Solution in a nutshell. 2th Public LB / 4th Private LB

posted in NOAA Fisheries Steller Sea Lion Population Count 8 months ago

32

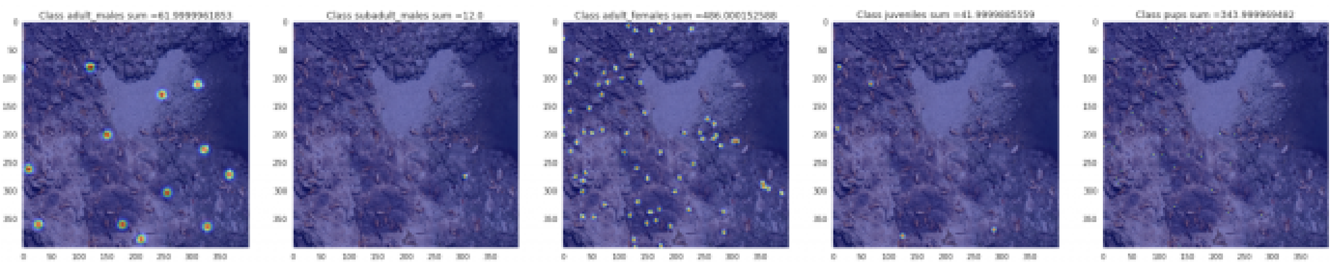
Here I will give a short description of the approach of our team (me and DmitryKotovenko).
The source code is available on GitHub https://github.com/asanakoy/kaggle_sea_lions_counting.

Preprocessing

Thanks to @Radu Stoicescu for his blob detection to get corrected counts from dotted images.

GT count for each tile was generated as a sum over heatmap (to overcome cases with lions on the border of the tile). On top of each lion we put a Gaussian with a standard deviation heuristically estimated by calculating the smallest distance between lion on the image.
We set the standard deviation to be 50 at least for each Gaussian and adjust it according to size of the animals from different classes (multiplied by 2 for adult males and by 0.5 for pups).

GT Gaussians for tile



Model and Training

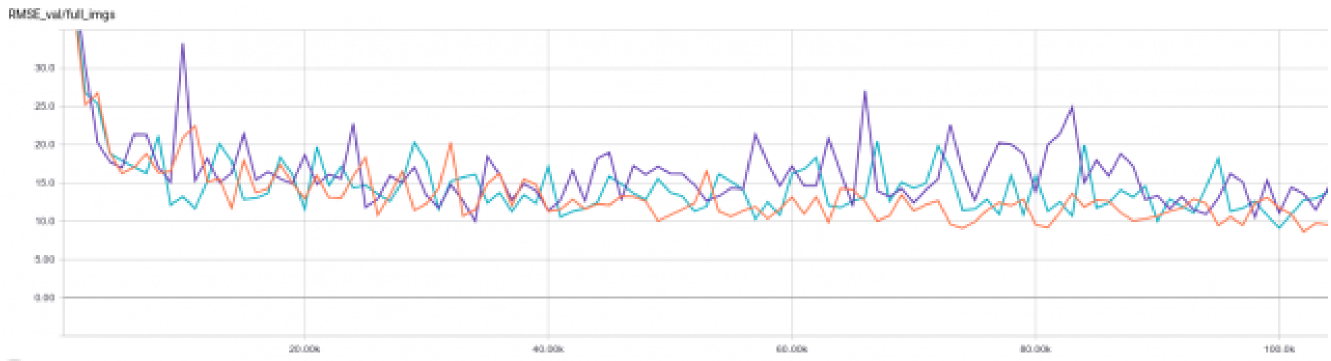
Our model incarnates **regression for 5 classes on tiles of the images**. (In similar spirit as the approach of @outrunner)

Inception Resnet v2 pretrainedon Imagenet.
We substituted the last layer with 256-way FC layer + dropout + 5-way FC layer on top. + RMSE loss. Then we fine-tuned the model on 299x299 image tiles with Adam optimizer.

Augmentations: random rotation on 90/180/270 grads, random flip left-right, bottom-up.

Scale augmentations: one model without them, one model with 0.83-1.25 random scaling, one model with 0.66 - 1.5 random scaling.

RMSE on val for 3 best models



Testing

During test we made predictions up to 5 times for each model using different shifts of the tiles in the image. Test images were downscaled in 0.4-0.5 times.

The final ensemble was made by averaging all the predictions.
Private LB RMSE: *13.18968*
Public LB RMSE: *13.29065*

Applying further postprocessing as suggested by @outrunner, could improve results.
Just increasing the number of pups by 20% gives a huge improvement:
Private LB RMSE: *12.58131*
Public LB RMSE: *12.75510*

Some negative experiments

We labeled some images from train set according to scale and trained a CNN to regress a scale of the image. This could unify all the images to have the same approximate size of the lions of corresponding classes and simplify the CNN training to count animals.

But it didn't work out. I reckon, the reason is the high variation in terrain and inability to estimate scale of objects if you look at them within a small spatial context (even with my own eyes).

eagle4 • (303rd in this Competition) • 6 months ago • Options • Reply 0

@ Artem, Thanks a lot for sharing the code. I am still intrigued by your solution which seems to combine a segmentation and a regression in the same model. I have a question since I am not tf fluent:

As far as I understand, you take the train images, apply the gaussian on the train pictures (are these gaussian pictures a secondary output or the input in your model?) and you do a regression using a resnet like neural network on them.

Can you re-explain please the part with the Gaussian kernels and how these gaussian kernels works at prediction time?

Thanks in advance for your help.

Artem.San... • (4th in this Competition) • 6 months ago • Options • Reply 2

Hey @eagle4. Gaussians are used to estimate the ground truth number of lions on the tile. Gaussians are not used during inference. Putting a gaussian on top of each sea lion allows us to cut this lion with the tile border and still have a fraction of the lion in each adjacent tile. Otherwise, we would have to assign this lion to the only one tile.