

# Waldboost balíček pro Python

Funkční vzorek TE01020415-V027

Roman Juránek, Petr Musil, Pavel Zemčík

Fakulta informačních technologií  
Vysoké učení technické v Brně  
Božetěchova 2, Brno 61266

## Popis

Balíček<sup>1</sup> implementuje trénování detektorů objektů pomocí algoritmu WaldBoost [1]. Lze jej jednoduše instalovat standardním způsobem pomocí nástroje pip. Závislosti jsou hlavně základní knihovny pro zpracování obrazu a strojové učení (numpy, opencv, skimage, sklearn). Podporovaná verze Pythonu je 3.6 nebo novější.

Jednou z důležitých vlastností knihovny je podpora FPGA - tedy trénování detektorů objektů implementovaných v programovatelném hardware.

Software je vydán jako open source s BSD licencí.

## Použití

Knihovna implementuje trénování, ale neřeší nijak strukturu trénovacích dat - toto je plně v režii uživatele, který musí vytvořit generátor trénovacích vzorků produkující obrázky a ohraničující obdélníky objektů. Výstupem je pak struktura s parametry detektoru, kterou je možné uložit do binárního souboru pomocí Protocol Buffers. Následující sekce ukazují zjednodušené použití knihovny, pro detaily je nutné použít dokumentaci.

## Generátor dat

Generátor slouží jako zdroj dat pro trénování. Jedná se o nekonečný cyklus produkující obrázky a k němu příslušné souřadnice objektů.

```
import waldboost as wb
def generator():
    While True:
        # ... read and transform data
        yield image, boxes
```

---

<sup>1</sup>Ke stažení zde <https://github.com/RomanJuranek/waldboost>

## Nastavení trénovacích parametrů

Mezi trénovací parametry patří velikost detekčního okna (shape) a parametry extrakce příznakových obrazových kanálů (channel\_opts) - ty definují jak se vstupní šedotónový obraz předzpracuje pro trénování klasifikátoru..

```
channel_opts = {
    "shrink": 2,
    "n_per_oct": 8,
    "smooth": 1,
    "target_dtype": np.float32,
    "channels": [ wb.channels.grad_hist ]
}
shape = (12,12,4)
```

## Trénování

Během trénování, algoritmus postupě čte obrázky z generátoru a extrahuje z nich trénovací vzorky. Pro trénování je použit algoritmus WaldBoost, který tvoří sekvenční klasifikátor pomocí jednoduchých slabých klasifikátorů.

```
model = wb.Model(shape, channel_opts)
wb.train(model, generator())
```

## Uložení a použití detektoru

Natrénovaný detektor je možné uložit do binárního souboru (a znovu jej načíst). Funkce `wb.detect` detekuje objekty na vstupním obrázku.

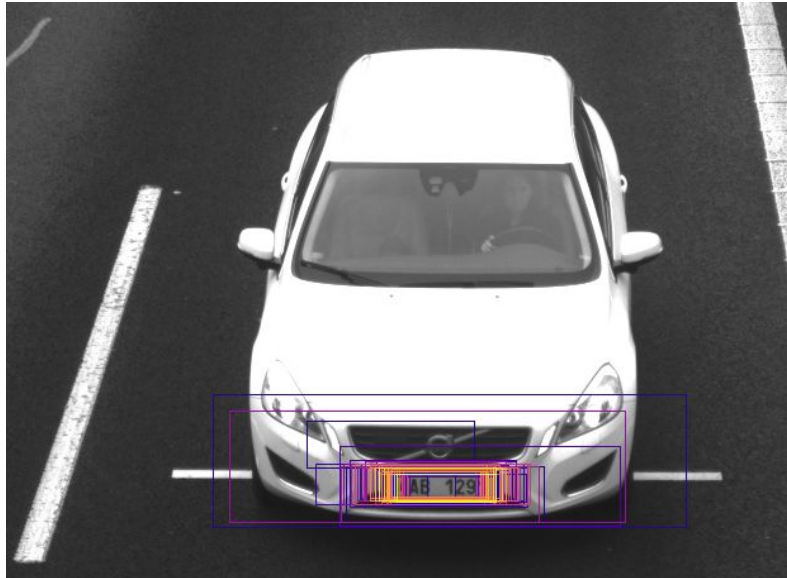
```
model.save("detector.bin")
# model = wb.load("detector.bin") # Load saved detector
Image = ... # load image
boxes = wb.detect(image, model)
# boxes contains locations and scores of detected objects
```

## Podpora FPGA

Detektory implementované v hardware mají zvláštní požadavky, které je nutné splnit aby byly takové detektory rychlé a zabíraly co nejméně zdrojů v cílovém hardware. Tyto požadavky jsou implementované v modulu `waldboost.fpga`. Jedná se zejména o výpočet příznakových kanálů pomocí integer aritmetiky, kvantizace odezev klasifikátorů a plánování přístupů do paměti během vyhodnocení klasifikátorů.

## Využití

Knihovna je úspěšně použita pro trénování detektorů poznávacích značek automobilů (demonstrováno na funkčním vzorku TE01020415-V027), ukázka na obrázku dole. A je využívána i v dalších projektech Fakulty informačních technologií.



## Reference

1. Sochman, Matas, WaldBoost – Learning for Time Constrained Sequential Detection, CVPR, 2005