How to fail hosting data science contests with images

Evgeny Nizhibitsky
Moscow State University
nizhibitsky@pm.me

Artur Kuzin
Lead Data Scientist, Dbrain
Moscow Institute of Physics and Technologies
kuzin.artur@gmail.com

Abstract

Recently, more and more data science contests are hosted that provide images with assigned labels or masks as the main source of truth to train models with. Nevertheless the images provided can contain even more truth than expected by the host that leads to not very random splits and thus uninformative cross-validation scores, partially private labels recovery and even close to 100% accuracy on the test split what is sure not to be the outcome expected. We review several examples of such contests hosted within past years on some of the top competition platforms we took part in and got a competitive advantage because of data leaks found, analyze popular errors organizers usually make and discuss how to avoid them in future.

1 Introduction

Since the widely-known Netflix data science competition was hosted, a lot of web platforms with this kind of competitions have been founded: Kaggle, DrivenData, Topcoder (mostly known for hosting coding challenges), challenger.ai and others. As the result we are having several ongoing data science competitions at any time now and they've become a noticeable alternative to public datasets in the terms of approaches benchmarking. Several popular open source software like vowpalwabbit or xgboost have gathered a big part of their glory being important parts of Kaggle winners' solutions.

As for now almost all the "table data" competitions have overcome most of the typical errors with data preparation leading to data leaks but, as we are going to discuss later in this paper, image-based competitions are still vulnerable to simple data explorations what leads to a variety of outcomes from unfair score improvements to total private targets recovery and full contest stop for a long period.

2 Contests and failures

In this section we are going to review several examples of image-based data science competitions hosted during last year on the platforms mentioned above to summarize the most popular errors that the competitions organizer make while hosting them that leads to overly optimistic contenders scores on the score board in the best case and competition stop in the worst scenario. Running ahead, it is worth noting that we have counted zero competitions with meaningful data preparation weaknesses absence among all the contests we've participated during the period covered.

2.1 "Planet: Understanding the Amazon from Space"

The data provided consisted of multi-labeled satellite images. As it was found out by several top scoring teams, the image patches could be grouped into several big mosaics where each of them would have samples from both train and test subsets.

Preprint. Work in progress.

The leak found leaded to score improvements based on two-stage algorithms that incorporate true labels data from the neighboring tiles. This is sure to be against the initial organizers' will to fully automate the forests analysis that eliminates two-stage approaches with partially labeled tiles.

2.2 "Pathological Image Segmentation"

Square images sized 500 pixels with labeled binary masks formed the original dataset.

As "Amazon" competition had just finished another attempt to re-create the mosaic was taken and it succeeded. It was found that the original images had 1000 pixels wide except several test images that appeared to be just rotated crops from the original train images. As the result the participants had both non-informative CV scores (as small pieces go to both train/val splits) and private scores (overfitting on the train could result into LB improvements because of train crops leaked into test).

2.3 "N+1 Fish, N+2 Fish"

Hundreds of labeled video sequences were initially offered to analyze. The aim of the competition was to reproduce the sequences of the fish species appearing on the videos. During video exploration it was found that similar frames are distributed among different video clips.

After video frames decomposition and analysis it occurred that all the clips had been randomly cropped from some longer sequences. As the result the contest score can be improved by partially test labels algorithmic recovery based on common subsequences with the train clips.

2.4 "Detecting Abnormality for Automated QAs"

The aim of the competition was to detect the algorithmically added dirt splashes on the images with "clean" reference images also provided. The main flow of the contest was in the alogithm the ogranizers used. Simple data exploration could show that generated dirt patterns were shifted compared to the "true" binary dirt mask that were attached to the train subset.

The winners' solution heavily exploited that feature via adding additional scale transforms to the model pipeline while having only basic non-pretrained U-Net segmentation network as an approach backbone compared to other top solutions with complicated architectures without masks fixing.

2.5 "Airbus Ship Detection Challenge"

This is the most recent competition hosted on Kaggle that also has the most noticeable leak in the image contests so far. The data provided consists of 768-sized satellite image tiles with (mostly) cargo ships on them. With all the experience from other contests it was necessary to try matching ships on different tiles and compiling the mosaic if the former succeeds.

The exploration leaded to astonishing conclusion – all the tiles had been cropped from some big mosaics with 256 step, what means that each 256 sub-tile of the 768 tile has 9 copies in train and test splits and each test mask is almost guaranteed to be reproducible from common train sub-tiles masks.

Leak-exploiting solution was compiled and got close to 100% score. As the result the competition was stopped and the process of gathering brand new test data was initiated.

3 Conclusion

Based on the competition failures we've discussed we propose a new check list for the organizers to minimize the data leaks in their image-based competitions:

1. If you crop samples from big images or crop clips from some long videos, make sure that the train and test splits are spatially or timely spared. In no case let train/test samples overlap.

   Otherwise some participants will waste time recreating the original mosaic or sequences while you will have useless approaches improvements they would create after that.
2. If you have only small amount of images, don't try to chop them to increase it in times — some will waste time to revert the process, some will fail trying to make fair CV.

3. If you have only small amount of images, don't try to crop new examples programmatically — the reverse-engineering of this process is sure to happen and it will happen.
4. If you generate the image or masks samples by task design, recheck several times that the output of the generator is really what you expect it to be.

References

[1] Kaggle: Your Home for Data Science. Kaggle Inc, https://www.kaggle.com/.

[2] DrivenData. DrivenData, https://www.drivendata.org/.

[3] Design & Build High-Quality Software with Crowdsourcing. Topcoder, https://www.topcoder.com/.

[4] Planet: Understanding the Amazon from Space. Planet on Kaggle, https://www.kaggle.com/c/planet-understanding-the-amazon-from-space.

[5] Konica-Minolta Pathological Image Segmentation Challenge, Konica-Minolta on Topcoder, https://community.topcoder.com/longcontest/?module=ViewProblemStatementrd=16950pm=14622.

[6] N+1 fish, N+2 fish. The Nature Conservancy on DrivenData, https://www.drivendata.org/competitions/48/identify-fish-challenge/.

[7] Detecting Abnormality for Automated Quality Assurance. Konika-Minolta on Topcoder, https://community.topcoder.com/longcontest/?module=ViewProblemStatementrd=17070pm=14799.

[8] Airbus Ship Detection Challenge. Airbus on Kaggle, https://www.kaggle.com/c/airbus-ship-detection.