TRƯỜNG ĐẠI HỌC
SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
HCMC University of Technology and Education
HCMUTE

KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
FACULTY FOR HIGH QUALITY TRAINING
FHQ

Identify and count the number of cows on farms using Region-Based Convolution Neural Network.

Tran Quoc Toan[1].

1. HCMC University of Technology and Education, Ho Chi Minh City, Vietnam.

ARTICLE INFO

Article history:

Received 21 June 2022

Keywords:

CNN
R-CNN
Artificial Intelligent
Python
Realtime
Object detection
Cow
Count object

ABSTRACT

This study object localization algorithm of Region-Based Convolution Neural Network (R-CNN) model, apply on identify and count the number of cows on farms, thereby easily monitoring the behavior, psychology and health of the cows in anytime, from which to plan on motivate cows on going for activities, improves their mood, to produces high quality meat and milk products.

I. Introduction

People, machines, and robots are all required in every factory to produce goods. Humans are in charge of everything, from the smallest to the largest objects, such as computers, robots, and machines. Robots and other electric equipment collaborate to do tasks assigned by humans, assisting the livestock industry's growth.

When appropriate, cutting-edge technologies are used in the livestock industry, the quantity and quality of products increase significantly and the industry continues to grow. In the cattle industry generally, keeping an eye on the animals' health can provide timely, accurate analysis to ensure that the cows are not afflicted with infectious diseases, anorexia, inactivity, or other conditions.

When hiring people to do the same task over and over, we run into a lot of issues. To create a robotic arm that grabs and places packaged food from the conveyer to the crate in this project. The following are some of the drawbacks of hiring employees: employees' health, emotions, conception, malpractice.

There are far more issues with humans than we can list. This is when the AI arrive to solve the issues. AI increase productivity because they do not require rest and follow all of your directions. The opportunity to invest in the surplus value is enormous. They would work continuously, precisely, and quickly for a longer amount of time than humans if the current was not cut off. Compared to today's camera surveillance technology, manual monitoring with the naked eye is very expensive, prone to mistakes, and ineffective.

Object detection algorithms developed from the R-CNN currently typically include Fast R-CNN, Faster R-CNN, Mask R-CNN and Yolo.

This article will use Semantic Segmentation Mask R-CNN algorithm aside with Google Colab to detect and count the number of cows on farms.

II. Methodology

1. Image labeling

The goal of image labeling, a type of data labeling, is to recognize and tag particular details in an image.

There are many image labeling tool, which Labelimg is one of them

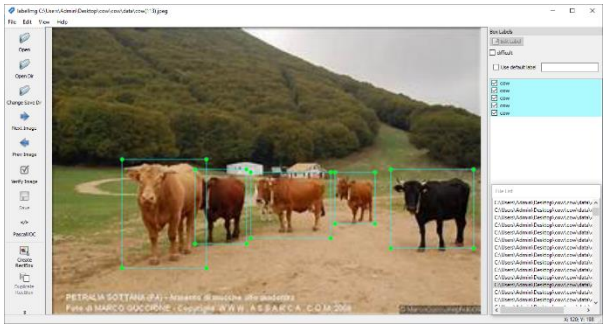


Fig.1 Image labeling using Labelimg

2. Semantic Segmentation

Clustering portions of an image that belong to the same object class together is known as semantic segmentation, also known as image segmentation. Since each pixel in a picture is assigned to a category, it is a type of pixel-level prediction. Pascal

Voc is a examples of benchmarks for this job. The Mean Intersection-Over-Union (Mean IoU) and Pixel Accuracy metrics are typically used to assess models.

3. Object Detection

Identifying an object in an image and its location within the image frame is the job of an object detection algorithm. Bounding boxes are commonly used to define an object's location. The smallest rectangle that completely encloses an object in an image is known as a bounding box.

A bounding box technically refers to a set of four coordinates that are associated with a label that identifies the object's class. Typically, a JSON file using a dictionary format is used to store the bounding box coordinates and their labels. The dictionary file's key is the image ID or number.

4. Selective Search algorithm

A region suggestion approach for object detection is called Selective Search. It is made to be quick and have a high recall rate. It is based on computing hierarchical grouping of related regions according to compatibility of color, texture, size, and form.

Using a graph-based segmentation method developed by Felzenszwalb and Huttenlocher, Selective Search begins by over-segmenting the image based on pixel intensity. Below is a display of the algorithm's output. Segmented zones are shown in solid color in the image to the right.

At each iteration, larger segments are formed and added to the list of region proposals. Hence we create region proposals from smaller segments to larger segments in a bottom-up approach. This is what we mean by computing “hierarchical” segmentations using Felzenszwalb and Huttenlocher’s oversegments [1].

5. Classification of region proposal

There are many region proposals that contain no objects as a result of the selective search algorithm for up to 2000 region proposals. Therefore, we must add a background layer (which contains no objects). For instance, we will categorize each bounding box in the image below as either a background, a horse, or one of the four region proposals [2].

4. Mask R-CNN

Faster R-CNN approximately 2000 bounding boxes with the potential to contain objects should be obtained using the Selective Search algorithm with faster speed than Fast R-CNN and R-CNN.

Mask R-CNN was built using Faster R-CNN with 3 output of each object: class label, bounding box and object mask. The additional mask output requires the extraction of a much more precise spatial layout of an object because it differs from the class and box outputs.

Mask R-CNN has 2 main types of image segmentation: Instance Segmentation and Semantic Segmentation.

III. Data

1. Datasets

Only 100 pictures with multiple number of cows was used due to the high RAM consuming, which is about 80 pictures for training and 20 for testing. The image was labeled into PascaVOC format.

Fig.2 Labeling by bounding boxes coordinates datasets

2. Model

The class was devided into 2 classes: cows and others. Each epoch need the step that equal to the training pictures. That mean the bigger datasets are, the longer it take to train the model.

We can use tourch.nn liberary to build a similar model, define layer 2 3 4, use _upsample_add or UpSampling2D, config.TOP_DOWN_PYRAMID_SIZE, but we will use Mask R-CNN model instead.

Mask R-CNN use 4 + 4 Feature Pyramid Network (FPN) layers to train the model:

```
Selecting layers to train
fpn_c5p5               (Conv2D)
fpn_c4p4               (Conv2D)
fpn_c3p3               (Conv2D)
fpn_c2p2               (Conv2D)
fpn_p5                 (Conv2D)
fpn_p2                 (Conv2D)
fpn_p3                 (Conv2D)
fpn_p4                 (Conv2D)
```

Fig.3 Mask R-CNN model's layers

In the 4 last FPN layers fpn_px have kernel size (3x3), they are working independence to each others, included in one bottom-up pathway with decreasing resolution/ increasing semantic value, and one top-down pathway.

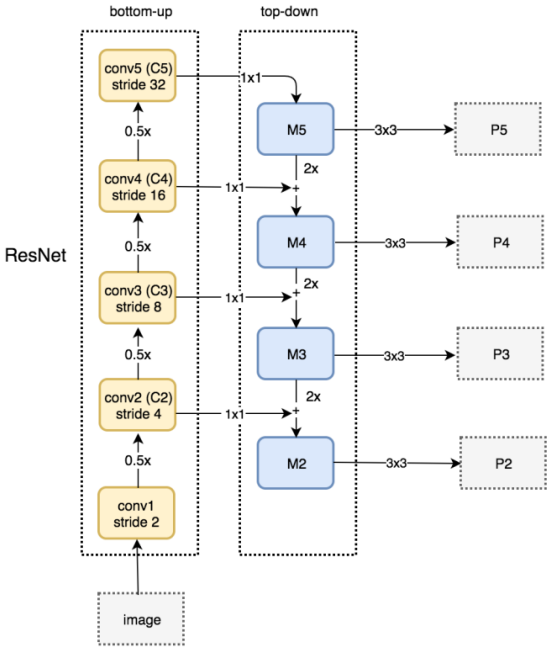
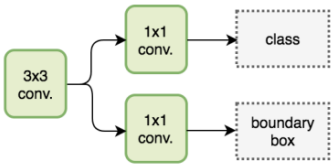


Fig.4 Data flow in Feature Pyramid Network

We can use tourch.nn liberary to build a similar model, define layer 2 3 4, use _upsample_add or UpSampling2D, config.TOP_DOWN_PYRAMID_SIZE, but we will use Mask R.

One FPN layer is include with one Conv2D 3x3, going through feature maps, extract it and convert data to one Conv 1x1 Regresstion, one Conv 1x1.



predictor head

On the other hand, the first 4 FPN layers fpn_cxpx only have kernel size (1x1) that is used to convert data to Conv 1x1 Mask.

3. Training process

Even when we use Colab training consume about 2 minute each step.

```
Epoch 1/5
80/80 [==============================] - 10546s 132s/step - loss: 1.4742 -
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/keras/callba

Epoch 2/5
80/80 [==============================] - 10340s 129s/step - loss: 1.0119 -
Epoch 3/5
24/80 [========>.....................] - ETA: 1:40:53 - loss: 0.8643 - rpn_
```

The loss in each epoch is decreasing, but because of the method of time, we will only run 5 epoch with 0.787 loss value

```
Epoch 1/3
80/80 [==============================] - 10152s 127s/step - loss: 1.6022
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/keras/cal

Epoch 2/3
80/80 [==============================] - 9991s 125s/step - loss: 0.8956 -
Epoch 3/3
80/80 [==============================] - 10024s 125s/step - loss: 0.7873
```

- Final result give: loss: 0.7873

rpn_class_loss: 0.0066 - rpn_bbox_loss: 0.1696

mrcnn_class_loss: 0.105 - mrcnn_bbox_loss: 0.209

mrcnn_mask_loss: 0.2972 - val_loss: 0.9043

val_rpn_class_loss: 0.012 - val_rpn_bbox_loss: 0.294

val_mrcnn_class_loss: 0.092

val_mrcnn_bbox_loss: 0.215

val_mrcnn_mask_loss: 0.2903

- Evaluate mAP:

Train mAP: 0.715

Test mAP: 0.751

IV. Results and Discussion

Here are some results from the model:

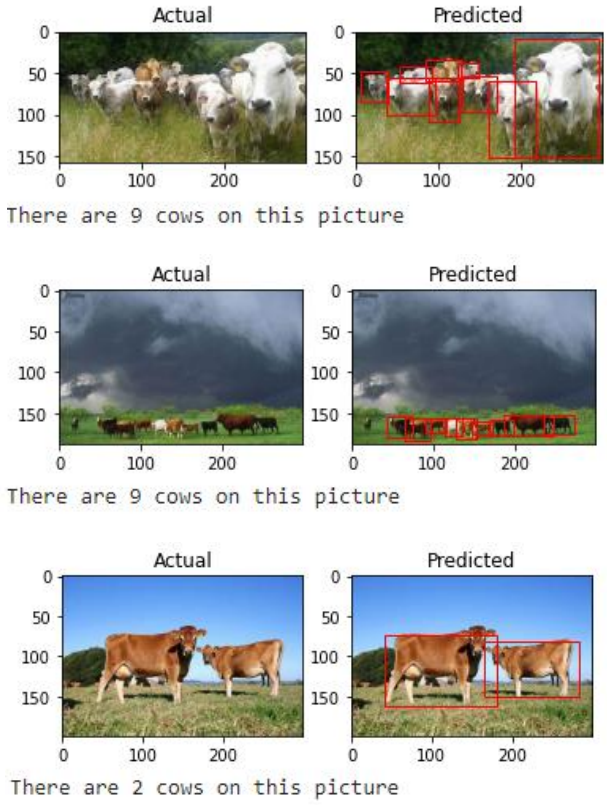


Fig.5 Predicted results

We can also extract frame from webcamera on Google Colab to run the model:

Fig.6 Webcam interface on Google Colab

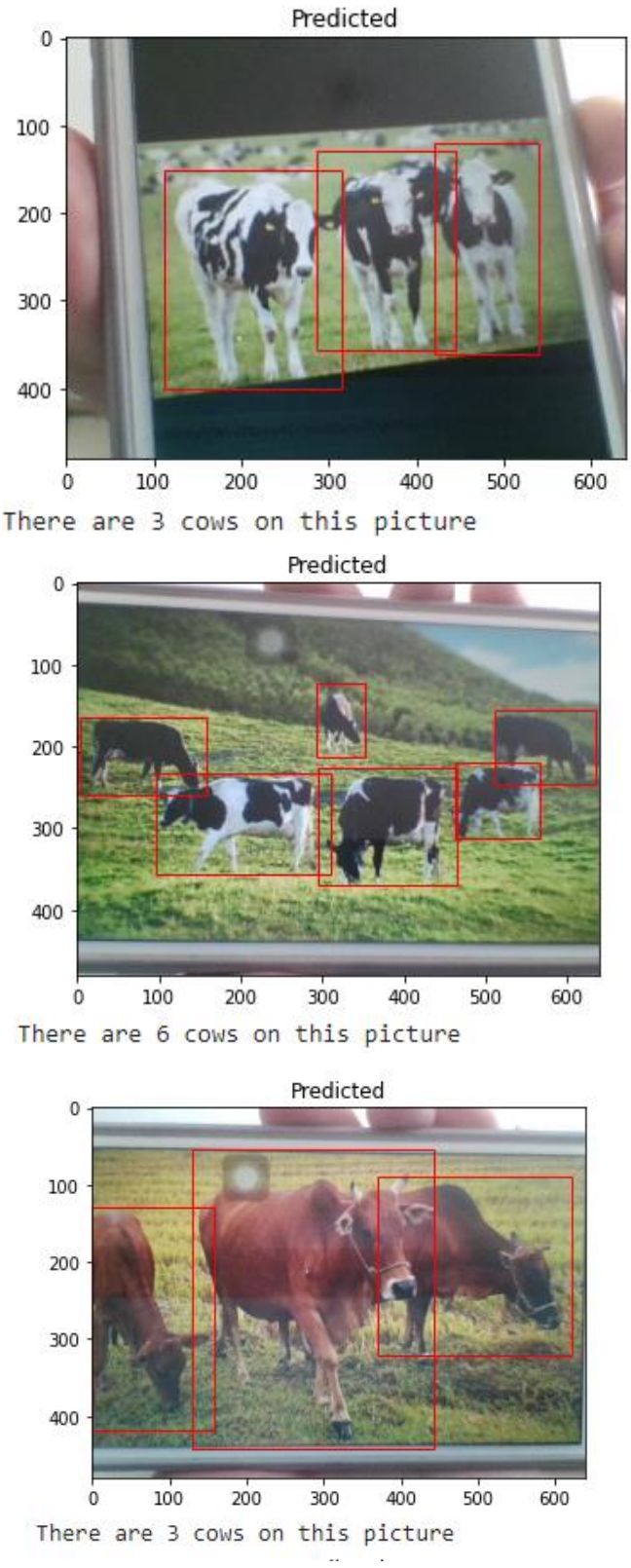


Fig.7 Webcam extracted frame predicted results.

Since the loss value is still higher than 0.1, there will be some error in classifying, and because only 1 class of cow define mean giving more loss in value, more Intersection over Union (IoU) when classifying the head, butt and parts of other cows. But the upper results is acceptable.

V. Conclusion

In general, the model met the minimum requirements for the problem of identifying and counting cows on farms. The next step is to create more topics and draw on more experiences in order to find more cattle.

IV. References

[1] Uniduc JSC. *"https://uniduc.com/vi/blog/object-detection-tim-hieu-ve-thuat-toan-r-cnn-fast-r-cnn-va-faster-r-cnn"*. 2020

[2] Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik. *"Rich feature hierarchies for accurate object detection and semantic segmentation"*. 2014