Building an algorithm for automatic recognition of images on the problem of recognition of loyalty cards

Anonymous Authors[1]

Abstract

In this article we describe a case of applying deep neural networks for solving the problem of image classification on the example of recognition of loyalty cards. The case was to classify the cards by their images.

The case was complicated by the large amount of data, namely more than 200,000 classes, which were divided into 3 different types, for each of them a separate model was trained. The total number of images exceeded 10 million objects, which is comparable to one of the largest image datasets: ImageNet

We present our approach for solving this problem, as well as the results on validation datasets. We also compare different convolutional Neural Networks architectures, their perfomance according to our case. We discuss the best quality model for our case - MobileNet network and compare it with others on three various datasets.

1. Introduction

In this article, we describe a practical example of using deep neural networks for the problem of multiclass object classification. The purpose of the work was to identify the brand that owns a loyalty card by its photograph. To solve the problem, we tested our neural network architectures, and also used well-known neural networks, training them from scratch or using the Transfer Learning (**??**) approach. Various well-known models have been tested, such as Resnet (**?**), VGG16 (**?**), Inception V3 (**?**), InceptionResNetV2 (**?**), NASNet (**?**), DenseNet (**?**), MobileNet (**?**), MobileNetV2 (**?**). The best quality in the validation dataset was demonstrated by the MobileNet architecture. Comparative assessments were made for each of the algorithms using such metrics as Accuracy, Precision and Recall. The MobileNet model showed the best results that we present in this article. The paper also describes the process of preparing training, test samples and the learning process.

2. Formulation of the problem

We had a business task of automatically recognizing the brand of a loyalty card from its image. The approximate amount of data was estimated as 150,000 photos per day. The total number of possible brands exceeded 230,000. We proposed to consider this task as a machine learning task, namely, a multiclass classification problem. Considering recent research in the field of Computer Vision and Deep Learning, we have chosen models of deep convolutional neural networks showing leading results at data analysis competitions such as ImageNet (**?**).

3. Data

The training dataset consisted of a set of tagged photos of loyalty cards. For each loyalty card, the answer is uniquely defined: the class (or brand) to which it belongs. It should be noted that the number of classes in this task exceeded 230,000 classes. All classes can be divided into 17 types, their distribution is given below:

Table 1. The percentage of types in the total volume of cards

Type	Percent
A	43.47
B	31.67
C	14.76
D	3.69
E	2.97
F	1.14
G	1.05
H	0.88
I	0.13
J	0.08
K	0.01
L	0.013
M	7.05e-03
N	4.73e-03
O	3.52e-03
P	7.71e-04
Q	4.40e-04

As can be seen from the table, the first three types cover more than 88% of cards. In total, these three types con-

tain about 3,000 classes. In other words, 3,000 of 230,000 classes cover more than 88% of the cards. Based on this data, and taking into account the server capacity that we had, we decided to work with types A, B and C, which together account for 3000 unique classes. Thus, we exclude from consideration classes of D-Q. It is noteworthy that we had the technical means to automatically determine type of the card (one of 17), therefore we did not have a corresponding technical task. Each type (A, B, C) was studied and selected separately. On average, one class had from 200 to 400 photos. Each of the photos was colored and had a dimension of 145x93 pixels. In this case, the photo does not always contain only a card, but also foreign objects, such as a table or tablecloth, which made the classification more complicated. Also, the low quality of the image and the presence of a flash light in the photo complicated our task.

Each class was divided into a training and validation sample in a ratio of 2:1. Then sequentially for types A, B and C training of different models was started. Before training, the pixel values were assigned to the range [0;1], and each photo was compressed or stretched to the size of 224x224 pixels.

4. Metrics

In order to assess the quality of classification, there are several generally accepted metrics, including: Accuracy, Precision, Recall, F1-Score, ROC-AUC, PR-Curve. In this task we decided to use Accuracy, Recall and Precision metrics. First, we give an example of Confusion Matrix for a multiclass classification problem:

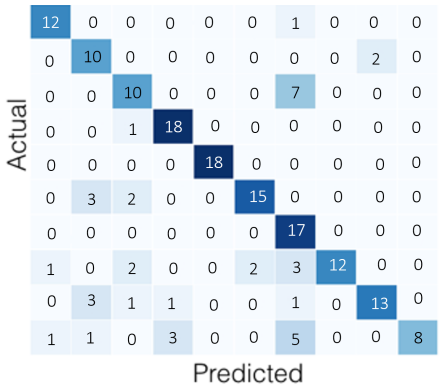

Figure 1. Confusion Matrix for a multiclass classification problem

We describe what each of the symbols TP, FP, TN, FN means in terms of the multiclass classification problem. Their illustration is shown in Fig. 2. For each fixed class N:

1. TP — is the number of objects for which the true class N, predicted class is also N

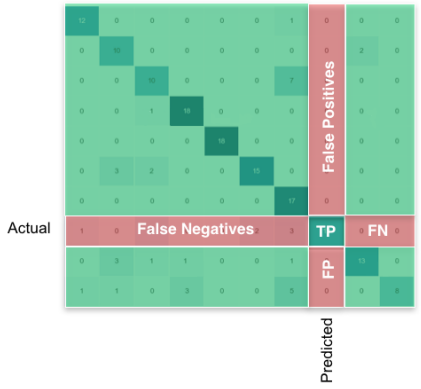

Figure 2. Confusion Matrix for a fixed class

2. TN — is the number of objects for which the true class $M \neq N$, predicted class is also M
3. FP — is the number of objects for which the true class $M \neq N$, but predicted is N
4. FN — is the number of objects for which the true class N, but predicted class is $M \neq N$

- *Accuracy*:Accuracy metrics, traditionally for a binary case, is calculated using the following formula: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$. For the problem of multiclass classification, we can define the metric by analogy to binary case: $Accuracy = \frac{\overline{TP}+\overline{TN}}{\overline{TP}+\overline{TN}+\overline{FP}+\overline{FN}}$, where, for example, $\overline{TP}$ - mean TP for all classes from the training set. However, it should be noted that for a multiclass classification $\overline{TN}$ may be very large. In particular, this happens with a large number of classes. In our experiments on 10,000 photos with the number of unique classes of 2,600 indicator $\overline{TN}$ was equal to 9 500 280. Because of such large values, Accuracy is very close to 1 and is not a correct indicator of the network's generalization ability. So that Accuracy has an interpretable value, we defined $Accuracy = \frac{\sum_{i=1}^{K} TP_i}{N}$, where N — total number of objects in the sample, K — number of unique classes
- *Recall*: Similarly, as we considered Accuracy, Recall will be considered according to the following formula:$Recall = \frac{\overline{TP}}{\overline{TP}+\overline{FN}} = \frac{\sum_{i=1}^{K} TP_i}{\sum_{i=1}^{K} TP_i + \sum_{i=1}^{K} FN_i} = \frac{\sum_{i=1}^{K} TP_i}{N}$. This means that in our problem the Accuracy

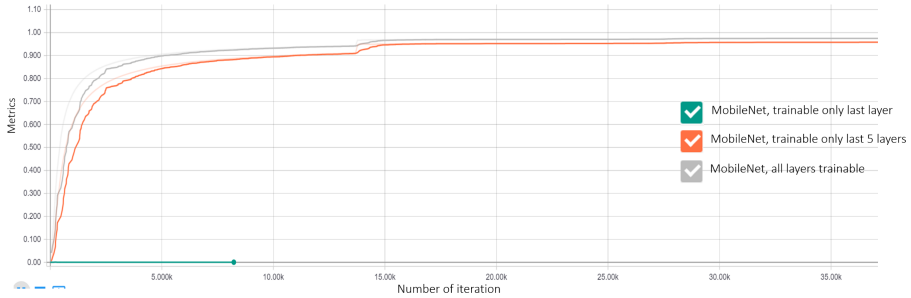

Figure 3. Comparison of different models, the dynamics of the mean Accuracy metric on type A. Comparison of the dynamics of learning models

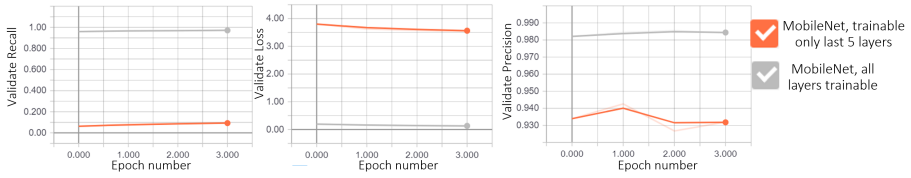

Figure 4. MobileNet metrics dynamics on type A. Comparison of test prediction quality for MobileNet-trained and pre-trained MobileNet

and Recall metrics coincide

- *Precision:* Precision will be considered analogous to the binary case: $Precision = \frac{\overline{TP}}{\overline{TP}+\overline{FP}}$
- *Mean_Accuracy*: data was submitted in batches during training session.Then, for the k-th batch, the Mean_Accuracy metric is calculated by the formula: $Mean\_Accuracy_k = \frac{\sum_{j=1}^{k} Accuracy_j}{k}$. After the completion of an epoch, the counter k is reset to zero.

5. Image classification

To solve the problem we tried three approaches:

1. Building a neural network architecture from scratch
2. Additional training of the pre-trained model, that is, using the Transfer Learning [1]
3. Learning from scratch well-known neural network architectures

In the first approach, several architectures were tested from scratch, but all tested architectures converges slowly or does not converge at all. Based on this, we decided to try various well-known models.

[1]The ability of the system to recognize and apply the knowledge and skills obtained in previous tasks to new tasks or data. For example, knowledge gained from learning to recognize cars can be applied when trying to recognize trucks

As known, convolutional networks show very good quality in classification problems. To achieve high quality, huge amounts of training samples are required, as well as computational power. However, in practice, often, there are problems with insufficient data, in such cases, you can take pre-trained models on a large data set, take the outputs of one of the last layers and use them as a feature vector of images, which learns the standard machine learning algorithm. This approach is called Transfer Learning. In the second approach, we decided to test a series of models using the Transfer Learning approach. However, the models either did not converge or retrained. The fig. 3 illustrates the results of the experiment on learning the MobileNet architecture from scratch, as well as the additional training of only on 1 last layer and last 5 layers.

As we can see from the graph above, the model with fixed weights on all layers except the last one does not learn. At the same time, the model with the additional 5 last layers that are being trained shows a similar learning dynamic with the model being trained from scratch, but it achieves a slightly lower quality on the training set. However, on the validation dataset, the model being trained from scratch has a clear advantage, this is illustrated on fig. 4

From the graph above, we can conclude that the pre-learning model has overfitted. Based on the results of the experiment, we decided to train models from scratch.

Taking into account this information, we decided to train also other existing known neural network architectures from scratch. In order to find out which architectures today show the best results, we turned to the latest achievements in

the field of Deep Learning, namely, we decided to study the results of the annual image recognition competition on the ImageNet data set. The ImageNet project is a large visual database designed for use in studies of the recognition of visual objects. Since 2010, an annual software contest is held in the ImageNet project — ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where various models compete with each other, solving the problem of classification and recognition of objects. As of August 2017, ImageNet contained 14,197,122 images, divided into 21,841 categories. At the same time, a typical category, such as balloon or strawberry, contains several hundred images.

We studied the winning models of different years, as well as models that showed high results on ImageNet. The table 2 shows the comparative characteristics of such models.

Here TOP-1 accuracy means the usual Accuracy metric: the response of the model (the one that has the highest probability) will be compared with the true answer. Top-5 Accuracy means that any of the first five most likely answers must match the true answer. After examining the table with the results of the best models, we decided to try all these models on our data set. All weights in architectures were initialized with weights found during training models on ImageNet. Almost all models converged to Accuracy above 90%. The best results and the fastest convergence was demonstrated by the MobileNet model with the AdaDelta (**?**) optimizer. For each of the A, B, C types, the MobileNet architecture performed better than the others, showing the best quality in the validation sample. The comparisons of learning dynamics of models MobileNet, InceptionV3, ResNet50, Inception resnet v2 are provided on figures 5, 6

As it can be seen from fig. 5, 6 all models, with the exception of Inception v3, converge to approximately the same quality. However, MobileNet 1) in the same number of steps as competing models achieves better quality and comes to a plateau earlier than its analogues 2) for any fixed number of steps N MobileNet passes them faster than analogs. Namely, MobileNet learns about 1.5 times faster than Resnet50 and 2.5 times faster than Inception ResNet2. Figures 5 and 6 refer to type B cards; for type A and C cards we get the same dynamics

We also conducted experiments to reduce the size of training set. The required volume of the validation sample was fixed as 33% of the data provided. Fig.7 illustrates the quality of the MobileNet model on the validation sample for the percentage of the training set: 0.66, 0.5, 0.15, 0.05 cards of type A.

Based on the graphs, the quality of the model on a deferred sample decreases as the training sample is reduced. Since the best quality was prioritised than the learning speed of the models, we decided not to reduce the size of the training sample and leave it at the rate of $\frac{2}{3}$ from source dataset. However, it is worth noting that the optimization of the partition boundary is one of the future problems.

During the verification of the model on the validation sample, it turned out that the original sample had many pairs of duplicate classes, that is, the dataset contained two classes with a different name, and the photos of cards of the same brand corresponded to them. In the process of testing a trained model, several classes were distinguished for which the model was wrong in most cases. Further manual verification of most images on which the neural network was wrong revealed these duplicate classes. After we formed a report on possible duplicate classes, the samples were cleared and the models were re-trained.

After clearing the sample from duplicates, we re-trained our model. Fig. 8, 9, 10 display are screenshots of the comparative learning dynamics of MobileNet on an untreated and duplicate-free datasets.

As can be seen from these graphs, the model has improved its quality and speed of convergence in the training set. For types A and C, the quality on the test sample increased, but for type B, the quality on the test sample fell. We hypothesized that the model began to overfit: by removing duplicate classes, we removed the effect of regularization. It was decided to add noise to the classes in which the model began to make mistakes. By noise we imply pictures from another class. 1% and 10% noise photos were added, but the quality did not improve. Nevertheless, the model trained only one epoch gave better quality than the the model trained two or more epochs. So model achieved best quality during first epoch and then overfited. Therefore, it was decided to train all models one epoch only

6. Results

According to results on validation dataset, MobileNet model was chosen as our final model. The architecture of the MobileNet network is presented in table 3.

Average metrics that we achieve on test dataset are following: Recall=0.92, Precision=0.94. Final metrics of the best model - MobileNet for three types: A, B, C: are presented in tables 4, 5, 6. As it can be seen from these tables, after cleaning the dataset from the duplicate classes, the dynamics of model training has improved: this can be seen from the Precision and Recall indicators on the training dataset.

This effect can also be seen in figures 8, 9 and 10: model achieves better quality on a cleaned sample than a model that trained on an uncleaned sample. However, the quality of the test sample improved only for type A. For type B, Precision decreased, Recall increased, for type C, Recall fell, and Precision remained the same. This effect is most

Table 2. Comparative characteristics of models trained on ImageNet

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	99 MB	0.749	0.921	25,636,712	168
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
NASNetMobile	23 MB	0.744	0.919	5,326,716	-

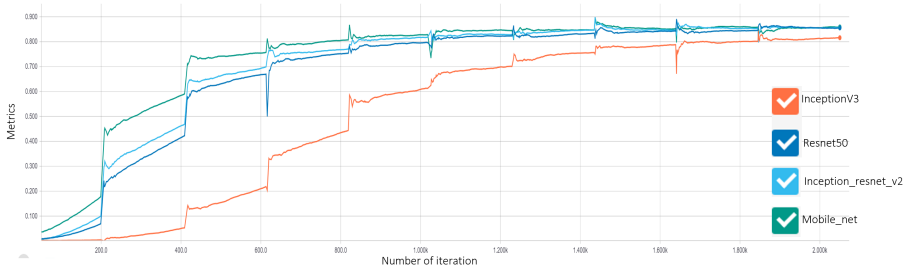

Figure 5. Comparison of different models, the dynamics of the mean Accuracy metrics on type B. Comparison of the dynamics of learning models

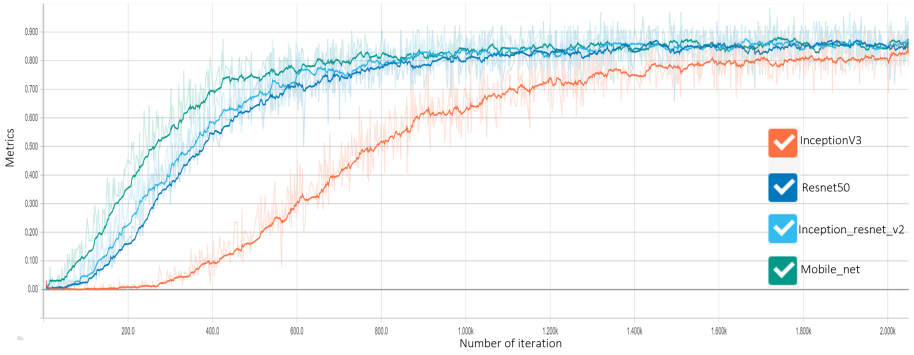

Figure 6. Comparison of different models, the dynamics of Accuracy metrics on type B. Smoothing coefficient 0.8. Comparison of learning patterns

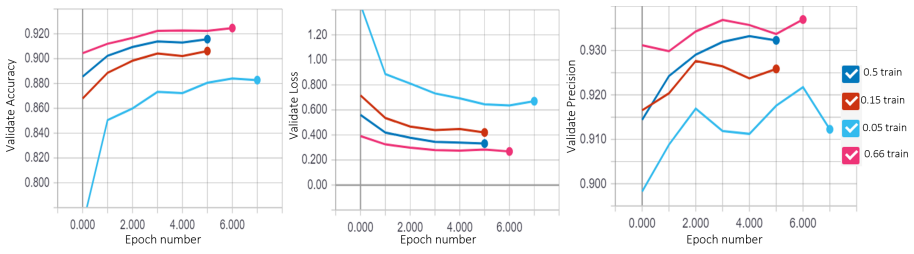

Figure 7. Dynamics of MobileNet metrics on type A. Comparison of test prediction quality for different sample sizes

Table 3. Architecture of MobileNet

Type/Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5 \times$ Conv dw / s1	$3 \times 3 \times 512$dw	$14 \times 14 \times 512$
$5 \times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times \mathrm{N}_{classes}$

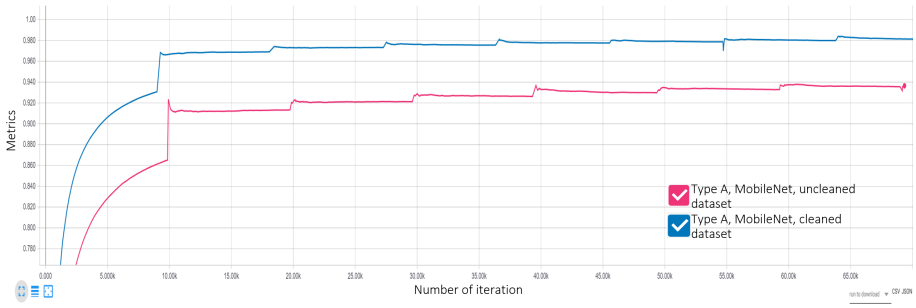

Figure 8. Dynamics of Mean Accuracy for MobileNet model on Type A. Comparison of learning Dynamics on cleaned and uncleaned datasets

Figure 9. Dynamics of Accuracy for MobileNet model on Type A. Smoothing Ratio 0.9. Comparison of learning dynamics on cleaned and uncleaned datasets

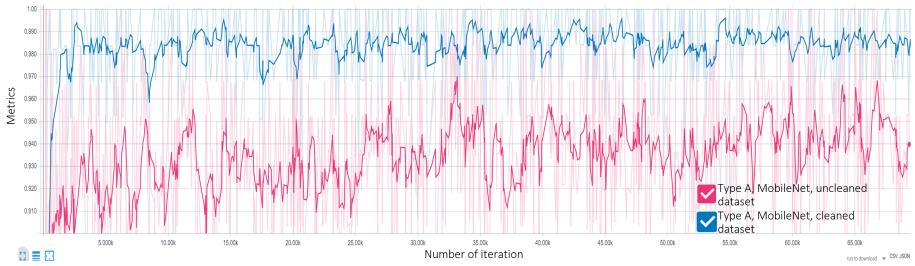

Figure 10. Dynamics of Precision for MobileNet model on Type A. Smoothing Ratio 0.9. Comparison of learning dynamics on cleaned and uncleaned datasets

likely due to the effect of overfitting. After the samples were cleared, the regularizer (duplicate classes) was lost, as a consequence the model began to overfit.

7. Conclusion

We have provided a report on the application of the deep learning approach for the real problem of image classification. We described the entire solution process: data collection, data cleaning, training and testing convolutional neural networks. Following approaches were tried: 1) Training our neural network architecture 2) Using the Transfer Learning approach 3) Learning from scratch well-known architectures. The third approach proved to be much better than the others, surpassing all the models both on the training and on the test samples. We received the final models with the Precision and Recall metrics for almost all types exceeding 90%. The best results on all three types showed MobileNet network As the next steps in our problem, we plan:

1. To study in more detail the regularization effect with uncleaned dataset, i.e when the dataset contains two equal classes with a different name
2. Consider other types of cards that we do not cover in this paper: D, E, F, G and so on. Together, these classes occupy about 10% in the distribution of card types.

Table 4. Quality for type A

DataSet	Before clean		After clean	
	train	test	train	test
Recall	0.93	0.916	0.98	0.943
Precision	0.94	0.945	0.98	0.961

Table 5. Quality for type B

DataSet	Before clean		After clean	
	train	test	train	test
Recall	0.91	0.876	0.98	0.881
Precision	0.93	0.921	0.98	0.907

Table 6. Quality for type C

DataSet	Before clean		After clean	
	train	test	train	test
Recall	0.93	0.945	0.98	0.936
Precision	0.94	0.954	0.985	0.954