Keyboard and Keys Image Recognition

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Author: Jan Lorenc
Supervisor: Jan Pluskal

Excel@FIT 2023

About

The goal of this thesis is to create a working solution for keyboard keys recognition to automate robotic writing on keyboards. The work is split into separate keyboard detection, single-character detection and post-processing of the results. Each of these parts required individual datasets.

Conditions

Due to extending an existing system AIVA, some factors can be taken into consideration:

- Full-HD cameras
- Camera calibration system
- Lab/office environment

Technologies

- PyTorch + OpenCV
- YOLOv7
- Canny edge detection

Datasets

Keyboards

- 615 keyboards of different types from various devices
- Data augmentation ➞ 20 000 images
- Scene background (COCO17) or single color
- 1280x720 px

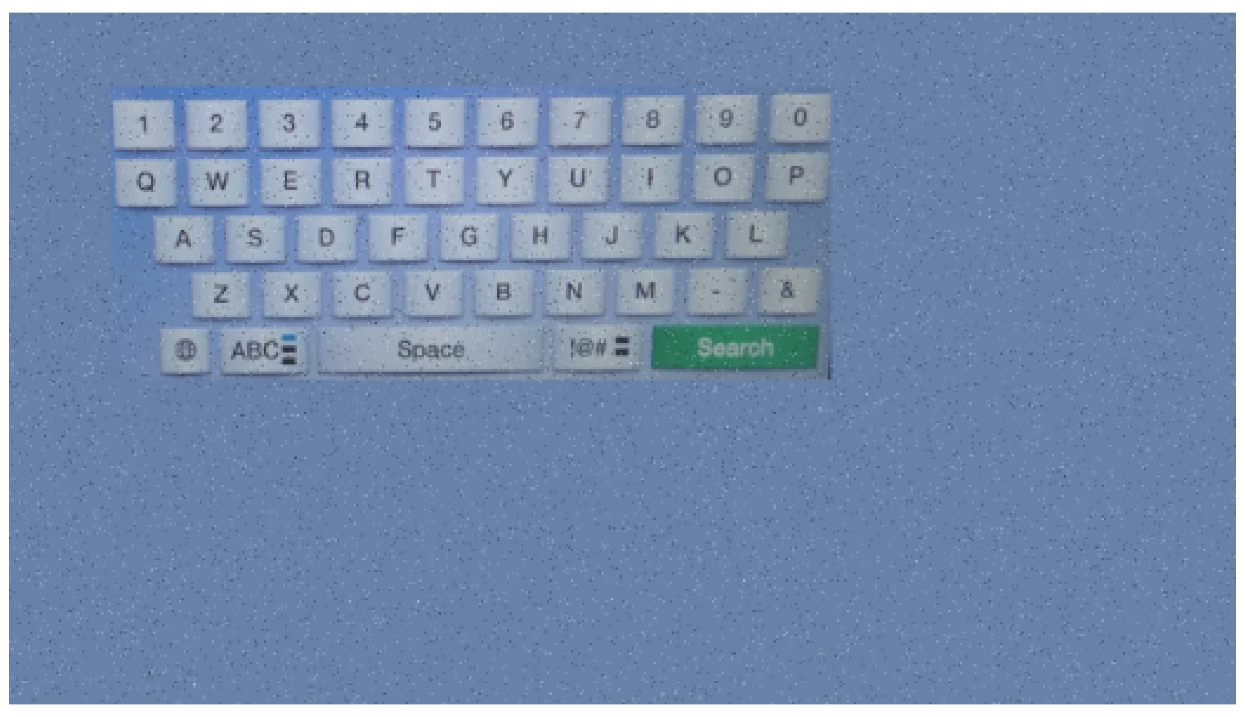


Figure 1: A keyboard is generated on a background of the same color as is its averaged background color

Characters

- 99 classes
- Data augmentation ➞ 50 000 images
- Grayscale (high contrast between keyboard background and characters)
- 640x640 px (prepared for YOLOv7)

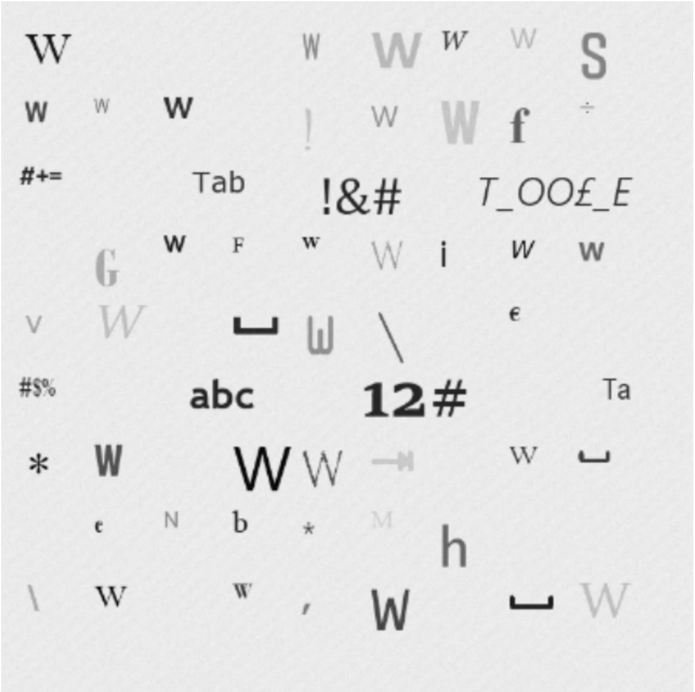


Figure 2: Main character W and other random characters generated for character detection training

Post-processing

- Validation dataset
- 120 manually annotated images
  - 60 for various layouts
  - 60 for missing character computation
- Any input image size (real-world examples)

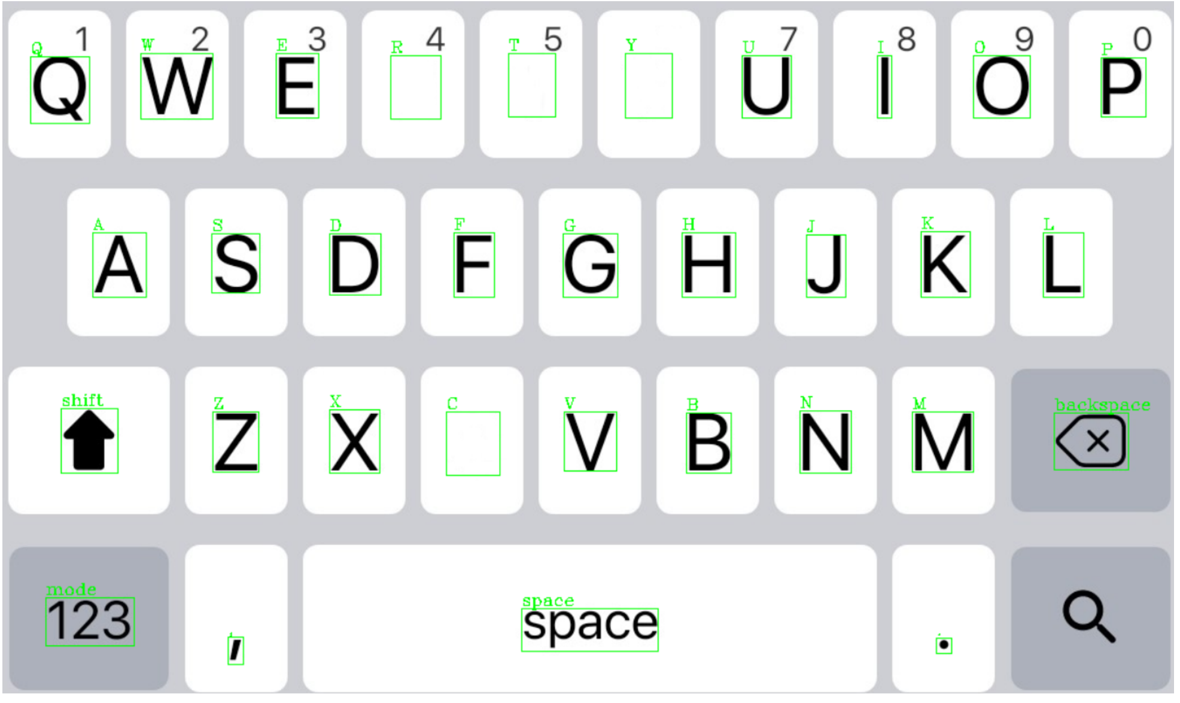
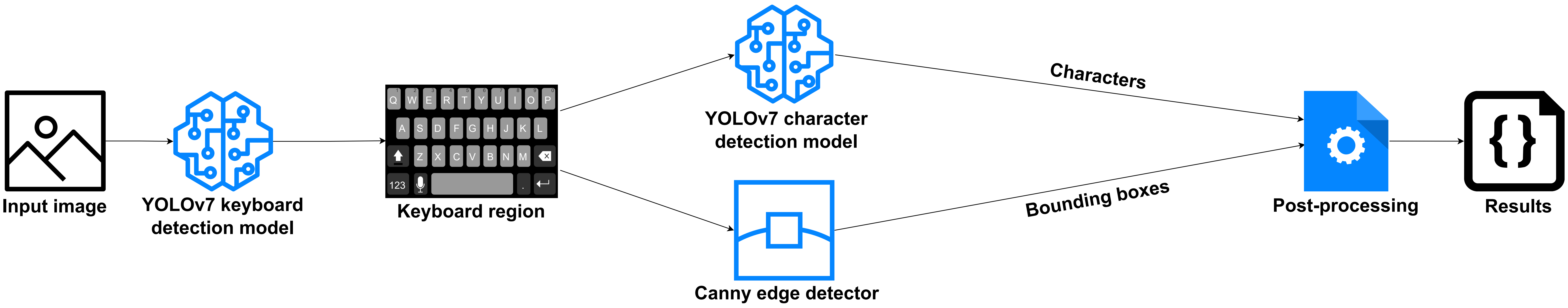


Figure 3: An annotated smartphone keyboard with some missing characters expected to be computed

Image processing



Input image

YOLOv7 keyboard detection model

Keyboard region

YOLOv7 character detection model

Canny edge detector

Characters

Bounding boxes

Post-processing

Results

Figure 4: Flowchart of the recognition process

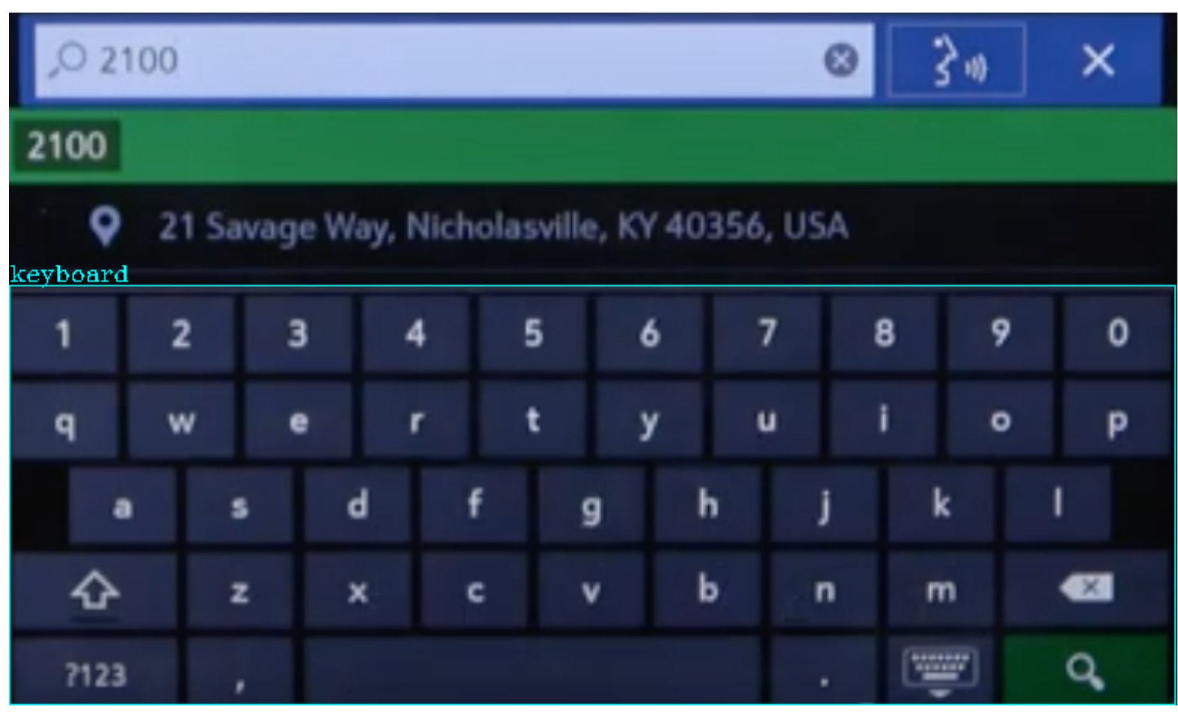


Figure 5: The first phase detects a keyboard region in an input image

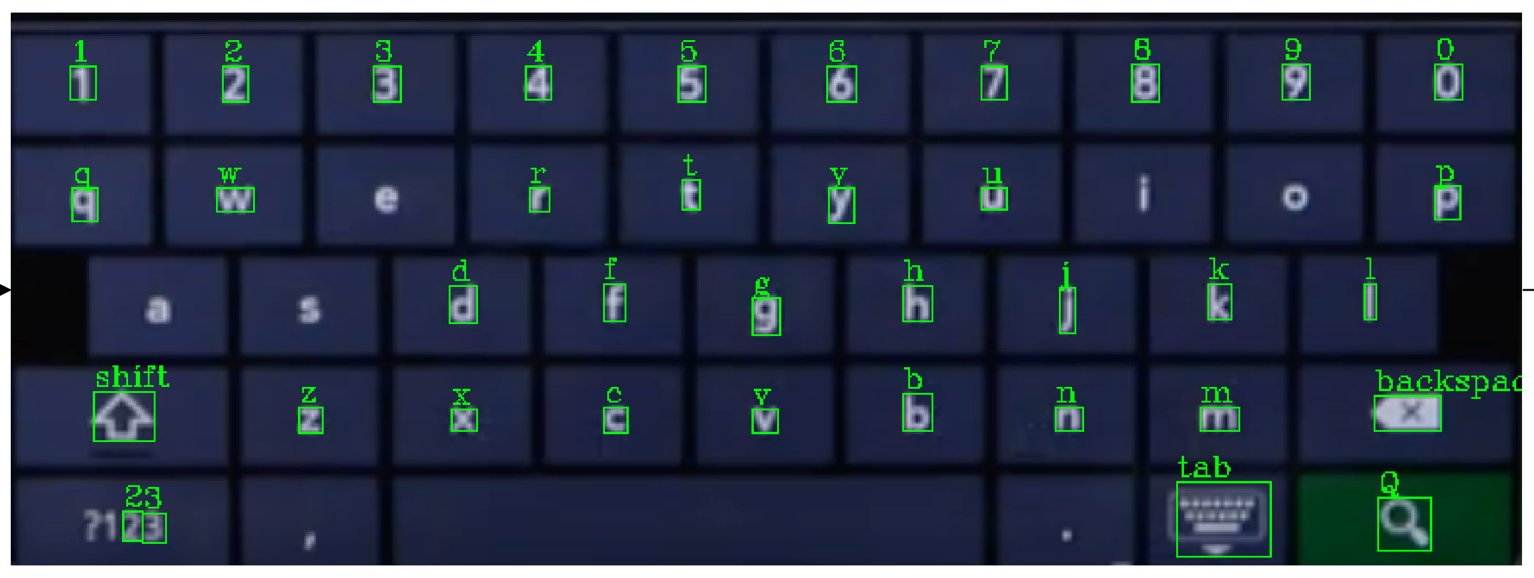


Figure 6: The character recognition is run on the detected keyboard region. Here it can be seen that some characters were undetected and there are also two false positives.

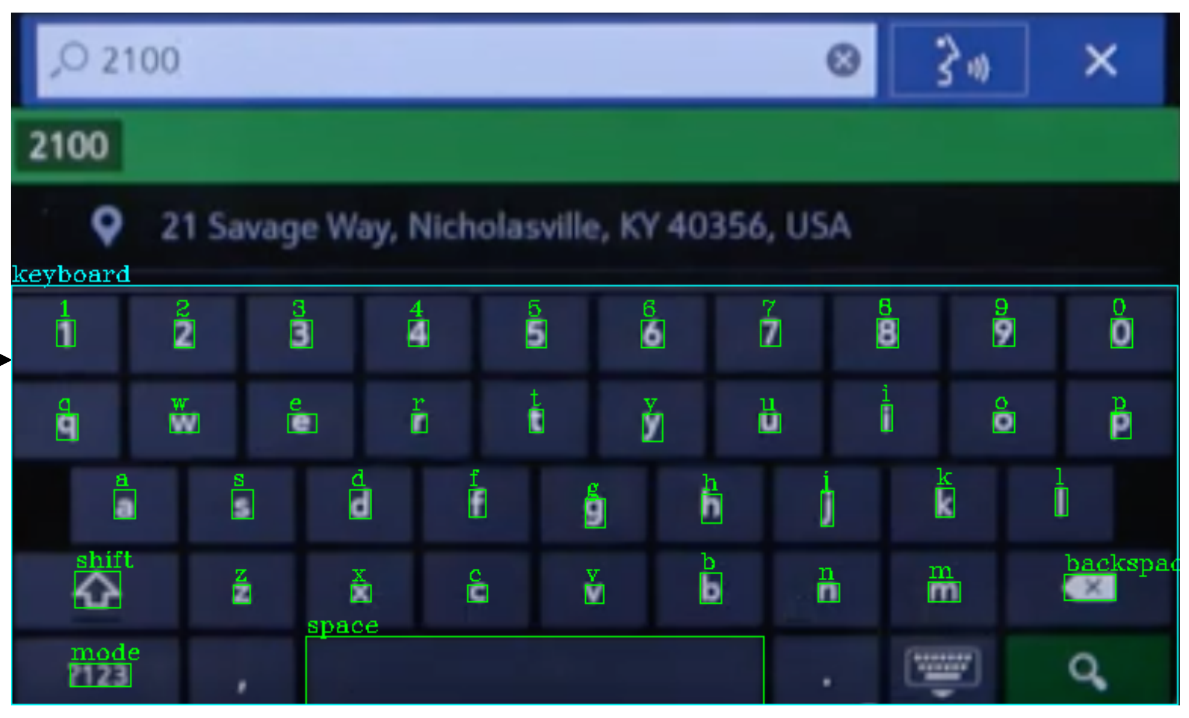


Figure 7: Final detection result after application of the post-processing algorithm. Missing characters were computed. False positives were removed. Special key "mode" was recognized and space was found thanks to the Canny edge detection.

Recognition results

Keyboard detection

Dataset	Precision	Recall	mAP@.5	mAP@.95
Validation	1	1	0.996	0.985
Test	1	1	0.996	0.971

Table 1: Results of selected tiny variation of the YOLOv7 model on the generated keyboard detection dataset

Character detection

Dataset	Precision	Recall	mAP@.5	mAP@.95
Validation	0.979	0.953	0.979	0.85
Test	0.979	0.951	0.977	0.848

Table 2: Results of the YOLOv7 model on the generated character detection dataset

Dataset	Precision	Recall	mAP@.5	mAP@.95
Validation	0.95	0.851	0.913	0.749
Test	0.948	0.852	0.912	0.748

Table 3: Results of tiny variation of the YOLOv7 model on the generated character detection dataset

Post-processing

	Normal model		Tiny model	
Charset	Precision	Recall	Precision	Recall
All	0.942	0.949	0.964	0.942
Alphabet	1	1	1	1
Numbers	0.998	0.987	1	0.993
Alphanumeric	**0.999**	**0.997**	**1**	**0.998**
Special keys	0.965	0.910	0.987	0.914
Special characters	0.684	0.767	0.762	0.705

Table 4: Results of post-processing on the validation dataset. Target alphanumeric characters outperform.