

PAPER • OPEN ACCESS

Training video surveillance system for the purpose of object identification with the help of neural networks

To cite this article: Kirill Zhigalov *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **537** 052009

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the **collection** - download the first chapter of every title for free.

Training video surveillance system for the purpose of object identification with the help of neural networks

Kirill Zhigalov^{1,2}, Karen Avetisyan³ and Svetlana Markova⁴

¹ V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia

² Moscow Technological Institute, Moscow, Russia.

³ Moscow University of the Ministry of Internal Affairs of the Russian Federation n.a. V. Ya. Kikotya, Moscow, Russia

⁴ Department of data analysis, decision making and financial technologies, Financial University under the government of the Russian Federation, Moscow, Russia

E-mail: kshakalov@mail.ru

Abstract. Research on the development of algorithms for the use of neural networks associated with video surveillance systems on the identification of the desired objects today go to a whole new technological level. Categorization of classes-objects of research, determination of actual zones of fixation, integration of functions-cloud computing, analysis of incoming data and their evaluation determine the relevance in the creation of a trainable model that allows to solve a wide range of problems when working with the existing and the flow of incoming data in real time both at the state and at the level of transnational companies.

1. Introduction

At the moment, there are a number of effective solutions presented by Microsoft and Google on the identification of objects on the market. This paper discusses the features and functionality of the learning model built on the basis of neural networks using Google tools [1].

The structure of training of the selected model includes: questions on the development of classes of objects that are learned by the model of software and hardware complex; selection of software products and preparation of data for the creation and training of the model, as well as evaluation of the results, on the basis of which the analysis of the performance of the model [2].

2. Development of the structure of object classes recognized by the model of the software complex

In order to create and train a system aimed at the search and identification of the desired objects, as well as further classification in the fixed zone, the following structure of defined classes was put forward in the analysis of incoming data.

By season: summer time; winter time. By age of defined objects and gender: age up to 18 years (boys; girls); age from 18 to 25 years (boys; girls); in the range from 25 to 55 years (men; women); over 55 years (men of advanced age); over 55 years (women of advanced age). According to the description: clothes of the summer season; clothes of the winter season; other forms of clothing. Categorization by classes allows you to organize the flow of data on a number of features, providing increased efficiency when working with the desired objects.



3. Preparing data for model creation and training

When preparing the training data set in popular Internet search engines, images that meet the following requirements were selected:

- JPEG image file type;
- resolution up to 1920x1200;
- file size up to 2 megabytes.

Assuming possible prospects for the use of software and hardware complex, as well as the actual position of the camera installation, the following subjects of images were selected:

- city streets in different weather conditions, shooting from web cameras;
- public transport (stops on the streets, ground transport salons, subway and commuter train cars, underground passages, escalators, transitions between stations);
- parks, squares, skating rinks, stadiums and sports complexes in different weather conditions;
- shopping malls, restaurants, hypermarkets shops and other areas hold a variety of objects and areas.

4. Selection of software products, suitable for model training

To learn using Tensorflow library models, in the collected data set it is necessary to select the coordinates of the objects found in the image and their class. This task is solved with the help of Labeling software. For each image, the program creates an annotation in the form of an XML file containing the coordinates of the areas of the found objects from the label class.

The next step is to convert the data to CSV (train_labels.csv, text_labels.csv) and file generation TFRecord (train.record, test.record).

Creation of CSV files describing objects found on images is presented in the following form:

```
import os
import glob
import pandas as pd
import xml.etree.ElementTree as ET
def xml_to_csv(path):
    xml_list = []
    for xml_file in glob.glob(path + '/*.xml'):
        tree = ET.parse(xml_file)
        root = tree.getroot()
        for member in root.findall('object'):
            value = (root.find('filename').text,
                    int(root.find('size')[0].text),
                    int(root.find('size')[1].text),
                    member[0].text,
                    int(member[4][0].text),
                    int(member[4][1].text),
                    int(member[4][2].text),
                    int(member[4][3].text))
            xml_list.append(value)
    column_name = ['filename', 'width', 'height', 'class', 'xmin', 'ymin', 'xmax', 'ymax']
    xml_df = pd.DataFrame(xml_list, columns=column_name)
    return xml_df
def main():
    for directory in ['train','test']:
        image_path = os.path.join(os.getcwd(), 'images/{}'.format(directory))
        xml_df = xml_to_csv(image_path)
        xml_df.to_csv('data/{}_labels.csv'.format(directory), index=None)
        print('Successfully converted xml to csv.')
    main()
```

The structure of the trained data set of the project model has the following form:

```
Object-Detection/
-data/
  --test_labels.csv
  --train_labels.csv
  -- train.record
  --test.record
  -- object-detection.pbtxt
-images/
  --test/
    ---testingimages.jpg
    ---testingimages.xml
  --train/
    ---testingimages.jpg
    ---testingimages.xml
```

For training of our model the model, "sd_mobile net_v1_crack", as the most suitable for fast operation of mobile devices and recognition of images on video was chosen [3]. In order to use a pre-trained model, you need to download its checkpoint to a folder that contains a training dataset converted to TensorFlow format, as well as the settings file of the retrained model (pipeline.config).

In this file you need to make changes depending on our dataset for training of own models (the number of designated classes, local file paths TFRecord and file classes).

5. Model training, evaluation of results

Since the training of the model using the TensorFlow ObjectDetection API capabilities requires high system technical characteristics on the part of the hardware, and at the same time provide for the possibility of cloud computing [7], including synchronization with Google Cloud products, the training of the model was carried out in the Google Cloud Machine Learning API service [4].

The above files of the test model of the software package describing our set of data images converted into TensorFlow form, the settings file and the training model were loaded into the Cloud Storage, and the tasks (so-called train-job) are launched from the local machine to train the training model on our data set and the class map of the recognized objects for the created model [6].

After successful start-up, the learning process of the model can be followed by the logs of Cloud Machine Learning, or in the TensorBoard console, which analyzes the learning process depending on the time or number of iterations [5].

For example, you can follow the graph of the loss percentage curve (not finding objects in the image) depending on the number of iterations, see figure 1.

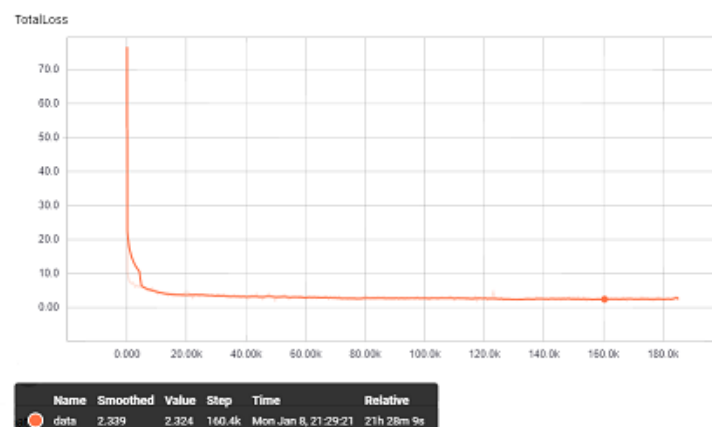


Figure 1. Graph of the loss percentage curve.

After achieving the specified results when training the model, the calculations are stopped, and from the control points for the entire period of iterations the point with the best performance is selected and converted into a file (Tensorflow graph proto), which is used for further calculations and identification of objects. [8]

6. The trained model performance analysis

An attempt was made to find out the limits of the minimum number of images for training. The result was influenced by the wide variability of image quality (technical characteristics) and their content. For classes with the highest number of annotated values in the images of the test data set (boys and girls 18-25 years old, summer and winter season), the quality of education is higher.

Let's compare the results of the test model and the trained model on the set.



Figure 2. Original image.



Figure 3. The trained model.

Conclusion 1: Correctly found and fixed objects with the position of the face-full face, also correctly determined the gender characteristic of the identified objects.



Figure 4. Original image.



Figure 5. The trained model.

Conclusion 2: Object recognition is correct. Note: it is advisable to add the class «family».



Figure 6. Original image.

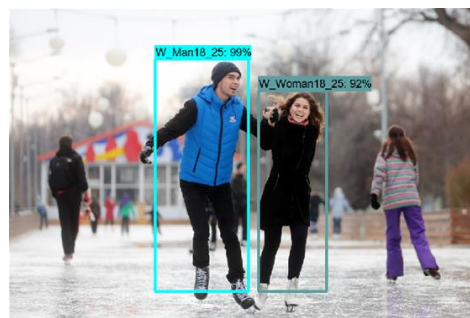


Figure 7. The trained model.

Conclusion 3: Trained model worked correctly, all objects are identified by the specified parameters.



Figure 8. Original image.



Figure 9. The trained model.

Conclusion 4: Trained model worked correctly, all objects are identified by the specified parameters.



Figure 10. Original image.



Figure 11. The trained model.

Conclusion 5: recommendation regarding the value of the coefficient for the category «kids».

7. Use additional Google tools

Let's consider the detection of the last image by means of Google Vision API:

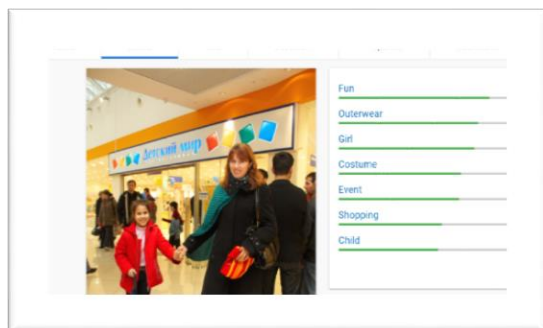


Figure 12. Original image.

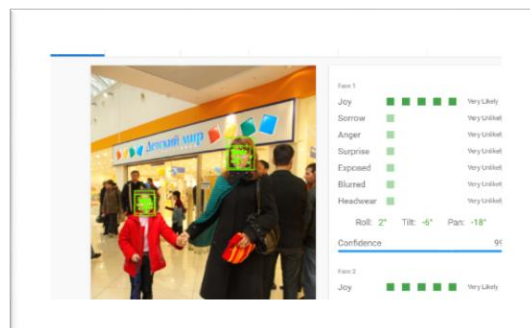


Figure 13. The trained model.

Conclusion 6: The use of tags will allow you to conduct a preliminary moderation of advertising content, if it was also pre-marked by the most popular tags.

Note: Presence of a label Child/Kids will probably help to solve the error of determination of the age of the children.

8. Conclusion

The use of the Faces tool provides a preliminary assessment of the emotional state of the found objects, as well as their number, this iteration is implemented in the form of an additional test of the quality of the trained model.

To solve the tasks set for the program complex, it is also necessary to Supplement the existing class diagram taking into account the style of the found people (business, sports, every day, etc.). It makes sense to annotate objects that are not facing the camera because they are defined by a test training model. To get a more workable and efficient model for classifying objects in an image, you need to increase the data set for training.

References

- [1] Sultan N 2014 Making use of cloud computing for healthcare provision: Opportunities and challenges *International Journal of Information Management* **34** 177–84
- [2] Brednik A V 2013 Security issues of cloud computing. Analysis of methods of protection of the clouds from cloud security alliance *Almanac of modern science and education* **10** 35–8
- [3] Sommerville I 2011 *Software engineering* (New York: Pearson, Addison-Wesley)
- [4] Hoff C 2011 In: Security guidance for critical areas of focus in cloud computing pp 12–20
- [5] Zhigalov K 2017 Using cloud computing technologies in IP-video surveillance systems with the function of 3d-object modelling *ITM Web of Conferences* **10** 02004
- [6] Zhigalov K 2016 Use of the game technologies engines for the purpose of modern geographic information systems creation *ITM Web of Conferences* **6** 030118
- [7] Rehman S, Kriebel F, Shafique M and Henkel J 2014 Reliability-driven software transformations for unreliable hardware *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **33** 1597-610
- [8] Pluzhnik E, Nikulchev E and Payain S 2014 Optimal Control of Applications for Hybrid Cloud Services *IEEE World Congress on Services (SERVICES)* (Anchorage, AK: USA) pp 458-61