

# Grafický CLI editor

Roman Princ

ČVUT-FIT

princrom@fit.cvut.cz

25. května 2024

## 1 Úvod

Práce je zaměřena na tvorbu grafického editoru ovládaného pomocí CLI. Aplikace na vstupu přijme parametry z příkazové řádky, které zpracuje a aplikuje na vstupní obrázek. Výstup se uloží do, uživatelem definované, cesty, přičemž vstupní obrázek zůstane beze změny.

## 2 Vstupní data

Aplikace na vstupu přijme obrázek v .jpg nebo .png formátu. Ostatní formáty nejsou podporovány. Obrázek se převede na pole z knihovny numpy. Pole má 2 dimenze pokud je obrázek černobílý (grayscale), a nebo 3 dimenze (RGB).

### 2.1 Operace

Operace jsou pomyslně rozděleny na dvě kategorie. První kategorie obsahuje méně výpočetně náročné operace, jako je zrcadlení, zesvětlení, ztmavení, inverze, konverze na černobílý (grayscale) obrázek a rotace. Druhá kategorie obsahuje více výpočetně náročné operace. Jsou nimi vyostření, rozostření, detekce hran a aplikace Roberts cross[2] filtru.

## 3 Výběr algoritmů

Nejdůležitější byl pro tuto práci výběr algoritmu pro aplikování filtrů.[1]

### 3.1 Základní operace

Pro implementaci základních grafických operací, jako jsou zrcadlení, zesvětlení, ztmavení, inverze a převod na černobílý obrázek, byly využity matricové operace knihovny numpy. Tím bylo dosaženo rozumné efektivity a minimalizace použití cyklů, které jsou v Pythonu pomalejší.

### 3.2 Použití FFT konvoluce

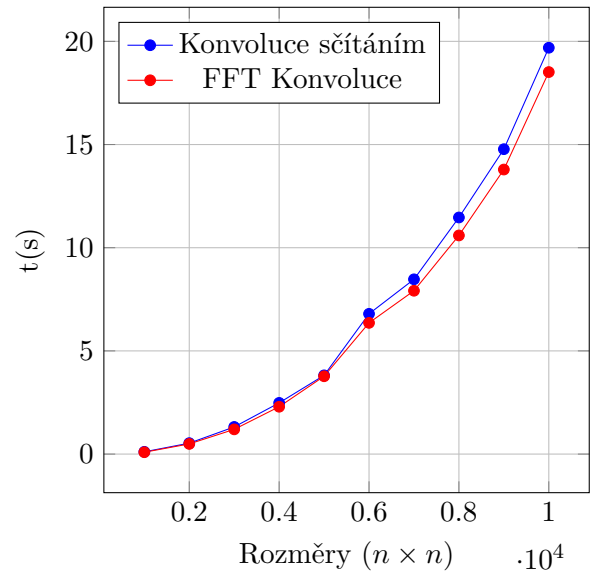
Pro operace vyostření, rozostření a detekce hran je použita FFT konvoluce místo té běžné. Časová složitost tohoto algoritmu je  $O(n \log n)$  ve srovnání

s  $O(n^2)$  u běžné konvoluce. Je to díky tomu, že FFT konvoluce využívá rychlou Fourierovu transformaci. Ta umožňuje provádět operaci násobení ve Fourierově prostoru. V porovnání s iterativním násobením a sčítáním přes všechny pixely je tato operace výrazně rychlejší.[4]

## 4 Výsledky

Implementace grafického editoru byla úspěšná a aplikace umožňuje provádět všechny požadované operace na vstupním obrázku. Testování ukázalo, že použití FFT konvoluce zrychlilo operace vyostření, rozostření a detekce hran ve srovnání s běžnou konvolucí.

### 4.1 Porovnání rychlosti



Výsledky testování ukazují, že FFT konvoluce je rychlejší než běžná konvoluce, což potvrzuje její výhody. Ovšem výrazné skoky v rychlosti jsou vidět až při větších vstupech. Pro testování byl použit Sobel–Feldman filtr[3]. Použitý počítač pro testování byl HP 250 G8.

## 5 Závěr

Tato práce je užitečná pro rychlou úpravu obrázků. V budoucnu se dá rozhodně rozšířit v několika smě-

rech. Jedno z možných rozšíření je rozhodně uživatelské rozrhaní, a nebo rozšíření kolekce filtrů a přidání další logiky.

## Reference

- [1] Wikipedia Contributors. Kernel (image processing) — Wikipedia, the free encyclopedia. online, 2024. [https://en.wikipedia.org/wiki/Kernel\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing)).
- [2] Wikipedia Contributors. Roberts cross — Wikipedia, the free encyclopedia. online, 2024. [https://en.wikipedia.org/wiki/Roberts\\_cross](https://en.wikipedia.org/wiki/Roberts_cross).
- [3] Wikipedia Contributors. Sobel operator — Wikipedia, the free encyclopedia. online, 2024. [https://en.wikipedia.org/wiki/Sobel\\_operator](https://en.wikipedia.org/wiki/Sobel_operator).
- [4] Autor neuveden. Convolution and fft. online, 2024. <https://www.cs.princeton.edu/courses/archive/spr05/cos423/lectures/05fft.pdf>.