{v}$ is completely filled with indices corresponding to the number of counts in each bin in F_{bg}^{-}. With the vector $\vec{v}$ in place, it is rather straight forward to create the full background histogram. In bullet-point form our method reads:

- Create an empty new background histogram F_{bg} with the same number of bins as F.
- Draw a random integer number N_{bg}, which constitutes the number of events in F_{bg}, from the distribution given by (14).
- Draw a random integer value R between 1 and n, and extract the corresponding value of $\vec{v}$ for this random number, i.e. $\vec{v}(R)$. This extracted value $\vec{v}(R)$ is an index to a bin in F_{bg}.
- If the number of events given by $F_{\text{bg}}(\vec{v}(R))$ is lower than $F(\vec{v}(R))$, then increase the value of bin $\vec{v}(R)$ in F_{bg} by one. If this is not the case, return to step 3.
- Repeat step 3-4 until the number of distributed events in F_{bg} reaches N_{bg}.

At this point, we have created the full background feature histogram with the approximate distribution of F_{bg}^{-}, which satisfies the requirement of having no bin values higher than F. As in appendix C.1 , it might be favorable to repeat step 1-5 above a number of times to obtain several F_{bg} histograms in order to be able to take the average and even out random fluctuations.

In order to further get rid of unwanted fluctuations, between neighboring bins in F_{bg} and F_{sig}, it appears that performing some sort of smoothing is favorable . The specific smoothing procedure applied to the two histograms in this study makes use of a triangular kernel [18], stretching over 3 bins in each direction from the bin of interest, weighting the value of the bins with linearly decreasing weights the further away they are from the bin of interest, and replace the value on the bin of interest with the average of the weighted sum. The choice of how such a smoothing procedure should be performed, assuming it should be done at all, is at this stage rather arbitrary, even though the idea of smoothing of histograms is in no way new [18, 30].

D The feature value

Here, let us discuss in more detail the feature value f and some associated technicalities that should be touched upon.

The feature value is introduced since the intensity alone does not provide enough information to classify a pixel as signal or background (see Fig. 2). Regarding the choice of feature value, we required that f should be independent of the intensity value I. In an illustrative purpose, a histogram showing the cross-correlation factor, created from the values in $I(\vec{x}_{\text{bg}}, \vec{y}_{\text{bg}})$ and $f(\vec{x}_{\text{bg}}, \vec{y}_{\text{bg}})$ representing the intensity and feature value for the pixels classified as background seen in Fig 3, is shown in Fig. 15.

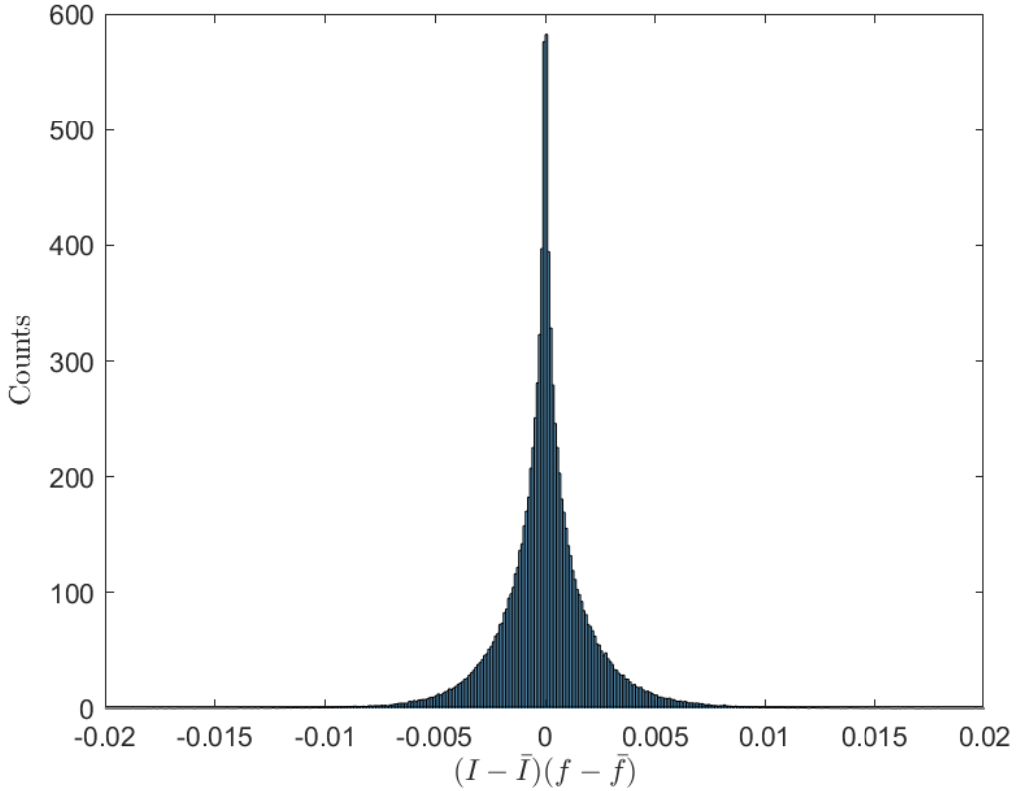


Figure 15: **Histogram of cross-correlation factor.** A histogram of the cross-correlation factor for the pixels classified as background from the fit shown in Fig. 2. A symmetric shape around zero would indicate that there is no correlation between the variables. Visually, it seems like there is no significant correlation between I and f, indicating that a good feature has been chosen. The feature value used here is the modified median filter value (see Fig. 5).

As seen in Fig. 15, the histogram is limited to a region close to zero, which indicates that the feature value f and corresponding intensity I in each pixel often show weak correlation.

Here, let us discuss why the feature value indeed should be uncorrelated with the intensity value. The fundamental principle of the new algorithm presented in this study builds on the fact that we can truncate the data at a certain intensity value $I_{\text{cut-off}}$, and from this extract the full distribution by fitting a truncated Gaussian (see step 1 in section 3.2). As the intensity alone, in some cases, does not provide enough information to classify pixels with a satisfying result, we introduced an additional parameter to further separate signal and background. When using the Bayesian approach (see page 7), we require that the training set of background pixels have to be representative for all background pixels in the image in terms of the feature. The thing now is, when we truncate the data in intensity space, we are not allowed to truncate the data in terms of its distribution in feature space, because if we do so, the Bayesian approach will not work. Furthermore, for the feature to bring new, valuable, information to the Bayesian decision theory,

the signal and background pixels have to form spatial coherent regions where the signal pixels are separated from background pixels. If this is not the case, the distribution of the feature values for the background training pixels would appear similar to the distributed for all pixels, but with lower variance [27], since we use a type of median filter. This could occur for images with spatial regions of interest only extending a single pixel. For most images, this concern might not be a problem due to another concern related to the point spread function (see section 5.3.2).

The best way to extract the feature value f, is left open in this study; even though preliminary tests have been performed in order make sure that the chosen method (modified median filter) performs on a level similar to competitive methods, e.g. a moving average.

E Errors in the truncated Gaussian fit parameters when reducing the number of pixels

Here, we are interested in the performance of the truncated Gaussian fitting procedure with a changing fraction of included pixels, and present some tests to investigate this matter. By sampling a set of N' unique pixels from an image, where $N' \leq N$, and on this set of pixels fit a truncated Gaussian, the parameters for the background distribution can be extracted and compared to the true values, as a function of N'/N. The reason why one should consider this, is the problem of large image sizes, when minimization of computational cost (see section 5.1) becomes important in relation to the trade-off in precision for truncated Gaussian fitting parameters.

For the scope of the tests considered here, the image seen in Fig. 1b has been used. The procedure of sampling N' pixels and to their intensity distribution fit a truncated Gaussian was repeated 20 times to obtain an average value and the corresponding standard deviation of the extracted parameters. The fraction of sampled pixels N'/N was changed in steps of 0.1 from 0.2 to 1.0. The result regarding the average of the mean of the background distribution $\bar{\mu}_{\text{bg}}$ can be seen in Fig. 16 and the result obtained for the average of the standard deviation $\bar{\sigma}_{\text{bg}}$ can be seen in Fig. 16.

We notice from the the plot seen in Fig. 16, that the standard deviation σ_μ tends to increase for decreasing values of N'/N, and the uncertainty is growing considerably large for values below $N'/N = 0.5$. On the other hand, the uncertainty seems to be very small for values of $N'/N \geq 0.8$, indicating that excluding 20 % of the pixels does not have a big effect on the precision of the fitted parameters. Furthermore, it appears as $\bar{\mu}_{\text{bg}}$ increases for decreasing values of N'/N and shows a biased behavior. This might seem unreasonable, since nothing in particular tells us that fewer data points, in general, should give this behavior. A potential, but not at all certain, explanation to this phenomena could be the fact that $I_{\text{cut-off}}$ is steadily increasing, which would result in more signal pixels being taken into the fitting. It may be that a higher sensitivity of the estimated value of $I_{\text{cut-off}}$ is needed to overcome this problem. To check if this indeed is the problem, one should fix $I_{\text{cut-off}}$ and redo the test seen in Fig. 16.

From the plot in in Fig. 17, it is clear that, in a similar way as seen in Fig. 16, the average value of $\bar{\sigma}_{\text{bg}}$ tends to increase for decreasing values of N'/N in a biased way. The deviation of $\bar{\sigma}_{\text{bg}}$ from its true value 0.1 is not very large, though noticeable and will most likely have an impact on the efficiency of the algorithm and the fraction of correctly classified pixels.

What has been left out of from the observations in both Fig. 16 and 17, is the fact that each individual value contributing to the mean of the parameters obtained as $\bar{\mu}_{\text{bg}}$ and $\bar{\sigma}$ has an uncertainty in themselves. For example, the point where all the pixels have been included in the fit, to the very far right in Fig. 17, one sees that the error is zero, which of course is not true. We know that the error in this point is ± 0.00025 from the fit in Fig. 2, which places it in the same order as the point obtained for $N'/N = 0.7$.

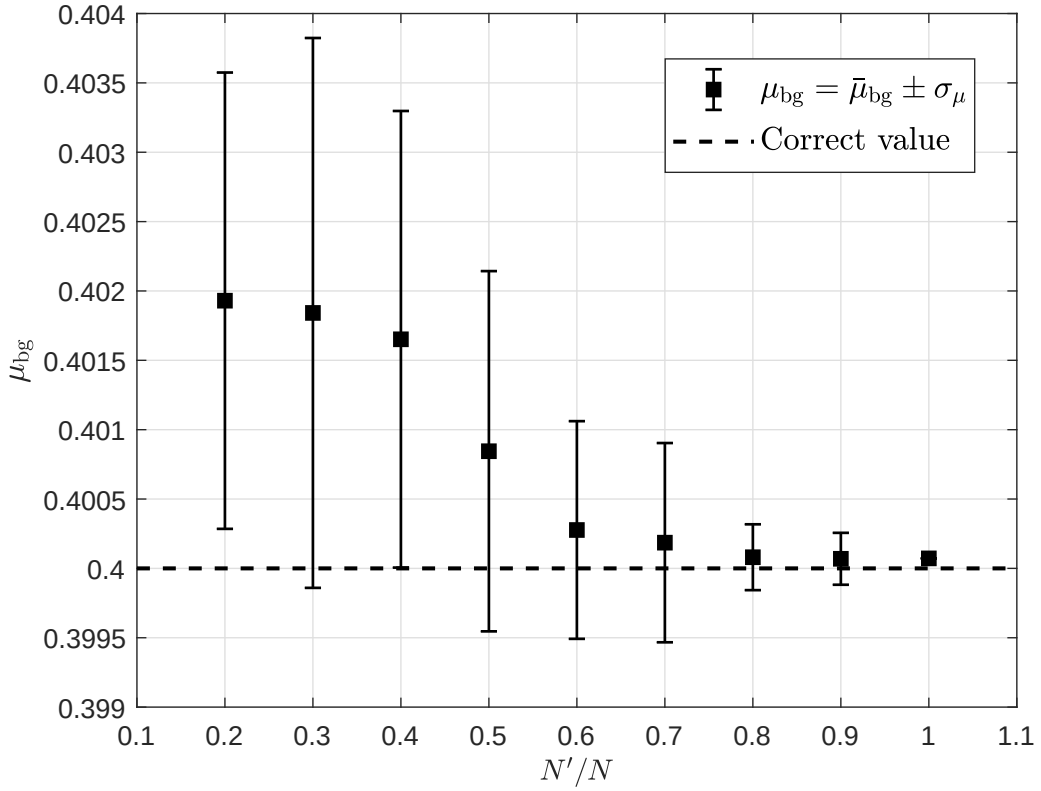


Figure 16: **Mean for truncated Gaussian fit for different sampling fractions.** The plot shows the average value of the obtained values for $\mu_{\rm bg}$, for different values of the sampling fraction N'/N. The mean $\mu_{\rm bg}$ is obtained by 20 times randomly sample N' pixels from the image in Fig 1b, and to their intensity distribution fit a truncated Gaussian, and calculate the mean of these different values of $\mu_{\rm bg}$. The standard deviation around this mean for the 20 different values is indicated with error bars in the plot. The total number of pixels is $N = 624448$. The correct value of $\mu_{\rm bg}$ is indicated with the dashed line, and it is obtained from distribution that the background intensities were sampled from (see Fig. 1b). from the plot, it is clear that the variance of $\sigma_{\rm bg}$ increases with fewer pixels. Furthermore, the value of $\bar{\mu}_{\rm bg}$ seems to approach the true value of 0.4 when more pixels are included in the fit.

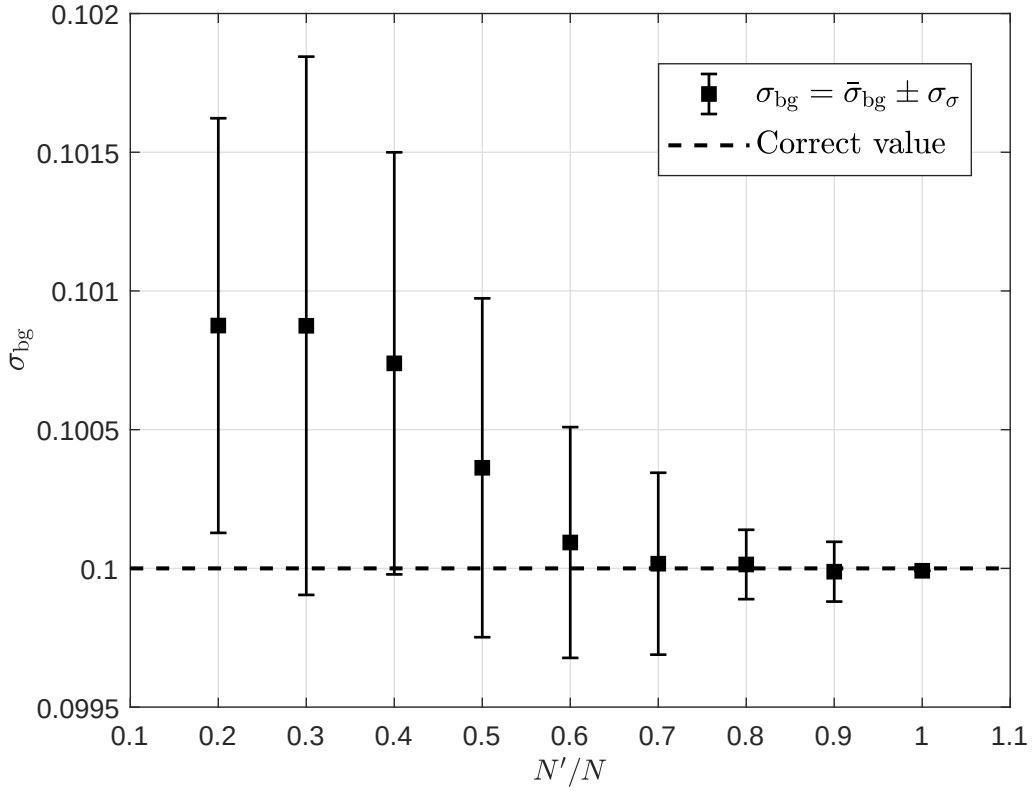


Figure 17: **Standard deviation for different sampling fractions.** In the figure, the mean of σ_{bg} is plotted against the sampling fraction N'/N. The mean of σ_{bg} is, as the values in Fig. 16, obtained by randomly sample N' pixels from the image in Fig. 1b 20 times. For each random sample, we fit a truncated Gaussian thereby extracting the parameter σ_{bg}, from which a mean can be calculated. The standard deviations of these 20 values from the mean $\bar{\sigma}_{\mathrm{bg}}$ are indicated by error bars. The true value of σ_{bg} is indicated with the dashed line and is obtained from the distribution of the background intensities (see Fig. 1b). Notice that the value of $\bar{\sigma}_{\mathrm{bg}}$ tends to decrease for larger values of N'/N. Further, we notice that the value of $\bar{\sigma}_{\mathrm{bg}}$ seems to approach the true value of 0.1 for values of N'/N on the same order as in Fig. 16.

F Complementary numerical data

Here, we give the numerical values for the results presented in Fig. 11 and 12 with the corresponding parameters μ_{bg}, σ_{bg}, μ_{sig} and σ_{sig}, as well as the standard deviation in the presented fraction of correctly classified pixels. Let us denote the true fraction of correctly classified pixels for the new algorithm with an intensity-independent prior probability by η_{nai}, and the true fraction of correctly classified pixels for the new algorithm with an intensity-dependent prior probability by η_{nad}. Furthermore, let us denote the estimated fraction of correctly classified pixels for the new algorithm with an intensity-independent prior probability by $\eta_{\mathrm{est,nai}}$, and the estimated fraction of correctly classified pixels for the new algorithm with an intensity-dependent prior probability by $\eta_{\mathrm{est,nad}}$. The exact values for all data points seen in Fig. 11 can be seen in Table 1, and all the exact values for the data points in Fig. 12 can be seen in Table 2.

Table 1: The fraction of correctly classified pixels given in numerical values for the results presented in Fig. 11, reported with one standard deviation (Δ) for different values of $\mu_{\rm sig}$. For all values of $\mu_{\rm sig}$ we have: standard deviation of signal intensity distribution $\sigma_{\rm sig} = 0.035$, mean of background intensity distribution $\mu_{\rm bg} = 0.4$ and standard deviation of the background intensity distribution $\sigma_{\rm bg} = 0.1$. The true fraction of correctly classified pixels is denoted by: $\eta_{\rm Otsu}$ for the Otsu method, $\eta_{\rm nai}$ for the new algorithm with an intensity-independent prior probability and $\eta_{\rm nad}$ for the new algorithm with an intensity-dependent prior probability. The estimated fraction of correctly classified pixels is denoted by: $\eta_{\rm est,nai}$ for the new algorithm with an intensity-independent prior probability and $\eta_{\rm est,nad}$ for the new algorithm with an intensity-dependent prior probability.

$\mu_{\rm sig}$	$\eta_{\rm Otsu} \pm \Delta$	$\eta_{\rm nai} \pm \Delta$	$\eta_{\rm est,nai} \pm \Delta$	$\eta_{\rm nad} \pm \Delta$	$\eta_{\rm est,nad} \pm \Delta$
0.4800	0.5722±0.0007	0.9653±0.0024	0.9712±0.0024	0.9628±0.0047	0.9356±0.0028
0.4900	0.5731±0.0006	0.9718±0.0009	0.9771±0.0055	0.9692±0.0027	0.9445±0.0036
0.5000	0.5869±0.0004	0.9776±0.0006	0.9821±0.0010	0.9771±0.0007	0.9545±0.0013
0.5100	0.6005±0.0005	0.9814±0.0004	0.9824±0.0026	0.9826±0.0007	0.9612±0.0026
0.5200	0.6140±0.0005	0.9834±0.0004	0.9836±0.0032	0.9850±0.0013	0.9661±0.0024
0.5300	0.6167±0.0056	0.9856±0.0004	0.9880±0.0040	0.9865±0.0006	0.9712±0.0023
0.5400	0.6272±0.0004	0.9869±0.0017	0.9872±0.0055	0.9890±0.0002	0.9761±0.0027
0.5500	0.6404±0.0004	0.9890±0.0002	0.9890±0.0016	0.9906±0.0002	0.9804±0.0014
0.5600	0.6532±0.0006	0.9895±0.0006	0.9896±0.0031	0.9911±0.0013	0.9820±0.0011
0.5700	0.6660±0.0005	0.9904±0.0012	0.9928±0.0029	0.9919±0.0004	0.9861±0.0005
0.5800	0.6789±0.0006	0.9922±0.0002	0.9933±0.0007	0.9932±0.0001	0.9883±0.0007
0.5900	0.6916±0.0006	0.9923±0.0017	0.9926±0.0023	0.9941±0.0001	0.9903±0.0008
0.6000	0.7043±0.0003	0.9932±0.0001	0.9929±0.0009	0.9948±0.0001	0.9917±0.0007
0.6100	0.7213±0.0063	0.9921±0.0020	0.9922±0.0023	0.9953±0.0001	0.9931±0.0008
0.6200	0.7400±0.0006	0.9920±0.0024	0.9915±0.0030	0.9955±0.0004	0.9935±0.0013
0.6300	0.7517±0.0006	0.9921±0.0023	0.9918±0.0051	0.9954±0.0006	0.9938±0.0016
0.6400	0.7633±0.0006	0.9934±0.0026	0.9933±0.0054	0.9960±0.0004	0.9952±0.0019
0.6500	0.7849±0.0006	0.9945±0.0027	0.9946±0.0041	0.9964±0.0005	0.9961±0.0014
0.6600	0.7954±0.0005	0.9951±0.0023	0.9946±0.0035	0.9970±0.0004	0.9965±0.0011
0.6700	0.8155±0.0004	0.9956±0.0020	0.9950±0.0041	0.9971±0.0010	0.9971±0.0012
0.6800	0.8327±0.0041	0.9950±0.0028	0.9944±0.0036	0.9973±0.0010	0.9973±0.0009
0.6900	0.8435±0.0005	0.9933±0.0032	0.9922±0.0051	0.9973±0.0010	0.9974±0.0010
0.7000	0.8607±0.0005	0.9963±0.0001	0.9961±0.0004	0.9981±0.0002	0.9983±0.0002
0.7100	0.8767±0.0004	0.9947±0.0027	0.9937±0.0042	0.9979±0.0006	0.9983±0.0005
0.7200	0.8911±0.0002	0.9952±0.0024	0.9954±0.0024	0.9983±0.0004	0.9988±0.0004
0.7300	0.9064±0.0034	0.9951±0.0026	0.9941±0.0039	0.9981±0.0008	0.9988±0.0005
0.7400	0.9201±0.0030	0.9951±0.0020	0.9945±0.0037	0.9987±0.0004	0.9991±0.0003
0.7500	0.9327±0.0003	0.9957±0.0018	0.9952±0.0034	0.9988±0.0005	0.9993±0.0004
0.7600	0.9420±0.0002	0.9946±0.0029	0.9935±0.0046	0.9988±0.0005	0.9992±0.0004
0.7700	0.9504±0.0002	0.9951±0.0021	0.9946±0.0029	0.9990±0.0005	0.9994±0.0003
0.7800	0.9578±0.0002	0.9952±0.0024	0.9947±0.0041	0.9990±0.0008	0.9995±0.0004
0.7900	0.9641±0.0003	0.9955±0.0018	0.9956±0.0017	0.9994±0.0003	0.9996±0.0002
0.8000	0.9697±0.0002	0.9953±0.0021	0.9945±0.0038	0.9993±0.0006	0.9996±0.0004
0.8100	0.9725±0.0009	0.9944±0.0025	0.9934±0.0039	0.9993±0.0005	0.9996±0.0004
0.8200	0.9768±0.0002	0.9949±0.0024	0.9935±0.0050	0.9993±0.0007	0.9996±0.0004
0.8300	0.9806±0.0001	0.9948±0.0025	0.9944±0.0032	0.9994±0.0008	0.9996±0.0004
0.8400	0.9840±0.0002	0.9963±0.0001	0.9963±0.0005	0.9998±0.0001	0.9999±0.0000
0.8500	0.9855±0.0001	0.9958±0.0016	0.9951±0.0040	0.9997±0.0006	0.9998±0.0003
0.8600	0.9880±0.0001	0.9951±0.0021	0.9947±0.0035	0.9995±0.0006	0.9998±0.0003
0.8700	0.9892±0.0001	0.9954±0.0019	0.9950±0.0029	0.9996±0.0006	0.9998±0.0003
0.8800	0.9912±0.0001	0.9954±0.0019	0.9947±0.0037	0.9996±0.0007	0.9998±0.0005
0.8900	0.9921±0.0001	0.9949±0.0023	0.9937±0.0046	0.9995±0.0008	0.9997±0.0005
0.9000	0.9936±0.0001	0.9959±0.0012	0.9957±0.0021	0.9998±0.0002	0.9999±0.0001

Table 2: The fraction of correctly classified pixels given in numerical values for the results presented in Fig. 12, reported with one standard deviation (Δ) for different values of μ_{sig}. For all values of μ_{sig} we have: full width of signal intensity distribution $s_{\mathrm{sig}} = 0.5$, mean of background intensity distribution $\mu_{\mathrm{bg}} = 0.4$ and standard deviation of the background intensity distribution $\sigma_{\mathrm{bg}} = 0.1$. The true fraction of correctly classified pixels is denoted by: η_{Otsu} for the Otsu method, η_{nai} for the new algorithm with an intensity-independent prior probability and η_{nad} for the new algorithm with an intensity-dependent prior probability. The estimated fraction if correctly classified pixels is denoted by: $\eta_{\mathrm{est,nai}}$ for the new algorithm with an intensity-independent prior probability and $\eta_{\mathrm{est,nad}}$ for the new algorithm with an intensity-dependent prior probability.

μ_{sig}	$\eta_{\mathrm{Otsu}} \pm \Delta$	$\eta_{\mathrm{nai}} \pm \Delta$	$\eta_{\mathrm{est,nai}} \pm \Delta$	$\eta_{\mathrm{nad}} \pm \Delta$	$\eta_{\mathrm{est,nad}} \pm \Delta$
0.6500	0.8160±0.0026	0.9798±0.0002	0.9747±0.0038	0.9861±0.0002	0.9846±0.0025
0.6600	0.8304±0.0040	0.9808±0.0002	0.9741±0.0038	0.9874±0.0001	0.9855±0.0023
0.6700	0.8445±0.0005	0.9832±0.0008	0.9835±0.0022	0.9886±0.0002	0.9904±0.0015
0.6800	0.8603±0.0004	0.9848±0.0002	0.9823±0.0030	0.9899±0.0002	0.9899±0.0018
0.6900	0.8750±0.0004	0.9855±0.0001	0.9839±0.0020	0.9910±0.0001	0.9914±0.0011
0.7000	0.8886±0.0005	0.9872±0.0007	0.9874±0.0038	0.9916±0.0003	0.9931±0.0017
0.7100	0.9022±0.0022	0.9892±0.0001	0.9880±0.0023	0.9929±0.0002	0.9934±0.0013
0.7200	0.9127±0.0018	0.9897±0.0001	0.9879±0.0015	0.9937±0.0001	0.9939±0.0008
0.7300	0.9232±0.0017	0.9889±0.0019	0.9882±0.0032	0.9939±0.0009	0.9948±0.0010
0.7400	0.9318±0.0017	0.9918±0.0018	0.9919±0.0044	0.9947±0.0002	0.9958±0.0011
0.7500	0.9385±0.0004	0.9927±0.0014	0.9916±0.0019	0.9956±0.0002	0.9960±0.0006
0.7600	0.9456±0.0004	0.9933±0.0001	0.9928±0.0013	0.9962±0.0002	0.9967±0.0006
0.7700	0.9514±0.0009	0.9924±0.0024	0.9914±0.0040	0.9965±0.0004	0.9969±0.0008
0.7800	0.9560±0.0002	0.9933±0.0011	0.9931±0.0018	0.9970±0.0002	0.9975±0.0005
0.7900	0.9605±0.0006	0.9931±0.0015	0.9926±0.0033	0.9973±0.0004	0.9978±0.0007
0.8000	0.9646±0.0005	0.9932±0.0025	0.9937±0.0042	0.9969±0.0004	0.9981±0.0006
0.8100	0.9682±0.0004	0.9956±0.0022	0.9954±0.0027	0.9979±0.0004	0.9984±0.0004
0.8200	0.9719±0.0003	0.9958±0.0017	0.9957±0.0022	0.9982±0.0002	0.9986±0.0003
0.8300	0.9753±0.0002	0.9950±0.0028	0.9953±0.0023	0.9983±0.0004	0.9988±0.0003
0.8400	0.9787±0.0001	0.9933±0.0032	0.9915±0.0057	0.9979±0.0007	0.9984±0.0007
0.8500	0.9820±0.0002	0.9951±0.0024	0.9943±0.0046	0.9984±0.0007	0.9988±0.0006
0.8600	0.9849±0.0001	0.9950±0.0027	0.9947±0.0034	0.9985±0.0007	0.9991±0.0003
0.8700	0.9868±0.0002	0.9957±0.0021	0.9952±0.0032	0.9988±0.0006	0.9992±0.0004
0.8800	0.9880±0.0001	0.9958±0.0015	0.9956±0.0020	0.9990±0.0003	0.9994±0.0002
0.8900	0.9892±0.0004	0.9946±0.0028	0.9936±0.0042	0.9988±0.0007	0.9993±0.0003
0.9000	0.9906±0.0006	0.9963±0.0001	0.9966±0.0004	0.9994±0.0001	0.9996±0.0001
0.9100	0.9918±0.0004	0.9951±0.0026	0.9949±0.0032	0.9992±0.0005	0.9996±0.0002
0.9200	0.9927±0.0004	0.9947±0.0026	0.9936±0.0041	0.9989±0.0008	0.9995±0.0003
0.9300	0.9934±0.0004	0.9963±0.0001	0.9963±0.0005	0.9995±0.0001	0.9997±0.0001
0.9400	0.9936±0.0003	0.9952±0.0023	0.9934±0.0061	0.9991±0.0009	0.9996±0.0004
0.9500	0.9943±0.0001	0.9954±0.0018	0.9951±0.0021	0.9995±0.0003	0.9997±0.0001