UDC 004.852

A. Yakovlev, O. Lisovychenko

AN APPROACH FOR IMAGE ANNOTATION AUTOMATIZATION FOR ARTIFICIAL INTELLIGENCE MODELS LEARNING

Abstract: The article focuses on the approach of image annotation automatization for artificial intelligence models learning. List of basic requirements for the image annotation process has been highlighted and analyzed with existing software applications. An application has been developed as a proposed solution for disadvantages elimination. The tables and graphs representing key applications parameters relate.

Key words: image recognition, image annotation, machine learning, artificial intelligence.

Introduction

Artificial intelligence-based image recognition systems are becoming popular nowadays. Scopes of neural networks application varies from speech recognition tasks to large data processing centers for decision-making and prediction.

Effective usage of image recognition systems requires software tools, providing enough possibilities for learning data annotation. Data requirements specifics for artificial intelligence models training is directly connected with recognition tasks. Thus, every single task requires its individual approach.

Attention increase for artificial intelligence models usage caused numerous available software applications appearance. Such programs provide tools for annotation processes automatization, hence, it's more effective execution. Artificial intelligence models implementation could vary with many parameters, but mostly train data quantity factor directly proportional to recognition precision and may require from hundreds and thousands of images till tenth of thousands. For enough performanced annotation of large images counts, all suitable for automatization processes should be performed by the software, so letting the annotation process last for a foreseeable period of time.

An example of annotated data array creation for neural network learning could be number plates recognition and vehicle identification based on Yolo image recognition system task.

The aim of the study is to create an approach for image annotation process automatization using computer software as a part of artificial intelligence models learning.

Problem statement

For image annotation tasks to be accomplished it is required that a full list of technological stages should be defined. All these stages will stand as a list of software application requirements and will lead, in the final, to annotation process automatization accomplishing. Existing software applications for image annotation research allows to determine the level of conformity to annotation process requirements and define disadvantages.

Because the image annotation, for enough recognition accuracy reaching, may require large quantities of images processing (up to thousands), this quantity factor should be taken into account during existing applications research.

Collected results should be analyzed for requirements conformance and a software solution should be proposed. This solution should match all requirements of the annotation process, eliminate disadvantages in existing software applications and increase annotation process time efficiency.

Software application for image annotation requirements:

– Region creation, editing (size and position), deletion;

– Region size templates;

– Annotated object classes processing (selection, change);

– Metric graphic coordinates to Yolo graphic coordinates conversion;

– Visual scaling (zooming) of image selected region for small annotation objects;

– Ordered images sequences processing;

– Annotation data saving in accordance with Yolo detection system;

– Tools for visual quality control;

– Create sequencing with actions repeatability minimization, thus human error probability minimization;

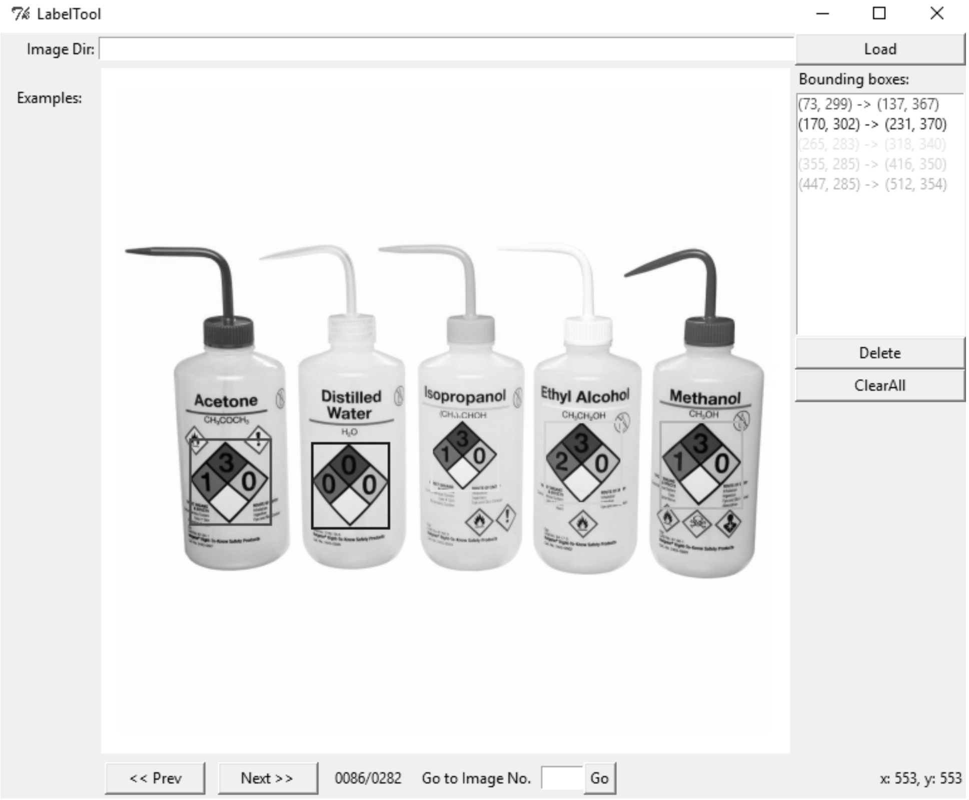
– Train files generation;

– All per-image annotated regions saving in single annotation file;

– Mass image processing requirement match.

Existing software applications analyzation

Label Tool developed with Python programming language for image regions annotation. After program interface exploration (Fig. 1.) and testing capability list was defined, which conforms to annotation process requirements, namely, work with images sequencing, regions coloring, region parametric view, regions deletion, metric region storage to external file. However, during analysis, some missing features were discovered: region editing and location change, different object classes interaction, only metric scales are available while working with regions, without Yolo format conversion, active region scaling, train data files creation, etc.

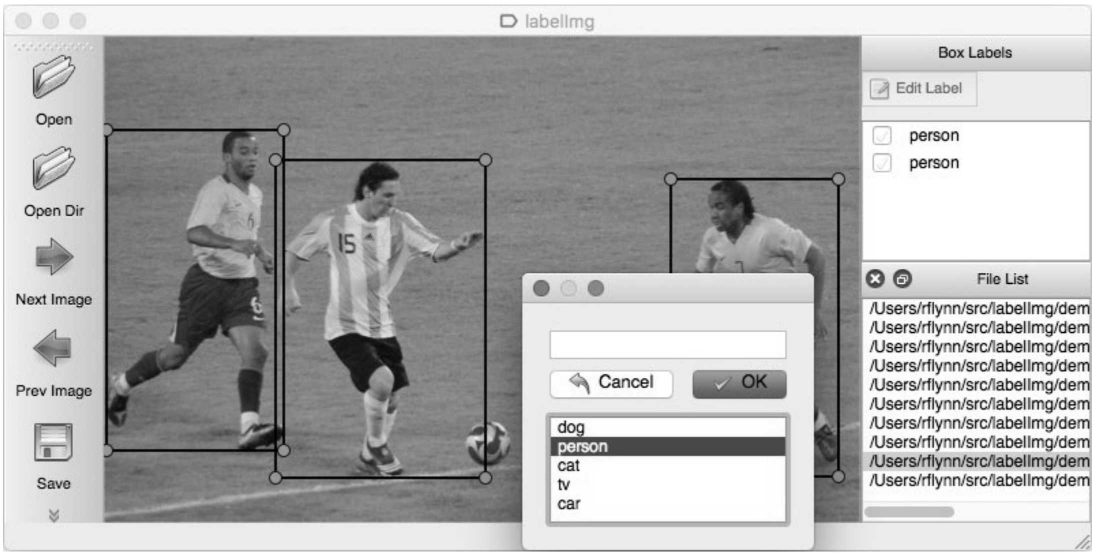


(73, 299) -> (137, 367)
(170, 302) -> (231, 370)
(265, 283) -> (318, 340)
(355, 285) -> (416, 350)
(447, 285) -> (512, 354)

0086/0282

x: 553, y: 553

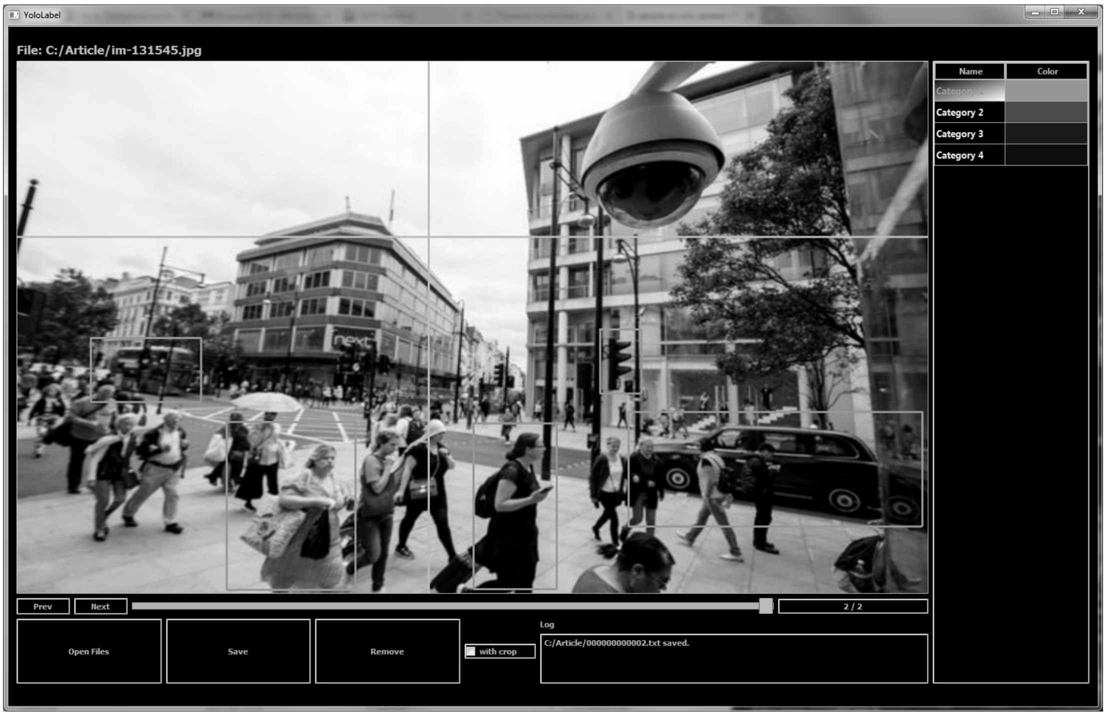
Figure. 1. Label Tool application main window [4]



/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem

Figure. 2. labelImg application main window [5]

labelImg application has the most conformance to annotation process requirement because of next features (Fig. 2.): multiple object classes support, region size and location change, images sequencing support, train data files has Yolo format, direct annotation files interaction, active region scaling, etc. But, still some key features for the annotation process are missing and could cause extra time consumptions, like: parametric region view, regions coloring differentiation, train files creation, suboptimal active region scaling procedure, etc.



File: C:/Article/im-131545.jpg

Name	Color
Category 1	
Category 2	
Category 3	
Category 4	

```
C:/Article/000000000002.txt saved.
```

Figure. 3. Yolo_Label application main window [6]

While examining Yolo_Label application user interface (Fig. 3.) and performed annotation abilities testing, list of advantages was compiled: different object classes support, region deletion, images sequencing, annotation files with Yolo format, additional components for better navigation, measurement rulers, regions color differentiation. In progress of work with an application testing images annotation was successfully accomplished, but numerous disadvantages were discovered, for example: parametric region view missing, created region size/position direct change is impossible, but only through the region re-creation, active region scaling is missing.

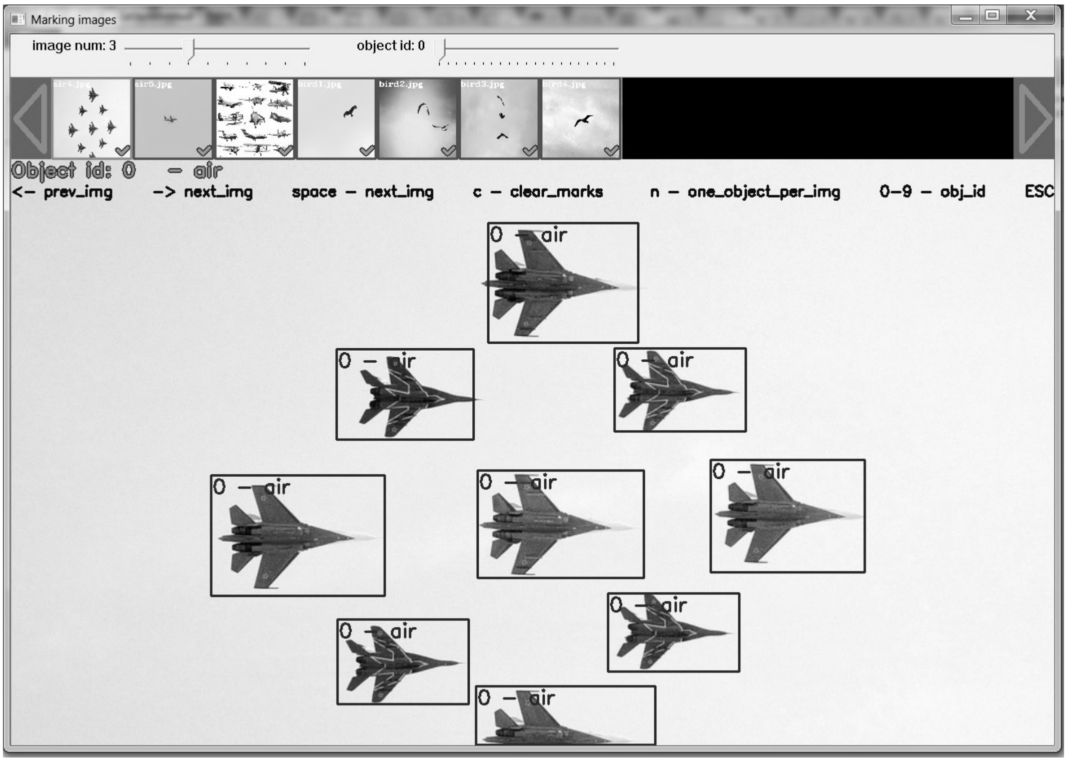


Figure 4. Yolo_mark application main window [7]

Yolo_mark provides significant amount of functional capabilities for annotation tasks accomplishing (Fig. 4.), namely: object classes support, region deletion, images sequencing, output files with Yolo data format, additional UI components for image files and classes navigation, measurements rulers, regions colouring, on-screen user tips showing. Missing or insufficient functions in given program are inavailability to edit region size or position (only through the deletion and creation), application freezing and hanging, even on modern and fast computers and significantly more time consumptions for annotation operations, compared to other software applications.

For the above statements, the list of typical software applications operations was defined and measured time consumptions required for its accomplishment, as well as number of user actions (clicks) required for such operations. Results of measurements are shown in Table 1 and Table 2.

Table 1.

Time characteristics of image annotation in modern programs

Product / Time consumption per operation (sec.)	Label tool	labelImg	Yolo_ Label	Yolo_ mark	Expected consumption values
Region creation	1.56	4.44	5.2	6.48	<=4

Ended table 1

Product / Time consumption per operation (sec.)	Label tool	labelImg	Yolo_ Label	Yolo_ mark	Expected consumption values
Class assignment	n / a	1.9	2.04	0.8	<=2
Class change	n / a	5.6	n / a	n / a**	<=3
Region position change	n / a	2.97	n / a	n / a**	<2
Region size change	n / a*	3	n / a	n / a**	<2
Region deletion	3.53	1.81	1.58	1.24**	<2
Next image transition	2.09	1.4	2.47	2.21	<=1
Yolo format annotations saving	n / a	3.28	2.3	2.21	<=1
Active region scaling	n / a	12.6	n / a	n / a	<=1
Average image processing time for two different object classes	6.15***	11.12	7.9	12.2	<=11

*** available only when delete and re-create**

**** all regions removal is available only**

***** single class processing is only allowed**

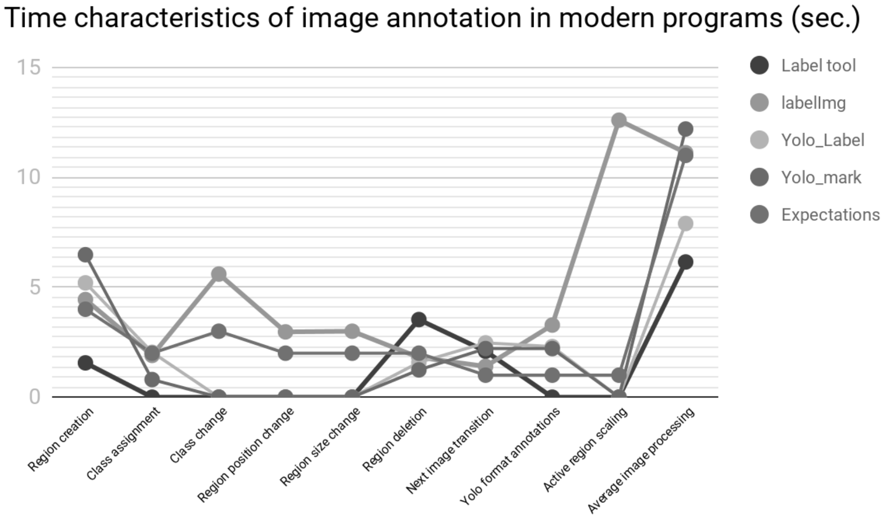


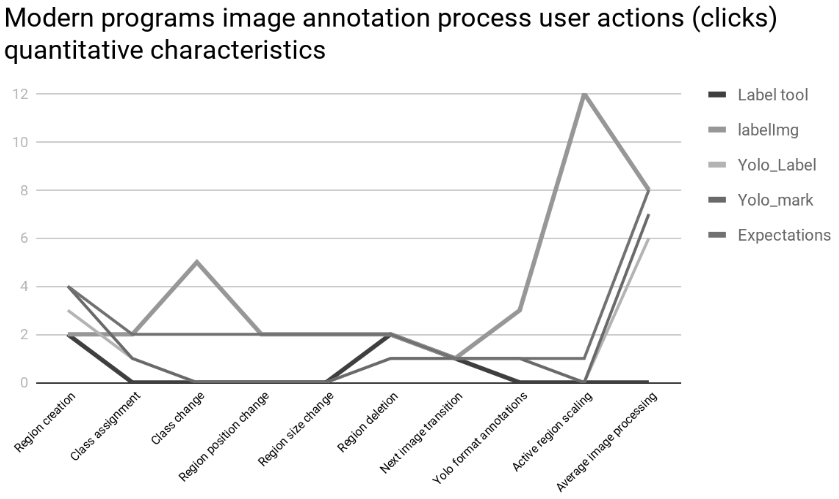
Table 2.

Modern programs image annotation process user actions (clicks) quantitative characteristics

Product / Actions count	**Label tool**	**labelImg**	**Yolo_Label**	**Yolo_mark**	**Expected values**
Region creation	2	2	3	4	<=4
Class assignment	n / a	2	1	1	<=2
Class change	n / a	5	n / a	n / a	<=2
Region position change	n / a*	2	n / a*	n / a	<=2
Region size change	n / a*	2	n / a*	n / a	<=2
Region deletion	2	2	1	1**	<=2
Next image transition	1	1	1	1	1
Yolo format annotations saving	n / a	3	1	1	1
Active region scaling	n / a	12	n / a	n / a	1
Average image processing time for two different object classes	n / a	8	6	7	<=8

*** available only when delete and re-create**

**** all regions removal are available only**



Conclusions

Image annotation process was examined as a part of artificial intelligence models based on Yolo detection system training, and all mandatory same as optional process stages were defined. Annotation process could be significantly automated using software applications. Major requirements for such applications are enough abilities for process coverage, minimal time consumption factor based graphical user interface, also user actions count value should be least possible without repetitions and excessiveness.

In the process of modern software application for image annotation and Yolo train data generation work specific research there was a number selected for testing, namely: Label Tool, labelImg, Yolo_Label, Yolo_mark.

In accordance with annotation process requirements mentioned applications were analyzed from influencing criterias point of view: time cost for single image annotation and minimal required application features set for the annotation task accomplishment. Mediocre criteria for application analysis could be defined as user actions count, which influences on final time cost and is directly proportional to human error probability. Results of analysis were provided in tables 1 and 2.

In accordance with results provided in tables it can be argued that examined modern software applications are only partially covering annotation process requirements and not always optimal from time consumptions point of view, as mass factor is not always taken into account in the process of design and development.

Optimal software application parameters were proposed in accordance with collected results, which could be reached by dint of modern software development tools. Yoloanno application was developed as a result of proposed parameters processing. In the process of application analysis, the possibility of requirements and hypotheses put forward implementation were proved in practice during mass image annotation for a knowledge-based image recognition systems training.

REFERENCES

1. Rajalingappaa S. . Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras - 312 p., Packt Publishing, January 23, 2018. ISBN-10: 1788295625, ISBN-13: 978-1788295628

2. Neurotechnologies and neurocomputer systems: textbook. Yampolskyi L. S., Lisovichenko O.I., Oliynyk V.V. Kyiv: Dorado-Druk, 2016. 576 p. (Ukr.)

3. How to train YOLOv2 to detect custom objects. URL: https://timebutt.github.io/static/how-to-train-yolov2-to-detect-custom-objects/

4. BBox-Label-Tool: A simple tool for labeling object bounding boxes in images. URL: https://github.com/puzzledqs/BBox-Label-Tool

5. labelImg: LabelImg is a graphical image annotation tool and label object bounding boxes in images. URL: https://github.com/tzutalin/labelImg

6. Yolo_Label: GUI for marking bounded boxes of objects in images for training neural network Yolo v3 and v2. URL: https://github.com/developer0hye/Yolo_Label

7. Yolo_mark: GUI for marking bounded boxes of objects in images for training neural network Yolo v3 and v2. URL: https://github.com/AlexeyAB/Yolo_mark

8. Yoloanno – The YOLO annotation tool for Windows. URL: https://github.com/AntonYakovlev/Yoloanno