Segmentation and graph generation of muzzle images for cattle identification

Lucas Wojcik
Federal University of Paraná
Email: lmlw19@inf.ufpr.br

Jorge Junior
Federal University of Paraná
Email: uni@aqua.rip

David Menotti
Federal University of Paraná
Email: menotti@inf.ufpr.br

João Hill
Institute of Rural Development of Paraná
Email: joaohill@idr.pr.gov.br

***Abstract*—The current methods for the organizing the records (i.e., cataloguing) of cattle are known to be archaic and inefficient, and often harmful to the animal. Such methods include the use of metal tags attached to the animal's ears like earrings and of branding irons on their necks. Previous research on new methods of livestock branding based on computer vision techniques utilized a mixture of texture features such as Gabor Filters and Local Binary Pattern as a means of extracting identifying features for each animal. The presented approach proposes a new technique using the muzzle image as an individual identifier as a novel technique, assuming that the muzzle RoI taken as input for the model pipeline is already extracted and cropped. This task is performed in three steps. First, the muzzle image is segmented via a convolutional neural network, resulting in a bitmap from which a graph structure is extracted in the second phase. The final phase consists of matching the resulting graph with the ones previously extracted and stored in the database for an optimal match. The results for the segmentation quality show a fidelity of around seventy percent, while the extracted graph perfectly represents the extracted bitmap. The matching algorithm is currently in progress.**

I. Introduction

The branding of livestock is of vital importance in the agribusiness, both for inner farm organization and due to legal matters. It is used not only to indicate the owner of the animal, but also as a means to identify each individual in the group, making sure there are no missing or extra animals. It is also important to track the location of each animal, to make sure it has not stepped out of bounds or got lost. Some special marks are also commonly used to indicate that a given animal has been correctly vaccinated.

In the last scenario, the marking is often done using a hot iron, where the animal is tied and held down and receives the mark in its neck through a heated piece of iron in the correct shape, a painful experience for the animal. Another common technique is the use of metal tags on the animal's ears, a sort of earring containing written, graphical or electronic information to correctly identify the animal. The main problem with this approach is that the earring can be easily lost or forged, defeating its purpose due to inefficacy.

Thus, using photos of the animals as the input for the identifier, it is possible to create a salutary and non-intrusive alternative. Computer vision techniques have been used in a myriad of scenarios, and are very popular in the problem of image identification [1], [2]. In the past years, research has been conducted to create an identification system that serves as a livestock cataloguer utilizing said techniques, such as in Kumar & Singh [3], where they presented a SIFT-based feature extraction and matching system, also previously explored by Noviyanto & Arymurthy [4], who also developed a different method using SURF features [5], and texture-based feature extractors such as Gabor filters and Local Binary Pattern histogram generator as described by Kusakunniran & Chaiviroonjaroen [6]. We can also cite the approach proposed by Gaber et al., using the Weber Local Descriptor [7].

According to our testing, the techniques using texture features have proven to be inconsistent in multi-session scenarios, that is, when the training data images were taken on an occasion distinct from the testing data. This is mainly due to the changes in lighting and other residues that drastically change from one day to another. Since the biometry does not change in a short time span, a method based on the muzzle patterns should be robust to cross-session scenarios. Furthermore, a study [8] has found that the muzzle biometry appears to suffice as an individual identifier. The same study presents a simple method for taking the nose prints that can be easily adapted for taking pictures, consisting of one person holding the animal's head while it is locked in the stanchion, while another person takes the picture.

A novel method based on the biometry of bovine animals (particularly cows) is proposed in this paper. The method follows the following structure: first, the RoI (region of interest) consisting of a muzzle image of a given animal is segmented into a bitmap, where each pixel indicates whether that position on the original image is part of a bead (zero) or a ridge (one). Then, the outputted bitmap is used to create a graph structure that uniquely identifies the given animal. Finally, the obtained graph is used to compute the similarity of the input biometry between all known animals, searching for the optimal match. The state of the inputs and outputs at each phase is illustrated in Fig. 1. To the best of our knowledge, the presented approach has not yet been implemented anywhere.

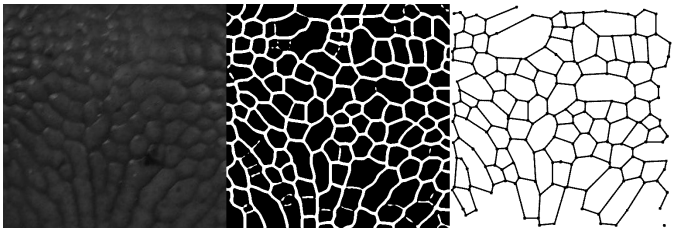

Fig. 1. The cropped RoI in the left, the segmentation map in the middle and the extracted graph in the right.

II. Proposed Method

The framework for the model is illustrated in Fig. 2. The pipeline takes as input the RoI cropped out of a muzzle image and outputs the predicted classification for that animal. The RoI detection and extraction is done through the YOLOv5 algorithm, which was trained to detect the nostrils and mouth of bovines. From these detections it is possible to extract the middle square as the RoI. The three intermediate states correspond to the phases of feature extraction (via the segmentation followed by the graph extraction) and matching (through a graph matching algorithm).

A. Segmentation and bitmap generation

The first module of the pipeline consists of a machine learning model for image segmentation. Two existing convolutional neural networks were evaluated for this task, and their results compared. The first is Ronneberger's U-Net [9] and the second is Liu et al's PoolNet [10].

The U-Net model consists of two symmetric paths, one consisting of two 3×3 convolution and a ReLU (Rectified Linear Unit) followed by a 2×2 pooling operation (downsampling the image by a factor of two), and the other one concatenating the current segmentation map with the corresponding output from the first path, and replacing the pooling operation with a 2×2 up-convolution layer (upsampling the image by a factor of two).

The PoolNet model also consists of two symmetric paths, but the output from the final pooling step in the end of the first half is also fused to the feature maps of all of the layers in the second half. The result of that fusion is then converted into multiple feature spaces (with the idea of capturing local context information at different scales), which are then combined back by means of pooling. The main point is to lessen the dilution of semantic information that happens as it is progressively transmitted back through the expansion layers.

Considering that the output of the thresholding step often suffered from incomplete lines (as can be seen in Fig. 3), experiments were also made with applying Watershed [11] to the distance transform of that output. The Watershed algorithm uses regional minima as seeds for segmenting regions, by "flooding" the minima and building watersheds in the places where "water" from different regions would merge. The main idea is to separate incorrectly merged regions, such as most of those where incomplete lines were present.

The models are trained with the same dataset, which consists of twenty-four manually annotated pairs of images (cropped RoIs) and ground truth labels divided into two subsets of twelve pairs labelled A and B. The label for a given RoI is a grayscale image where each pixel is classified as black, white, or gray. The black and white classes correspond to the beads and ridges, while the gray class denotes a region of uncertainty. The models are trained in two distinct modalities, one where the gray pixels are considered as beads (negative, or gray as black) and another where the gray class maps to the ridges (positive, or gray as white).

The output of both models upon evaluation consists of a real valued image, which can be interpreted as a grayscale one, corresponding to the segmentation map of the tested RoI.

The bitmap is acquired from the outputted segmentation map using an adaptive thresholding operation with the block size parameter set to 99 [12]. The resulting bitmap is then skeletonized using the Zhang-Suen thinning algorithm [13]. The results of each of these operations is shown in Fig. 3.

B. Graph extraction

Finally, a graph is extracted from the skeleton obtained in the last step. The algorithm is divided in two parts. First, the vertices are identified, and after that, each path between vertices is computed and the result is compiled as a graph structure.

In the context of biometrics, the vertices will be defined as the points where three or more ridges meet. So in order to identify the vertices in the bitmap, it is necessary to identify every white pixel that has three or more white neighbours. These vertices are uniquely defined by a tuple containing its geometrical coordinates in the image, i.e., x and y. After that, the algorithm iterates through all of the vertices, following the paths that protrude from it until reaching either another vertex, in which case we consider that there is an edge connecting the two, or a dead end, in which case no edge is generated as the immediate borders of the image are not taken into account.

At the end of the algorithm, the vertices and edges identified represent the biometry of the animal whose RoI was segmented at the start. This is the graph that will be used in order to uniquely identify the bovine, and therefore also the structure that will be used in order to perform the matching between known animals. Fig. 4 illustrates the graph obtained from the bitmap generated at the last step.

C. Matching

The matching part of the proposed methods then utilizes the extracted graph for the task. In particular, the Elastic Bunch Graph Matching algorithm [14] will be implemented to wrap it all up. In general terms, a bunch graph will be created for every known individual, containing the features extracted from various images of the same animal. A new image will then be elastically fitted (moved and re-scaled, searching for the optimal transform) to the known animal bunch models, and a graph similarity measure computed with features extracted from the new image in the best fitted bunch graph vertices.

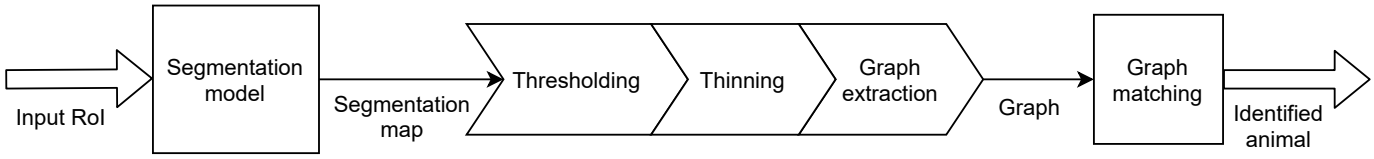

Fig. 2. The entire pipeline of the proposed method.

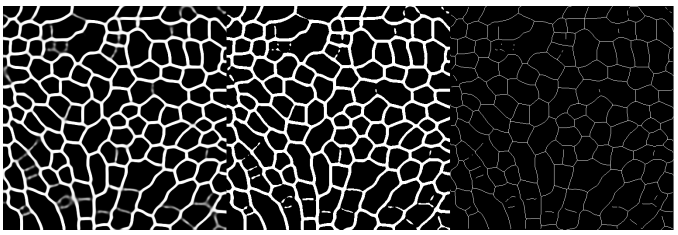

Fig. 3. Left: the output segmentation map, middle: the thresholded bitmap, right: the skeleton.

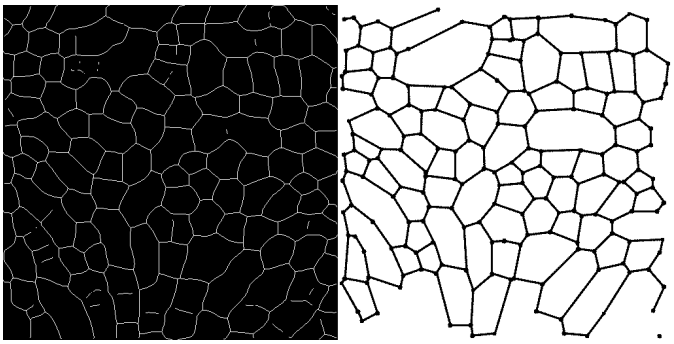

Fig. 4. The prediction skeleton on the left and the graph extracted from it on the right.

III. Experiments and results

For the segmentation part of the pipeline, two disjoint sets of twelve muzzle images and manually annotated label pairs, named A and B were used as training and testing data in alternated cycles. For the refinement of the annotations, a third class was introduced. It designates a region of uncertainty in the original image, and is represented with the gray color.

The training and testing cycles consisted of four rounds. First, the models were trained on the A set, with the gray class set to zero (black) and tested on the B set. On the second round, B was the training set and A the testing set, again with gray set to zero. The third and fourth rounds were equivalent to the first two, but the gray class was set to white. On every round, U-Net training ran for 6 epochs on average with 300 steps (with 2 epochs of tolerance for early stopping), and the PoolNet model was trained for 3000 epochs of 1 step. The watershed technique was used on the outputs of both networks, and the results with and without it were evaluated with a novel metric described in the next two paragraphs.

Since the goal for the segmentation is to create a graph representation as faithful as possible to the ground truth label, we define the accuracy in terms of the recognized regions (beads) and ridges. A region is correctly segmented if the model prediction encloses the same region on all sides by the ridges in the same way that the ground truth does so. Therefore we define two distinct kinds of error: under and over segmentation, where ridges are wrongfully absent and present, respectively. Border regions are ignored.

Given a ground truth label and its prediction, we then evaluate the representation fidelity by iterating through the regions of each, mapping every region on the ground truth to the region with highest IoU (intersection over union) on the prediction, and vice versa. The under segmentation error is identified when more than one region on the ground truth is mapped to a single region on the prediction, and the over segmentation error is identified when more than one region on the prediction is mapped to a single region on the ground truth. Fig. 5 illustrates the correspondence between a ground truth annotation and the corresponding section of a U-Net prediction. Table I presents the average of the percentage of regions in each class in every scenario, for all images in the testing set.

Evaluation on segmentation and watershed averaged below a quarter of a second for both models. The graph extraction took around five seconds per image, all on an 8th gen i5 intel processor.

The matching experiments will be done by matching the testing images one-to-all in the training sets. An optimization for the final version will be matching at first a few of the central vertices in order to discard bad matches faster, and so only compute the full similarity measure in a reduced set.

TABLE I
Performance of models according to the described metric. Neg is gray as black, Pos is grey as white. WS indicates Watershed usage.

	Train: A, Test: B			Train: B, Test: A		
Model	Good	Over	Under	Good	Over	Under
U-Net_Neg	72.4	6.7	20.9	72.3	24.7	3.0
U-Net_Pos	71.7	20.1	8.2	69.6	25.1	5.3
PoolNet, Neg	67.7	22.4	9.9	61.9	34.9	3.2
PoolNet, Pos	44.2	54.6	1.2	39.3	60.6	0.1
U-net + WS, Neg	62.8	35.1	2.1	75.9	18.6	5.5
U-net + WS, Pos	73.8	20.1	6.1	76.5	14.1	9.4
PoolNet + WS, Neg	71.1	25.1	3.8	75.1	21.1	3.8
PoolNet + WS, Pos	73.3	6.2	20.5	72.5	5.9	21.6

In terms of the graph extraction, a binary image containing the resulting graph is generated, such that the vertices are in the same position as the output segmentation or ground truth. We evaluate the fidelity of the graph regarding the prediction or ground truth label from which it was extracted. The algo-

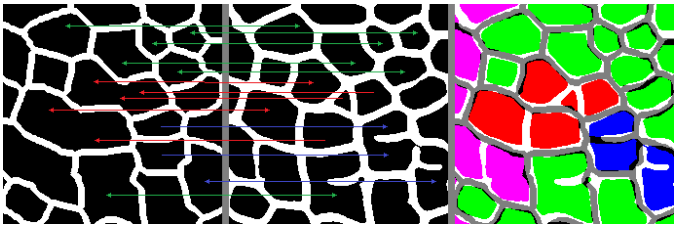

Fig. 5. A section with the ground truth on the left and prediction in the middle. The right image is the superposition of both, color-coded: red is oversegmented, blue is undersegmented, green is ideal. Pink is a border region (which we ignore). White is a false positive (pixel wrongly classified as ridge, black is a false negative and grey is a hit.)

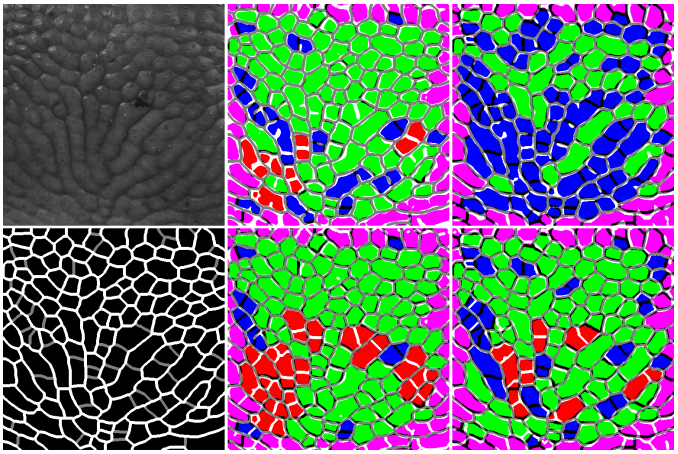

Fig. 6. Visual representation of segmentation results on a single image from set A for both models for round two (grey as white, training on B and evaluating on A). Top left: Original image. Top center: U-net. Top right: PoolNet. Bottom left: Ground truth. Bottom center: U-net result into Watershed. Bottom right: PoolNet result into Watershed.

rithm is the same as the one that creates the correspondence between two bitmaps as described above, differing only in the metric that is used to map the region of one bitmap into the corresponding region in the second. Here, instead of IoU, the algorithm takes the centroid of the region to be mapped in the query image, and maps it to the region that contains the centroid pixel in the target image. For all 24 predictions of both PoolNet and U-Net in all modalities, the accuracy in this stage reached the ideal 100%.

IV. Conclusion

Given the results achieved, we conclude that the most efficient developed approach for the segmentation task is the one using the PoolNet model followed by the Watershed algorithm. The achieved efficiency of over 70% seems to result in a satisfactory representation. However, cross-session robustness of the representation is yet to be proven in future work, that is, when the described pipeline is fully implemented.

Acknowledgment

The authors would like to thank the Institute of Rural Development of Paranáfor all the support, especially for providing a large amount of bovine images and zoological expertise. The NVIDIA GTX Titan X used for this research was donated by the NVIDIA Corporation.

References

[1] T. Burghardt, N. Campbell, P. J. Barham, I. C. Cuthill, and R. Sherley, "A fully automated computer vision system for the biometric identification of african penguins (spheniscus demersus) on robben island," in *6th International Penguin Conference (IPC07)*. University of Tasmania, Australia, 2007.

[2] D.-H. Jang, K.-S. Kwon, J.-K. Kim, K.-Y. Yang, and J.-B. Kim, "Dog identification method based on muzzle pattern image," *Applied Sciences*, vol. 10, no. 24, p. 8994, 2020.

[3] S. Kumar and S. Singh, "Automatic identification of cattle using muzzle point pattern: a hybrid feature extraction and classification paradigm," *Multimedia Tools and Applications*, vol. 76, pp. 1–30, 12 2017.

[4] A. Noviyanto and A. M. Arymurthy, "Beef cattle identification based on muzzle pattern using a matching refinement technique in the sift method," *Computers and Electronics in Agriculture*, vol. 99, pp. 77–84, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169913002093

[5] A. Noviyanto and A. Arymurthy, "Automatic cattle identification based on muzzle photo using speed-up robust features approach," *Proceedings of the 3rd European Conference of Computer Science*, pp. 110–114, 01 2012.

[6] W. Kusakunniran and T. Chaiviroonjaroen, "Automatic cattle identification based on multi-channel lbp on muzzle images," in *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*, 2018, pp. 1–5.

[7] T. Gaber, A. Tharwat, A. E. Hassanien, and V. Snasel, "Biometric cattle identification approach based on weber's local descriptor and adaboost classifier," *Computers and Electronics in Agriculture*, vol. 122, pp. 55–66, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016816991600003X

[8] W. Pedersen, "The identification of the bovine by means of nose prints," *J. Dairy Sci*, vol. 5, p. 249–258.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[10] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *IEEE CVPR*, 2019.

[11] S. Beucher and F. Meyer, *The Morphological Approach to Segmentation: The Watershed Transformation*. Marcel Dekker Inc., 01 1993, vol. Vol. 34, p. 433–481.

[12] D. Bradley and G. Roth, "Adaptive thresholding using the integral image," *Journal of graphics tools*, vol. 12, no. 2, pp. 13–21, 2007.

[13] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[14] D. S. Bolme, "Elastic bunch graph matching," Ph.D. dissertation, Colorado State University, 2003.