Sensor Applications

# A Vision-Based Cattle Recognition System using Tensor-Flow for Livestock Water Intake Monitoring

Amin Biglari[1] and Wei Tang[1*]

[1]Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM, 88001, USA

[*]Member, IEEE

Manuscript received June 7, 2017; revised June 21, 2017; accepted July 6, 2017. Date of publication July 12, 2017; date of current version July 12, 2017.

Abstract—This paper presents a new method for identifying individual cattle for the purpose of tracking and efficiently measuring their daily water intake using a vision-based machine-learning system. In addition to the current solution of using Radio Frequency Identification (RFID), the proposed system uses TensorFlow Object Detection to detect labels on the RFID tag. The proposed system can be integrated into the current water intake monitoring system and alleviate the errors introduced by the unreliable RFID readers. The system allows users to train an object recognition model that can recognize and differentiate the labels on the ear tags of individual farm animals, so the drinking events can be recorded by the water intake monitoring system. The models are trained using custom data sets of manually annotated tag images with pre-trained model architectures from TensorFlow 2 Model Zoo. The system is tested using images from event-triggered weather-proof cameras deployed in the grazing site. Experimental results of the system showed an accuracy of around 90% In comparison to other present methods, this newly proposed system provides scalability and flexibility making it an attractive vision based solution for machine learning systems in agriculture.

Index Terms—Tensor Flow, Computer Vision, Machine Learning, Water Management, Object Detection, Agriculture.

## I. INTRODUCTION

Water is the most important resource and nutrient for ruminants in animal agriculture [1], which consumes 30% of the overall water in food production [2]. In order to improve the efficiency of water usage, especially in drought areas, technologies in water management have been introduced to track the water usage of individual cattle. Such technology can help track water consumption by each cow in the grazing area, and therefore be able to monitor the drinking pattern and behavior which is linked to growth and health issues. As shown in Fig. 1, a water intake monitoring system applies microchips implanted into the animals' skin and uses Radio Frequency Identification (RFID) or Electric Identification (EID) technologies to identify individual cattle. However, such systems are not quite reliable due to the detection range and weather conditions. Therefore, other technologies can also be added to improve the reliability and accuracy of the system. For example, a smart water intake monitoring system in [3] is equipped with a motion sensor and a weather-sealed camera. When a cow is presented at the drinking station, the motion sensor triggers the camera, which takes a few pictures to confirm the drinking event. In order to improve the detection capabilities of the system even further, this paper presents a method for visual identification of cows via their ear tags using object recognition

In order to process images using vision-based identification systems, several methods have been proposed with deep-learning technologies. For example, [4] presented an automatic cattle identification system using multi-channel local binary pattern (LBP) on muzzle images, which identifies individual cattle using the unique biometric pattern presented on their muzzle. Another example of vision-based

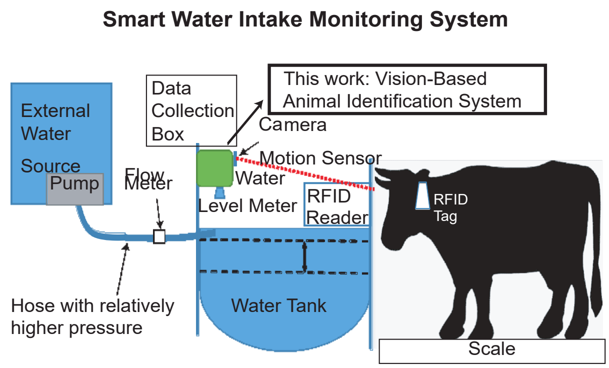


Fig. 1: Water Intake Monitoring System [3].

identification uses a local binary pattern descriptor to perform face recognition of cattle [5]. These systems show advancements in the utility of image recognition and computer vision for animal identification in agriculture. However, there are several challenges in the vision-based identification system. First, the image quality varies based on the environment and the device limitations, which may require high-resolution cameras. Second, for each new cattle, the model needs to be retrained since each animal has unique biometric features, which limits the scalability of the system. Therefore, a new vision-based method is expected to provide a robust classification of images while reducing the efforts in training models for new subjects. In this work, we propose a novel vision-based recognition method that focuses on the RFID tag numbers instead of the muzzle or face of the animal. This is because when the cow is drinking water, the RFID tag numbers are usually much easier to be identified. Another advantage of this method is that a group of RFID tag numbers can be trained in prior and then assigned to individual cattle. Therefore a new cattle can use the same tag as a prior cattle, which increases the scalability of the system and saves time for retraining the model.

Corresponding author: Wei Tang (e-mail: wtang@nmsu.edu).
Associate Editor: Alan Smithee.
Digital Object Identifier 10.1109/LSENS.2017.0000000

Moreover, field testing demonstrates that the accuracy of identifying tag numbers in the image is better than the face or muzzle of the cows under different environmental conditions. Images of ear tags do not need to have a very high resolution for the model to be able to detect them. Our system is designed based on Google TensorFlow Object Detection Application Programming Interface [6]. The system can be implemented in combination with the water intake monitoring system through the addition of a embedded computing system to handle tag detection. There are many systems that can fulfill this task, such as google Coral Edge TPU, the NVIDIA Jetson series, and Raspberry Pi devices. We speculate the overall cost of this system to be between $130.00-$150.00 for implementing the system onto a Raspberry Pi and anything images using an Adafruit weatherproof camera. In combination of the rest of the system including sensors, controllers, and automatic drinking systems, the overall cost of this method for potential customers would be around $2000.00. In the remaining of this paper: Section II describes the system design and training process; Section III presents experimental results; Section IV concludes this paper and proposes future works.

## II. SYSTEM DESIGN AND TRAINING

The proposed system operates with the following procedures. First, the image dataset is collected and then expanded using PyTorch image augmentation, which enhances the dataset by generating cropped and color-converted copies of each image. Next, the RFID ear tags in the image dataset are manually annotated using LabelIMG [7]. These annotations are then saved in PASCAL VOC format as xml files alongside their corresponding images. Afterward, a pre-trained model from the “TensorFlow 2 Detection Model Zoo" [8] is selected for initializing the custom mode. After that, the model is trained for a set amount of iterations and finally evaluated. Once the model has been trained, the checkpoint with the highest detection accuracy and recall is selected and used for tag detection. All of these steps are based on the dataset generation and training code provided in the TensorFlow Object Detection GitHub repository [9].


Fig. 2: Data collection and processing procedures of the proposed system.

The performance of the proposed system is evaluated using four custom models trained with two pre-trained models from the “TensorFlow 2 model zoo" [8] which are trained with hundred of thousands of images and 90 image classes models. The custom models contain five classes representing five tags used for training and analysis of the system. Two datasets were used to compare the training and overall accuracy of Tensor Flow API for this particular application. The models are trained by each model-zoo model and dataset once. The data collection and operating procedures of the proposed system are summarized in Fig. 2. The following context in this section describes details of each step.

### A. Dataset Image Collection

The images used for the training were taken using a Logitech C922 Pro HD Stream Webcam. They were all collected horizontally with an aspect ratio of 640 x 480 and a horizontal and vertical resolution of 96 dpi. Two different datasets were used for testing this method: a small dataset, containing only images taken from the webcam, and a large dataset that also held images from the image augmentation step. 30 images of five different tags were taken this way to test the method proposed in the paper, making the final size of the small dataset 150 images. The tags were “8526", “8527", “8528", “8529", and “8530". The images were upscaled to 640 x 640 during training. More images, as well as images of new tags, are being taken to expand the dataset even further for future experimentation. For the implementation demonstration, the tags “910" and “949" were used. Note that the tags are recognized as an image instead of the combination of individual numbers.

The large dataset was generated using Pytorch image augmentation, this resulted in cropped copies of the smaller dataset images with different hues, saturation, brightness, and contrasts. Initially, 4 images were generated per 1 image in the small dataset, however, not all of these images were in a usable condition as due to the cropping some did not contain the tags within them. Overall, the expanded dataset set contained a total of 414 images. Since the annotated tag objects in these images were much more contrasting and varied than the ones in the small dataset, as well as the far longer training time for the large dataset models, we also believed that it would be beneficial to model’s overall performance to increase the relative size of the testing to that of the training, as larger testing files allow for better assessment of machine learning model accuracy.

### B. Image Annotation

The tags within the dataset images were annotated manually using LabelIMG [7]. These annotations are saved xml files in PASCAL VOC format and are named identical to their respective image files, which allows LabelIMG to automatically tag the annotations to the right image. While manual annotating would reduce the speed at which a dataset is tagged, it allows for users to have more oversight over their training data. An example of an annotated image from the small dataset is provided in Fig. 3. 80% of the generated datasets are used for training, and 20% of data are used for testing the models.

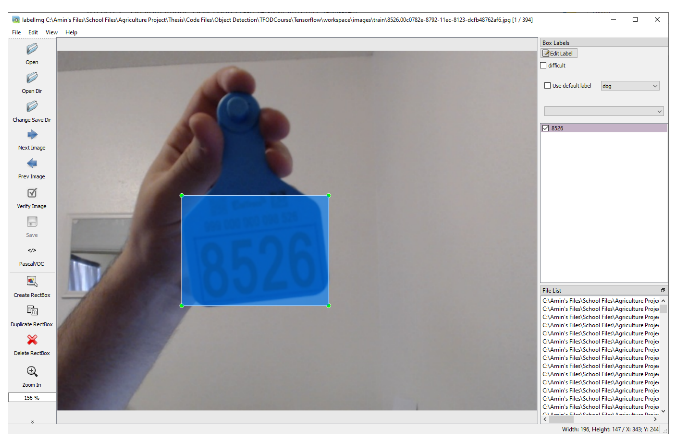

Fig. 3: Annotated image of tag “8526" in LabelIMG, small dataset.

### C. Model Training

To train the custom tag identification models of this method, we first select a pre-trained TensorFlow Object Detection model with Feature Pyramid Network (FPN) architecture from the “TensorFlow 2 Detection Model Zoo" [8], whose models were trained with COCO 2017 image dataset which included hundreds of thousands of images

Table 1: Used Model Attributes.

| Model Name | Speed (ms) | COCO mAP | Outputs |
|---|---|---|---|
| SSD MobileNet V1 FPN | 48 | 29.1 | Boxes |
| SSD MobileNet V2 FPNLite | 39 | 28.2 | Boxes |

with annotations for 90 different object classes. The models that were selected for this method were “SSD MobileNet V1 FPN 640x640" and “SSD MobileNet V2 FPNLite 640x640", both of which were chosen for their relatively high accuracy and low training speed. These models are used twice in our experiment: once for the smaller dataset, and once for the larger dataset each. All of the training and testing steps of this method were performed on TensorFlow 2.8.0, its compatible CUDA ToolKit version 11.2 and cuDNN 8.1, and NVIDIA RTX 3060 ti GPGPU. More detailed descriptions of these models’ speed and accuracy are summarized in Table 1. After selecting the model, a verification script ensures all of the packages related to TensorFlow are properly installed and that there are no compatibility issues, then the selected pre-trained model is cloned onto the training machine. Next, the labels and IDs of the custom object classes as well as the TFrecords (TensorFlow Records) of the “train" and “test" files are created. Finally, the training Pipeline file is configured and the custom model training begins with a given number of iterations. For every 1000 training cycles, the model generates a checkpoint. For the purpose of this paper, all four custom models were trained for 20,000 iterations.

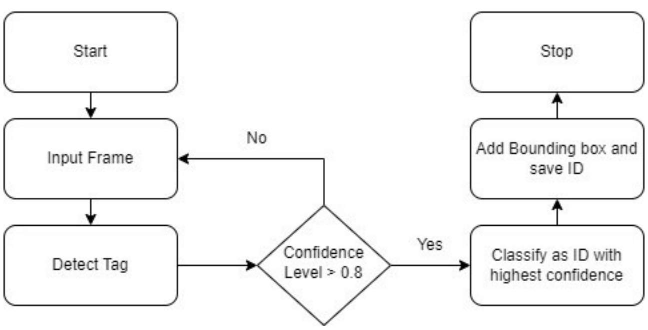


Fig. 4: Overall System Flowchart.

## D. Model Evaluation

The model is evaluated when it has been trained for the given amount of training cycles. The files resulting from the evaluation not only include the mean average precision (mAP) of the Precision and average recall (AR) of the very last checkpoint generated, but also the loss and learning rate of the entire model throughout the training. After the evaluation is complete, the resulting files are uploaded to TensorBoard [10] to generate the graphs of the model performance. Due to the fact that TensorBoard only generates the evaluation data of the very last checkpoint, to evaluate the entire model’s precision and recall, the evaluation scripts need to be manually run every 1000 training iterations. This reduces the speed of the evaluation stages.

Once a custom model has been trained for the given number of iterations, it can be used for tag detection. To build the detection model, the pipeline configuration file is loaded in and the model checkpoint with the highest evaluated mAP detection precision and recall is restored, this is usually the latest checkpoint. After the checkpoint is restored, the system can perform tag detection using the custom model within a given accuracy threshold, The output of which is

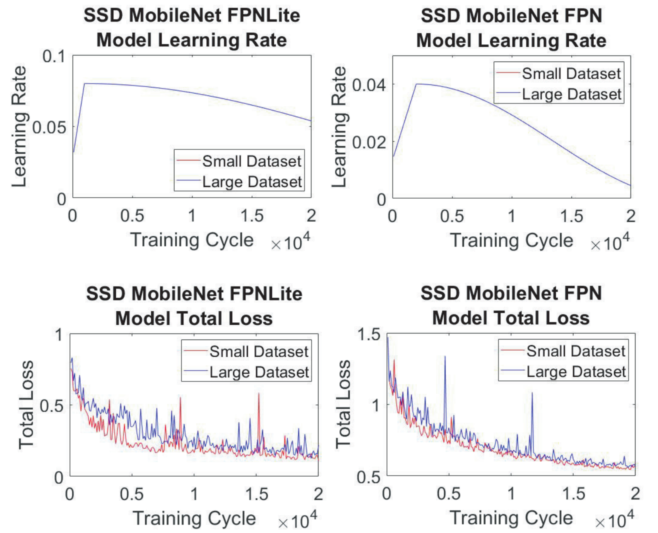


Fig. 5: Learning Rate and Total Loss of FPNLite and FPN models.

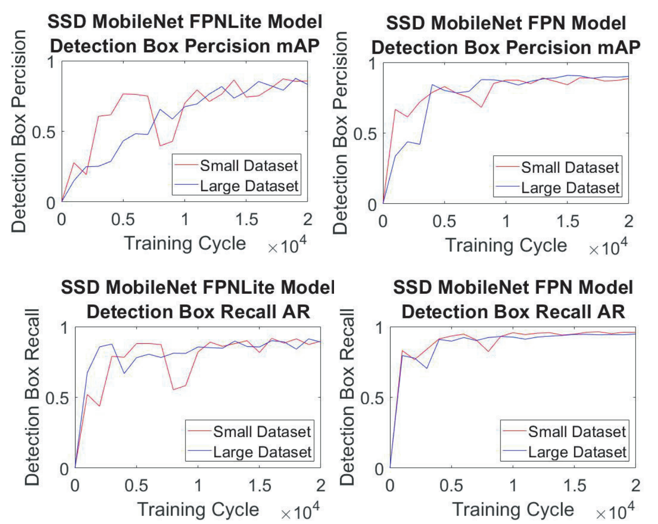


Fig. 6: Detection Box Recall AR and Detection Box Precision mAP for FPN and FPNLite Models.

the given image with the detected object placed in a bounding box that indicates both the identified class and the confidence rate. This could be either detection in single-frame images or live detection of tags in videos. The threshold that was generally used in this experimentation was 0.8. The entirety of the detection process is shown in the flowchart in Fig. 4.

## III. EXPERIMENTAL RESULTS

The small dataset contained 120 images for training and 30 images for testing. The large dataset contained 344 images for training and 109 images for testing. The overall performance during training and evaluation of both pre-trained models and datasets is displayed in Fig. 5 and Fig. 6. These graphs show that there are no differences in learning rate between the large and small dataset models and the learning rate drops significantly faster for the FPN model than the FPNLite model. The total loss of the larger dataset models stayed higher than the smaller dataset throughout the training. While the total losses of the FPN models were far higher than those of FPNLite when training started, they both stabilized at below 0.2. A similar trend is observed in the precision and recall of the two different models trained with the separate datasets. While in the earlier stages

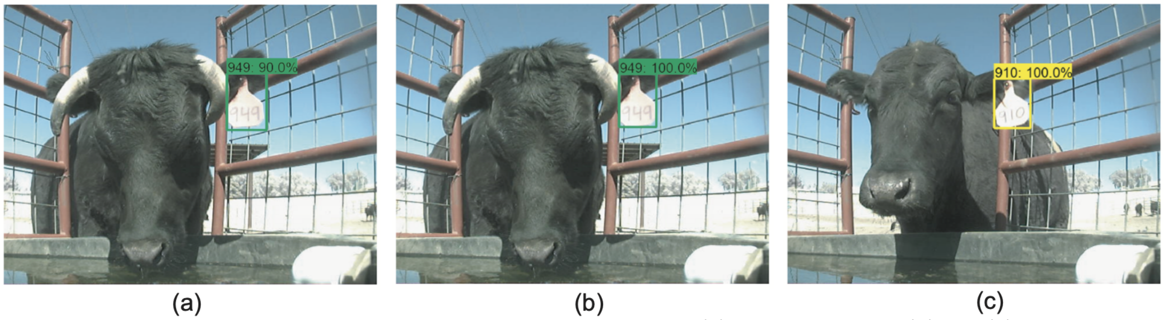


Fig. 7: Example detection results using FPNLite model (a) and FPN models (b) and (c).

Table 2: Comparison of recent cattle object detection systems

| | This Work | [4] | [5] |
|---|---|---|---|
| Method Target | Ear Tag | Muzzle | Face |
| Architecture | FPN | LBP | LBP |
| Accuracy | 90.00% | 100% | 95.3% |
| Retraining | NO | YES | YES |
| Camera | Regular | High Precision | High Precision |

of the training the precision and recall of the FPNLite model were higher than those of FPN, at 20000 cycles, the FPN model both had higher recall and precision by 5%.

While both models have achieved a high accuracy in detecting the tags as shown in Fig. 7, the models trained on using the ModelNet FPN model had a higher detection confidence. The confidence threshold for these images was around 80%. While the dataset that was used for this training was small with few numbers of training iterations. This study showed the overall effectiveness of TensorFlow Object Detection in identifying similarly shaped objects with slightly different patterns.

Table 2 compares this work and other works for the similar application in [4] and [5] in terms of overall accuracy. Methods in [4] and [5] focused on identifying the cows through their biometric features such as their muzzle pattern and face structure. Both of them require high precision cameras for multi-channel LBP histograms, which increases computing overhead. Moreover, when new cows move into the farm, the models need to be retrained. In our system, the use of FPN via TensorFlow lowers the computing overhead and provides flexible options to adapt new cows since the tags can be trained without cow biometric features.

## IV. CONCLUSION

A novel vision-based identification recognition system was developed for livestock water intake monitoring. The system is based on TensorFlow models to identify ear Tag numbers. Two models were compared in both larger and smaller datasets and achieved levels of accuracy around 90%. The proposed system provides scalability and reliability compared to existing methods, which is an attractive solution for vision-based machine learning systems in agricultural applications. In future work, the system should be implemented by combining other sensing technologies such as RFID and weight scales which can enhance detection accuracy and perform reliable water-intake monitoring of cattle.

## ACKNOWLEDGMENT

This project was partially supported by United States National Science Foundation ECCS-1652944 and USDA Western Sustainable Agriculture Research and Education program SARE OW19-341. This study was done with the collaboration of Dr. Marcy Ward of the NMSU College of Agricultural, Consumer, and Environmental Sciences.

## REFERENCES

[1] National Academies of Sciences, Engineering, and Medicine (NASEM). "Nutrient Requirements of Beef Cattle", The NationalAcademies Press: Washington, DC, USA, 2016.

[2] Gerbens-Leenes, P.; Mekonnen, M.; Hoekstra, A. "The water footprint of poultry, pork and beef:A comparative study in different countries and production systems", Water Resources, Ind. 2013, 1–2, 25–36.

[3] Wei Tang, Amin Biglari, Ryan Ebarb, Tee Pickett, Samuel Smallidge, and Marcy Ward. 2021. "A Smart Sensing System of Water Quality and Intake Monitoring for Livestock and Wild Animals" Sensors 21, no. 8: 2885.

[4] J. W. Kusakunniran and T. Chaiviroonjaroen, "Automatic Cattle Identification based on Multi-Channel LBP on Muzzle Images," 2018 IEEE International Conference on Sustainable Information Engineering and Technology (SIET), 2018, pp. 1-5,doi: 10.1109/SIET.2018.8693161.

[5] C. Cai and J. Li, "Cattle face recognition using local binary pattern descriptor," 2013 IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, 2013, pp. 1-4, doi: 10.1109/APSIPA.2013.6694369.

[6] Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K, "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors", proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017, pp. 7310-7311

[7] Tzutalin, LabelImg. Git code(2015). https://github.com/tzutalin/labelImg

[8] Hongkun Yu, Chen Chen, Xianzhi Du, Yeqing Li, Abdullah Rashwan, Le Hou, Pengchong Jin, Fan Yang, Frederick Liu, Jaeyoun Kim, Jing Li, "TensorFlow Model Garden", 2020, https://github.com/tensorflow/models

[9] Nicknochnack, TFOD. Git code (2021). https://github.com/nicknochnack/TFODCourse

[10] "Tensorboard." https://www.tensorflow.org/tensorboard.

[11] Ibarra, Vicente, Marcelo Araya-Salas, Yu-ping Tang, Charlie Park, Anthony Hyde, Timothy F. Wright, and Wei Tang. 2015. "An RFID Based Smart Feeder for Hummingbirds" Sensors 15, no. 12: 31751-31761. https://doi.org/10.3390/s151229886.

[12] Liu, Yifei, Paul M. Furth, and Wei Tang. 2015. "Hardware-Efficient Delta Sigma-Based Digital Signal Processing Circuits for the Internet-of-Things" Journal of Low Power Electronics and Applications 5, no. 4: 234-256. https://doi.org/10.3390/jlpea5040234

[13] Wei Tang, Paul M. Furth and Venkat Harish Nammi, Gaurav Panwar, Vicente Ibarra, Xiaochen Tang, Graciela Unguez, and Misra, Satyajayant, "An Aquatic Wireless Biosensor for Electric Organ Discharge With an Integrated Analog Front End," in IEEE Sensors Journal, vol. 19, no. 15, pp. 6260-6269, 1 Aug.1, 2019, doi: 10.1109/JSEN.2019.2908822.

[14] S. Liu, P. Reviriego, X. Tang, W. Tang and F. Lombardi, "Result-Based Re-computation for Error-Tolerant Classification by a Support Vector Machine," in IEEE Transactions on Artificial Intelligence, vol. 1, no. 1, pp. 62-73, Aug. 2020, doi: 10.1109/TAI.2020.3028321.