Data Distillation for Traffic Sign Detection*

Alexey Popov[1], Vlad Shakhuro[1], and Anton Konushin[1,2,3]

[1] Lomonosov Moscow State University, Moscow, Russia
[2] NRU Higher School of Economics, Moscow, Russia
[3] Samsung AI Center, Moscow, Russia
{alexey.popov,vlad.shakhuro,anton.konushin}@graphics.cs.msu.ru

Abstract. This work is devoted to the traffic sign detection on images using deep learning methods. We focus on the problem of detector transfer to new datasets with different road signs. We present an algorithm for distilling a set of unlabelled data to select the most informative images to be labeled. This method allows to significantly reduce the amount of data labeling with a small decline of detector performance.

Keywords: Data distillation, Traffic sign detection, Detector adaptation.

1 Introduction

Modern object detection methods are based on deep learning methods. Deep neural network require large labelled datasets to be trained. Labelling large enough datasets is usually difficult and expensive.

Consider traffic sign detection task. Traffic sign recognition system which is used, for instance, in self-driving cars, should be universal and work in several countries. Traffic sign is a standardized object, but it looks a little different in different countries. Moreover, there exist unique signs in some countries. Traffic sign detection system should be somehow trained on different datasets on several datasets and be applicable in different countries. Another problem, which we explore in this work, is training sample size. Nowadays, there is no common opinion in research community on how much data is sufficient to train a traffic sign detector.

In the first part of our work we explore traffic sign detector finetuning. We use two large traffic sign datasets RTSD (Russian signs) and TTK100 (China signs) to find out how detector performance changes with and without pretraining, how training sample size affects performance.

In the second part of our work we explore several methods which allow to reduce training sample size. We explore random sampling and propose two new methods for data filtering. First method requires labelled training sample and uses detector loss to filter out uninformative frames. It chooses the most informative frames and works better than random sampling.

* Publication is supported by RFBR grant 19-07-00844.

Second proposed method regresses a number which describes how informative frame is. This method chooses most informative frames without any labelling.

2 Related work

Modern object detection methods use deep neural networks. They can be divided into several types.

First type, two stage detectors are represented by Faster R-CNN method [8] and its' successors. These methods consist of two parts: object location hypotheses generator and object-background classifier. Such algorithms work quite slowly, but usually achieve best detection quality.

Second type is single stage detectors. YOLO [11], SSD [1] and other methods aim to simultaneously generate object location hypotheses and classify them. These methods achieve higher inference speed at price of lower quality. Such detectors are widely used in real-time applications.

In addition to basic object detection methods there exist methods like RetinaNet [2] that improve baseline detection methods using pyramid of features and loss function designed specially for object detection.

All detectors require a large amount of data for training. For example there are several data sets available for the task of detecting road signs [3, 4]. There are rare and frequent classes of road signs in data sets.

Conventional machine learning approaches are not suitable for recognizing rare road signs. There are special methods are offered for processing rare road signs, for example, based on generating synthetic data [9, 10]. Early approaches to generating synthetic data used heuristic methods based on computer graphics [12]. These methods contained some parameter selection algorithms for creating synthetic samples. Methods for evaluating the quality of target data samples were also an important part of them. Currently, the main direction of methods for generating synthetic data collections. It is to use the idea of the generative adversarial network (GAN) [13], which allows us to create high-quality synthetic data sets for the task of road signs detecting [14].

Data augmentation algorithms are also available to increase the number of rare data. For example [15]. In this article propose augmentation strategy: patches are cut and pasted among training images where the ground truth labels are also mixed proportionally to the area of the patches. By making efficient use of training pixels and retaining the regularization effect of regional dropout.

But on the other hand, with a large amount of data, detection algorithms take a very long time to learn. Therefore, there are algorithms that reduce the amount of data for training. However, the quality of the algorithm remains the same. This class of algorithms includes data distillation algorithms.

Data distillation is a relatively new method in computer vision. In detection task there is method described in [5]. This article propose method that ensembles predictions from multiple transformations of unlabeled data, received using single model, to automatically generate new training annotations and, in particular, is applied to the detection task.

There are method based on the idea of reducing training data by choosing the most informative examples. For example, in [6] proposed method, using the adapted loss function of the target algorithm for obtaining complexity value of frame. Frames are sorted by complexity and the most complex ones are selected for training target algorithm. It is possible to significantly increase the quality of the detection algorithm in the problem of detecting road signs.

Also there are method using two different neural network for data distillation. First neural network is teacher and second is student. For example in [7] this approach using for data distillation approach to learning optical flow estimation from unlabeled data. The approach distills reliable predictions from a teacher network, and uses these predictions as annotations to guide a student network to learn optical flow. This approach is data-driven, and learns optical flow for occluded pixels. This enables to train model with a much simpler loss function, and achieve a much higher accuracy.

3 Training data filtering

An important task is to select the most informative images for finetuning. Opposite, finetuning of the detector is also possible on a full collection of data, but it is time-consuming. Therefore, we offer our own methods for selecting the most informative frames. In our work, we consider several such methods and show that in some cases it is possible to increase the quality of the algorithm. The practical purpose of the methods is to select data from an unmarked collection for further markup. The selected data should be the most informative for the target algorithm and the algorithm must achieve high quality performance on these test data.

3.1 Random sampling

Random sampling is the simplest method for data filtering. Let X be a set of data and X_i is image from dataset. Let y be a vector witch consist of 0 or 1 with size like first X dimension, $|y| = n = |X|$, q - a selection parameter that defines the part of data to select. Next, we require that $\sum_{i=0}^{n} y_i \sim n * q$ and than $Y = \{X_i | y_i = 1\}$. Thus we have chosen a random part of the images defined by the q parameter and we will continue to call this technique RS (Random sample).

3.2 Data filtering using loss ranking

Based on the assumption that the loss function of the binary detector determines the complexity of the frame. The loss function determines the complexity of the entire frame, regardless of the number of characters on this frame. The loss function determines the difficulty of detecting signs on a frame without reference to classes of specific signs on this frame. We can construct an algorithm for selecting the most complex frames. In this paper, we propose an algorithm that can be used to estimate the complexity of frames that already have markup.

1. Loss function $L(f(Y_i), gt(Y_i))$ of the detector algorithm $f(Y_i)$ that was trained on data from the basic domain and Y set of data from the target domain images, where $L(f(Y_i), gt(Y_i)) \in R, Y_i \in Y$, $gt(Y_i)$ is ground truth detections on the Y_i frame.
2. Compute $y : y_i = L(f(Y_i), gt(Y_i))$ and sort y and Y in descending order respectively.
3. Create $\widetilde{Y} = \{Y_i | i \leq n * q\}$.
4. $\widetilde{Y}$ - this is the final dataset that will be used for traning or finetuning.

The use of the algorithm described above has a good effect on further training of the detection algorithm. Because, complex examples are often the most common, which allows you to train the algorithm faster without losing its final quality. It is important to note that the presented algorithm has a significant drawback - requirement for data markup.

3.3 Filtering unlabelled data

Based on the previous point, we can get the complexity of marked frames based on the values of the loss function of the trained algorithm. It is important to note that the main burden on data collection is just imposed by their mark up. Therefore, we need to reduce the number of images that need to be marked up. To reduce the number of markup data, we need to select the most informative frames without any additional information about these frames. It is proposed to train a small neural network that can predict the complexity of frames without using pre-markup.

Let's develop the idea of determining the complexity of images. To do this, we will use a small neural network. In the following algorithms, we used the general architecture Fig. 1 but different loss functions. This neural network model is designed specifically for predicting the complexity of frames. This is a simple convolutional neural network that receives a 300x300 image as input. It has a pair of fully connected layers and returns a single real number at the output.

Based on the results of the previous part, we got markup for all our data that has the original markup. We formalize this approach. We have $\widetilde{X} = \{f(x) | x \in X\}$ and than we use it as training sample. We will use them as a training sample, for an algorithm that we will train to predict the complexity of the image for our detector. In total, we have the following algorithm:

1. Using a trained detection algorithm, we run the marked-up data set of the base domain and get the complexity of each image for our algorithm.
2. Based on the obtained data, we train a small neural network that predicts the complexity of the image for our initial algorithm.
3. Predicting complexity for images from the target data set and forming them in descending order of complexity.
4. Select the appropriate part of the most complex images and mark them up.

Let us get the function $g_{loss}(x) \in R, x \in Y$ where $g_{loss}(x)$ is our algorithm.

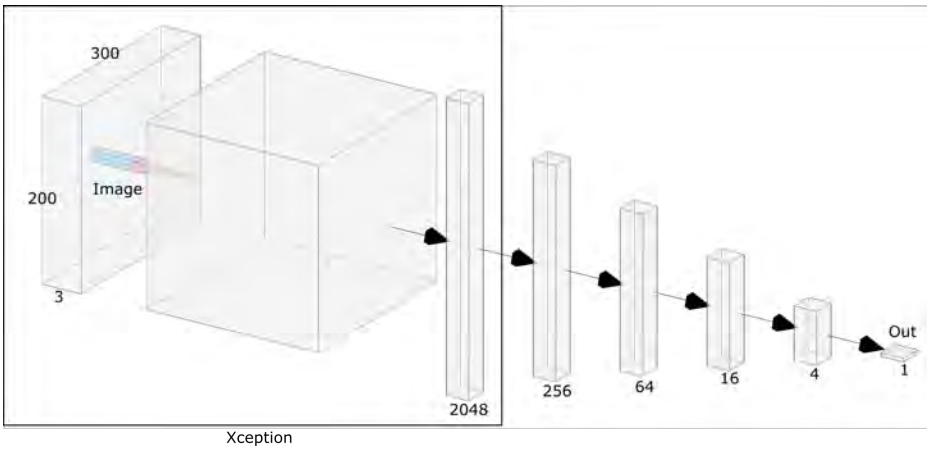

Fig. 1. The architecture of neural network for predicting the complexity of the image.

MSE loss For sample version of our algorithm we use MSE loss. This algorithm will be trained to predict the exact value of the image complexity. Here is an example of calculating MSE for a pair of images:

$$MSE = \frac{(y^2 - \widetilde{y}^2)}{2} \tag{1}$$

Where $y, \widetilde{y}$ are pairs of numbers from the markup and predicted by the network respectively.

DupletLoss As part of this work, we developed the DupletLoss loss function, which allows us to train the algorithm to predict complexity (the information content of the frame for the neural network detector). This loss function based on comparing the complexity of a pair of images from a data set.

Here is an example of calculating the DupletLoss (DL) function for a pair of images:

$$DL = \begin{cases} \varphi(\widetilde{d}, T) & (d * \widetilde{d}) \leq 0 \\ \Psi(\widetilde{d}, T) & (d * \widetilde{d}) > 0 \end{cases} \tag{2}$$

$$\varphi(\widetilde{d}, T) = \begin{cases} 100 + \frac{\widetilde{d}}{4} & \widetilde{d} \leq T \\ 100 + 4(\widetilde{d} - \frac{15}{16} * T) & \widetilde{d} > T \end{cases} \tag{3}$$

$$\Psi(\widetilde{d}, T) = \begin{cases} -100 - \frac{\widetilde{d}}{4} & \widetilde{d} \leq T \\ 100 - 4(\widetilde{d} + \frac{15}{16} * T) & \widetilde{d} > T \end{cases} \tag{4}$$

Where T - parameter of the loss function, $d = y[1] - y[0]$ - the difference between the complexity of the image 1 and 0, which are obtained from markup, and $\widetilde{d} = \widetilde{y}[1] - \widetilde{y}[0]$ - the difference between the complexity of the image is 1 and 0, which are predicted by the algorithm.

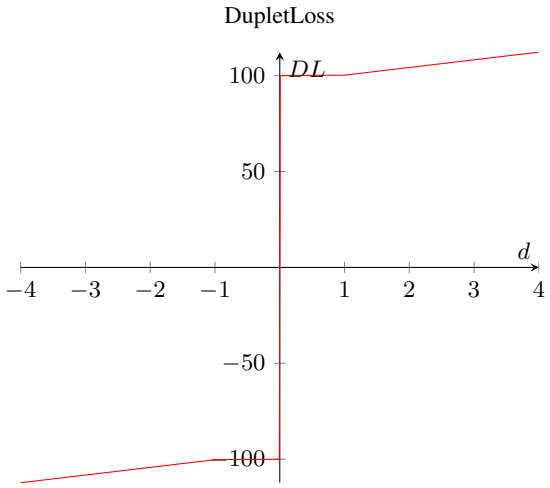

Fig. 2. Graph of the DupletLoss function with $T = 1$.

This function allows you to order images by their complexity, using the gradation of complexity in the markup. This function assumes a penalty for incorrectly ordered images, but it does not impose any restrictions on the difference module, and does not force the network to optimize values for selecting accurate accuracy values.

4 Experimental evaluation

All experiments to train or finetuning the RetinaNet detector were performed on two GeForce GTX 1080 Ti. The choice was made in favor of RetinaNet based on the following advantages:

1. Thanks to the Focal Loss function, the detector demonstrates high quality performance on rare or small objects.
2. High speed of operation.
3. Its accuracy and completeness is not inferior to other modern algorithms that are relevant at the time of writing this work.

For the experiments we used RTSD dataset [4] which contains 59028 images, and TTK100 dataset [3] for finetuning containing 9182 images. In TTK100 there are several major classes of characters Fig. 3.

The metric calculated in all experiments is the AUC. The quantitative interpretation of ROC is given by the AUC indicator-the area bounded by the ROC curve and the axis of the proportion of false positive classifications, where the AUC - area under ROC curve.

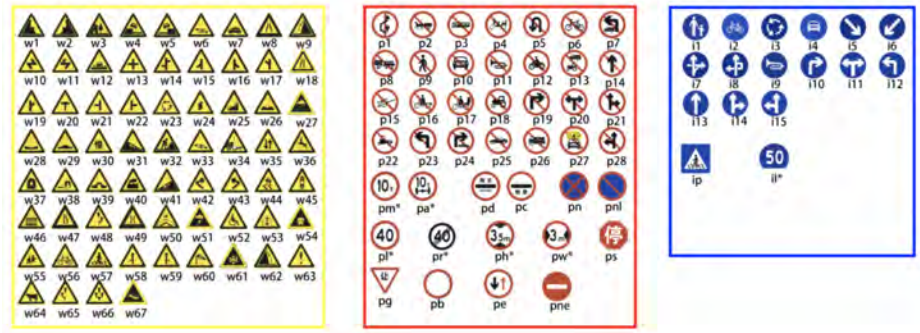

Fig. 3. The main classes of the TTK100 dataset.

4.1 Baseline

In the initial training, experiments were performed to train the detector on mixed data sets in equal proportions, which allowed us to evaluate the detector's ability to finetuning.

In further experiments, we will determine that not only is finetuning as good as learning on a mixed dataset, but in some cases it is significantly better (Table 1).

Table 1. Baseline of our algorithm.

BaseTrain/Finetuning	Group name	Base only	TTK100	Base+TTK100
RTSD train	Average	**0.7110**	**0.8904**	-
	Yellow	0.7059	0.8886	-
	Red	0.7145	0.8910	-
	Blue	0.7043	0.8891	-
Without train	Average	-	**0.8908**	**0.8908**
	Yellow	-	0.8902	0.8873
	Red	-	0.8928	0.8918
	Blue	-	0.8914	0.8889

Thus, based on the results obtained, it is possible to evaluate the further effectiveness of the proposed algorithms.

4.2 Training detector by random data selection algorithm

In order to obtain an experimental estimate (Table 2), which will be compared with the trained algorithms for selecting the most complex examples for the detector, experiments were initially conducted with a random selection of data portions from the main data set.

These results allow us to evaluate the quality of random selection methods in comparison with the selection algorithms of the most complex algorithms.

Table 2. Result of detector on different part of DS (DS is TTK100).

BaseTrain/Finetuning	Group name	$\frac{1}{2}$ of DS	$\frac{1}{4}$ of DS	$\frac{1}{8}$ of DS	$\frac{1}{16}$ of DS	$\frac{1}{32}$ of DS
RTSD train	Average	0.8909	0.8990	0.8827	0.8787	0.8735
	Yellow	0.8890	0.7976	0.7998	0.8703	0.7933
	Red	0.8917	0.8856	0.8851	0.8802	0.8773
	Blue	0.8894	0.8828	0.8811	0.8782	0.8698
Without train	Average	**0.8863**	0.8806	0.8721	**0.8466**	0.7792
	Yellow	**0.8826**	0.8986	0.8936	**0.8699**	0.8908
	Red	**0.8872**	**0.8831**	**0.8751**	**0.8533**	0.7816
	Blue	**0.8845**	**0.8787**	**0.8678**	**0.8426**	0.7727
Count of images	Images	3003	1500	754	374	184
	Yellow	**353**	**148**	**92**	50	18
	Red	**5399**	**2763**	**1433**	628	318
	Blue	**1111**	**534**	**247**	133	71

Fig. 4. On the left, the most complex frame from TTK100, on the right, the simplest frame, based on the loss function ranking algorithm.

4.3 Training detector by data selection with loss ranking algorithm

Results of the algorithm based on data filtered out using an algorithm that uses the original markup (Table 3). This algorithm shows good results compared to random selection, as it improves the balance of the data set.

It increases the number of sample images from rarer classes at the expense of more frequent ones. The results of the algorithm significantly improved on yellow signs, but slightly deteriorated on red signs. This is a consequence of the fact that the dataset has become more balanced. The number of yellow characters in the dataset has increased, but the number of red characters has decreased, which is logical, since initially there are few yellow characters in the data set, but a lot of red ones.

It is experimentally proved that this method improves the balance of samples by adding examples from the rarest classes to the data set. This allows you to improve the quality of the final algorithm for rare character classes.

It is important to note that the deterioration of results in small subsamples is due to the fact that ranking data by the loss function determines the most complex images,

Table 3. Result of detector on different part of DS (DS is TTK100) which are obtained using the algorithm from 3.2.

BaseTrain/Finetuning	Group name	$\frac{1}{2}$ of DS	$\frac{1}{4}$ of DS	$\frac{1}{8}$ of DS	$\frac{1}{16}$ of DS	$\frac{1}{32}$ of DS
Without train	Average	**0.8873**	0.8832	0.8813	**0.8077**	**0.5305**
	Yellow	**0.8854**	0.8797	0.8784	**0.7216**	**0.3957**
	Red	**0.8879**	**0.8839**	**0.8821**	**0.8218**	**0.5814**
	Blue	**0.8869**	**0.8820**	**0.8810**	**0.7386**	**0.4538**
Count of images	Images	3000	1499	749	374	186
	Yellow	**480**	**261**	**112**	48	25
	Red	**5964**	**2721**	**1237**	563	288
	Blue	**1232**	**612**	**288**	133	54

which are the most noisy. Due to this, the quality of the algorithm on small parts of the dataset significantly decreases.

Fig. 5. On the left, the most complex frame from TTK100, on the right, the simplest frame, based on the MSE ranking algorithm.

4.4 Training detector by data selection without marking algorithm

According to the results of experiments (Table 4), you can see that the quality of the model in some cases improves, but at the same time, with a very small number of images, this method sometimes produces outliers, which we discussed in the previous part. We'll talk about what you can do when you need to select very few images later. If you pay attention to how the sample balance has changed, it is obvious that this method has a positive effect on this parameter in the data set.

For cases where there are some outliers in small portions of data, you can try to randomly select portions of data from the sample in which such outliers are not observed, for example, we conducted experiments with a random selection of part of the data from a larger one, on which the algorithm showed some improvements (Table 5).

It is important to note that a very important advantage of our method is the improved balance of training data compared to random selection.

Table 4. Result of detector on different part of DS (DS is TTK100) which are obtained using the algorithm from 3.3.

BaseTrain/Finetuning	Group name	$\frac{1}{2}$ of DS	$\frac{1}{4}$ of DS	$\frac{1}{8}$ of DS	$\frac{1}{16}$ of DS	$\frac{1}{32}$ of DS
Without train	Average	0.8858	0.8814	0.8716	0.7899	**0.6938**
	Yellow	0.8797	0.8721	0.8967	**0.7887**	**0.7630**
	Red	0.8867	**0.8831**	0.8748	**0.7917**	**0.7567**
	Blue	0.8857	0.8781	0.8661	**0.7861**	**0.6842**
Count of images	Images	3001	1501	751	376	188
	Yellow	**423**	**249**	**146**	93	55
	Red	**5779**	**2688**	**1343**	603	328
	Blue	**1238**	**574**	**265**	105	38

Table 5. Result of detector on different part of DS (DS is TTK100) which are obtained using a combination of random selection and algorithm 3.2 or 3.3 (based on $\frac{1}{8}$ DS called BDS), which is compared with DupletLoss.

BaseTrain/Finetuning	Group name	$\frac{1}{2}$ of BDS with 3.2	$\frac{1}{2}$ of BDS with 3.3	$\frac{1}{4}$ of BDS with 3.2	$\frac{1}{4}$ of BDS with 3.3	$\frac{1}{16}$ of DS with DupletLoss
Without train	Average	0.8394	0.7923	0.7309	0.7779	**0.7833**
	Yellow	0.7657	0.7791	0.6926	0.6919	**0.7852**
	Red	0.8456	0.7945	0.7406	0.7826	**0.7845**
	Blue	0.8269	0.7895	0.7077	0.7760	**0.7800**
Count of images	Images	374	374	187	187	376
	Yellow	**65**	**64**	**39**	36	55
	Red	**656**	**668**	**310**	370	614
	Blue	**155**	**138**	**80**	55	108

5 Conclusion

In this paper, we consider the problem of training a traffic sign detector and the related problem of the detector's demand for the amount of training data. Considering algorithms for detecting road signs, we offer a number of algorithms for selecting the most informative frames with different approaches. An approach with pre-markup of data and an approach without pre-markup of data. Thanks to the algorithms proposed in this paper, it is possible to reduce the amount of data required for training the detector. However, the quality of the final algorithm does not deteriorate.

References

1. Liu W. et al. Ssd: Single shot multibox detector //European conference on computer vision. – Springer, Cham, 2016. – C. 21-37.
2. Lin T. Y. et al. Focal loss for dense object detection //Proceedings of the IEEE international conference on computer vision. – 2017. – C. 2980-2988.
3. Zhu Z. et al. Traffic-sign detection and classification in the wild //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016. – C. 2110-2118.

4. Shakhuro V. I., Konouchine A. S. Russian traffic sign images dataset //Computer optics. – 2016. – T. 40. – V. 2. – C. 294-300.
5. Radosavovic I. et al. Data distillation: Towards omni-supervised learning //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2018. – C. 4119-4128.
6. Sofiyuk K.: Neural network model for object detection in images //Lomonosov-2018. - 2018.
7. Liu P. et al. Ddflow: Learning optical flow with unlabeled data distillation //Proceedings of the AAAI Conference on Artificial Intelligence. – 2019. – T. 33. – C. 8770-8777.
8. Ren S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks //Advances in neural information processing systems. – 2015. – C. 91-99.
9. Chigorin A., Konushin A. A system for large-scale automatic traffic sign recognition and mapping // ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. — 2013. — no. II-3/W3. — P. 13–17.
10. Faizov BV, Shakhuro VI, Sanzharov VV, Konushin AS. Classification of rare traffic signs. Computer Optics 2020; 44(2): 237-244. DOI: 10.18287/2412-6179-CO-601.
11. Redmon J. et al. You only look once: Unified, real-time object detection //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016. – C. 779-788.
12. Moiseev B. et al. Evaluation of traffic sign recognition methods trained on synthetically generated data //International Conference on Advanced Concepts for Intelligent Vision Systems. – Springer, Cham, 2013. – C. 576-583.
13. Goodfellow I. et al. Generative adversarial nets //Advances in neural information processing systems. – 2014. – C. 2672-2680.
14. Shakhuro V., Faizov B., Konushin A. Rare Traffic Sign Recognition using Synthetic Training Data //Proceedings of the 3rd International Conference on Video and Image Processing. – 2019. – C. 23-26.
15. Yun S. et al. Cutmix: Regularization strategy to train strong classifiers with localizable features //Proceedings of the IEEE International Conference on Computer Vision. – 2019. – C. 6023-6032.