Transfer learning for road-based location classification of non-residential property

Lev Mizgirev[1], Edward Galiaskarov[1], Irina Astrakhantseva[1], Sergey Bobkov[1] and Roman Astrakhantsev[2]

[1]Ivanovo State University of Chemistry and Technology, 7, Sheremetevskiy Avenue, Ivanovo, 153000, Russian Federation
[2]National Research University "Higher School of Economics", 20, Myasnitskaya st., Moscow, 101000, Russian Federation
galiaskarov_eg@isuct.ru

Abstract. This article reveals the method of the property classification by its location relative to highways using transfer learning approach. Solving this problem is crucial for non-residential real estate market value assessment automation in the course of market analysis, pre-trial appraisal and other aspects of making managerial decisions in the field of financing and lending. Instead of a standard approach based on models developed using machine learning libraries and programming, this work considers the use of Google's Teachable Machine service. This article examines the aspects of initial data preparation, the use of Teachable Machine for model training and the results obtained. The parameters and results of training classification models in different conditions are presented, the classification accuracy is analyzed. The results obtained generally indicate the validity of this approach and recommends it for solving similar problems.

Keywords: Transport highway, Intra-quarter location, Non-residential property, Road-based location classification, Transfer learning, Teachable machine.

1 Introduction

Machine learning algorithms have been around for a long time and are being actively implemented in the practice of making financial decisions. These include making valuation models of the real estate market. They are more actively used to assess the residential real estate market due to mortgage lending market development and the interest of financial and credit institutions in a high-quality independent and quick assessment of such properties. The market for non-residential real estate is less active, these properties have more unique characteristics in contrast to residential real estate, therefore, data analysis of the market is less developed and is of utmost interest.

One of the important pricing factors, especially for the non-residential real estate market in Moscow, is the location of the facility. All other things being equal, real estate remote from the city center can be sold at a lower price per unit area. Fig. 1 shows a heat map of commercial real estate prices in the city of Moscow. The map was generated using Yandex.Realty service.

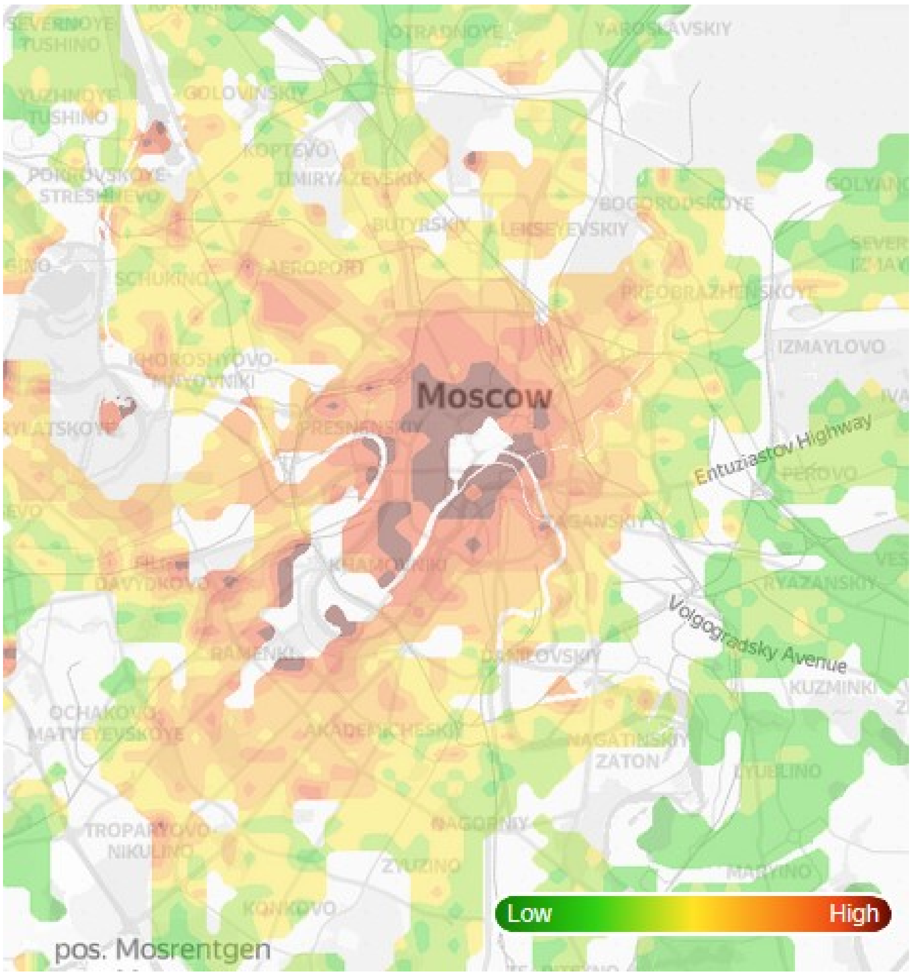

Fig. 1. Heatmap of commercial real estate prices in the city of Moscow.

The property location in Moscow can be defined by its location relative to the administrative districts, city center, distance from the metro, as well as its location relative to roads and highways. As a rule, from the practice of appraisal activities, the cost of property located on the city's main roads, the so-called "red line", exceeds the cost of similar properties located in the block [1-2].

The location analysis of the property relative to the "red line" road is carried out manually based on Yandex or Google maps imagery. As part of a general project to develop an intelligent service for Moscow non-residential real estate price appraisal it is proposed to automate this process using machine learning. The purpose of this study is to develop a model that determines the location of non-residential properties in relation to the city's highways based on its address.

2 Materials and methods

2.1 Determining property location

Based on the business practices of appraisal activities and standards, it is customary to classify the location of the property relative to highways into three categories [3-4]:

– The property is located on the main street (city’s transit or exit roads) or has a direct access to the main street (a, so called, "red line");
– The property is located on the front line of a minor street;
– The property is located on the second building line, or further, from the street lines (or inside the block).

A building is located on the red line if it is adjacent to the city’s main road (map systems usually mark such roads in yellow). An example of such an arrangement is shown in Figure 2.

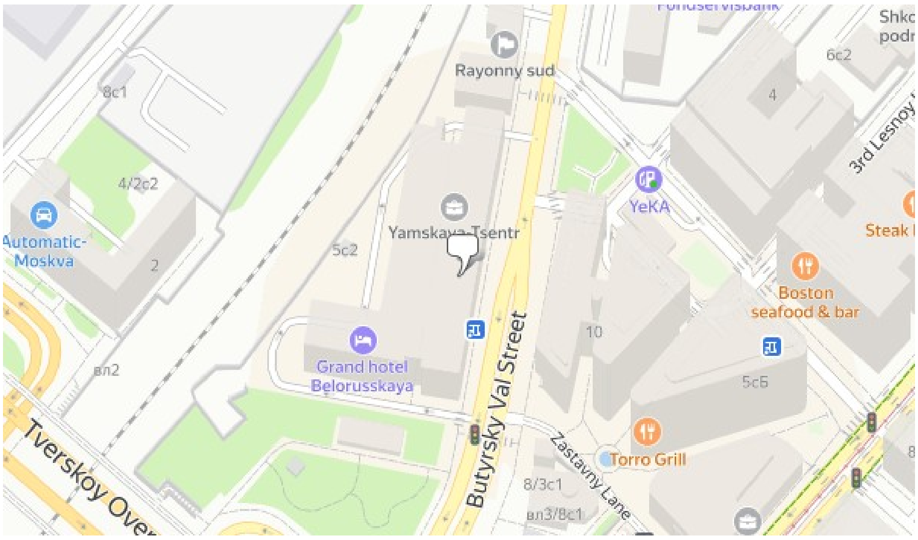

Fig. 2. The marked building is located on the “red line”.

A building located on the front line of a minor road is usually adjacent to a secondary road in a city (map systems usually mark such roads in white). An example of such an arrangement is shown in Figure 3.

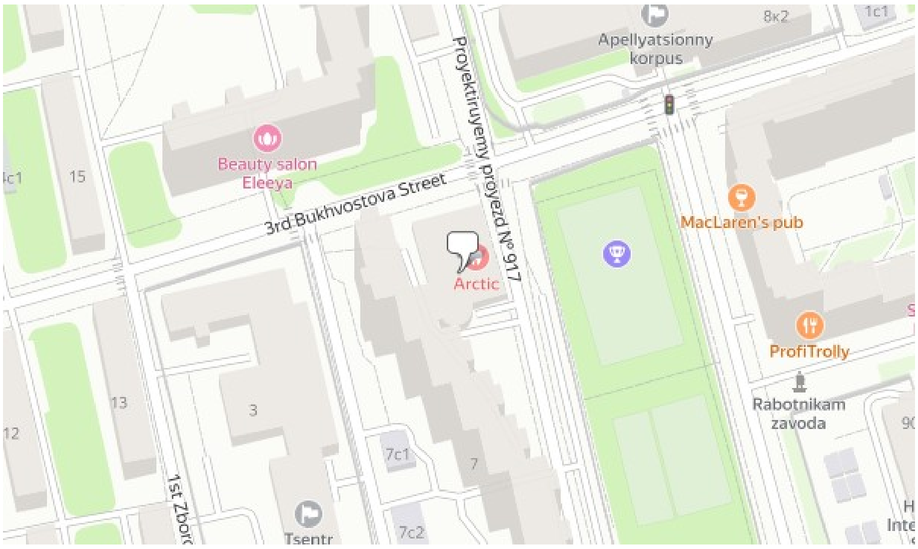

Fig. 3. The marked building is located on the "Front line of a minor road".

A property inside a block is building that does not have access to main or minor roads and is located among other buildings. An example of such an arrangement is shown in Figure 4.

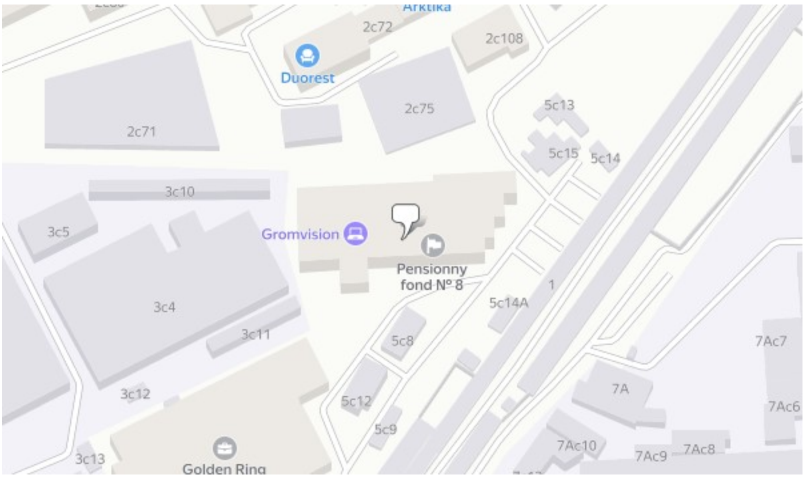

Fig. 4. The marked building is located inside a block.

Thus, our problem can be represented as a classification task. The designations for the previously identified road-based location classes are as follows:

- RLC – "Red line", the property is located on the main street;
- MRC – "Minor road", the property is located on the front line of a minor street;
- IQC – "Inside quarter", the property is located on the second building line, or further, from the street lines (or inside the block).

2.2 Initial data

To solve the road-based property location classification problem, the data collected from the real estate website Realto.ru was used. The website provides users with up-to-date information on supply and demand in the real estate market gathered through the Realto program, created specifically for the Moscow real estate market [5]. At the moment, this database covers the market for the sale and lease of residential real estate in Moscow and the Moscow region, the market for the sale and lease of non-residential real estate (shops, premises for offices, restaurants, etc.), the market for the sale and lease of suburban real estate (land, houses, cottages, for individual housing construction).

The initial dataset downloaded from the website was an MS Excel file and included many features of the property, from which only the address was selected for the task. Using the map service Sputnik.ru (based on OpenStreetMap) [6], the property's address was converted into geographic coordinates. An extract from the dataset after the transformation is shown in Table 1.

Table 1. Extract from the original dataset.

#	Address	Longitude	Latitude
0	13 2-ya Rybinskaya Street	55.788868	37.659927
1	18-2 Tvardovsko Street	55.798820	37.410603
2	2 Zubovskiy Boulevard	55.73767	37.58983
…	…	…	…
4997	23 Shmitovskiy Proyezd	55.755848	37.544495
4998	20-1 Nelidovskaya Street	55.846703	37.436054
4999	15-1 New Arbat Avenue	55.751940	37.592533

Based on the property’s geographical coordinates, map fragment images were obtained from the Yandex.Maps Static API service [7]. The property of interest was located strictly in the center of the image and was marked with a white flag. Examples of such images are shown in Figures 2-4.

As a result, 5000 images with a resolution of 600x450 pixels were obtained. The image size was determined by the capabilities of the software used and was selected in such a way that the analyzed property, as well as the surrounding roads and buildings, would fit on the fragment.

Then the resulting set of images was labeled manually, i.e., each fragment was assigned one of three location class labels. To speed up the labelling process, a dedicated app was developed. The application displayed a map fragment with the analyzed property. The user manually determined the property location ("Red line", "Minor road", "Inside quarter") then, in the drop-down menu, the appropriate location class is chosen and saved. If the fragment turned out to be incorrect (for example, the property was not on the map or the wrong property was marked), then this fragment was labeled "Unknown" for further clarification or exclusion from the dataset. The application source code is provided in [8].

As a result of labelling, 2363 samples and the following class distribution were obtained (Table 2). The significant decrease in the number of samples compared to its initial value (5000) is primarily due to the fact that many analyzed properties had the same addresses.

Table 2. Class distribution in the dataset.

Class	# Samples
Read line (RLC)	978
Minor road (MRC)	466
Inside quarter (IQC)	919

It is well known that deep neural networks can be used to solve the image classification problem [9]. At the same time, such networks require large amounts of data for the convergence of training. As shown in Table 2, the amount of data is not enough to train a neural network well from scratch. The Transfer Learning technique should help us to solve this problem.

2.3 Transfer Learning

Transfer Learning is a machine learning technique in which knowledge gained by a model (usually a neural network) to solve one problem can be applied to solve other similar problems [10].

The formal definition of the description method is described through the concepts of subject area and tasks.

A domain D consists of: a feature space X and a marginal probability distribution P(X), where X = {x1, … xn} $\in X$.

Given a specific domain D = {X, P(X)}, a task consists of two components: a label space y and an objective predictive function f: $x \rightarrow y$. The function f is used to predict the corresponding label f(x) of a new instance x.

This task, defined as τ = {y, f(x)}, is learned from the training set consisting of pairs {xi, yi}, where xi $\in$ X и yi $\in y$. Given a source subject area Ds и and learning task τ_s, a target domain Dt and learning task τ_t, where Ds $\neq$ Dt or $\tau_s \neq \tau_t$, transfer learning helps to improve the learning of the target predictive function, using the knowledge in Ds и τ_s [10].

For example, the knowledge gained by the model that recognizes cars can be used for recognizing trucks. This method makes it possible to achieve high accuracy on a small dataset and with a small number of training epochs. However, it is not without its drawbacks – the model is often subject to overfitting with suboptimal training parameters [11].

Transfer learning is used to classify and cluster text and images. The method is successfully applied in medicine (in classification of diseases by fluorographic images [12, 13]), in translators from sign languages [14] and in game bots [15] (AlphaGo from DeepMind).

One of the implementations of the transfer learning is carried out in the following way: the pretrained network is deprived of the classification layers and new empty classification layers are attached [11]. After that, the model is trained on a new dataset. The stages of the transfer learning are shown in Figure 5.

Transfer learning can be implemented in different ways. One of the typical approaches is to implement the method using programming languages, for example, using Tensorflow and Keras for Python. This method is quite versatile, but requires a mastery of the language and programming experience. Back in 2017, Google developed the Teachable Machine [16], which offers an "out of the box" transfer learning implementation. Conveniently, the service does not require programming and greatly simplifies the process of data preparation, training and deploying models, thanks to its interactive environment and a rich set of parameters.

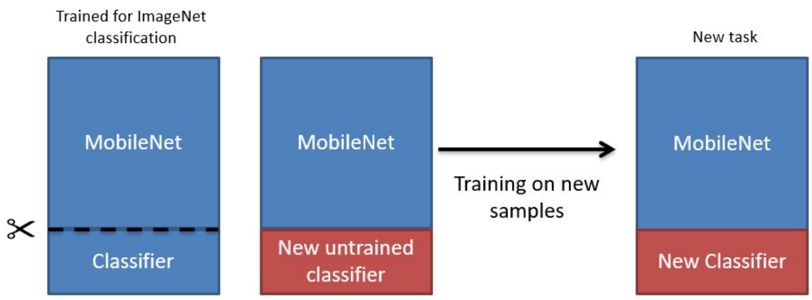

Fig. 5. The main stages of Transfer learning implementation.

2.4 Teachable Machine

Teachable Machine is a service that allows users to create, train and deploy machine learning models based on the transfer learning technique [16].

Users can create models that recognize images and sound. Building and training a model does not require programming. The service makes it easy to export trained models to the TensorFlow and TensorFlow.js libraries and use them in other applications.

The platform is based on TensorFlow.js and uses the MobileNet convolutional neural network pre-trained on the ImageNet dataset as an initial model for image classification [17]. The Teachable Machine uses Adam optimizer (stochastic gradient descent with adaptive moment estimation) and the cross-entropy as a loss function [18].

Thus, the main disadvantage of Teachable Machine is that other pre-trained models, optimizers or the loss functions cannot be used.

The service allows us to easily and quickly create models for our task, set classes for our subject area, load samples into the appropriate classes, select and set model parameters: Batch Size (the number of images used per training iteration); Epochs (the number of learning epochs) and Learning Rate.

Teachable Machine is able to generate model performance reports which include [16]: accuracy per class and confusion matrix on the test samples and a graph of change in model accuracy during training both on train and test sets.

2.5 Model training

Experimenting with each model consisted of three stages: preparing samples, training the model, analyzing the results and exporting the model for further use.

Sample preparation was performed as follows: images for each class were uploaded to the Teachable Machine platform. At the same time, the service automatically randomly selected the loaded samples at a ratio of 85% for the training set and 15% for the test set [16].

The model training involved selecting parameters for training that provide the best classification accuracy. Training parameters selection was carried out as follows: the number of samples (Batch Size) was sequentially selected from the specified range of values [16; 32; 64; 128; 256; 512]. Thus, the number of samples dictated a series of experiments. Then, for each series in range [30; ...;150], the number of learning epochs (Epochs) was fixed, and for each set, the number of learning epochs, the learning rate was fixed from the range [0,0005; 0,001; 0,005]. A total of six series of fifteen experiments were conducted. The results were tabulated and the set of parameters for which the classification accuracy was the best was selected.

3 Results

First model. Initially, the model was trained on a dataset of all three classes. Training was performed according to the method described above. Six series of fifteen experiments were conducted. The best results for each series are presented Table 3.

Table 3. Parameter selection for the first model.

No	Batch Size	Epochs	Learning Rate	Accuracy
1	16	50	0.001	0.62
2	32	70	0.001	0.71
3	64	70	0.001	0.77
4	128	150	0.001	0.85
5	**256**	**100**	**0.001**	**0.86**
6	512	100	0.001	0.83

As you can be seen here, the highest classification accuracy was achieved with the following model parameters: Epochs = 100; Batch Size = 256 and Learning Rate = 0,001.

The model training results for each class are presented in Table 4.

Table 4. Classification accuracy of the first model on the test set.

Class	Accuracy	# Samples
Read line (RLC)	0.89	147
Minor road (MRC)	0.70	70
Inside quarter (IQC)	0.93	138

The overall model accuracy was 86%. The model distinguishes "Inside quarter" (93%) and “Read line” (89%) classes with a fairly high degree of confidence, but the "Minor road" class has a significantly smaller accuracy (70%). The confusion matrix in Fig. 6 also shows that the model distinguishes well between "Red line" (RLC) and "Inside quarter" (IQC) classes (the first 2 elements of the matrix main diagonal are intense), but it can make a mistake when classifying a property as the "Minor road" (MRC) class.

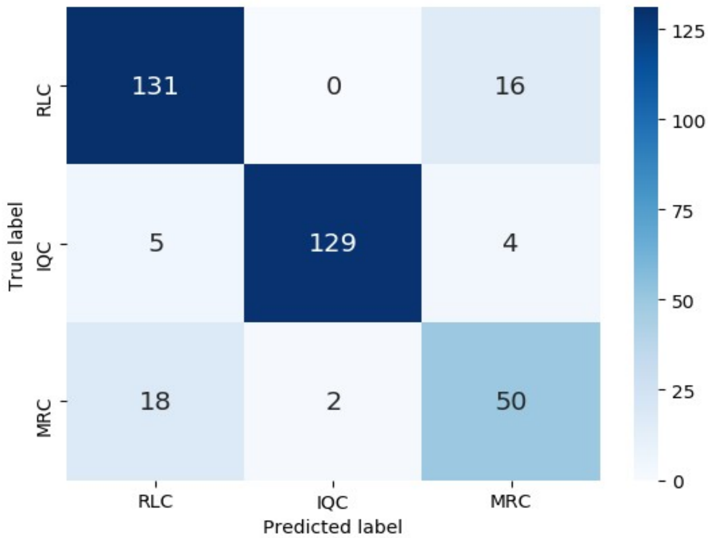

Fig. 6. Model training results on the initial 3-class dataset.

Second model. The probable reason of such difference in accuracy per class might be that the number of samples for the "Minor road" class is significantly smaller than in the other two classes (see Table 3). Now if we balance the number of samples for each class, in our case, by cutting the number of samples for 2 classes to the smallest one). The training parameters for the new model were taken from the first model: Epochs = 100; Batch Size = 256 and Learning Rate = 0,001

The model training results for each class are presented in Table 5.

Table 5. Classification accuracy of the second model on the test set.

Class	Accuracy	# Samples
Red line (RLC)	0.88	69
Minor road (MRC)	0.77	69
Inside quarter (IQC)	0.86	69

The overall model accuracy was 83%. The class accuracy of "Red Line" slightly decreased (88%), and "Inside quarter" accuracy noticeably dropped (86%). At the same time, the accuracy of the "Minor road" class significantly increased (77%). The confusion matrix in Figure 7 shows that there is an intense major diagonal, but the model still confuses the "Inside quarter" (IQC) and "Minor Road" (MRC) classes.

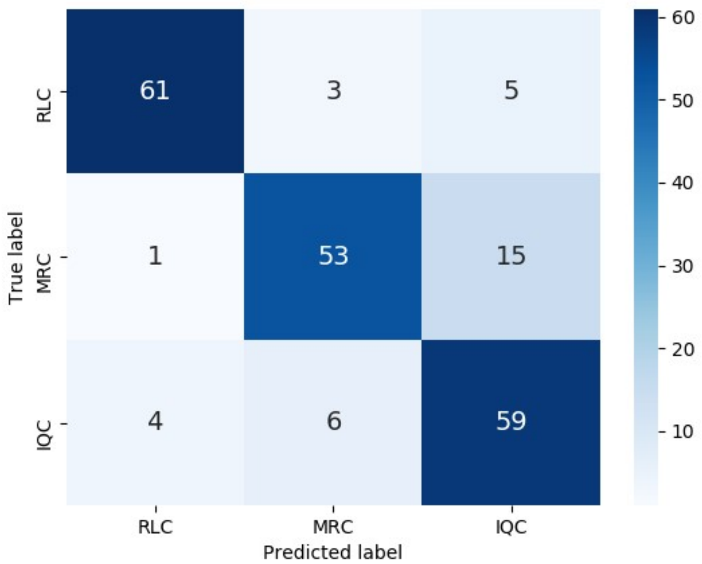

Fig. 7. Model training results on a balanced 3-class dataset.

The reason for such model behavior may lie in the similarity of "Inside quarter" and "Minor road" classes (compare Figs 3 and 4), which is why the model cannot extract any distinctive features from the data provided.

To better understand the issue, we will perform a number of experiments, constructing three more models, sequentially removing one of the classes.

Third model. The next model was trained on a set of two classes, "Red Line" (RLC) and "Inside quarter" (IQC). The parameter selection was carried out in a similar way to the previous experiments. The results of the parameter selection are presented in Table 6.

Table 6. Parameter selection for the third model.

No.	Batch Size	Epochs	Learning Rate	Accuracy
1	16	150	0.001	0.76
2	32	100	0.005	0.79
3	64	50	0.001	0.78
4	**128**	**50**	**0.001**	**0.94**
5	256	30	0.001	0.93
6	512	70	0.0005	0.91

The highest classification accuracy on the test set was achieved with the following parameters: Epochs = 50; Batch Size = 128 and Learning Rate = 0,001. The model training results for each class are presented in Table 7.

Table 7. Classification accuracy of the third model on the test set.

Class	Accuracy	# Samples
Red line (RLC)	0.96	69
Inside quarter (IQC)	0.93	69

The overall model accuracy was 94%. The class accuracies are quite high and amounted to 96% for the "Red Line" class, and 93% for the "Inside quarter" class. In the confusion matrix in Fig. 8, an intense main diagonal is visible, which indicates the model is good at distinguishing between these 2 classes.

Fourth model. The fourth model is generally similar to the third and was trained on a set of two classes: "Minor road" (MRC) and "Red line" (RLC). The parameter selection was carried out in a similar way to the previous experiments. The parameter selection results are presented in Table 8.

Table 8. Parameter selection for the fourth model.

No	Batch Size	Epochs	Learning Rate	Accuracy
1	16	50	0.001	0.84
2	32	70	0.001	0.88
3	64	50	0.001	0.93
4	**128**	**70**	**0.001**	**0.94**
5	256	70	0.001	0.93
6	512	50	0.001	0.91

The highest classification accuracy on the test set was achieved with the following parameters: Epochs = 70; Batch Size = 128 and Learning Rate = 0,001. The model training results in the fourth experiment are presented in Table. 9. The overall model accuracy was 94%; this coincides with the accuracy of the third model. Accuracies per class are also quite high, although slightly different: "Red Line" (RLC) – 94%, "Inside quarter" (MRC) – 93%.

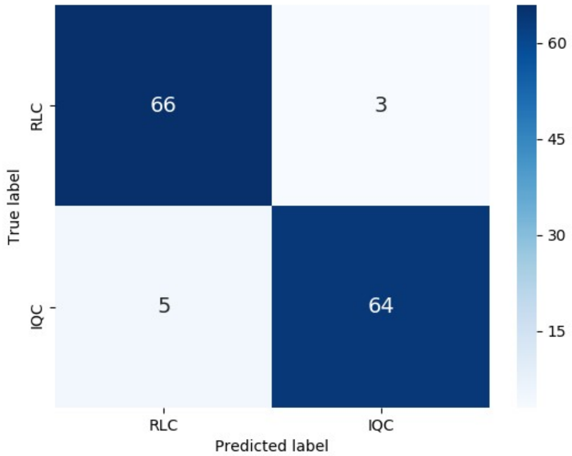

Fig. 8. Model training results on a 2-class dataset: "Red line" and "Inside quarter".

Table 9. Classification accuracy of the fourth model on the test set.

Class	Accuracy	# Samples
Red line (RLC)	0.94	69
Minor road (MRC)	0.93	69

In the confusion matrix in Figure 9, an intense main diagonal is visible, which indicates the model is good at distinguishing between these 2 classes.

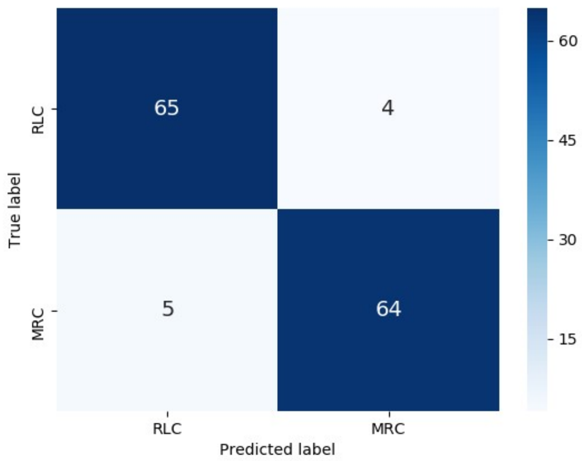

Fig. 9. Model training results on a 2-class dataset: "Red line" and "Minor road"

Fifth model. This model was trained on a set of two classes ("Minor road" (MRC), "Inside quarter" (IQC)). The parameter selection was carried out in a similar way to the previous experiments. The results of parameter selection are presented in Table 10.

Table 10. Parameter selection for the fifth model

No	Batch Size	Epochs	Learning Rate	Accuracy
1	16	30	0.001	0.85
2	32	100	0.0005	0.84
3	64	70	0.001	0.84
4	128	70	0.001	0.83
5	**256**	**50**	**0.001**	**0.86**
6	512	50	0.005	0.85

The highest classification accuracy on the test set was achieved with the following parameters: Epochs = 50; Batch Size = 256 and Learning Rate = 0,001.

The model training results are presented in Table 11. The overall model accuracy was 86%. Accuracy per class tuned out to be "Minor road" (MRC) - 82%, "Inside quarter" (IQC) - 90%.

Table 11. Classification accuracy of the fifth model on the test set.

Class	Accuracy	# Samples
Minor road (MRC)	0.82	69
Inside quarter (IQC)	0.90	69

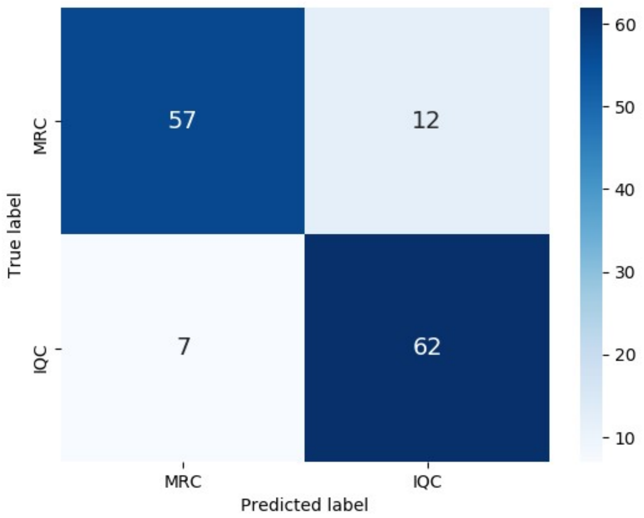

Fig. 10. Model training results on a 2-class dataset: "Minor road" and "Inside quarter"

The confusion matrix in Fig. 10 shows that the model is significantly worse at distinguishing between classes in the MRC and IQC pair than in the "RLC and IQC" and "RLC and MRC" pairs (Figures 8 and 9).

4 Discussion

The training results of the models show that transfer learning is generally suitable for classifying properties by their location relative to roads according to given classes. In general, as shown by the experimental results, transfer learning with a proposed dataset provides a fairly confident and accurate classification of properties for at least two classes: "Red line" and "Not red line", which is the sum samples of "Minor road" and "Inside quarter" classes.

The model trained on three classes gives a fairly accurate classification result for properties located on the "red line" and within the block, but quite often makes mistakes when recognizing located on the first line of secondary roads. Balancing the number of samples in each class increases the accuracy at determining properties located on the front line of minor roads, but at the expense of a general decrease in the model accuracy and the accuracy of other classes.

The probable reason, as shown by experiments with Models 3-5, is that there are significantly fewer differences between the "Inside quarter" and "Minor road" classes than between these classes and the "Red line" class. To increase the overall accuracy and improve the classification quality, it is proposed to increase the number of samples for the "Minor road" class, as well as to use several training models simultaneously.

5 Conclusions

The use of Teachable Machine to solve the problem of property location classification using transfer learning is justified and can be recommended for solving similar problem solution.

At the same time, transfer learning remains sensitive to the quantity and quality of samples, as well as the balance in the number of samples for each class in the training set.

The resulting models make it possible to fairly accurately classify properties that are on the "Red line" or outside it.

To improve classification accuracy for "Minor road" and "Inside quarter" classes, it is necessary to increase the number of samples, simultaneously use several classification models or use other pre-trained networks that are more appropriate for the subject area.

References

1. Astrakhantseva, I.A., Balandina, E.V.: Rights assessment methodology of collateral under loan agreements, Izvestia of higher educational institutions. Series "Economics, finance and production management", 3(33), 3-9 (2017).
2. Burinskienė, Marija, Rudzkiene, Vitalija, Venckauskaite, Jurate: Models of factors influencing the real estate price. 8th International Conference on Environmental Engineering, ICEE (2011).
3. International Valuation Standards (IVS), https://www.rics.org/globalassets/rics-website/media/upholding-professional-standards/ sector-standards/valuation/international-valuation-standards-rics2.pdf, last accessed 2020/12/21.
4. Order of the Ministry of Economic Development of the Russian Federation, http://www.consultant.ru/document/cons_doc_LAW_160678/76df77d9a6013003f54d591e78864ebd67908f32/, last accessed 2020/12/22.
5. Realto application description, http://www.realto.ru/about_program/description/, last accessed 2020/12/21.
6. Geocoding service, http://api.sputnik.ru/maps/geocoder/, last accessed 2020/12/21.
7. Static Maps API, https://yandex.com/dev/maps/staticapi/, last accessed 2020/12/21.
8. Mizgirev L, Galiaskarov E.: Estate position estimation, v0.3, Zenodo, 10.5281/zenodo.4400637 (2020).
9. Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Hasan, M., Van Essen, B.C., Awwal, A.A.S., Asari, V.K.: A State-of-the-Art Survey on Deep Learning Theory and Architectures. Electronics, 8, 292 (2019).
10. Bozinovski Stevo, Fulgosi Ante: The influence of pattern similarity and transfer learning upon training of a base perceptron B2. Proceedings of Symposium Informatica, 3-121-5 (1976).
11. Yunho Jeon, Yongseok Choi, Jaesun Park, Subin Yi, Dongyeon Cho, Jiwon Kim:. Sample-based Regularization: A Transfer Learning Strategy Toward Better Generalization (2020).
12. Taresh, M., Zhu, N., Ali Ali, T.: Transfer learning to detect COVID-19 automatically from X-ray images, using convolutional neural networks medRxiv (2020).

13. Pan S. J., Yang Q.: A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22, 10, 1345-1359 (2010).
14. Dhruv Rathi: Optimization of Transfer Learning for Sign Language Recognition Targeting Mobile Platform arXiv, 1805.06618 (2018).
15. Silver, D., Schrittwieser, J., Simonyan, K., et al.: Mastering the game of Go without human knowledge. Nature, 550, 354–359 (2017).
16. Phillips, K.: Teachable Machine 2.0 makes AI easier for everyone, https://blog.google/technology/ai/teachable-machine/, last accessed 2020/12/23.
17. Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A., Chen, A.: Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification. In Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, 1–8 (2020).
18. Kacorri, Hernisa: Teachable machines for accessibility. ACM SIGACCESS Accessibility and Computing, 10-18 (2017).