PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Supporting artificial intelligence with artificial images

Lars Aurdal, Alvin Brattli, Eirik Glimsdal, Runhild Aae Klausen, Kristin Hammarstrøm Løkken, et al.

Lars Aurdal, Alvin Brattli, Eirik Glimsdal, Runhild Aae Klausen, Kristin Hammarstrøm Løkken, Hans Christian Palm, "Supporting artificial intelligence with artificial images," Proc. SPIE 10794, Target and Background Signatures IV, 107940M (9 October 2018); doi: 10.1117/12.2324969

SPIE.

Event: SPIE Security + Defence, 2018, Berlin, Germany

Invited Paper

Supporting artificial intelligence with artificial images

Lars Aurdal[a], Alvin Brattli[a], Eirik Glimsdal[a], Runhild Aae Klausen[a], Kristin Hammarstrøm Løkken[a], and Hans Christian Palm[a]

[a]Norwegian Defence Research Establishment (FFI), P.O. Box 25, 2027 Kjeller, Norway

ABSTRACT

Infrared (IR) imagery is frequently used in security/surveillance and military image processing applications. In this article we will consider the problem of outlining military naval vessels in such images. Obtaining these outlines is important for a number of applications, for instance in vessel classification.

Detecting this outline is basically a very complex image segmentation task. We will use a special neural network for this purpose. Neural networks have recently shown great promise in a wide range of image processing applications, image segmentation is no exception in this regard. The main drawback when using neural networks for this purpose is the need for substantial amounts of data in order to train the networks. This problem is of particular concern for our application due to the difficulty in obtaining IR images of military vessels.

In order to alleviate this problem we have experimented with using alternatives to true IR images for the training of the neural networks. Although such data in no way can capture the exact nature of real IR images, they do capture the nature of IR images to a degree where they contribute substantially to the training and final performance of the neural network.

Keywords: naval vessel infrared image segmentation neural network

1. INTRODUCTION

In security/surveillance and military applications, IR imagery often plays an important role.[1–3] IR images can be obtained in low light/night conditions and concealing objects under IR observation is difficult since IR sensors capture both emitted and reflected light.[4] In this paper we will consider the problem of outlining military naval vessels as seen in IR images. Such outlines have a number of applications, and can serve in identifying the vessel, classifying the vessel as belonging to a particular design class, determining its orientation relative to the observation vector, can be used in determining whether it is a military or civilian vessel, etc.

Recent developments in image processing have shown the value of different types of neural networks for a number of complex applications in this domain.[5] In particular, object outlining (or the equivalent problem of image segmentation) is one of the areas where neural networks perform very well.[5,6] One of the challenges when using neural networks for this purpose is the number of training images required in order to obtain good performance. When considering outlining of naval military vessels seen in IR images, this problem is aggravated by the difficulty in obtaining real IR images of such vessels. IR images are in general hard to acquire due to the need for specialized hardware for image capture. IR images of naval vessels are also typically considered classified military information since such images can reveal underlying information about the vessel being imaged. Usually only images of vessels from collaborating nations can be obtained.

In order to alleviate this problem we consider alternatives to true IR images for training our network. Such data cannot replace real IR images, but for our purpose we only need images that are sufficiently close to IR images in order for them to have value in the training of the neural network.

In this article we will consider several possible sources of alternative data for our purpose, and discuss the advantages and disadvantages of the different sources when it comes to supporting training of neural networks for image segmentation. We will also show how a network trained on such alternative data performs well on true IR images.

Further author information:
Lars Aurdal: E-mail: lars.aurdal@ffi.no, Telephone: +47 91 50 30 03

Target and Background Signatures IV, edited by Karin U. Stein, Ric Schleijpen,
Proc. of SPIE Vol. 10794, 107940M · © 2018 SPIE · CCC code:
0277-786X/18/$18 · doi: 10.1117/12.2324969

In Section 2 we will provide an overview of the existing body of work related to detecting vessel outlines. We then discuss the various sources for alternative IR imagery that we have considered in Section 3. In Section 4 we will detail our current neural network based approach to vessel segmentation based on alternative data. In Section 5 we present and discuss the results we have obtained and in Section 6 we conclude.

2. RELATED WORKS

Outlining maritime vessels based on input from various types of sensors has been a longstanding field of research driven both by military applications as well as applications in civilian surveillance and law enforcement. Our focus here will be on sensors observing the vessels in the IR part of the electromagnetic spectrum.

Earlier attempts at solving the outlining problem typically rely on some thresholding approach applied to the IR images under analysis.[7–14] Once a threshold, either local or global, has been established, the image is thresholded and the vessel outline is simply the outer contour of the foreground object in the thresholded image. This approach is theoretically very well understood, and has the great advantage of being computationally efficient. However, threshold based methods have numerous drawbacks leaving them of little value today:

- Thresholding as reported in the cited works[7–14] is a purely local process in that each pixel is treated separately from its neighbors making these methods prone to error in the presence of noise which is frequently an issue in IR images.
- Modern military vessels will typically employ different sophisticated methods all aimed at reducing the vessel's signature in different parts of the electromagnetic spectrum.[1] Signature reduction in the IR bands has received considerable attention, and modern military vessels typically have low signature in IR. Simple algorithms such as thresholding that specifically rely on the foreground object to have some level of contrast to its background will typically not work well on images of such vessels.
- Finally, for thresholding to work, the vessel under analysis must stand out, in terms of graylevel values, from some local background. In complex scenarios, such as with vessels in port or close to land, this condition is frequently not met.

In spite of these shortcomings, variations of thresholding have remained popular as a processing step for vessel outlining to this day.[15–18]

In order to address the problems inherent in thresholding, more advanced image processing approaches have recently been applied to the problem of naval vessel segmentation. One such approach is to use *active contours* where a contour is actively deformed using deformation constraints imposed both by the underlying image and by the shape of the contour itself. The aim of the final deformation is a contour that outlines the desired object in the underlying image. This approach was originally suggested by Kass et. al.[19] and has since been refined in a number of ways. Since the deformation is guided by deformation constraints calculated globally for the entire contour, this method is relatively resilient to noise and also to occlusions of parts of the target vessel outline.

Variations of deformable contours are used for vessel segmentation in a number of recent works.[20–23] These and similar methods solve some of the problems encountered with the simpler thresholding approaches, but they still have weaknesses that make them prone to error. The deformable contour must be initialized somewhere in the image and these methods are typically sensitive to the position of the initial contour. After the initial contour is positioned, it is deformed through a series of deformation steps in the hope that it will finally lock on to the desired object outline. In order to achieve this, the contour is typically deformed in a force field derived from the gradients in the image under study. If strong gradients arise from objects in the image apart from the target object, this might guide the contour to an erroneous final position.

Recent developments in the use of neural networks for image processing has brought these methods to the forefront of research in the field of image processing. In particular, from (approximately) year 2010 the performance of neural networks, in a large number of application domains, has been very impressive, frequently surpassing previous methods by a wide margin.[5] At the core of this development have been the so called *convolutional neural networks* (CNN) and the field of research dealing with these and similar networks is frequently termed

deep learning based on the sheer size and number of layers of such networks. We will not discuss the theoretical foundations for these methods here, but refer the interested reader to one of the many excellent sources on this matter.[5]

One very interesting application of different neural network architectures is that of image segmentation and a large number of recently reported works deal with this problem.[24–28] In the work reported here, our aim is to train a segmentation network to outline naval vessels in IR images. An example of the type of outline that we want to achieve is shown in Figure 1 . For this purpose we will use a network architecture described as a *u-net*.[29] The process of training such networks consists in showing the network an image of a given object and an image containing a mask for the object. The training of the neural network then consists in showing the network such image/mask pairs in an iterative fashion training the network to produce the mask on its own given a previously unseen input image.

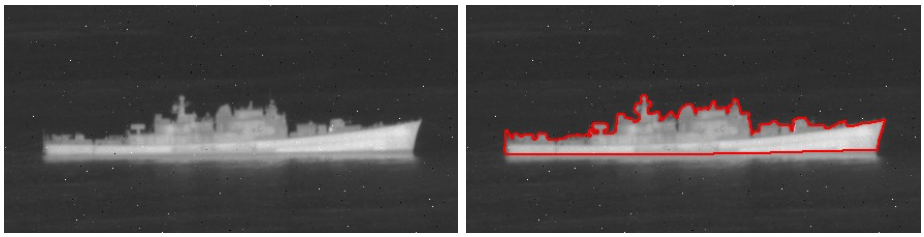

Figure 1. The left part of this figure shows a long-wave (LW) IR image of the Norwegian frigate HNoMS Bergen (HNoMS Bergen was decommissioned in 2005), the type of outline we want to obtain for such an image is shown on the right.

3. POSSIBLE TRAINING IMAGE SOURCES

When developing algorithms for outlining naval vessels seen in IR images one will rapidly realize that such images are hard to obtain. This is so for a number of reasons. IR images of naval vessels show the heat signature of the imaged vessel and can as such reveal signature weaknesses such as the placement of the engine room. Also, IR images can reveal heat signatures related to exhaust gases (the plume) and also the operational state of onboard equipment such as radars. Considering that a number of modern anti-surface missiles will use IR imagery for target detection and homing,[1] it is obvious that IR images of naval vessels are carefully guarded and hard to come by.

Another issue is that those images that are available typically are acquired during different military trials where the participating vessels are imaged during very specific conditions in relation to backgrounds, presence of other vessels, etc. This makes the number of available images small and the variability in these images low. This is of course further complicated by the fact that during such trials, only certain nations will participate making the selection of imaged vessels low. This problem is particularly severe in relation to the training of neural networks. An inherent issue in the training of most modern neural network solutions for image processing is the need for large quantities of training and validation data. The problem is then simply how to obtain the required number of such images for the effective training of the network.

One possible solution to this problem is to use alternative data for the training of the network. This would involve training the neural network on alternative data in the hope that the final network would perform its segmentation task well on both the alternative and the real IR images. This is the approach selected for this study and in the following subsections we will briefly describe different sources of alternative IR images that we have explored.

Before considering the different methods for generating alternative data it is important to clarify exactly what we need to obtain. The most critical output of the selected method is an IR look-alike image of a naval vessel observed in a credible sea scenario including a realistic sea surface and possibly land backgrounds. A very desirable byproduct of the process would be a mask for the vessels above-water structure. This mask can be drawn manually, but this is a time consuming and tiresome process that should ideally be avoided.

3.1 SIMDIS

SIMDIS[*] is a set of software tools developed by the US Naval Research Laboratory that provide two and three-dimensional interactive graphical and video display of live and post processed simulation, test, and operational data.

With SIMDIS it is possible to render a number of 3D ship models in realistic sea scenes and to capture 2D images of these scenes. These images can in turn be given an IR-like appearance by image processing manipulations (see Section 4.1 for details concerning this). We have chosen not to use SIMDIS images in this study for the following reasons:

- SIMDIS is not designed for this kind of use and it was difficult to script the use of SIMDIS such that the images could be generated with any degree of automation. This means that a considerable amount of time must be spent on generating each image.
- We did not find any simple way to obtain the vessel mask automatically from SIMDIS. This means that the masks must be drawn after the image has been generated, a very time consuming process.
- The number of available ship 3D models is rather low. This means that the variability in the generated data is small and thus the final trained neural network will be less performant.

3.2 ShipIR

ShipIR[†]is a software tool specifically designed for generating realistic IR data for naval vessels.

With ShipIR it is possible to render 3D ship models in such a way as to obtain a realistic scene radiance including that of the vessel and of the background. Given these scene radiance data it is a simple matter to obtain an image of the scene. ShipIR allows for inserting the model in the scene under different meteorological conditions and ambient temperatures. With ShipIR it is also possible to obtain the mask for the above water structure of the vessels. We have chosen not to use ShipIR images in this study for the following reasons:

- The number of available ship 3D models is lower than for SIMDIS. This in turn is strongly limiting for the training of the neural network. Generating novel 3D models is possible, but this is a very labor intensive process.
- The seastate in the generated images is always zero, a completely calm sea surface. Likewise, no clouds or more sophisticated backgrounds like land are possible. Again, this strongly limits the value of these images for the training of a neural network.

3.3 Using visual imagery from shipspotting.com

shipspotting.com[‡] is a web site for shipspotters all over the world. As a user one can upload ship images along with the location of the shot, the name and type of the vessel, etc. There are millions of images available and a large fraction of the images shows military vessels. Also, since the users are from all over the world and all that is needed to participate is a simple digital camera and internet access, the range of vessels is enormous.

We were able to download a large number of images of certain military vessels (frigates) from shipspotting.com in the form of RGB images (some images, especially those of older vessels may be graylevel images). A typical image size would be 600 rows by 800 columns with the vessel normally filling a substantial part of the image. The images show vessels in all aspect angles as well as in all kinds of operating scenarios: in port, in harbors, close to land, at sea, etc. Having downloaded the images, they can be converted to look like IR images (see Section 4.1 for more information about this conversion).

There are two main drawbacks when using images from shipspotting.com:

[*]See: https://simdis.nrl.navy.mil/index.aspx
[†]See: http://www.davis-eng.com/ntcs.html
[‡]See: http://www.shipspotting.com

- Obviously, the images are not true IR images and can only be made to look somewhat like IR images through image processing. Although such images are valuable for training, they can not fully replace true IR images.
- As was the case with the SIMDIS images, there is no way to obtain the vessel masks automatically thus forcing us to make these masks by hand, a very time consuming and labor intensive process.

When we still ended up basing our approach on images from shipspotting.com, it is primarily due to the amount of and enormous variability in the data. This in turn allows us to train the neural network better.

4. METHOD

For this article we will consider images of frigates. Our aim will be to produce good outlines of frigates as seen in long-wave (LW) IR images. For simplicity, we will consider images showing such vessels more or less perpendicularly to the starboard or port sides (broadside views). For this purpose we will train a neural network of the *u-net* type. We will describe the network in more detail in Section 4.3. The training of such networks requires a substantial number of pairs of raw images of vessels and corresponding masks positioning the vessel in the image. We do not have the required number of true IR images showing frigates from this angle, and as we have pointed out above, the lack of such a substantial set of true IR images has led us to use alternative images for this purpose.

4.1 Data

Our approach for generating the training and test data is as follows:

1. We have downloaded a large number of images of frigates from `www.shipspotting.com` as described above.
2. We make a subselection of these images keeping only those images that show the vessels from the desired aspect angles (broadside views).
3. For these images, we also manually draw a mask indicating relatively precisely the main outline of the vessel. Thin antennas etc. are omitted, except for that the outline is drawn as carefully as possible. We have a total of 500 such image and mask pairs. Figure 2 shows two such masks, one drawn for an open sea scenario and one for a complex scenario with land background.

Having drawn the masks, we then proceed to transform the RGB images of the selected images into images bearing a good visual resemblance with an LWIR image of the same scene. The procedure for transforming the images is as follows:

1. The network requires the input images to be of the same size, thus we scale all images (and naturally the masks) to a fixed size of 256 rows by 512 columns. Neural networks are surprisingly resistant to such changes in aspect ratio.
2. The images are then converted to graylevel images by simply keeping the green (G) channel in the original RGB image. Other, more sophisticated, methods for converting from RGB to graylevels are of course possible, but this very simple approach works well enough for our purpose.
3. The graylevels in each image are scaled to span the interval $[0.0, 1.0]$.
4. These graylevels are then inverted so that previously dark regions become light and vice versa.
5. The images from `www.shipspotting.com` are very sharp, we therefore blur the graylevel images using a Gaussian blur kernel with $\sigma = 1.0$.
6. We add uniformly distributed noise in the interval $[0, 1]$ according to the formula $i_N = i + 0.05n$ where i_N is the final noisy image, i the input image and n is a matrix the size of the image i with the uniformly distributed noise samples.

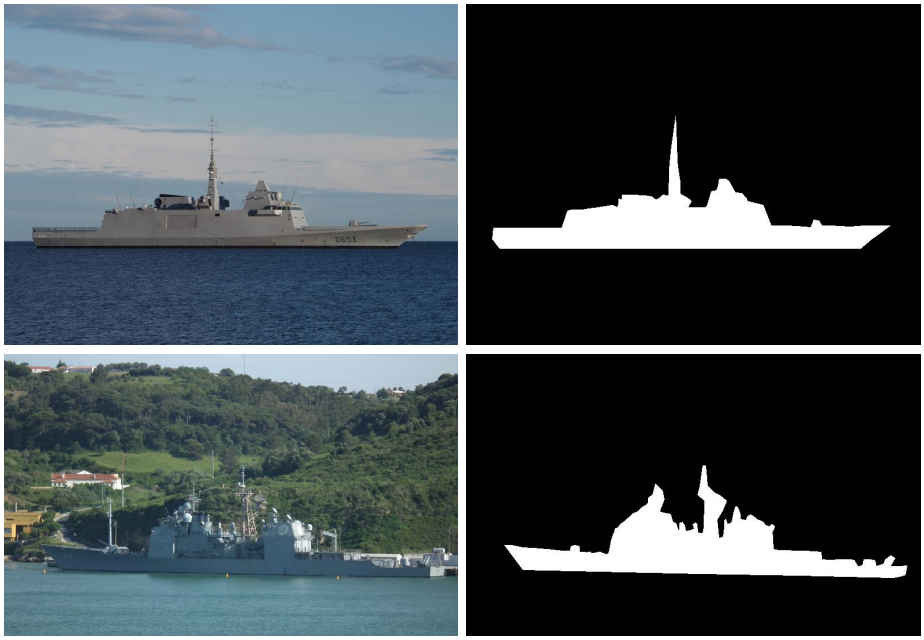

Figure 2. Top left: FS Lanquedoc D-653 (photographer: Soumcouy). Bottom left: USS Anzio CG-68 (photographer: MattyBoy). Both images from `www.shipspotting.com`. Right: the corresponding manually drawn masks that we will use to indicate the precise location of the vessel in the images.

7. The image is then clipped so that the values lie in the interval $[0.0, 1.0]$.
8. We finally add salt and pepper noise by randomly clamping either to 0 or 1 a total of 0.1% of the pixels.

The selection of steps and the choice of values in this procedure might seem somewhat arbitrary but was selected to make the generated images as similar as possible, in a qualitative sense, to available real LWIR images of similar vessels. This whole procedure is illustrated in Figure 3.

4.2 Data augmentation

A standard approach in training neural networks consists in performing so called data augmentation. This consists in performing simple modifications of the data in order to increase the size of the training data set. One simple such approach consists in flipping the images (and of course the corresponding masks) left to right so as to simulate a boat showing its opposite side. We use the following additional approaches in order to increase our data set from an initial size of 500 images to a final size of 5000 images:

1. Pick a random image, 4500 times, among the original 500 images.
2. Randomly select if the image should be flipped left/right or not.
3. Randomly select a factor in the interval $[0.5, 1.1]$ and multiply all pixel values in the image by this value. After this operation, all pixels with values greater than 1.0 are clamped to 1.0. As an example, if the random factor is chosen as 1.05, this will shift all pixels in the image towards lighter values. Values that end up greater than 1.0 in the scaled result end up clamped at 1.0. This process would create a brighter image with some pixels saturated at 1.0.
4. Apply further Gaussian blurring with σ selected randomly from an exponential distribution with a scale parameter $\beta = 0.6$.

Figure 3. The process of turning an original RGB image into an image resembling an LWIR image of the same scene. Original image from `www.shipspotting.com`, photographer: Igor Dilo. The effect of adding uniform noise and salt and pepper noise is hardly discernible.

4.3 Neural network

Convolutional neural networks have recently shown excellent performance on different image processing tasks.[5] Image segmentation is no exception to this. We have chosen to implement a network architecture originally proposed by Ronneberger et. al.[29] This architecture is commonly referred to as a *u-net*, the specific architecture used by our net is shown in Figure 4. As is easily seen from this figure, there are two main paths through the network. Through a series of convolutions information is first condensed and then unpacked through a series of deconvolution layers. Additionally, horizontal network connections make less condensed data available during the deconvolution stages. All critical network parameters should be apparent from Figure 4. During training we apply 50 % dropout after the two last convolutional layers in order to reduce the risk of overfitting to the training data.

All convolution and deconvolution layers use the standard 'relu' (rectified linear unit) activation function except for the output layer which uses a sigmoid activation function.

It is important to pause and consider the nature of the network output at this point. The network will provide an output image with values ranging from 0.0 (low likelihood of a pixel belonging to a vessel) to 1.0 (high value of a pixel belonging to a vessel). Values in the interval between these two extrema are common. A value of e.g. 0.8 could be interpreted as an 80 % likelihood of that pixel belonging to a vessel. The output can therefore be thought of as a probability map giving the probability for each pixel belonging to the vessel. Thresholding such a map at for instance 0.3 will provide a two-level image with values of 0 representing pixels that are 30 % or less likely to belong to the vessel, and pixels with values of 1 representing pixels that are 30 % or more likely to belong to the vessel. The outline of the region(s) of pixels mapped to 1 should ideally be a close approximation to the vessels outline. Examples of such maps are provided in Section 5.

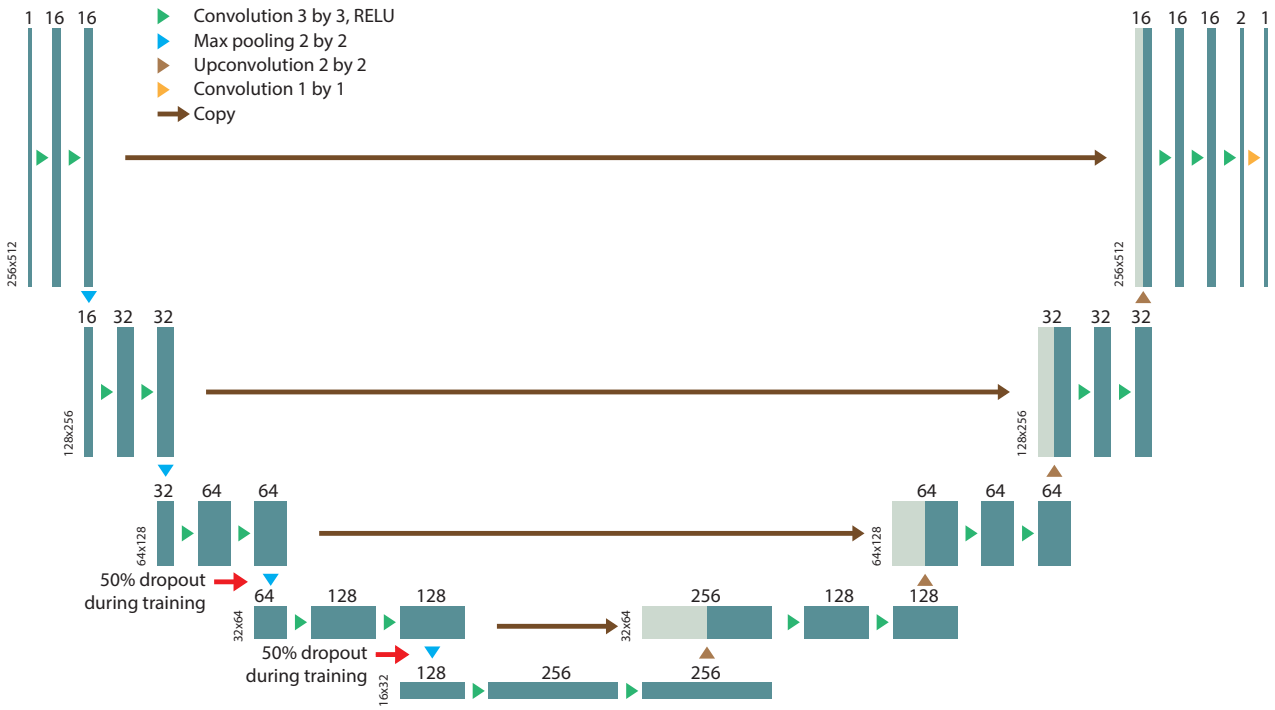

Figure 4. The implemented u-net architecture used in this work.

4.4 Training

The training of the network is implemented in Python (3.6) using the *Tensorflow*[30] library by Google Inc. (API version 1.7). We also employ the *keras* library[31] (version 2.1.5). Training is performed on a Windows 7 standard desktop computer using an Intel Xeon E5-1630 v3 CPU with 32GB RAM and using an NVIDIA Geforce GTX Titan GPU for computational speedup.

Before training, 10 % of the training data are set aside as validation data in order to monitor training progress. The neural network architecture is as shown in Figure 4. We train the network for 100 epochs with a batch size of 4 images randomly selected among the 4500 training images. The loss function is the standard *mean squared error (MSE)* loss metric and we use the *stochastic gradient descent (SGD)* optimizer. After each epoch the current classifier is saved as the current best classifier if it outperforms (in the MSE sense) previous classifiers on the validation data set (consisting of 500 images). Training of the network takes approximately 10 hours.

5. RESULTS AND DISCUSSION

Having trained the network as described above, two tests are important in order to assess performance. Initially we test the network on mock LWIR images generated from visual images, that is, the same kind of images used for training the network. We obviously select these test images among images that are previously unseen by the network (images present neither in the training nor the validation data). This will indicate whether the network performs well on the exact type of data for which it was trained. In a second test step, we test the network on real LWIR images. This will indicate whether the network is able to transfer its performance to images that are somewhat different compared to the type of images used for training.

In order to assess performance, we will show results as outlines made by the network and overlaid on top of the original images. However, this type of illustration can be misleading since it can be quite hard to judge the quality of the outline just through visual observation. For this reason we will use two separate quality metrics, these metrics are described in Section 5.1.

5.1 Quality metrics

For all the test cases we dispose of a set of three images:

1. One image showing a mock or real LWIR image.
2. A hand drawn contour for the vessel seen in the mock or real LWIR image (we will refer to this contour as the **M**anual **C**ontour, MC). The MC serves as the ground truth.
3. An automatically generated contour for the vessel seen in the mock or real LWIR image (we will refer to this contour as the **A**utomatic **C**ontour, AC). The AC is obtained by first using the network to calculate the vessel probability map, we then threshold this map at a value of $t = 0.5$ and keep only the largest thresholded component. The outline of the resulting region of pixels with value 1 are considered vessel outlines.

The two metrics we propose for assessing our results basically compare the MC and the AC for the different test cases. We will use the following two approaches for this:

1. Calculate, for all pixels in the image containing the MC, the distance to the closest point on the MC. Having this distance map it is now possible to sum, over all the pixels on the AC, their Euclidean distance to the closest point on the MC.

   By considering distances interior of the MC to be negative and distances exterior to the MC to be positive we can then generate histograms of the distance from the pixels on the AC to the MC. Figure 5 illustrates how the distance maps are calculated.

   Having this histogram it is also possible to generate the cumulative distribution of the absolute value of the distances from the AC to the MC. This cumulative distribution makes it easy to assess the fraction of points on the AC that lie within a specific distance (in pixels) of the MC.

2. By considering the geometric mean of all the pixels of the regions defined as the insides of the MC and AC, we get a crude measure of the overlap of the contours by comparing the position of this geometric mean for the two contours. Since we know the length of the frigates we consider and since the images are (roughly) broadside images, it is possible to calculate the size of a pixel, in meters, for each of the original mock and real LWIR images. This makes it possible to estimate the distance, in meters, between the two geometric means of the MC and the AC.

5.2 Results using mock LWIR data

For the purpose of testing the network on images derived from visual data we select six images of different frigates seen in increasingly complex scenarios: from open sea situations with good contrast via land background scenarios to in-harbor situations with low contrast. For simplicity we will refer to these six images as $F1_{\text{mock}}$ through $F6_{\text{mock}}$. Figure 6 illustrates results on a land background scenario. This figure shows $F4_{\text{mock}}$ against a land background, the resulting contrast after conversion to a mock LWIR image is moderate. The neural network predicts a relatively good probability map with some minor errors in the bow structure. Figure 7 shows the histograms of distances for the mock LWIR images. From this figure two observations can be made: There is a slight preponderance of pixels outside of the manually drawn contour. This is in part due to the fact that the MC will follow every detail as carefully as possible along the ship contour whereas the AC often will take 'shortcuts' across such details thus passing outside the MC. Secondly, from the cumulative histograms it is clear that for all the cases, $F1_{\text{mock}}$ through $F6_{\text{mock}}$, 80 % of the pixels on the AC will be within 2.4 pixels (or less) of the MC.

In Table 1 we summarize the distance, in pixels and estimated meters, between the geometric mean of the MC compared with that of the AC for the mock LWIR images. Notice the good overlap of these geometric centers.

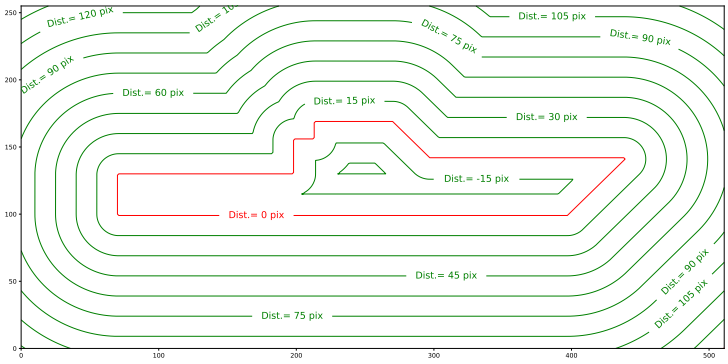

Figure 5. We assume we have a carefully made hand drawn contour of the vessel observed in the IR image being analyzed. This contour appears red in the figure above were we have drawn the contour of a conceptual vessel. Using standard methods from image processing it is easy to determine the distance from any pixel in this image to the closest point on this contour. The original contour will obviously be at a distance of zero pixels from itself. Pixels outside the contour are given a positive distance to the hand drawn contour and those inside receive a negative distance. By inscribing, in the same distance map, a new contour it is now easy to determine the distance from each of the pixels on the inscribed contour to the closest pixel on the hand drawn contour.

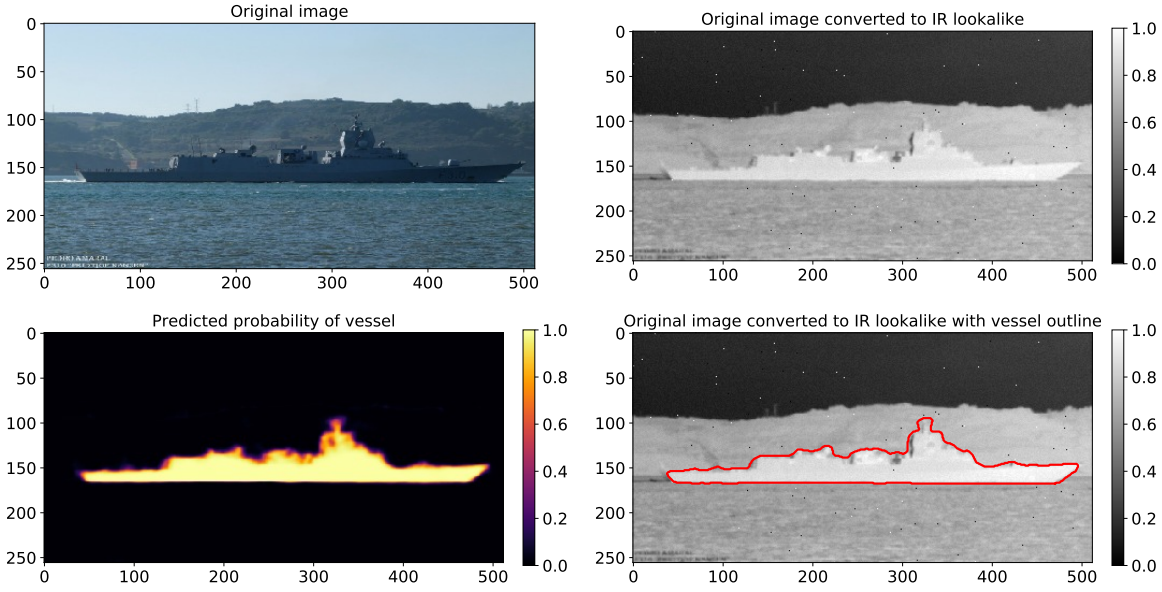

Figure 6. From top to bottom, left to right: Frigate $F4_{\text{mock}}$ (HNoMS Fridtjof Nansen, image from `www.shipspotting.com`, photographer: Pedro Amaral), image converted to mock LWIR image, vessel probability map generated by neural network, final outline (obtained by thresholding the probability map at $t = 0.5$.

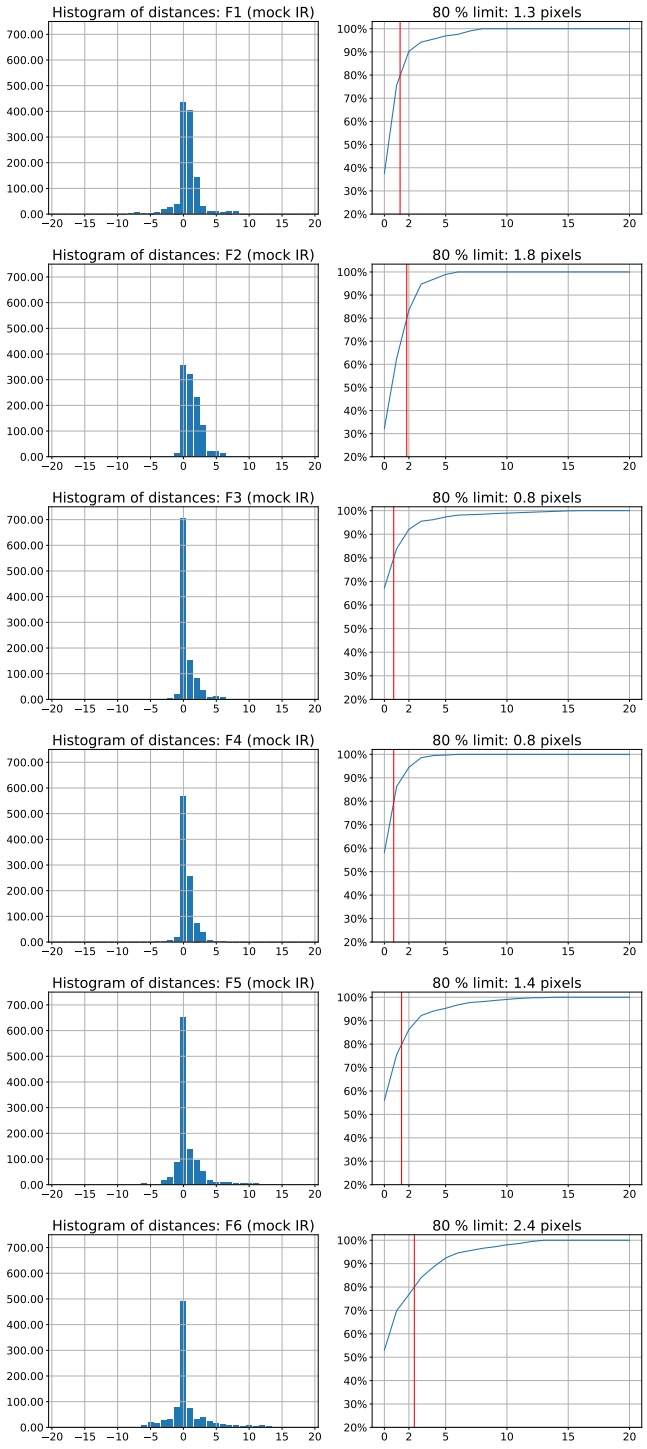

Figure 7. Results for mock LWIR data: The left parts of this figure shows the distribution of distances (in pixels) from the automatically calculated contour (AC) to the closest point on the manually drawn contour (MC). Negative distances signify that the AC lies inside the MC. The right parts of these figures show the cumulative distribution of the absolute value of the distances from the AC to the MC along with the distance limit within which, starting at the MC, we will find 80% of the AC.

Vessel	Dist. [pixels]	Dist. [m]
$F1_{\text{mock}}$	1.1	≈ 0.4
$F2_{\text{mock}}$	5.5	≈ 1.7
$F3_{\text{mock}}$	2.9	≈ 0.9
$F4_{\text{mock}}$	0.4	≈ 0.1
$F5_{\text{mock}}$	3.3	≈ 1.0
$F5_{\text{mock}}$	1.5	≈ 0.5

Table 1. Results for mock LWIR data: Euclidean distance between the geometric center of the filled manually and automatically generated vessel contours (MC and AC respectively). The distance is calculated originally in pixels, and converted to an estimate of this distance in meters by observing the length of the vessel in pixels in each image while knowing its true length in meters. Notice the close overlap of these two geometric centers.

5.3 Results using real LWIR data

For the purpose of testing the network on real LWIR images we likewise select six images of different frigates seen in increasingly complex scenarios: from open sea situations with good contrast to situations with low contrast, land in the background or sea surface reflections. For simplicity we will refer to these six images as $F1_{\text{real}}$ through $F6_{\text{real}}$. Figure 8 illustrates results on the last type of scenario. This figure shows $F6_{\text{real}}$ in an open sea scenario but with sea surface reflections. The neural network predicts a relatively good probability map. Notice that the detection of the upper contour is good, whereas the lower part of the contour is less precise. This is due to the strong reflection of the vessel in the sea surface, apparent right below the hull. This is a problem appearing in some real LWIR images that is currently not mimicked in the IR look-alikes obtained from images acquired in the visual bands. The neural network is therefore poorly trained to handle such situations. We will address this problem in future work.

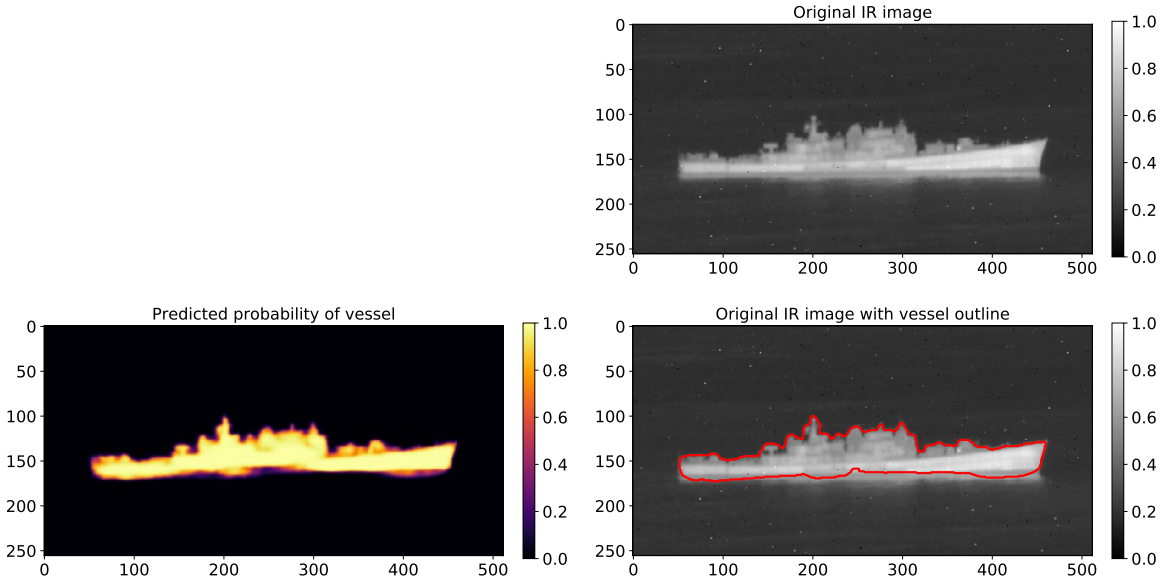

Figure 8. From top to bottom, left to right: $F6_{\text{real}}$ (decommissioned Norwegian frigate HNoMS Bergen), vessel probability map generated by the neural network, final outline (obtained by thresholding the probability map at $t = 0.5$.) The upper part of the vessel is well delineated whereas the delineation of the lower part is less good due to the strong sea reflection of the vessel. Notice also that the upper left part of this figure is left blank intentionally, in this case there is no original RGB image since this image originates as a true LWIR image.

Figure 9 shows the histograms of distances for the real LWIR images. From this figure it is clear that the performance of the network on real LWIR data remains comparable to the performance on mock LWIR data suggesting that the training is transferable from mock to real LWIR data. The worst result in this figure is for $F5_{\text{real}}$, where the distance at which 80 % or more of the pixels on the AC are within the MC is 3.7 pixels. The underlying image for $F5_{\text{real}}$ is of low contrast and with land background.

In Table 2 we summarize the distance, in pixels and estimated meters, between the geometric mean of the MC compared with that of the AC. Again, notice that the results for real LWIR data is comparable to the results for mock LWIR data.

Vessel	**Dist. [pixels]**	**Dist. [m]**
$F1_{\text{real}}$	1.6	≈ 0.4
$F2_{\text{real}}$	1.5	≈ 0.4
$F3_{\text{real}}$	1.7	≈ 0.5
$F4_{\text{real}}$	0.8	≈ 0.2
$F5_{\text{real}}$	7.2	≈ 2.2
$F6_{\text{real}}$	4.2	≈ 1.0

Table 2. Results for real LWIR data: Euclidean distance between the geometric center of the filled manually and automatically generated vessel contours. The distance is calculated originally in pixels, and converted to an estimate of this distance in meters by observing the length of the vessel in pixels in each image while knowing its true length in meters.

6. CONCLUSIONS

In the work reported here we have investigated how to train neural networks to produce contours of naval vessels when observed in LWIR images. Due to the scarcity of real LWIR images of such vessels we investigate how to use alternative data in order to train the network. We consider a number of approaches for this. Currently our selected approach is to artificially transform RGB images of such vessels into IR image look-alikes. We then train the network on these mock LWIR images and observe the network performance on real LWIR images.

Performance on the mock LWIR images is very promising as illustrated by the close approximation of the calculated contour to the real vessel outline. From observing Figures 7 and 9 and Tables 1 and 2 it is clear that performance on real LWIR images of naval frigates remains good indicating that what the neural network learned while training on mock LWIR data is transferable to real LWIR data.

We plan to further investigate how to make the best possible mock LWIR images, it is for instance clear that modeling vessel reflections in the sea surface is of value. Also, we will experiment with more advanced network architectures to achieve even more precise vessel outlines.

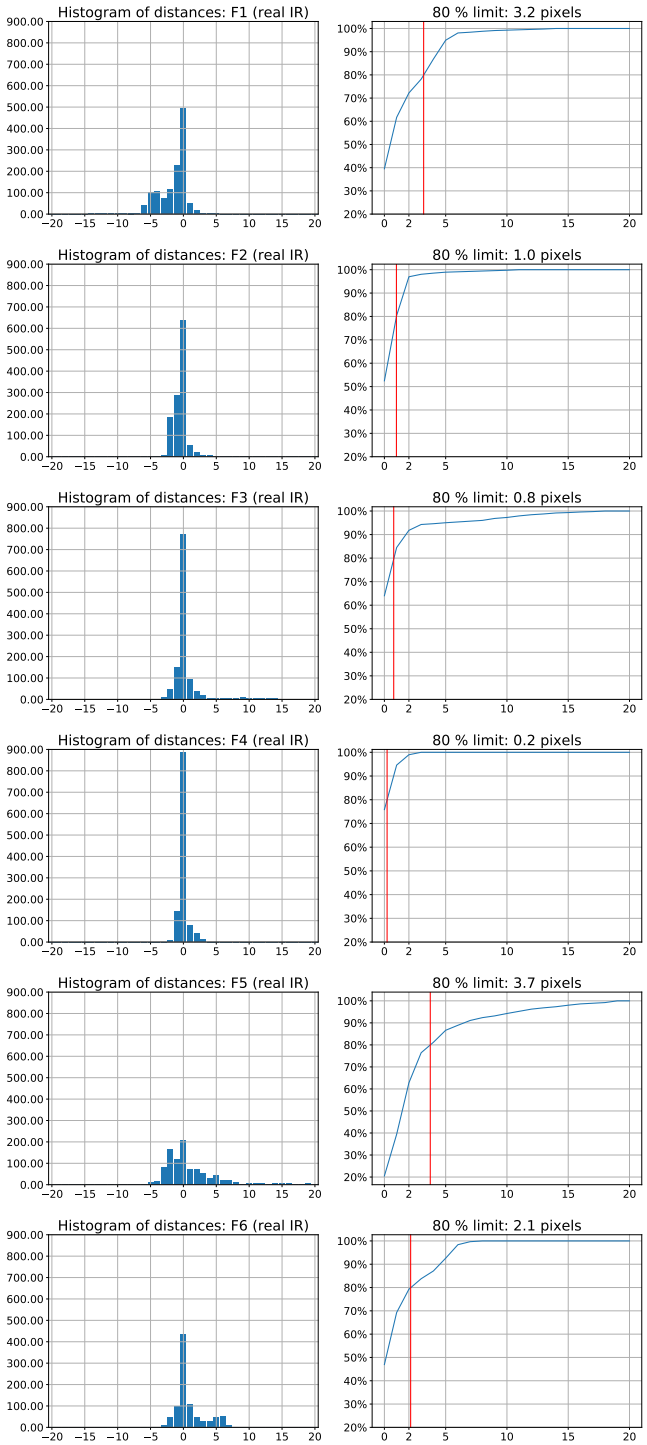

Figure 9. Results for real LWIR data: The left parts of this figure shows the distribution of distances (in pixels) from the automatically calculated contour (AC) to the closest point on the manually drawn contour (MC). Negative distances signify that the AC lies inside the MC. The right parts of these figures show the cumulative distribution of the absolute value of the distances from the AC to the MC along with the distance limit within which, starting at the MC, we will find 80% of the AC.

REFERENCES

[1] Kok, S., *Naval survivability and susceptibility reduction study-surface ship*, Master's thesis, Naval postgraduate school, Monterey, California, USA (2012).

[2] Gade, R. and Moeslund, T., "Thermal cameras and applications: A survey," *Machine vision and applications* **25**(1), 245–262 (2014).

[3] Ibrahim, S., "A comprehensive review on intelligent surveillance systems," *Communications in science and technology* **1**(1), 7–14 (2016).

[4] Budzier, H. and Gerlach, G., [*Thermal infrared sensors*], John Wiley and Sons Ltd., Hoboken, New Jersey, USA (2011).

[5] Goodfellow, I., Begio, Y., and Courville, A., [*Deep Learning*], The MIT Press, Cambridge, Massachusetts, USA (2016).

[6] Guo, Y., Liu, Y., Georgiou, T., and Lew, M., "A review of semantic segmentation using deep neural networks," *International journal of multimedia information retrieval* **7**(2), 87–93 (2018).

[7] Zvolanek, B. and Kessler, I., "Autonomous Ship Classification from Infrared Images," in [*IEEE Aerospace and Electronic Systems Society Conference*], 76–80 (1980).

[8] Zvolanek, B., "Autonomous ship classification by moment invariants," in [*SPIE 292, Processing of Images and Data from Optical Sensors*], 241–248 (1981).

[9] Casasent, D., Pauly, J., and Fetterly, D., "Infrared ship classification using a new moment pattern recognition concept," in [*SPIE 302, Infrared Technology for Target Detection and Classification*], **302**, 126–133 (1982).

[10] Luce, F. and Schaming, W., "Automatic classification of ship targets," *RCA Engineer* **31**, 18–23 (1986).

[11] Kato, D., Holben, D., Politopoulos, A., and Yin, B., "Ship classification and aimpoint maintenance," in [*SPIE 890, Infrared Systems and Components II*], **890**, 174–180 (1988).

[12] Zhongliang, Q. and Wenjun, W., "Automatic ship classification by superstructure moment invariants and two-stage classifier," in [*Singapore ICCS/ISITA '92*], 544–547 vol.2 (1992).

[13] Withagen, P., Schutte, K., Vossepoel, A., and Breuers, M., "Automatic classification of ships from infrared (FLIR) images," in [*AeroSense'99*], 180–187, International Society for Optics and Photonics (1999).

[14] Alves, J., Herman, J., and Rowe, N., "Robust recognition of ship types from an infrared silhouette," tech. rep., Defense Technical Information Center Document (2004).

[15] Li, H. and Wang, X., "Automatic recognition of ship types from infrared images using support vector machines," in [*International Conference on Computer Science and Software Engineering*], **6**, 483–486, IEEE (2008).

[16] Noroozi, M., Ramezani, A., and Aghababaee, M., "Automatic ship types classification in silhouette images," *International Journal of Engineering and Advanced Technoogy (IJEAT)* **4**(1), 52–56 (2014).

[17] Liu, Z., Zhou, F., Chen, X., Bai, X., and Sun, C., "Iterative infrared ship target segmentation based on multiple features," *Pattern Recognition* **47**, 2839–2852 (2014).

[18] Liu, Z., Bai, X., Sun, C., Zhou, F., and Li, Y., "Infrared ship target segmentation through integration of multiple feature maps," *Image Vision Comput.* **48**, 14–25 (Apr. 2016).

[19] Kass, M., Witkin, A., and Terzopoulos, D., "Snakes: Active contour models," *International journal of computer vision* **1**(4), 321–331 (1988).

[20] Ren, Z., "Variational level set method for two-stage image segmentation based on morphological gradient," *Mathematical Problems in Engineering* (2014).

[21] Zhang, R., "A gradient vector flow snake model using novel coefficients setting for infrared image segmentaion," in [*Proceedings of the 3rd International conference on material science and mechanical engineering*], 142–146 (2016).

[22] Zhang, R., Zhu, S., and Zhou, Q., "A novel gradient vector flow snake model based on convex function for infrared image segmentation," *Sensors* **16**(10), 1756 (2016).

[23] Fang, L., Zhao, W., Li, X., and Wang, X., "A convex active contour model driven by local entropy energy with applications to infrared ship target segmentation," *Optics Laser Technology* **96**, 166–175 (Nov. 2017).

[24] Koltun, V., "Efficient inference in fully connected crfs with gaussian edge potentials," *Adves in Neural Information Processing Systems* **24**, 109–117 (2011).

[25] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L., "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062* (2014).

[26] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. S., "Conditional random fields as recurrent neural networks," in [*Proc. IEEE Int. Conf. Computer Vision (ICCV)*], 1529–1537 (Dec. 2015).

[27] Noh, H., Hong, S., and Han, B., "Learning deconvolution network for semantic segmentation," in [*Proc. IEEE Int. Conf. Computer Vision (ICCV)*], 1520–1528 (Dec. 2015).

[28] Long, J., Shelhamer, E., and Darrell, T., "Fully convolutional networks for semantic segmentation," in [*Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*], 3431–3440 (June 2015).

[29] Ronneberger, O., Fischer, P., and Brox, T., "U-net: Convolutional networks for biomedical image segmentation," in [*Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*], 234–241, Springer International Publishing (2015).

[30] https://www.tensorflow.org/.

[31] https://keras.io/.