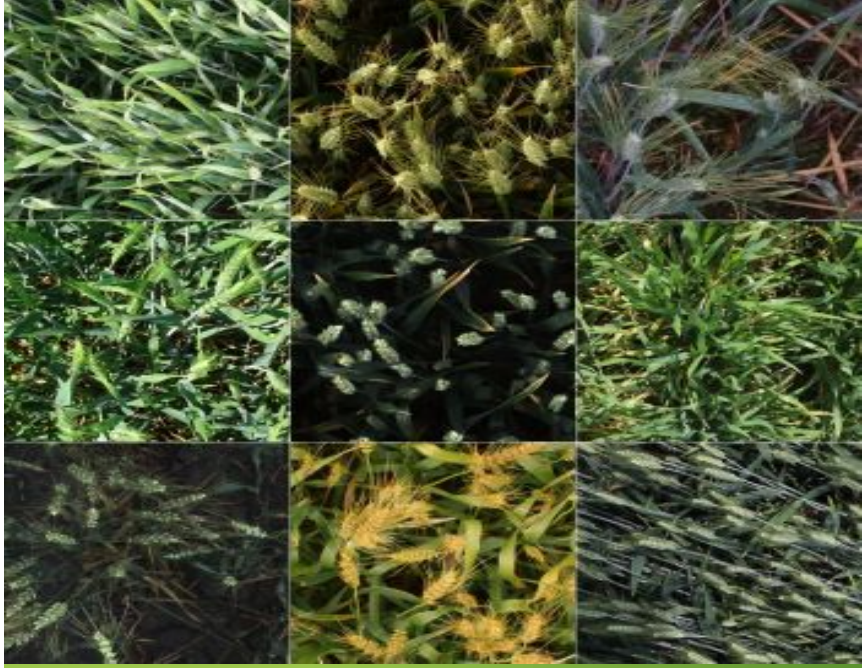


GLOBAL WHEAT DETECTION

Anna Nikolaeva | Maria Begicheva | Javier Penuela

Competition:
https://www.kaggle.com/c/global-wheat-Detection
Data analysis:
https://colab.research.google.com/drive/17Umd1lmlG6g2xPLVCcjL0SPDiwcgbFwq?usp=sharing
Whole pipeline:
https://colab.research.google.com/drive/1amtyo7uLU-fWPRoDhWD9Mffy5NgVRmxc?usp=sharing

MOTIVATION

To get large and accurate data about wheat fields worldwide, plant scientists use image detection of "wheat heads"—spikes atop the plant containing grain.

THE DATASET

The data is images of wheat fields, with bounding boxes for each identified wheat head. Not all images include wheat heads / bounding boxes. The images were recorded in many locations around the world

The CSV data is simple - the image ID matches up with the filename of a given image, and the width and height of the image are included, along with a bounding box (see below). There is a row in train.csv for each bounding box. Not all images have bounding boxes.

Files

- train.csv - the training data
- sample_submission.csv - a sample submission file in the correct format
- train.zip - training images
- test.zip - test images

WHAT ARE WE PREDICTING AND WHY IS IT USEFUL ?

We are attempting to predict bounding boxes around each wheat head in images that have them. These images 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.

CODE REQUIREMENTS

This is a Code Competition

Submissions to this competition must be made through Notebooks. In order for the "Submit to Competition" button to be active after a commit, the following conditions must be met:

- CPU Notebook <= 9 hours run-time
- GPU Notebook <= 6 hours run-time
- TPUs will not be available for making submissions to this competition. You are still welcome to use them for training models.
- No internet access enabled
- External data, freely & publicly available, is allowed. This includes pre-trained models.
- No custom packages enabled in kernels
- Submission file must be named "submission.csv"

INITIAL DATAFRAME

bbox

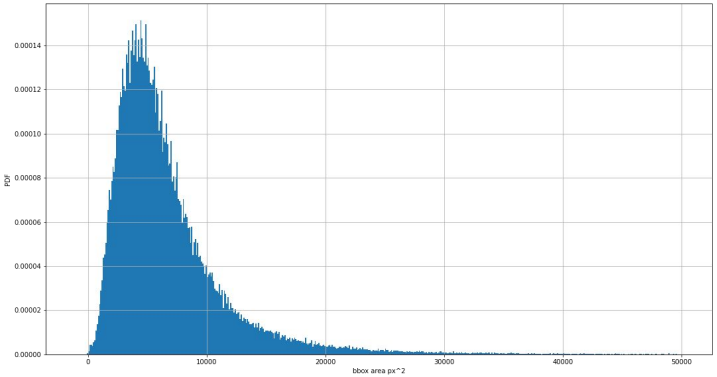
[x min, y min, height, width]

	image_id	width	height	bbox	source
0	b6ab77fd7	1024	1024	[834.0, 222.0, 56.0, 36.0]	usask_1
1	b6ab77fd7	1024	1024	[226.0, 548.0, 130.0, 58.0]	usask_1
2	b6ab77fd7	1024	1024	[377.0, 504.0, 74.0, 160.0]	usask_1
3	b6ab77fd7	1024	1024	[834.0, 95.0, 109.0, 107.0]	usask_1
4	b6ab77fd7	1024	1024	[26.0, 144.0, 124.0, 117.0]	usask_1

3373 training images
8* testing images for processing
x* testing unavailable images in public and private collection

BBOXES
AREA DISTRIBUTION

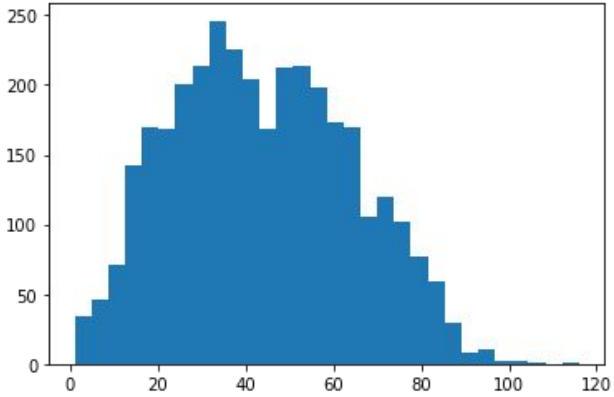
Area value distribution



As we see there are some irregular boxes

- some are too close to zero
- some have an area abnormally large we analyzed them to eliminate in case of being necessary.

Max number of bounding boxes is 116, whereas min (annotated) number is 1



BBOXES EVALUATION

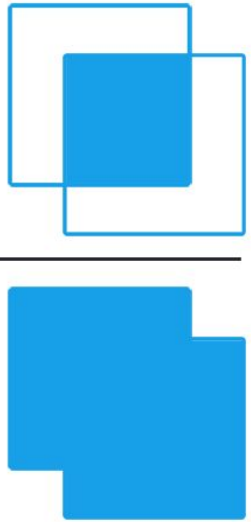
The submission looks like:

image_id	PredictionString
aac893a91	0.9776 551 542 129 185 0.9765 613 918 87 101 0.9726 356 535 97 78 0.9685 234 849 157 97 0.9564 581 7...
51f1be19e	0.9857 278 480 138 113 0.9717 760 888 154 102 0.9690 491 475 211 105 0.9651 797 764 119 97 0.9602 81...
f5a1f0358	0.9754 687 203 115 90 0.9709 146 760 154 122 0.9625 63 462 135 130 0.9553 815 409 98 91 0.9485 147 2...

The aim is to detect all wheat seeds with the proper accuracy of box boundaries.

The quality metric will be Intersection over Union (Коэффициент Жаккара):

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



THE METHODS

From ML point of view, we are solving “Binary classification and object localization problem”

fig 1.

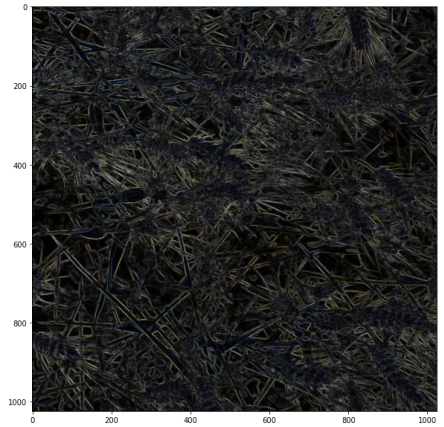


fig 2.

Classical CV preprocessing (separately):

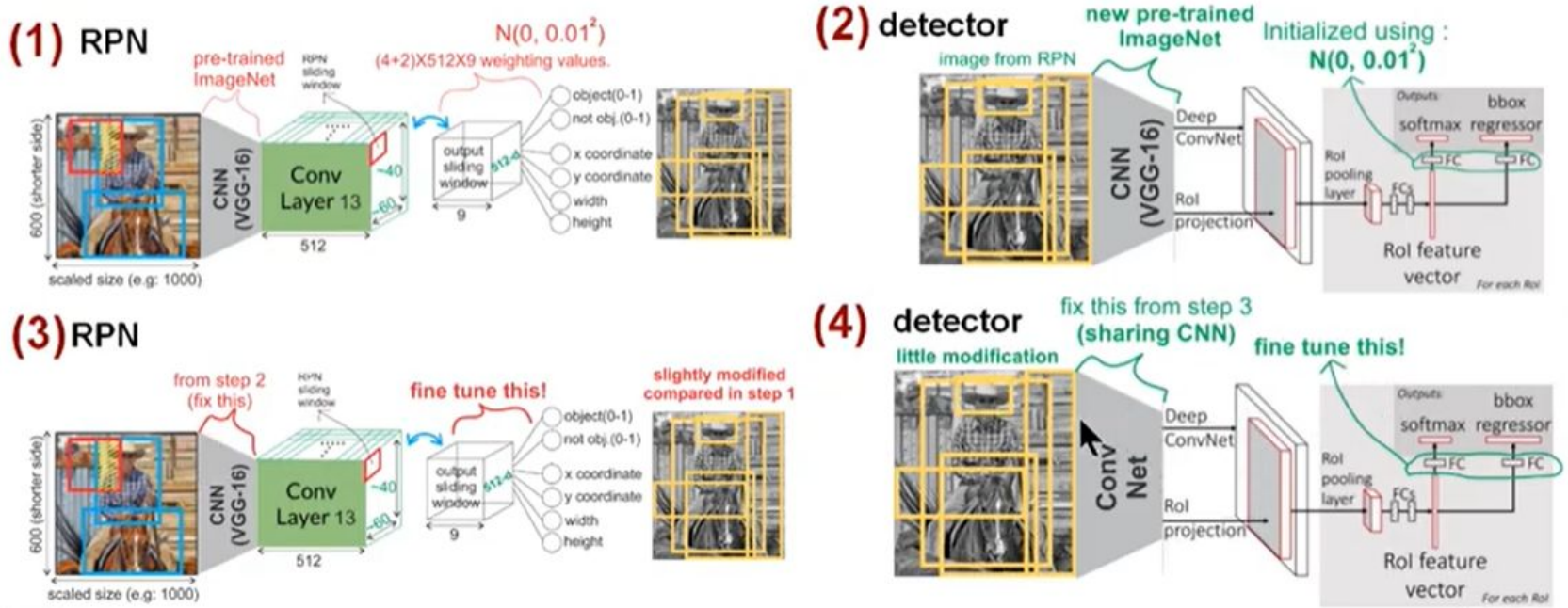
- Gaussian Blur (kernel size=3, 5, 7)
- Canny edge detector (search edges in a picture, figure 1)
- Histogram of Oriented Gradients (figure 2)
- Viola–Jones object detection (for working with faces)

Deep Learning

- Faster Region-based Convolutional Network (2015) (https://arxiv.org/abs/1506.01497)

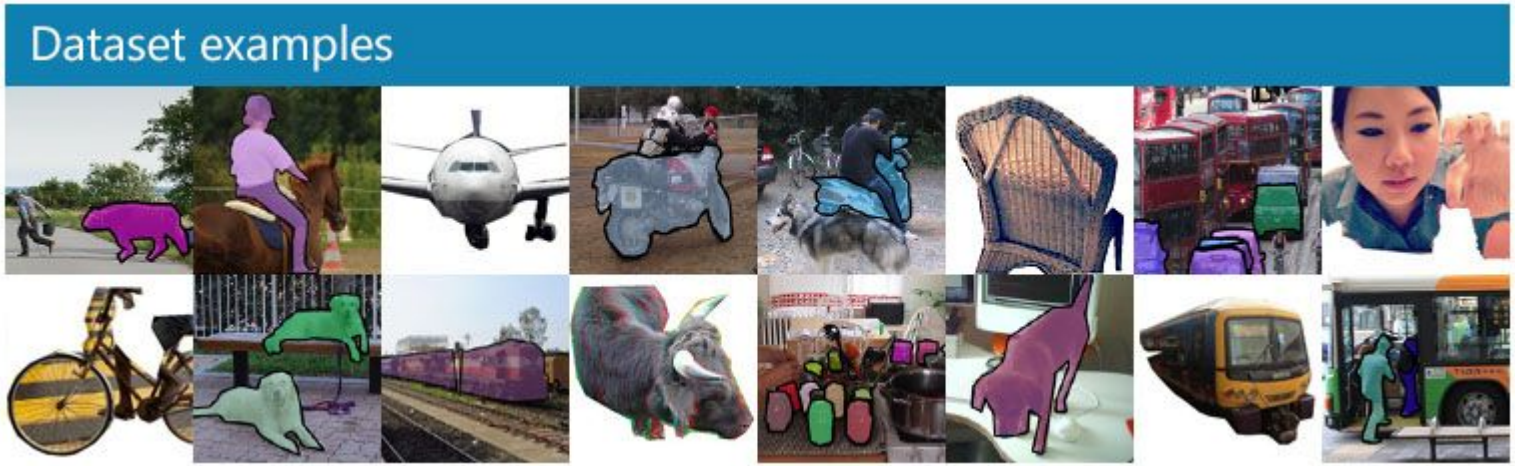
Faster Region-based Convolutional Network

RPN - Region Proposal Network



We have used Faster RCNN with pretrained ResNet50 as autoencoder.
For this we have used Common Objects in Context (COCO) dataset

Experimentally we sow the the increase in performance by 20% when used pretrained variant



MODEL

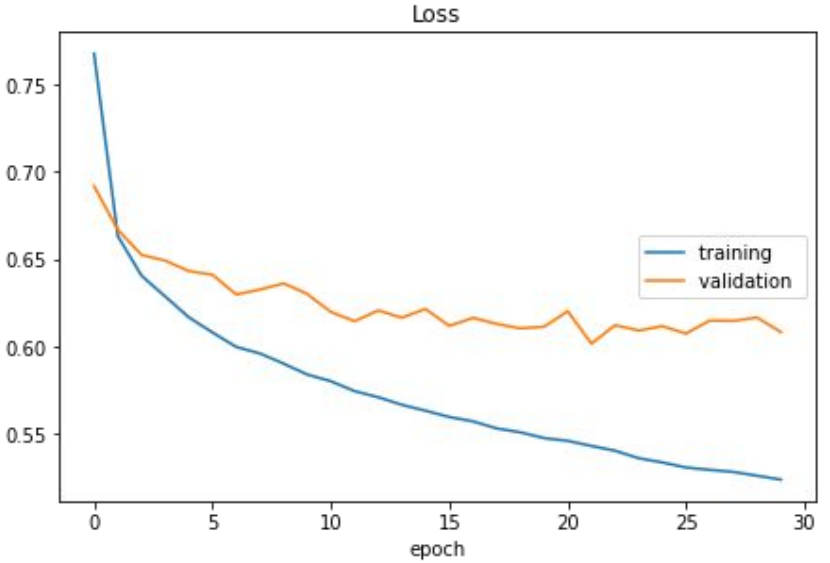
torchvision

fasterrcnn_resnet50_f

pn pretrained

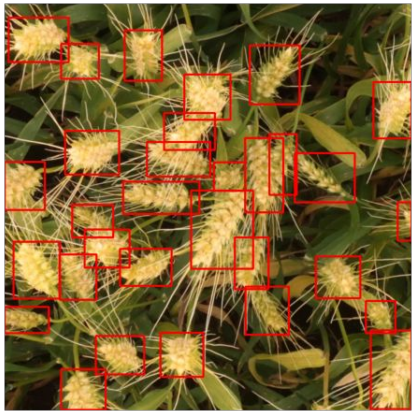
optimizer : SGD

PyTorch




```
training         (min:    0.524, max:    0.768, cur:    0.524)
validation       (min:    0.602, max:    0.692, cur:    0.608)
```

Test sample

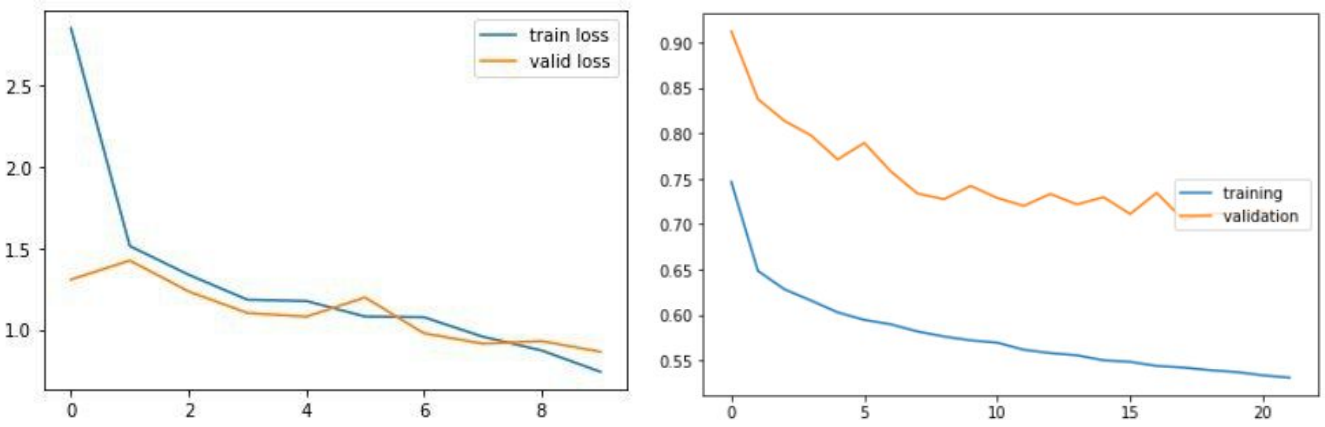


IoU: 0.6882 on test data
(Top 47% on Leaderboard)

DATA AUGMENTATION

Using

- Flip
- Gaussian Blur (kernel size=3, 5, 7)
- Canny edge detector (search edges in a picture)



Your questions please

Skoltech

thx.

Skoltech