



KATEDRA KYBERNETIKY

SEMESTRÁLNÍ PRÁCE Z PŘEDMĚTU KKY/ZDO

VARROA DETECTOR

Vítek Poór – poorv@students.zcu.cz
Petr Trnka – trnkpe@students.zcu.cz

28. května 2021

Obsah

1	Zadání	2
2	Návrh	2
3	Program	2
4	Experimenty	3
5	Závěr	6

1 Zadání

Navrhněte a naprogramujte nástroj pro detekci kleštíka (*varroa destructor*) v obraze. Vstupními daty jsou snímky směsného odpadu ze dna úlu (tzv. měl). Cílem je detekovat tvar kleštíků v daném snímku s úspěšností F1-skóre alespoň 55%.

2 Návrh

V semestrální práci budeme vycházet z vizuální podoby varroázy, tj. její velikosti (cca 2x2 px na QR kódu), tvaru (kulovitý, elipsoidický) a barevného odstínu.

Nejprve provedeme segmentaci podle předem odhadnuté škály hnědé barvy v šedotónu (odhad bude proveden na základ referenční varroázy). Dalším možným přístupem k segmentaci je použití metody Otsua. Tímto krokem bychom měli vyloučit pozadí a výrazně světlejší či výrazně tmavší objekty než varroáza.

V segmentovaném obrazu poté budeme hledat objekty, které mají eliptický tvar odhadnutého poloměru. Eliptické objekty budeme hledat jako kružnice pomocí vlastnosti kompaktnost s určitou mírou tolerance.

3 Program

Program byl vytvořen v prostředí Python verze 3.8.5 (64 bit). Program byl vyvíjen za pomoci vývojářského prostředí Visual Studio 2019 a veškeré testování v rámci vývoje bylo prováděno pomocí modulu s testovacími případy. Odkaz na git

[https : //github.com/vitekpoor/ZDO2021](https://github.com/vitekpoor/ZDO2021)

Vstupem programu je n-dimenzionální pole, jehož tvar je počet obrázků, výška, šířka a barevné kanály. Program také pracuje se souborem *config.json*, který načítá při inicializaci. Tento soubor musí existovat v adresáři, kde je program spouštěn a definuje hyperparametry programu.

Výstupem programu je obdobné n-dimenzionální pole poli na vstupu ale bez posledního prvku určující barevné kanály. Výstupem je tedy binární maska, která označuje detekované kleštíky.

Program obsahuje modul **test_zdo2021**, který obsahuje testovací scénáře jednonásobného a vícenásobného spuštění programu. Jelikož je program psán formou knihovničky, je doporučeno jeho spuštění provádět pomocí testů.

Jako testovací prostředí byl použit pytest verze 6.1.1. v prostředí Visual Studio.

Konfigurační soubor obsahuje následující parametry:

- **threshold** - Určuje ruční práh pro segmentaci vstupního obrázku prahováním (v procentech).
- **circ-min** - Určuje dolní mez odhadu kruhovosti jedné detekované oblasti.
- **circ-max** - Určuje horní mez odhadu kruhovosti jedné detekované oblasti.
- **area-min** - Určuje minimální plochu jedné detekované oblasti.
- **area-max** - Určuje maximální plochu jedné detekované oblasti.

Výše zmíněné parametry byly odvozeny na základě experimentálních měření (viz. Experimenty).

Samotné jádro programu, které provádí detekci je v modulu `zdo2021` ve skriptu `main.py`. Jedná se z pohledu objektového programování o třídu s názvem **VarroaDetector**. Detekce je potom určena její metodou **predict(self, data)**. V rámci inicializace se načítá konfigurační soubor.

4 Experimenty

Jak již bylo zmíněno výše, veškeré experimenty byly prováděny výhradně pomocí modulu s testovacími případy. V rámci experimentů jsme pracovali s vlastnostmi:

- **Práh** - Parametr segmentace prahováním.
- **Eulerovo číslo** - Vlastnost oblasti získané metodou třetí strany (`region props`).
- **Kruhovost** - Odhad kruhovosti pomocí vzorce $(4 * \pi * \text{Obsah}) / (\text{prop.perimeter}^2)$.
- **Obsah** - Vlastnost oblasti získané metodou třetí strany (`region props`).

Experimenty byly prováděny vždy nad celou datovou sadou (adresář `test/` `test_dataset`). Výsledné F1-skóre je tedy aritmetický průměr všech dílčích výsledků celé datové sady. Konfigurace s nejvyšším průměrným F1-skórem byla zvolena jako optimální pro náš detektor. Určení hodnot shrnuje následující tabulka konfigurací:

Threshold	Circ-min	Circ-max	Area-min	Area-max	Euler	F1-skóre
Otsu	0.7	1.1	100	-	-	0.0143
0.15	0.7	1.1	100	-	-	0.0512
0.25	0.7	1.1	100	-	-	0.1155
0.30	0.7	1.1	100	-	-	0.1396
0.40	0.7	1.1	100	-	-	0.0841
0.30	0.7	1.1	250	-	-	0.1693
0.30	0.7	1.1	500	-	-	0.0148
0.30	0.7	1.1	300	-	-	0.1352
0.30	0.7	1.1	200	-	-	0.1911
0.30	0.7	1.3	200	-	-	0.1911
0.30	0.7	1.0	200	-	-	0.1911
0.30	0.7	1.0	200	-	1	0.1653
0.30	0.8	1.0	200	-	-	0.1584
0.30	0.8	0.9	200	-	-	0.1645
0.30	0.7	1.1	200	1000	-	0.2259
0.30	0.7	1.1	200	750	-	0.2392
0.30	0.7	1.1	200	400	-	0.2171
0.30	0.7	1.1	200	500	-	0.2411
0.30	0.7	1.1	200	500	1	0.2101

Tabulka 1: Tabulka parametrů a průměrné F1-skóre pro jednotlivé konfigurace programu.

Nutno dodat, že ne všechny experimenty obsahují kompletní konfiguraci programu. Například Eulerovo číslo je použito pouze u dvou experimentů, jelikož při jeho použití vždy F1-skóre pokleslo. Obdobně maximální povolený obsah oblasti byl použit až při posledních experimentech, jelikož jsme na začátku vycházeli pouze z hodnot prahu, kruhovosti a minimální oblasti.

Z Tabulky 1 je patrné, že optimální hodnoty konfigurace z provedených experimentů jsou: práh na 30% jasového histogramu, kruhovost od 70% podobnosti kruhu a obsah oblasti mezi 200 až 500 body obrazu. Lze si povšimnout, že například rozšíření horního omezení kruhovosti nemá dopad při určité konfiguraci na výsledné skóre.

Pro vybranou konfiguraci bylo F1-skóre počítáno z těchto dílčích výsledků:

0.3802
0.1471
0.0605
0.0000
0.3592
0.0000
0.5665
0.1184
0.0000
0.4342
0.6165
0.0000
0.1098
0.0000
0.3630
0.0000
0.0000
0.1746
0.0000
0.0835
0.4976
0.6929
0.3381
0.4419
0.3258
0.5425
0.3887
0.3501
0.0000

Lze vidět, že jeden z obrázků použité datové sady měl úspěšnost detekce 69%. Na druhou stranu existují i takové případy, kde je úspěšnost detekce nulová.

5 Závěr

V rámci této práce byl vytvořen program, který se pokouší detekovat kleštíka na specifickém obrázku. Jak lze vidět z výsledků experimentů, úspěšnost F1-skóre tohoto programu je pro některé konkrétní vstupní obrazy až 69%. V průměru nad danou datovou sadu dosahuje jeho úspěšnost však pouhých 24%.

Zásadní nedostatek programu je v odhadu konfigurace programu, jelikož jsme z časových důvodů zdaleka neprokryli všechny možnosti. Další problém vidíme v QR kódu, který obsahuje každý obraz a jehož části se po segmentaci a následné detekci mohou jevit jako falešně pozitivní jedinci kleštíka. Tento problém je řešitelný například použitím některé z morfologických operací (uzavření) a následné odstranění oblasti výrazně větší, než maximální odhadnutá velikost kleštíka.

Dalším přístupem, vzhledem k požadovanému F1-skóre, by bylo nastavit výslednou masku na samé pozitivní případy a poté v ní odmazávat oblasti, které určitě kleštík nejsou. Tímto přístupem bychom začínali pokaždé na 50% skóre. Domníváme se totiž, že čím víc jsme modifikovali spodní mez obsahu oblasti, tím víc nám rostlo výsledné skóre. To znamená, že jsme na výstupu označovali více falešně pozitivních vzorků a odchylovali jsme se od koncepčně "správné" detekce, ve které chceme detekovat co nejméně objektů s co největší přesností.