Research and utilization of the convolutional neural network YOLO for automated testing of desktop and mobile applications

Abstract

This work is aimed at solving the problem of detecting desktop and mobile application controls. The problems arising in solving this problem are presented. The paper shows the ways of forming datasets, as well as the applicability of the neural network approach in the tasks of automated software testing.

Methodology

The source data can be globally divided into two groups: images of real applications and synthetic images. The first include snapshots of windows of existing applications marked up to detect objects in YOLO format. Any application that has a need for automated testing can act as a target application.

Synthetic sets are randomly generated snapshots of desktop applications with the corresponding YOLO markup. Three generators written in programming languages are engaged in the production of synthetic data: C# (WPF (Windows Presentation Foundation)), Python (PySide), Java (Swing). It is impossible to cover all the variety of desktop application design, however, there are controls that have similar features. A synthetic dataset is required for:

1) increasing the generalizing ability of the neural network;
2) some balancing of classes by generating scarce components (Figure 1).

Based on the requirements of automated testing, 11 classes of components for desktop applications were allocated: Button, CheckBox, Input, Label, List, RadioButton, Table, TextArea, TreeView, ScrollBar, Calendar. For mobile applications: Button, CheckBox, Input, Label.

The training of the network took place in two stages:

1) preliminary training of a neural network on a synthetic dataset;
2) network training based on snapshots of the target application.

The goal of the first stage is to cover as many varieties of components as possible. At the second stage, the dataset of the target application is collected with a small proportion of new synthetic snapshots. Thus, on the validation sample for the desktop application mAP (mean Average Precision) ≈ 0.95; for the android application mAP ≈ 0.98. Figures 2 and 3 shows some examples of work.

The detected components have the corresponding attributes. Knowing the attributes allows you to interact correctly with the control. The controls cut from the source image are fed to the input of simple convolutional neural networks. Networks consist of three consecutive blocks: a convolutional and a pooling layers. The last layer is fully connected.

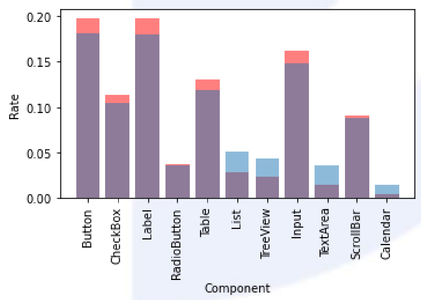

Figure 1. A little classes balancing

Results

The result of this work is a developed solution that allows you to detect the controls of desktop and mobile applications. The relevance of the applicability of computer vision for solving automated testing tasks is shown. The main problems are missing a component, which leads to a failed test, a wide variety of user interface designs, as well as the task of correlating controls. In addition, methods for generating datasets are presented. The allocation of control classes for each application should be individual, although the main component classes will be identical.

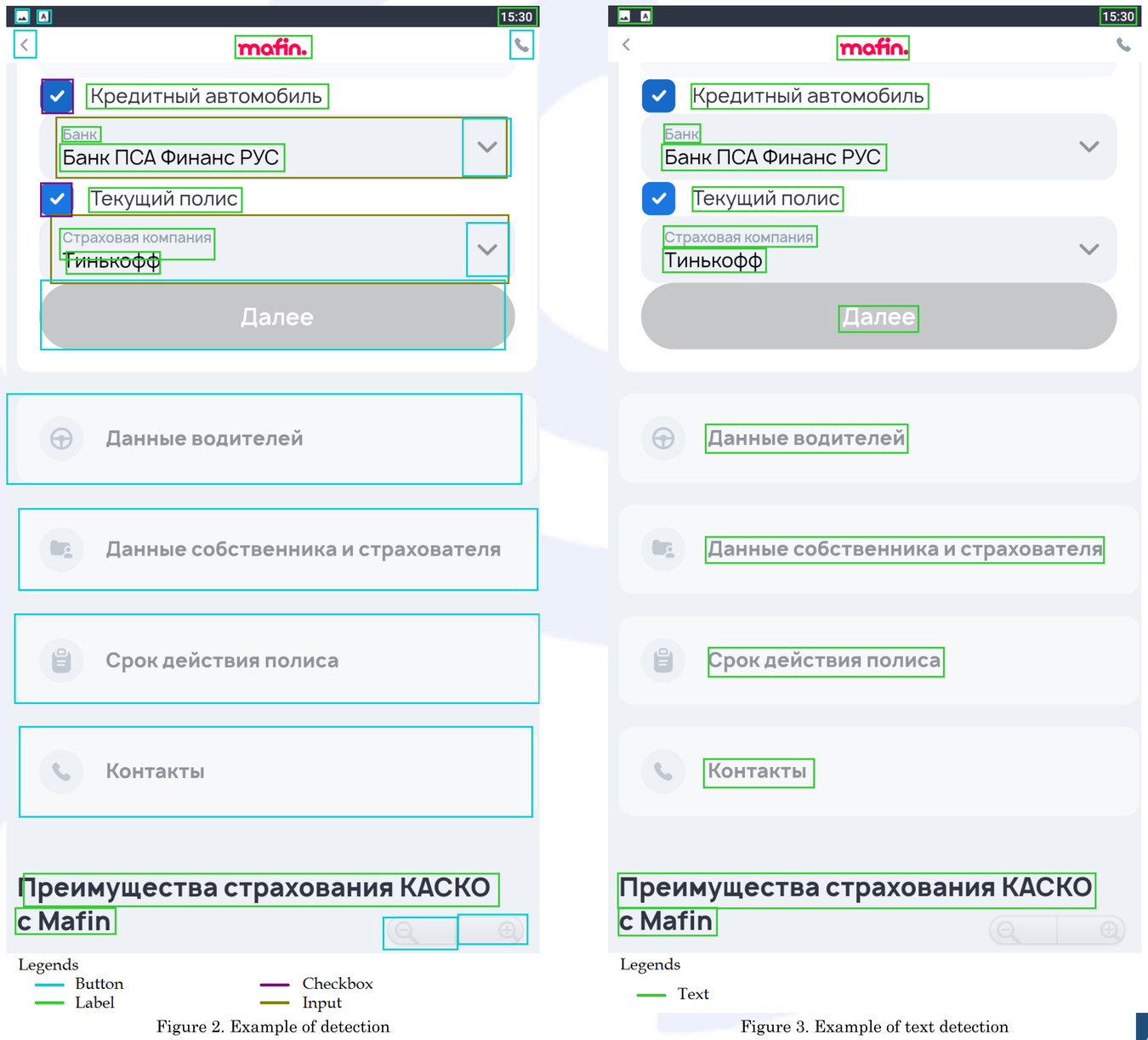

Figure 2. Example of detection

Figure 3. Example of text detection

Contact: Rud' A. A. – sasha_.96@mail.ru.