

Aplikované evoluční algoritmy

6. Kartézské genetické programování - obrazový filtr

David Hudák, xhudak03

23. března 2023

1 Úvod

Jako zadání jsem zvolil projekt na téma Kartézské genetické programování - obrazový filtr, kdy je cílem právě s pomocí kartézského genetického programování (dále jen CGP) vytvořit obrazový filtr pro zašuměné obrázky výstřelovým šumem. Tedy konkrétně na obrázku se vyskytují šumy, které dávají na poškozené pozice náhodné hodnoty – například pokud pixely nabývají hodnot $< 0, 255 >$, pak se pixely na náhodných pozicích změní do nějaké hodnoty z tohoto intervalu $[1]$.

Konvenční metody řešení tohoto problému se jistým způsobem dělí na dva odlišné přístupy – jeden mediánový (či jiný) filtr, nebo dva filtry. První varianta dělá všechno najednou, druhá detekuje šumy jedním filtrem a druhým je odstraňuje. Při řešení tohoto projektu bude cílem vytvořit právě takový filtr, který nejprve detekuje, zda se má opravovat, a pokud ano, pak teprve pixel opraví (jinak bude dávat identitu).

2 Použité nástroje a technologie

Pro implementaci daného projektu bude použit programovací jazyk Python ve verzi 3.7 (z důvodu kompatibility s knihovnami), nástroj pro práci s CGP HAL-CGP¹. Při instalaci bylo nutné nainstalovat také následující knihovny:

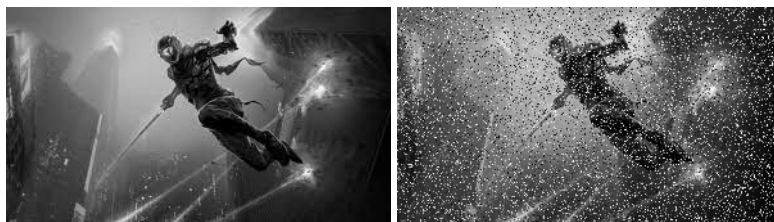
- numpy
- OpenCV
- matplotlib
- scipy

Jako pracovní prostředí slouží operační systém Linux v prostředí Ubuntu 20.04.

3 Použitá data

Jelikož je autor velkým fanouškem videoher, pro experimenty bude používat herní wallpapery převedené na černobílý obrázek s pomocí nástroje GIMP. V tomto nástroji pak bude také použita tvorba šumu. Viz wallpaper ze hry Ghostrunner na obrázku 1.

¹Viz <https://github.com/Happy-Algorithms-League/hal-cgp>



Obrázek 1: Ghostrunner – nalevo původní obrázek, vpravo obrázek se zhruba 22.5% ostrým zašuměním.

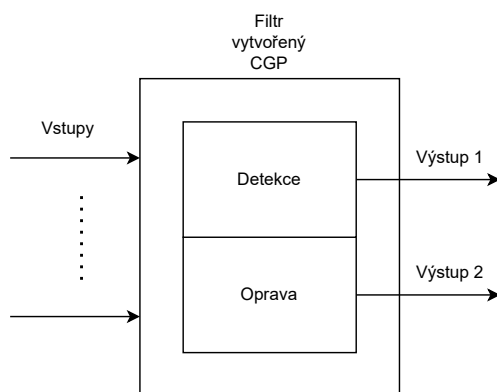
4 Úvodní implementace/experimenty

Před odevzdáním této zprávy bylo implementováno základní (baseline) řešení, které načítá obrázek, ten dává jako trénovací vstup do nijak významně neupravované příkladové struktury CGP opět s výchozími parametry s jedním výstupem (opravujícím), fitness funkcí typu MSE (mean square error) a funkčními bloky se základními funkcemi (sčítání, násobení, chráněné dělení apod.). S touto implementací bylo dosaženo velmi mírného překonání identity v kratším trénovacím čase (asi 15 minut).

Fitness je počítána tím způsobem, že se vezme původní obrázek a obrázek prohnaný vyevoluovaným filtrem, které se následně porovnají s pomocí fitness funkce. Z tohoto důvodu musela být importována knihovna scipy, která umožňuje filtrovat obrázky generickým filtrem.

5 Plány

V plánu je upravit strukturu CGP pro dva výstupy (jeden detekční, druhý opravující, viz obrázek 2), přidat/odebrat další funkční bloky, vyzkoušet různé fitness funkce (PSNR, MAE) a různé parametry učení. V rámci experimentů také bylo pracováno s omezeným okolím 3×3 , přičemž v plánu je vyzkoušet i okolí 5×5 či případně jiné vzory. Schéma řešení by velmi zjednodušeně mělo vypadat následovně:



Obrázek 2: Ideový přehled fungování filtru vytvořeným CGP.

Reference

- [1] Vasicek, Z.; Bidlo, M.: Evolutionary design of robust noise-specific image filters. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011, s. 269–276, doi:10.1109/CEC.2011.5949628.