IMAGE PROCESSING IN LOBSTER FARMING – CLASSIFICATION OF JUVENILE LOBSTERS

Hjørdis Høimyr, Tormod Drengstig and Thor Ole Gulsrud

Department of Electrical and Computer Engineering,
Stavanger University College, N-4091 Stavanger, Norway.

ABSTRACT

The work presented in this paper is partly practical and partly theoretical. The purpose is to develop and implement algorithms to detect and classify juvenile lobsters. In addition a prototype which includes a camera, some containers, pipes and tubes, an autonomous controller unit, and a computer is built. The controller unit handles the detection of juveniles, and acts as an external trigger to the computer software which is then responsible for the process of capturing, processing and classifying each image. The controller unit may also be extended with a mechanism to physically separate the classified juveniles.

Figure 1: A typical juvenile at its 3rd stage.

Figure 2: A typical juvenile at its 4th stage.

1. INTRODUCTION

One of the activities in a lobster farm is the sorting of juvenile lobsters. All of the juveniles are transferred to a container, or incubator, where they molt three times. When the juveniles reaches the 4th stage, they are to be transferred to other units. Each juvenile is siphoned out of the incubator and into a small transparent rectangular chamber, where an image snapshot is captured. The image is analyzed, and if it is a 4th stage juvenile it is transferred to the next unit in the lobster farm. If not, it is brought back to its siblings in the incubator.

Juvenile lobsters that have reached the 4th stage have a size that clearly separates them from the earlier stages. Intuitively, this yields the length as a discriminating feature. Likewise, the area that is covered by a juvenile in the acquired images is another discriminating feature. Keeping it simple, it is shown that these two features are the only ones needed to classify a juvenile lobster as being 4th stage or not.

A typical 3rd stage juvenile is shown in Figure 1, whereas Figure 2 shows a 4th stage juvenile. Notice the difference in size and placement of the claws. A 3rd stage juvenile cannot stretch its claws in the same way as the 4th stage juvenile is able to do.

The prototype is built to accomplish the task of capturing an image of each juvenile. The image is then analyzed, processed and segmented, and finally, sent through a back propagation feed forward neural network trained to do the classification of the juvenile lobster.

Two hatches of juvenile lobsters are used to make the training set and the test set of the classifier. The weight of the mother of the first hatch, which gives the training set, is 750 gram. The weight of the mother for the second hatch, which gives the test set, is about 1 kilogram. This is important for two reasons. First, the juveniles ability to survive depends on their mother's age, and hence, weight. Second, and most important in this work, the juveniles size at the different stages depends on their mother's weight. This fact makes it necessary to adjust the decision boarder for each new hatch of juveniles based on the weight of that particular mother.

A small number of different types of code has been developed in Microsoft®Visual Basic and MATLAB™ to facilitate the necessary processing tasks. The graphical user interface (GUI) shown in Figure 3 is used solely for acquiring images. The GUI shown in Figure 4 is used to extract the discriminating features and manually classify each juvenile of the training hatch to make the training set for the neural network. Yet another GUI is used to accomplish classification in real time, and is shown in Figure 5.

Figure 3: A GUI used to capture images of the juveniles. The juvenile shown is at its 4th stage.

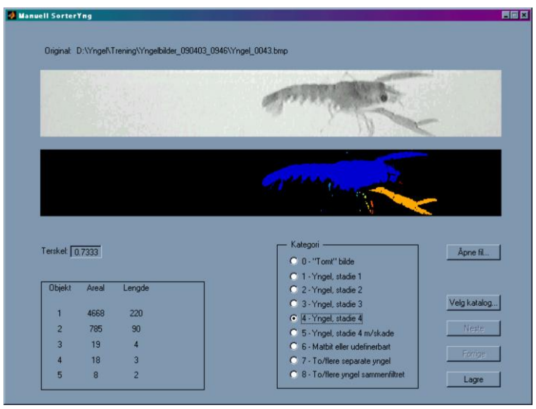

Figure 4: A MATLAB GUI used to manually classify the juveniles. The juvenile shown is at its 4th stage.

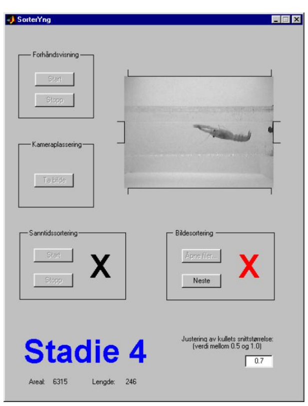

Figure 5: A MATLAB GUI used to classify juveniles in real time. The juvenile shown is at its 4th stage.

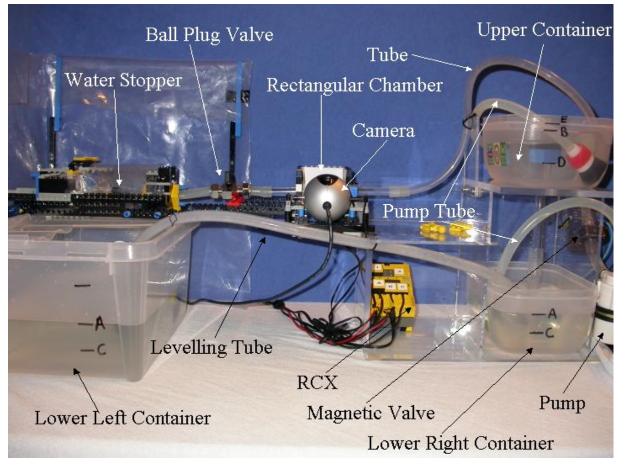

Figure 6: The prototype. Photo by Anna Koltveit

2. THE PROTOTYPE

The functioning of the prototype is essential for the final results of the work. Much effort is therefore put into the construction of the prototype. A picture of the prototype is shown in Figure 6. A siphon is used to keep the water in each of the lower containers at a stable level. Another siphon is used to transfer the juveniles from the **upper container** through the **rectangular chamber** and into the **lower left container**. The **water stopper** is used to stop the flow of water when the image is captured. This is necessary because the **camera** used has a quite slow reaction of the trigging. The **ball plug valve** installed between the rectangular chamber and the lower left container is simply used to stop the flow of water without having to empty the **tube**. The **pump** at the right side of the prototype pumps water from the **lower right container** and into the upper container. The **magnetic valve** behind the pump is used to stop the water from returning when the pump is stopped.

A LEGO® MINDSTORMS™ RCX™is used to control the prototype. The RCX is a programmable LEGO brick and an ideal controller unit because it makes it possible to implement functionality into the prototype that otherwise would have made the prototype far to complex for this work. A detailed description of the prototype can be found in [4]. Further information on the RCX can be found in [1] and at [5].

2.1. How Things Work

After water has been filled in all of the containers and tubes and the pump has been started, the juveniles are poured into the upper container. The ball plug valve is opened and each juvenile is siphoned out of the upper container through the tube and into the small rectangular chamber where an image snapshot is captured. A LEGO light sensor is placed in front of the chamber to detect the juvenile. The sensor is connected to the RCX which starts the motor of the water stopper mechanism and outputs a signal to the computers serial interface. This signal acts as an external trigger for the software responsible for capturing the image.

After the image is captured it is analyzed and its features are extracted and finally, it is classified. The result of the final classification gives one of two possibilities. Either the juvenile is a 4th stage juvenile or it is not. The classification result is presented in the GUI.

3. IMAGE PROCESSING AND CLASSIFICATION

The overall process, from capturing to final classification, is shown in Figure 7. Each of the indicated steps are briefly discussed below.

3.1. Preprocessing and Segmentation

The necessary tasks of the **preprocessing** step is converting the original RGB image to a 8-bit gray level image and choose the region of interest (ROI) of the image. The ROI in each juvenile image is proportionally equal in size to the inside area of the rectangular chamber, which is 6×44 millimeters. The web camera used in the prototype does not support hardware choice of ROI, so this is accomplished in software. The images are originally 480×640×3 pixels RGB images converted to 8 bit 88×613 pixels gray level images.

The light conditions in and around the rectangular chamber and hence, the contrast in the images are invariant. In the **segmentation** step a global threshold value is calculated for each image and then used to determine the connected objects in the image. Each connected object's pixels is given an integer label. These methods are well known in image processing, and are not discussed further here. A detailed theoretical description can be found in [3] and for practical use [6] is recommended.

3.2. Feature Extraction

After preprocessing and segmentation, the image, which is then proportionally equal in size to the rectangular chamber, contains labels for the connected objects in the image. In the **feature extraction** step the discriminating features, area and length, are extracted for each object. The area is simply the number of pixels the object covers in the image. The length is the horizontal width of the object's smallest possible bounding rectangle. These methods are fully described in [6].

3.2.1. Coarse Classification

The calculated area and length of the two largest objects are sent to a coarse classification algorithm which rejects images containing more than one juvenile and most of the images containing 3rd stage and smaller juveniles. This is indicated in Figure 7 by the second output ($k = 0$) of the feature extraction step.

In some of the images, one, and in rare cases both, of the 4th stage juvenile's claws are interpreted as being separate objects, as shown in Figure 4. If this is the case, the coarse classification algorithm adds the separate claw's area to the juvenile's area before the final classification of the juvenile. This extra preprocessing clearly reduces the possibility of 4th stage juveniles not being correctly classified.

3.3. Classification

Images that have not been rejected by the coarse classification algorithm, are sent to a back propagation feed forward neural network trained to do the final **classification**. The theoretical background material of neural networks is thoroughly described in [2] and [7] and are not discussed further here.

Input to the neural network is the non-rejected output vector of the feature extractor, containing the calculated area and length of the juvenile. The scalar output from the network is used to make the final decision of the juvenile. If a is the scalar output from the network, and t is the threshold calculated from the weight of the mother, then the final decision function is given as

$$k = \begin{cases} 0 & \text{when } a \leq t, \\ 1 & \text{when } a > t. \end{cases}$$

The threshold value, t, will vary with each hatch of juvenile lobsters depending of the weight of the mother. It's outside the scope of this work to find the correlation between the weight of the mother and the size of the juveniles, and for the time being, t must be set manually.

4. RESULTS

In Figure 8 a plot of the training set together with the decision boarders for $t = 0.5$ (leftmost) and $t = 1.0$ (rightmost) are shown. Each circle (o) marks a juvenile at the 4th stage, whereas each cross (+) marks a juvenile that is not at the 4th stage, or a juvenile at the 4th stage that is missing one or both of its claws. [8] states that the survival rate of the juveniles with both claws missing is less then 56%, whilst the survival rate of the juveniles with one missing claw is 80%, and the survival rate of juveniles with both claws intact is 82.5%. So the solution is to accept juveniles

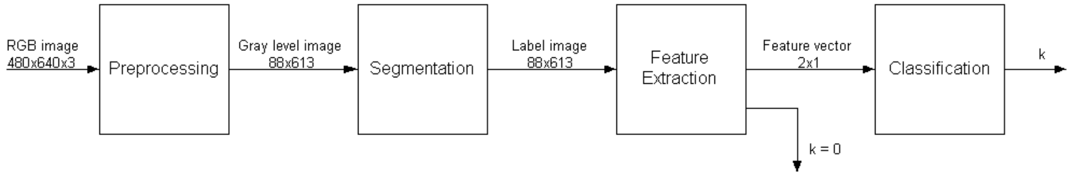

Figure 7: The overall process from the original RGB image until the final classification where k has been given the value of either 0 or 1.

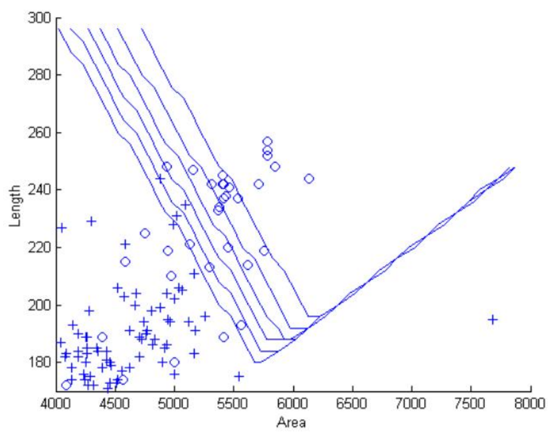

Figure 8: A plot of the training set together with the decision boarders for $t = 0.5$ (leftmost) and $t = 1.0$ (rightmost). Each circle (o) marks a juvenile at the 4th stage, whereas each cross (+) marks a juvenile that is not at the 4th stage, or a juvenile at the 4th stage but missing one or two of its claws.

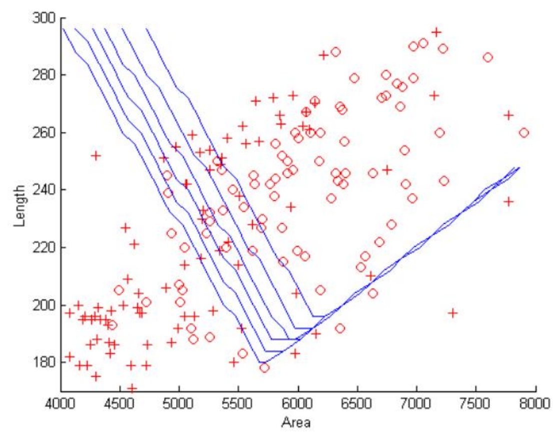

Figure 9: A plot of the test set together with the decision boarders for $t = 0.5$ (leftmost) and $t = 1.0$ (rightmost). Each circle (o) marks a juvenile at the 4th stage, whereas each cross (+) marks a juvenile that is not at the 4th stage, or a juvenile at the 4th stage but missing one or two of its claws.

with one claw missing, but reject juveniles with both claws missing. Since the initial idea of rejecting all of 4th stage juveniles with missing claws, is an all to harsh demand, some of the rejected juveniles marked with a cross (+) should instead have been accepted and marked with a circle (o).

This is also true for the plot of the test set, which is shown in Figure 9, and here the consequences of the initially harsh demand is clearly seen.

5. CONCLUSIONS

The work presented in this paper has shown that the task of sorting 3rd stage from 4th stage lobster juveniles in real time is successfully accomplished in a prototype by the use of a camera and image processing routines. The discriminating features used are length and area of the lobsters.

6. REFERENCES

[1] Dave Baum. *Dave Baum's Definitive Guide to Lego® Mindstorms™*. APress, 2000.

[2] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, second edition, 2001.

[3] Rafael C. Gonzales and Richard E. Woods. *Digital Image Processing*. Addison Wesley, second edition, 2002.

[4] Hjørdis Høimyr. Bildebehandling i hummeroppdrett. Sortering av hummeryngel. Hovedoppgave. Institutt for elektroteknikk og databehandling, Høgskolen i Stavanger, 2003.

[5] The LEGO Group. LEGO MINDSTORMS. http://www.legomindstorms.com.

[6] The MathWorks, Inc. *Image Processing Toolbox User's Guide*, 2001.

[7] Jürgen Schürmann. *Pattern Classification*. Wiley-Interscience, 1996.

[8] Ingebrigt Uglem. *Håndbok i hummeryngeloppdrett*. Havforskningsinstituttet, Bergen, Norway, 1995.