Name: Balaji Chippada

Email address: balajichippada.20@gmail.com

Contact number: 8129990317

Anydesk address: **286 698 756**

Date: **15th Jul 2020**

Self Case Study -2: Global Wheat Detection

Overview

Goal: The goal is to predict bounding boxes around each wheat head in images that have them. If there are no wheat heads, you must predict no bounding boxes.

This case study is focussed on detecting wheat heads from outdoor images of wheat plants, including wheat datasets from around the globe. Using worldwide data, you will focus on a generalized solution to estimate the number and size of wheat heads.

These images in the training data are used to estimate the density and size of wheat heads in different varieties. Farmers can use the data to assess health and maturity when making management decisions in their fields.

The Global Wheat Head Dataset is led by nine research institutes from seven countries: the University of Tokyo, Institut national de recherche pour l'agriculture, l'alimentation et l'environnement, Arvalis, ETHZ, University of Saskatchewan, University of Queensland, Nanjing Agricultural University, and Rothamsted Research.

Data can be found from Kaggle - https://www.kaggle.com/c/global-wheat-detection/data

Difficulties in the problem:

Wheat head detection in outdoor field images can be visually challenging. There is often overlap of dense wheat plants, and the wind can blur the photographs. Both make it difficult to identify single heads.

Additionally, appearances vary due to maturity, color, genotype, and head orientation. Finally, because wheat is grown worldwide, different varieties, planting densities, patterns, and field conditions must be considered.

This competition is evaluated on the mean average precision at different intersections over union (IoU) thresholds.

Research-Papers/Solutions/Architectures/Kernels

1. https://www.kaggle.com/c/global-wheat-detection/discussion/165393

   This kaggle discussion gave an understanding on how to interpret bounding boxes in the given data. As i haven't done any bounding boxes before this gave me on how one can draw bounding boxes and how you can submit your final predictions in the case study.

   And different models have different representation of bounding boxes like for example

   COCO Bounding box: (x-top left, y-top left, width, height)

   Pascal VOC Bounding box :(x-top left, y-top left,x-bottom right, y-bottom right)

2. https://stackoverflow.com/questions/55268762/how-to-accumulate-gradients-for-large-batch-sizes-in-keras

   This talks about Accumulated Gradients in kears. This comes into play when we are dealing with low ram space or gpu space. Loading more images(Batch size) at a same time will be memory consuming and will get your notebook to stop abruptly.

So one way is to use accumulated gradients where gradients will not be updated at every batch, instead they will be updated after a certain number of batches. This will help a lot in case you want to use less batch size and avoid overfitting.

3. https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/

   Albumentations is a fast and flexible image augmentation library. Albumentations can be used like Imagegenerators where we can transform images with different parameters like scaling, zooming, cropping etc…

   Albumentations is much better in this case study as this technique can be used to generate multiple images along with bounding boxes.

   Albumentations support a wide variety of transformations that we can use and there are certain parameters that we use to sort bounding boxes after transformation like min_area, min_visibility, Class_labels.

   Albumentations support COCO Bounding box,Pascal VOC Bounding box formats as well. Albumentations is faster than other commonly used image augmentation tools on most image transform operations.

4. https://towardsdatascience.com/test-time-augmentation-tta-and-how-to-perform-itwith-keras-4ac19b67fb4d

   We generally use ImageDataGenerators to train the model for getting more accuracy. But in fact we can use ImageDataGenertors to test the images to increase test accuracy.

   TTA -Test Time Augmentation is a technique that lets us use that.

   But we can't use TTA on every dataset because it might even reduce accuracy as the model will get confused on what the actual test image is. MNIST is an example dataset where you don't want to use TTA.

   Use TTA and average the predictions to get final predictions.

5. https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

   Faster RCNN model training:

We first take a pre-trained convolutional neural network.

Then, this model is retrained. We train the last layer of the network based on the number of classes that need to be detected.

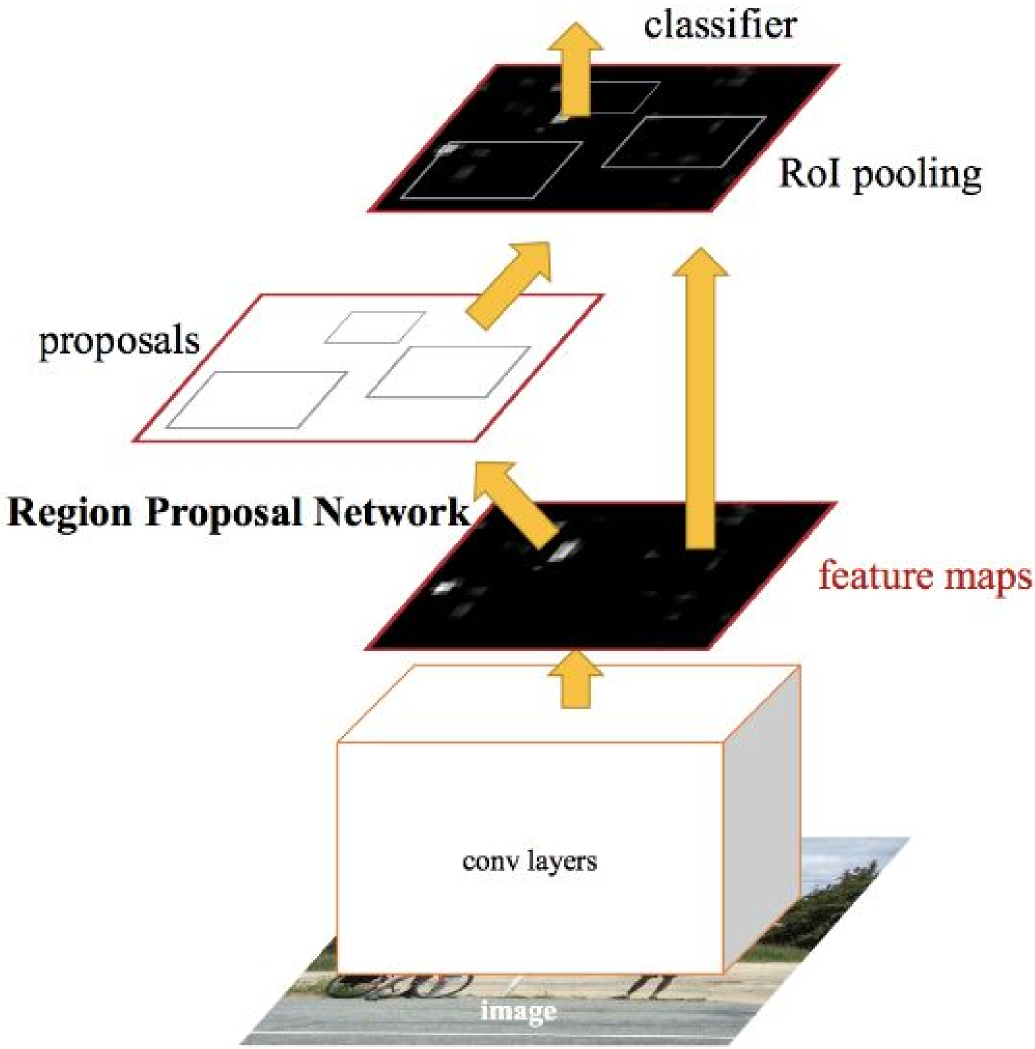
We take an image as input and pass it to ConvNet which returns the feature map for that image.

The third step is to get the Region of Interest for each image. We then reshape all these regions so that they can match the CNN input size.

After getting the regions, we train SVM to classify objects and backgrounds. For each class, we train one binary SVM.

Finally, we train a linear regression model to generate tighter bounding boxes for each identified object in the image.

A softmax layer is used on top of the fully connected network to output classes. Along with the softmax layer, a linear regression layer is also used parallely to output bounding box coordinates for predicted classes.



First Cut Approach

1. Load the Data.
2. Check how many images we have for training and testing.
3. As this problem is regarding bounding boxes and wheat detection , we have to do some sanity check on images in the given data.

4. Check weather bounding boxes are present for all the images in the training data or not. If by chance there are images where there are no bounding boxes i think it's better to represent them with 0 labels. [Will get to this while i do EDA].
5. In this task we just need to detect wheat heads. So if a given image there will only be 2 classes , either it will be wheat head or it will be considered as background.so num_classes in this task is 2.
6. So create a dataframe with labelling classes as wheat_head for the images that contain bounding boxes and 0 for the images that don't contain bounding boxes.
7. For EDA steps we can start to check distributions of the train data.
8. Check from what sources the images came and try to observe different kinds of images from different sources.
9. Try to find the max and min number of boxes present in a given image. From this we can conclude about the density of wheat heads in the images.
10. Visualize and see if the given ground truth bounding boxes are present in every wheat head of the image or check if they missed to label some.
11. Once EDA is done, we split the dataset into train and cv and use image augmentations for training the final model.
12. Most well versed object detection techniques are Faster RCNN and Yolo.
13. I will try to implement both of these techniques to train and will see which model works best and what needs to change to further improve the score.
14. Metric uses are - IOU over bounding boxes.
15. I can also use TTA while making predictions for the test image dataset.
