Object Detection with Mask RCNN for Holstein and Angus Cows

Anonymous Authors[1]

Abstract

The Dairy Industry is a critical aspect of agriculture as it is a billion-dollar industry. This industry makes the U.S. the world's largest producer of dairy and beef products according to the United States Department of Agriculture. Needless to say, cattle's well-being is paramount to the survival of this industry. However, contagious diseases pose a major threat and have been known to decimate cattle farms. Therefore, farmers must identify infected animals from the healthy ones. And it must be done in a quick and efficient manner. One of the most efficient methods to solve this issue is Visual Analysis. This is because sick animals behave differently, and their behavior can be analyzed visually. Therefore, to solve this problem I propose using Mask RCNN for this project. Mask RCNN is an object detection and instance segmentation method that is an extension of Faster RCNN.

1. Introduction

For this project, researchers are looking for particular actions or data that are tell-tale signs about sick or injured animals. This includes how many times a cow eats grains, or walks up to the food area, tracking the speed of the cow, and tracking the position of the cows. Furthermore, the data that the researchers have given us is in the form of a video, and therefore, we have to create something that works for videos. The reason for using Mask RCNN is that it uses the weights from coco and already has been pretrained to recognize cows. However, the main reason for using mask RCNN is that I have seen multiple people use this method successfully. Therefore, all these points led me to believe that this was the ideal method to use for cow detection and tracking.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Project Breakdown

2.1. Initial Findings

Here I detail what inspired me and what initial research I did in order to create my project.

2.2. Changes

In this section I detail the different changes I made to existing architectures.

2.3. Issues

Here I talk about the different challenges and problems I faced while making this project.

2.4. Future Goals

I lay out future goals so that someone can take over this project is necessary.

2.5. Reflection

Here I detail whether taking this course of action is the best way to go and also lay out some alternative ideas that might be more efficient.

3. Intial Findings

3.1. Neuromation

I was first inspired to use the Mask RCNN after reading about this company, Neuromation, and their success of tracking cows. Their project statement is very similar to ours in that they would like to track cattle and determine any injuries or sickness from cows. However, they used drones to track cows rather than a still video looking at a cage. So therefore, their data would not be useful. However, looking at the different images using different models (which can be seen in Figures 1-3), we can see that when they retrained the model with just cows they gained a higher accuracies. Therefore, I thought it would be a great idea to try to reimplement their work but with out data.

Figure 1. detection of cows using YOLO detection

Figure 2. detection of cows using MaskRCNN detection

Figure 3. detection of cows using retrained MaskRCNN detection

3.2. MaskRCNN

Before using the Mask RCNN I wanted to go in more depth about what it is and how it works. There are two stages of Mask RCNN. First, it generates proposals about the regions where there might be an object based on the input image. Second, it predicts the class of the object, refines the bounding box and generates a mask in pixel level of the object based on the first stage proposal. Both stages are connected to the backbone structure. The backbone structure used in this case was the FPN style deep neural network. The FPN style because it maintains strong semantically features at various resolution scales. Another reason I went with Mask RCNN is because it had already been trained on cows and will therefore, hypothetically, be able to detect cows from our data set.

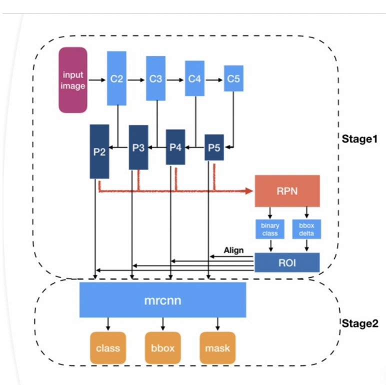

Figure 4. Architecture of Mask RCNN and the different stages included

4. Changes

4.1. MaskRCNN used for Videos not Images

Since Mask RCNN works only on one image at a time, as seen in Figure 5, I implemented a program that used the Mask RCNN to create a video. One of the first changes I made was treating every tracked object as either a cow or person. Sometimes, the mask RCNN detected one of the cows as a sheep as seen in Figure 6, however, we know for a fact that there are only cattle and people present in the video, and therefore, I decided to change all the detected classes as cows. First, I ran the Mask RCNN on one image, and stopped it during the second stage just before it overlays the pixel level segmentation and extracted the bounding boxes from it. Then I rendered the bounding box over the cows rather than the max and also saved the values in a file. I repeated this process for all the images and then combined them to create a video. However, you can see in the Figure 7, that this model does not do a good job of detecting all the cows in the image. Therefore, I tested the Mask RCNN on some cropped image. The cows in the back were not being detected, however, after cropping the image and focusing on those cows it detected better as seen in Figure 8. After seeing the entire video, it seemed that the Mask RCNN was not able to detect sitting cows or cows clumped together as well. However, after cropping the image and focusing on just laying cows it seemed to detect laying cows with higher accuracies as seen in Figure 10 . Yet, looking at the extracted data from the regular images, I saw that it had detected 20 cows, but the researchers estimate that there are 50-60 cows. Therefore, I now had to retrain the model to detect only cows and also detect cows that were clumped together and overlapping.

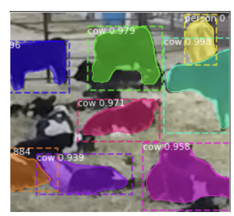

Figure 5. segmentation done on image using Mask RCNN

Figure 6. Mask RCNN detecting cow as sheep

Figure 7. Mask RCNN detecting cows on videos or series of images rather than one image

Figure 8. segmentation done on cropped image with cows in the back of the pen using Mask RCNN

Figure 9. segmentation done on Angus cows using Mask RCNN

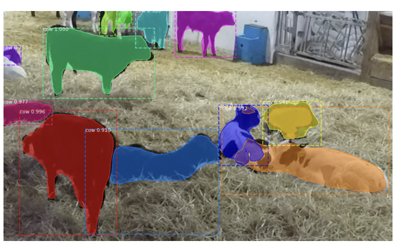

Figure 10. segmentation done on cropped image with cows laying down using Mask RCNN

4.2. Existing Repository

When researching how to train the model I found a repository that had already retrained the model on a cow data set. This cow data set included challenging image which were: overlapping cows, partial occlusion, similarity between cows and background, and bad lightening. The two that was important to me was the overlapping of the cows and partial occlusion since I believed that was the main issue in our dataset as well. Therefore, I tried using the repository to retrain the model.

5. Issues

For issues, I previously talked about the current Mask RCNN model that couldn't detect all the cows in the image and thus the need to retrain the model. However, retraining the model was the most challenging part of this project. In order to get the training model to run, I had to change the Cow Data class in repository, which was the class used just to retrain the model, change the backend libraries of TensorFlow and Keras, various functions used, and also changing the number of losses used where the repo had used 5 losses, and I just used the mrcnn class loss. However, after getting the training script to work it took a very a long time to retrain and therefore, I got disconnected and kicked off of Google Colab after a couple of hours. In addition, I did not have enough time to do my own annotation for our own data set.

Figure 11. Image from data set of Cows used in existing repository

Figure 12. Existing repository's data set included images of partial occlusion, overlapping of cows, and different colors of cows

6. Future Goals

For this project, future goals include annotating the data set we currently have and retrained the model from our data. There are also three different types of data or videos which include different cow species which is Holstein, Angus, and a mix of Holstein and Angus cows. I believe retraining the model on an annotated data set from the videos provided, would be a success since so many people have reproduced similar success with different data sets. However, I have laid out some doubts about using specifically Mask RCNN and gave some ideas about different models to use.

- Email: vaaCowTracking@gmail.com

- Password: vaatrackCows2020

7. Reflection

Reflecting on this project I learned that using pytorch instead of an existing repo would have been better to use. Pytorch already has the model and certain settings already created and therefore would have been easier to use instead of changing backend libraries or a repo's files. Also, reflecting on this project, I am thinking about whether or not the Mask RCNN is worth it. It is accurate in finding an object and rendering pixel level segementation, and even though I was not able to reproduce the results, if it is retrained on just a cow data set, it would work more accurately in finding all the cows. However, in order to produce a single image it takes quite a bit of time, and to make a video that is around 1 minute it took a couple of hours. However, it seems that researchers are looking for real time data and the mask rcnn might not be the best to use. When critiquing my project, a grad student suggested using faster RCNN. I do not know much about faster RCNN, but Mask RCNN is a derivative of it. However, it does not seem it has been trained on cow data and one will most likely have to start from scratch. After finding the right architecture, I advise the next person to also start annotating the data at the same time as well–training takes a long time and it would be best to use that time annotating data.

References

Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. Simple online and realtime tracking. *2016 IEEE International Conference on Image Processing (ICIP)*, Sep 2016. doi: 10.1109/icip.2016.7533003. URL http://dx.doi.org/10.1109/ICIP.2016.7533003.

danish07me. danish07me/mask-rcnn v0.5, May 2019. URL https://doi.org/10.5281/zenodo.3533303.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn, 2018.

Hui, J. Image segmentation with mask r-cnn, April 2018. URL https://jonathan-hui.medium.com/image-segmentation-with-mask-r-cnn-ebe6d793272.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection, 2017.

Neuromation. Tracking cows with mask r-cnn and sort, July 2018. URL https://medium.com/neuromation-blog/tracking-cows-with-mask-r-cnn-and-sort-fcd4ad68ec4f.

Redmon, J. and Farhadi, A. Yolo9000: Better, faster, stronger, 2016. URL https://arxiv.org/pdf/1612.08242.pdf.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection, 2016.

Ren, S., He, K., Girshick, R., and Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

Zhang, X. Simple understanding of mask rcnn, April 2018. URL https://alittlepain833.medium.com/simple-understanding-of-mask-rcnn-134b5b330e95.