



Bakalářská práce

Program pro poloautomatickou analýzu obrazu

Studijní program:

B0613A140005 Informační technologie

Studijní obor:

Aplikovaná informatika

Autor práce:

Jan Hartig

Vedoucí práce:

doc. Ing. Josef Chaloupka, Ph.D.

Ústav informačních technologií a elektroniky

Liberec 2024



Zadání bakalářské práce

Program pro poloautomatickou analýzu obrazu

<i>Jméno a příjmení:</i>	Jan Hartig
<i>Osobní číslo:</i>	M23000057
<i>Studijní program:</i>	B0613A140005 Informační technologie
<i>Specializace:</i>	Aplikovaná informatika
<i>Zadávací katedra:</i>	Ústav informačních technologií a elektroniky
<i>Akademický rok:</i>	2023/2024

Zásady pro vypracování:

1. Seznamte se s problematikou zpracování a rozpoznávání obrazu.
2. Navrhněte a realizujte systém (v Pythonu) pro poloautomatické zpracování a rozpoznávání obrazu.
3. Tento systém by měl obsahovat přívětivé prostředí, ve kterém budou použity jednotlivé algoritmy pro zpracování a rozpoznávání obrazu z knihovny OpenCV. Celý systém musí být navržen tak, aby byl ?otevřený?, tj. aby se v budoucnu daly do programu přidávat nové moduly.

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: 30–40 stran
Forma zpracování práce: tištěná/elektronická
Jazyk práce: čeština

Seznam odborné literatury:

- [1] Šonka, M., Hlaváč, V., Boyle, R.: Image processing, analysis, and machine vision. Fourth Edition. Australia: Cengage Learning, ISBN 978-1-133-59369-0, 2015.
- [2] Gonzales, R. C., Woods, R: Digital image processing. Global edition. New York: Pearson, ISBN 978-1-292-22304-9, 2017.
- [3] Hlaváč, V., Sedláček, M.: Zpracování signálů a obrazů. 2. přeprac. vyd. Praha: ČVUT, 255 s. ISBN 978-80-01-03110-0, 2007.
- [4] Liu, Y.: Deep Learning Based Image Processing: Recent Advances and Future Trends. In Eliva Press, 2022. ISBN 978-9994982554.

Vedoucí práce: doc. Ing. Josef Chaloupka, Ph.D.
Ústav informačních technologií a elektroniky

Datum zadání práce: 6. září 2023
Předpokládaný termín odevzdání: 14. května 2024

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Ing. Josef Chaloupka, Ph.D.
garant studijního programu

V Liberci dne 6. září 2023

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

10. 5. 2024

Jan Hartig

Program pro poloautomatickou analýzu obrazu

Abstrakt

Tato bakalářská práce se zaměřuje na problematiku zpracování a rozpoznávání obrazu a navrhuje systém pro poloautomatickou analýzu obrazu v jazyce Python. Práce se věnuje úvodu do oblasti zpracování a rozpoznávání obrazu, s důrazem na využití knihovny OpenCV. Navržený systém disponuje přívětivým uživatelským prostředím, které umožňuje interaktivní používání jednotlivých algoritmů. Klíčovým požadavkem je otevřenost systému pro budoucí rozšíření, což umožňuje snadné přidávání nových modulů pro další zpracování obrazu. Tímto způsobem práce přispívá k vytvoření flexibilního nástroje pro poloautomatickou analýzu obrazu s potenciálem pro další rozvoj a aplikace.

Klíčová slova: Python, GUI, OpenCV, zpracování obrazu, počítačové vidění, analýza obrazu

Program for semi-automatic image analysis

Abstract

This bachelor thesis focuses on the problem of image processing and recognition and proposes a system for semi-automatic image analysis in Python. The thesis provides an introduction to the field of image processing and recognition, with an emphasis on the use of the OpenCV library. The proposed system has a user-friendly environment that allows interactive use of the different algorithms. A key requirement is the openness of the system for future extensions, allowing new modules to be easily added for further image processing. In this way, the work contributes to a flexible tool for semi-automated image analysis with potential for further development and applications.

Keywords: Python, GUI, OpenCV, image processing, computer vision, image analysis

Poděkování

Rád bych zde poděkoval vedoucímu této bakalářské práce doc. Ing. Josefu Chaloupkovi, Ph.D. za čas a trpělivost, kterou mi věnoval při zpracování projektu, věcné rady a poskytnutí studijních materiálů a zdrojů.

Děkuji také své přítelkyni Mgr. Barboře Šimkové za podporu a trpělivost při psaní této práce.

Obsah

Seznam zkratk	8
1 Úvod	9
2 Návrh programu	10
2.1 Popis prostředí	12
2.2 Logika prostředí	13
2.3 Tvorba prostředí	15
3 Použité technologie	16
3.1 Python	16
3.2 OpenCV	16
3.3 PyQt	16
3.4 Pomocné knihovny	17
4 Použité funkce	18
4.1 Statické funkce	18
4.1.1 Historie úprav	18
4.1.2 Výběr části a zvětšení	19
4.1.3 Histogram obrazu	21
4.2 Konfigurovatelné funkce	22
4.2.1 Barevné prostory	22
4.2.2 Geometrická transformace	28
4.2.3 Filtrace obrazu	32
4.3 Funkce segmentace obrazu	35
4.3.1 Prahování	36
4.3.2 Srovnávání se vzorem	38
4.4 Funkce matematické morfologie	39
4.4.1 Dilatace a eroze	40
4.4.2 Otevření a uzavření	43
4.4.3 Gradient	43
4.4.4 Vrchní a spodní část klobouku	47
5 Příklad použití programu	50
Závěr	55

Použitá literatura	56
A Přílohy	57
A.1 Příklady změny násobícího jádra	57

Seznam zkratek

TUL	Technická univerzita v Liberci
FM	Fakulta mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci
GUI	Graphical User Interface – grafické rozhraní
UI	User Interface
RGB	Red, green, blue - barevný model
HSV	Hue, saturation, value - barevný model
YCbCr	Luma, Chroma blue-difference, Chroma red-difference - barevný model
CV	Computer vision – počítačové vidění
OpenCV	Open Source Computer Vision Library
VS Code	Vývojové prostředí Visual Studio Code
BSD	Berkeley Software Distribution - druh licence
PyQt	Spojení programovacího jazyka Python a multiplatformního grafického rozhraní
R / W	Read / Write - čtení a zapisování

1 Úvod

Zpracování obrazu a je oblastí v informatice a umělé inteligenci, které získávají stále větší význam v dnešní digitální éře. Tyto disciplíny nejenže nabízejí široké spektrum aplikací od biometrické identifikace po autonomní vozidla, ale také poskytují základy pro inovace v mnoha odvětvích, přičemž některé aplikace mohou mít zásadní dopady na lidský život a společnost jako celek.

Zpracování obrazu je prováděno pomocí široké škály technik a algoritmů, které umožňují manipulaci a analýzu digitálních obrazů za účelem získání užitečných informací. Tyto techniky zahrnují filtrování, segmentaci, detekci objektů a zvýraznění rysů. Od jednoduchých úprav a zlepšení obrazu po pokročilé analýzy a rozpoznávání vzorů, zpracování obrazu nachází uplatnění v medicíně, bezpečnostních systémech, průmyslové kontrole, robotice, virtuální realitě, rozšířené realitě a mnoha dalších oblastech.

Tato bakalářská práce je zaměřena na vývoj a vytvoření programu pro poloautomatickou analýzu obrazu využívajícího prostředí Python a knihovnu OpenCV. Cílem tohoto programu je poskytnout uživatelům nástroje pro efektivní zpracování a úpravu obrazů, což může být klíčové pro mnoho aplikací od lékařské diagnostiky po průmyslovou automatizaci.

Důležitým aspektem této práce je také otevřenost systému pro rozšiřování o nové funkcionality a technologie. Flexibilita a modularita jsou klíčové pro zajištění dlouhodobé udržitelnosti a přizpůsobitelnosti systému měnícím se potřebám a technologickým trendům, což je obzvláště důležité v oblasti, která se rychle vyvíjí a mění.

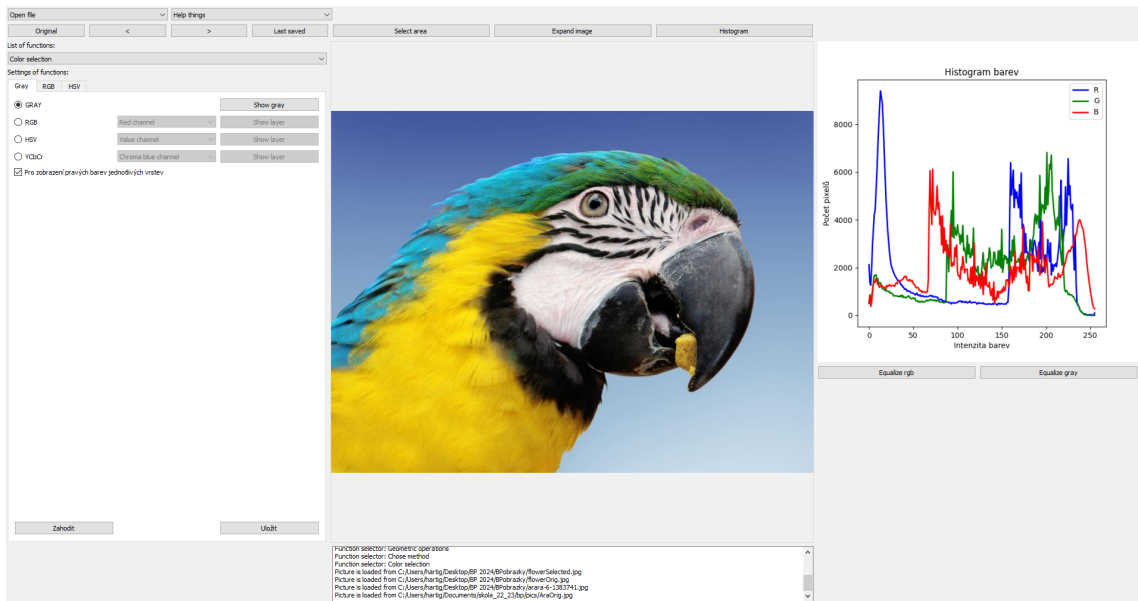
V rámci této práce bude kladen důraz nejen na efektivitu a spolehlivost algoritmů, ale také na uživatelskou přívětivost a intuitivnost uživatelského rozhraní. S ohledem na různé uživatele s různými úrovněmi znalostí v oblasti zpracování obrazu a programování je klíčové zajistit, aby systém byl snadno použitelný a přístupný pro širokou uživatelskou základnu.

V následujících sekcích této práce budou podrobněji představeny nejen samotný systém a jeho struktura, ale také jednotlivé metody z oblasti zpracování a rozpoznávání obrazu, včetně jejich implementace a možností přizpůsobení a rozšíření o další funkcionality a moduly.

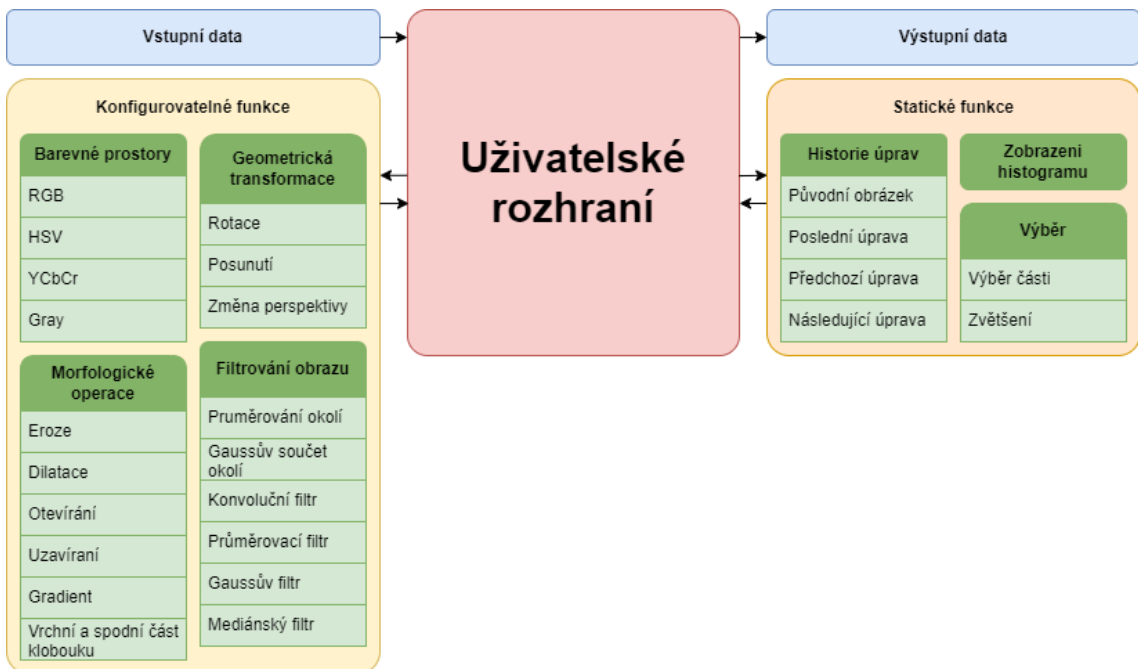
2 Návrh programu

Tato práce se zabývá návrhem programu pro grafické uživatelské rozhraní. V rámci práce bylo použita řada již vyvinutých funkcí, z toho důvodu je rešeršní a praktická část provázána dohromady. Jedná se o využití metod sloužících pro předzpracování obrazu z důvodů zlepšení obrazu k jeho následnému lepšímu zpracování. (obr. 2.2) Cílem je minimalizovat šum a další negativní vlivy, které vznikly při digitalizaci či přenosu obrazu, odstranit zkreslení či potlačit či zvýraznit žádoucí prvky. Výsledkem však nejsou nové informace, nýbrž, dle Shannonovy teorie, potlačené či zdůrazněné zacílené informace.[4]

Pro práci bylo zvoleno Visual Studio a programovací jazyk Python. Do GUI jsou nahrána určitá vstupní data, která si žádají zpracování signálů a obrazů. V rámci inicializace jsou volány jednotlivé funkce, které jsou oddělené v samostatném souboru. Do souboru s inicializací GUI se vloží možnost, způsob volání a uložení do globálních proměnných, v souboru s funkcemi se definuje nová funkce s patřičnými vstupy a výstupy. Navržené grafické uživatelské rozhraní rozděluje funkce na konfigurovatelné a statické. Mezi konfigurovatelné jsou zařazeny takové, které upravují barevné prostory. V této oblasti aplikují modely RGB, GRAY, HSV, YCbCr. Geometrická transformace implementuje rotace, posunutí a změnu perspektivy. Funkce zabývající se morfologickými operacemi jako je eroze, dilatace, otevírání, uzavírání, gradient a vrchní a spodní část klobouku. Co se týká filtrování obrazu bylo zařazeno průměrování okolí, Gaussův součet okolí, konvoluční, Gaussův a mediánový filtr.



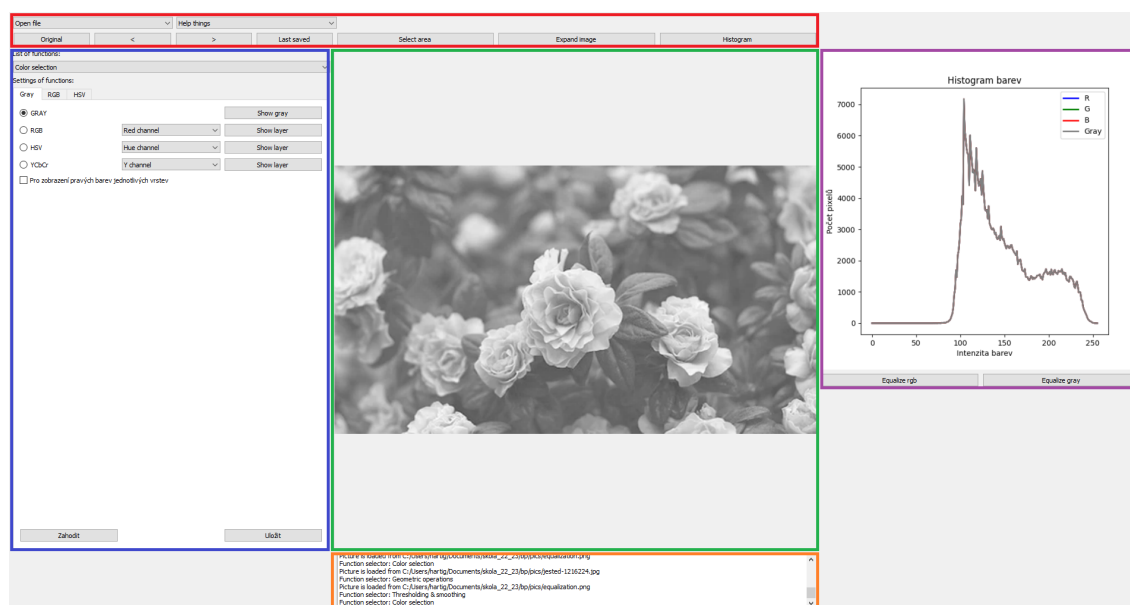
Obrázek 2.1: Zobrazení výsledné celé aplikace



Obrázek 2.2: Diagram zobrazující použité funkce

2.1 Popis prostředí

Pomocí knihovny PyQt bylo navrženo přehledné uživatelské prostředí naprogramované v prostředí Visual Studio Code. Prvním krokem při práci v navrženém GUI je potřeba se v něm orientovat. Horní, červená část, nabízí možnost nahrávání souborů, zobrazení nápovědy, historii úprav, výběr části a zvětšení a zobrazení histogramu. V hlavním okně je na levé straně, v modrém obdélníku, je výběr konfiguračních funkcí. Uprostřed, v zeleném poli, se zobrazuje nahraný obrazový soubor a na pravé straně, zvýrazněno fialově, je možné zobrazit histogram aktuálního obrazu. Ve spodní části okna v oranžovém obdélníku se nachází konzole, kde se zobrazují chyby, které mohou nastat při manipulaci s obrazem či při volání funkcí.



Obrázek 2.3: Hlavní funkční části programu

Při kliknutí na tlačítko „File manipulation” se zobrazí základní výběr, který je všeobecně známý i z jiných prostředí. Zařazuje možnosti otevření a uložení dat i jednotlivých procedur. Uživateli umožní nahrát soubor, případně exportovat výsledek do složky. Formáty vstupního a výstupního obrazu jsou limitovány knihovnou PyQt, která se stará o manipulaci s daty a převádění obrazu na datový typ QPixmap, který je vstupem a výstupem pro všechny implementované funkce v programu.

Zkratka	Celý název	Podpora v PyQt
.bmp	Window Bitmap	R / W
.gif	Graphic Interchange Fromat	R
.jgg	Joint Photographic Group	R / W
.jpeg	Joint Photographic Experts Group	R / W
.png	Portable Network Graphics	R / W
.pbm	Portable Bitmap	R
.pgm	Portable Graymap	R
.ppm	Portable Pixmap	R / W
.xbp	X11 Bitmap	R / W
.xpm	X11 Pixmap	R / W

Tabulka 2.1: Seznam povolených vstupních souborů

Při zvolení možnosti „Help” se zobrazí list všech dostupných funkcí. Pod těmito dvěma tlačítky je zařazen soubor statických funkcí, které mohou být zavolány po každé manipulaci s obrazem. Nachází se zde možnost zobrazení originální, původního obrazu, zároveň možnost zobrazení poslední úpravy a uživateli je nabídnuto prohlížet historii úprav pomocí šipek vpřed a vzad. Dále v těchto funkcích je obsažen histogram po aktuální provedené operaci, výběr části z obrazu a jeho rozšíření do maximálního prostoru. Pod těmito funkcemi je na levé straně seznam souboru konfigurovatelných funkcí, mezi ně patří výběr barevného prostoru, geometrická transformace, filtrování obrazu a morfologické operace. Pod každým souborem jsou implementovány jednotlivé základní grafické operace, například eroze, dilatace apod. U každé z těchto operací jsou k nalezení možnosti úpravy dané funkce, například nastavení stupně při otočení, výběr barvy pomocí dané metody (RGB, HSV, ...). Po každé operaci je možné nastalou změnu uložit do paměti programu či naopak dané změny vymazat a následně zavolat jinou funkci.

2.2 Logika prostředí

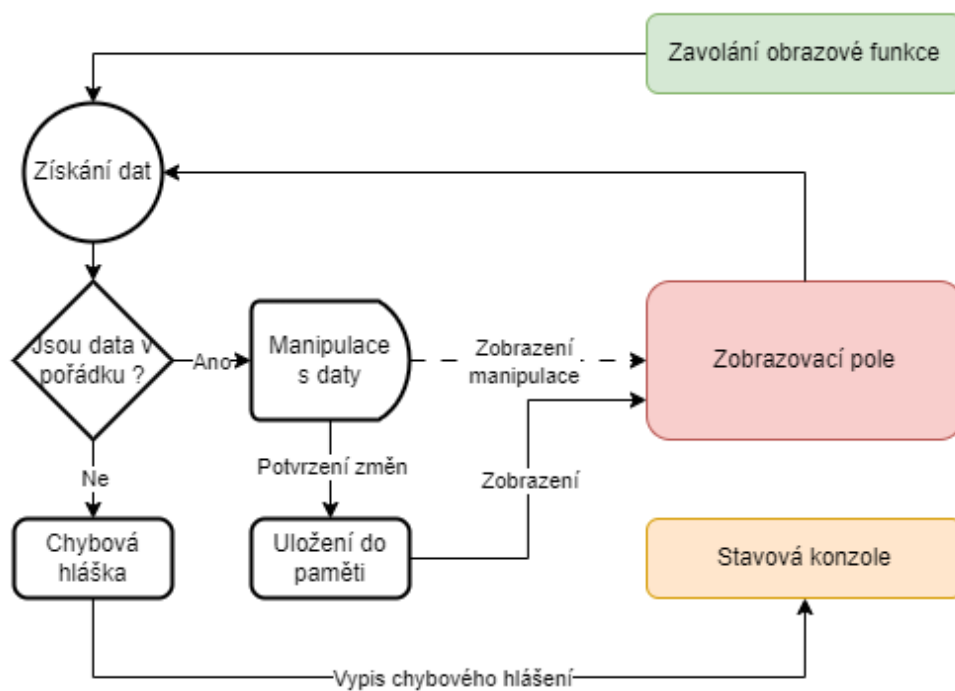
Soubory funkcí sdílí informace o aktuálně zpracovávaných datech pomocí předávání si obrazu z hlavního zobrazovacího okna. Například, při načtení počátečního obrazu se přidá položka do paměti programu, která funguje podobným způsobem jako struktura slovník (dict) v Pythonu. Kde klíč je slovo „original“. Hodnota na klíči je objekt typu QPixmap vytvořený převodem počátečního obrazu. Obraz je převeden z důvodu zobrazení v předdefinovaném zobrazovacím poli QLabel.

Může následovat další práce s obrazem, tedy zavolání jiné funkce. V následujícím příkladu je stručně popsán výběr části obrazu, podrobněji popsán v kapitole [4.1.2]. Při aktivaci funkce, v tomto případě tedy kliknutí na příslušné tlačítko, si funkce převezme aktuální stav z hlavního zobrazovacího okna. Pokusí se z daného okna získat objekt QPixmap a převést ho na matici hodnot (pokud se tento převod nepodaří, je vypsána příslušná chybová hláška do konzole pod hlavním zobrazovacím polem). Předpokládáme, že se obrazová data podařilo převést, potom následuje provedení samotné funkce. Z knihovny OpenCV se zavolá příkaz pro vytvoření okna

pro výběr ROI (rectangle of interest) se správnými vstupními parametry, které jsou v tomto případě pouze dva. Jedná se o jméno okna pro výběr a vstupní obraz, který je převeden do akceptovatelného typu. Následně už uživatel může opakovaně zkoušet výběr pomocí kurzoru. V okně se zobrazuje obdélník ohraničující výběr obrazu. Pokud uživatel s výběrem spokojený, potvrzuje ho stisknutím klávesy „Enter“. Obraz se automaticky nastaví na místo původního obrazu, následně je možné výběrové okno uzavřít. V případě, když chce uživatel zrušit výběr a pokračovat jinou úpravou, stiskne tlačítko „ESC“ a okno uzavřít.

Spolu s potvrzením proběhne zavolání vnořené funkce pro uložení do paměti, kde je vstupní parametr název provedené operace. V tomto případě „selectedArea“. Tato vnořená funkce, přidá ke vstupnímu parametru index, který zastupuje počet operací. Tím vznikne nový klíč, ke Uložená data v paměti programu je možné zpětně zobrazovat pomocí šipek zpět a vpřed („<“ a „>“). Díky tomu je umožněn opakovaný výběr nebo provedení jiné obrazové funkce.

Podobným způsobem fungují i ostatní funkce, které jsou popsány v kapitole [4 Použité funkce.] Jen s tím rozdílem, že funkce, která ukládá změněná data do slovníků, je provedena až po potvrzení tlačítkem „Uložit změny“, které je vždy na vybrané kartě s funkcí.



Obrázek 2.4: Diagram implementace zavolání grafické funkce

2.3 Tvorba prostředí

Celý program byl napsán ve vývojovém prostředí Visual Studio Code (běžně zkracováno jako VS Code). Jedná se o textový editor s pokročilými funkcemi pro vývoj software v mnoha programovacích jazycích, který byl vyvinut společností Microsoft. VS Code je distribuován pod licencí MIT (Massachusetts Institute of Technology). Jedná se o nejméně restriktivní open source licenci. Takto licencovaný program je možné bez omezení používat a šířit, pokud z programu není odstraněna kopie licence a jméno autora. VS Code poskytuje velké množství rozšíření pro kterýkoliv programovací nebo značkovací jazyk. Hlavním rozšířením je kontrola a zvýrazňování syntaxe právě psaného kódu nebo interaktivní dokončování příkazů. V případě psaní funkce nabízí příklady vstupů a jejich typy. Dalším užitečným doplňkem je podpora Gitu, což je název systému pro správu verzí kódu při psaní a vývoji software. Lze tak bez použití další externí knihovny revidovat změny v kódu, vrátet se zpět k verzím jednotlivých souborů a implementovat potřebné změny. Hlavním výhodou Gitu je možnost spolupráce více lidí na jednom kódu nebo jednoho člověka z více zařízení. Kód nebo ostatní potřebné soubory jsou díky tomu uloženy na cloudovém úložišti neboli vzdálených serverech. V případě soukromých projektů lze na Gitu omezit viditelnost pro veřejnost nebo v případě projektů s vyšší bezpečností (příkladem může být práce na projektu v soukromém sektoru) je možným řešením využít gitovací služby, kterou poskytuje zadavatel na vlastním serveru. Tento projekt je volně dostupný na webové službě [GitLab](#) a [GitHub](#).

3 Použité technologie

3.1 Python

Python je populární univerzální programovací jazyk. Oblíbenost si získal především svou jednoduchostí, čitelností a široké škálou využití. Python spadá pod licenci BSD (Berkeley Software Distribution), pro šíření software má podobné požadavky jako licence MIT, jméno autora a kopii licence, avšak povinností navíc je upozornění na zřeknutí se odpovědnosti za dílo. Python lze využít jako nástroj pro vývoj široké řady aplikací od webových stránek, přes desktopové a mobilní aplikace až po nástroje datové analýzy. Další užitečné uplatnění má v oblasti vývoje skriptů pro automatizaci úloh nebo v systémové administraci. Jelikož se jedná o interpretovaný jazyk, nepotřebuje před spuštěním kompilaci do strojového kódu. Umožňuje rychlý vývoj, testování programů a případné ladění chyb. Pro tuto bakalářskou práci byl zvolen z důvodu velkého množství rozšíření nebo-li knihoven.

3.2 OpenCV

Jednou z těchto knihoven, které napomohly realizovat tuto práci, je knihovna OpenCV (Open Source Computer Vision Library). Nabízí široké spektrum funkcí v oblasti počítačového vidění, detekce objektů, rozpoznávání tváří, sledování pohybu, analýzy obrazu nebo videa. Obsahuje implementace velkého množství algoritmů pro zpracování vstupních obrazových dat. Je distribuována pod Open source licenci, typu Apache 2, která má podobné nároky jako výše zmíněná BSD. Její výhodou je, že umožňuje udělování patentových licencí. Znamená to tedy, že je volně dostupná pro použití, úpravu a sdílení, může být použita volně i v komerčním využití.

3.3 PyQt

Značnou část, převážně obrazových funkcí, práce demonstruje v prostředí, které je realizované pomocí knihovny PyQt. Distribuována je pod licenci GPLv3 (General Public License verze 3). Je to jedna z dalších licencí pro svobodný software, s rozdílem toho, že v některých případech i za nekomerční využití mohou být distribuovány poplatky. V případě této bakalářské práce je kód zpřístupněn a zveřejněn. Rozšíření nabízí nástroje pro tvorbu a vývoj grafických uživatelských prostředí (graphical user interface, zkráceno a na GUI). Jedná se o knihovnu postavenou na platformě Qt,

kteřá je řiroce pouřívána v řrůmyslu a nabízí obsáhlou sadu nástrojů pro tvorbu moderních a responzivních GUI. Kombinace těchto dvou knihoven umožňuje vývoj desktopových, webových, mobilních nebo vědeckotechnických aplikací, které mohou fungovat bez ohledu na operační systém.

3.4 Pomocné knihovny

NumPy (zkratka z anglického Numerical Python), distribuována pod licenci BSD stejně jako Python, je jednou z podpůrných knihoven výsledného programu. Zprostředkovává efektivní práci s vícerozměrnými poli dat. Knihovna je optimalizována v jazyku C, disponuje velkou rychlostí zpracování dat. Pod pojmem data si v kontextu této práce lze představit obrazová data, která jsou reprezentována jako víceúrovňové matice.

Knihovna **Pillow** (dříve PIL Fork) se stala nezbytnou součástí této bakalářské práce. Poskytována je pod HPND licenci (z anglického Historical Permission Notice and Disclaimer). Jedná se o zastaralejší verzi, která je však plně kompatibilní s výše zmíněnými licencemi. Nabízí vývojáři funkce pro zpracování obrazu, kompatibilní s OpenCV. Využita byla především na převody mezi různými výstupními a vstupními typy jednotlivých funkcí v programu.

Nepostradatelným rozšířením od samotného zobrazení zpracovávaných dat se stala knihovna **Matplotlib** [6]. Knihovna spadá pod licenci GPLv3, avšak bez nutnosti zpřístupňovat zdrojový kód při jakémkoliv druhu použití. Umožňuje vytvářet dynamické a interaktivní zobrazení dat. Nabízí širokou škálu funkcí pro tvorbu grafů, přes lineární, sloupcové, kruhové až po 2D/3D grafy. V této práci byla využita především možnost zobrazení a vykreslení histogramu (kapitola 4.1.3). Čímž umožňuje snadnější pochopení komplexních informací. Poskytuje popsání os, nastavení barev pro jednotlivé soubory dat nebo zobrazení několika grafů vedle sebe a sledování změn.



Obrázek 3.1: Python



Obrázek 3.2: OpenCV



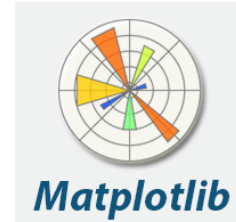
Obrázek 3.3: PyQt



Obrázek 3.4: NumPy



Obrázek 3.5: Pillow



Obrázek 3.6: Matplotlib

4 Použité funkce

Pro základní potřeby uživatelů, kteří pracují s obrazovým materiálem, ve kterém došlo při jeho pořizování, digitalizaci či přenosu ke vzniku šumů, zkreslení nebo přesunu zásadních informací do pozadí, byly využity základní funkce, které napomáhají k odstranění těchto nežádoucích procesů nebo naopak poslouží k vyzvednutí informací, na které uživatel cílí. Pro potřeby GUI byly zmíněné funkce rozděleny na statické a konfigurovatelné, které se dále dělí, viz schéma a následný podrobný popis.

4.1 Statické funkce

Tyto funkce jsou zpřístupněny a připraveny k použití při každé upravě obrazu v programu. Například lze zobrazovat histogram po každé úpravě výsledku a sledovat, jak jednotlivé konfigurovatelné funkce mění jeho průběh nebo nastavenou změnu umožní ihned vrátit o krok zpět nebo zobrazit vstupní obraz pro porovnání výsledků.

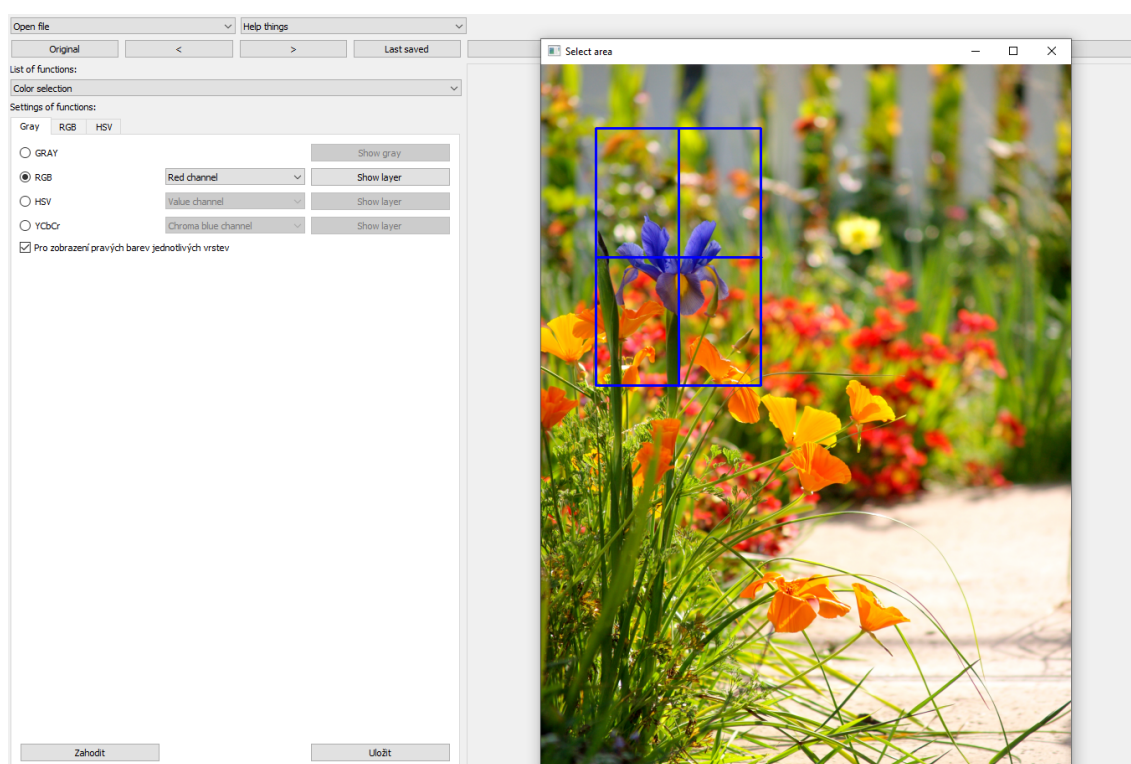
4.1.1 Historie úprav

Jedná se o nástroj, který je často používaný v nejrůznějších druzích softwaru. Hlavním úkolem této funkce je sledovat změny, které nastaly v průběhu práce a úprav, a to v časové souslednosti. Funkce historie úprav je charakterizována následovně. Poskytuje záznam změn provedených na objektu, zobrazuje změny, porovnává jednotlivé verze, obnovuje předešlé verze. Funkce toto nečiní samostatně, ale díky ní může uživatel v těchto krocích nadále pokračovat. V praktickém využití funkce historie úprav slouží k poskytnutí přehledu, ke kterým krokům při práci s obrazovým materiálem již došlo. Jeví se jako pomocník při dohledávání možných chyb a redukci nahodilostí, které s v procesu úprav mohou vyskytnout. Je umožněno jejich prohlížení a zobrazení originálního obrazu nebo poslední úpravy, jejich případné porovnání. Při zobrazení lze na jednotlivé úpravy volat další funkce.

Prohlížení Je umožněno pomocí tlačítek „Zpět“ a „Vpřed“. Další dvě tlačítka z tohoto souboru jsou pro zobrazení originálního obrazu nebo poslední provedené úpravy a jejich případné porovnání. Při zobrazení obrazu z historie úprav lze zobrazená data opět upravovat jiným způsobem.

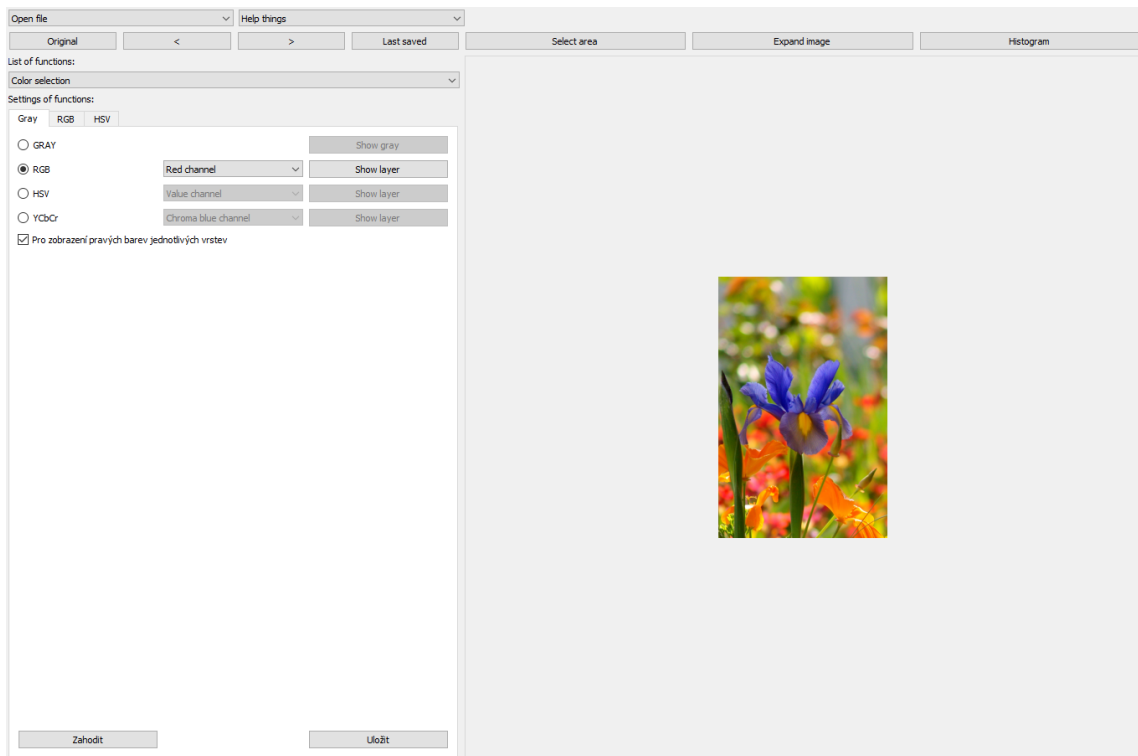
4.1.2 Výběr části a zvětšení

Výběr části je funkce, díky které je možné vybrat specifickou část obrazu s cílem s ní dále manipulovat. Tedy pokud uživatel potřebuje vybrat část obrazu, kterou chce upravit, kopírovat, přesunout nebo jinak upravovat bez toho, aniž by byl poškozen původní obraz. Při zvolení této možnosti se uživateli zobrazí nové okno, ve kterém se objeví aktuální obraz. V okně je umožněn opakovaný výběr části obrazu a následně z něho lze vystříhnout část pomocí ohraničení výběru, které probíhá pomocí tahání myši obdélníkového výběru. Finální výřez obrazu je potvrzen stisknutím klávesy „ENTER“, poté bude výstřížek převeden zpět do hlavního programu a výběrové okno se bude moci uzavřít. Po výběru je umožněno s ním nadále pracovat, zároveň se automaticky uloží do paměti. Pokud je výběr nesprávný, lze se ke stavu před výběrem vrátit pomocí funkce „Zpět“ a výběr opakovat.

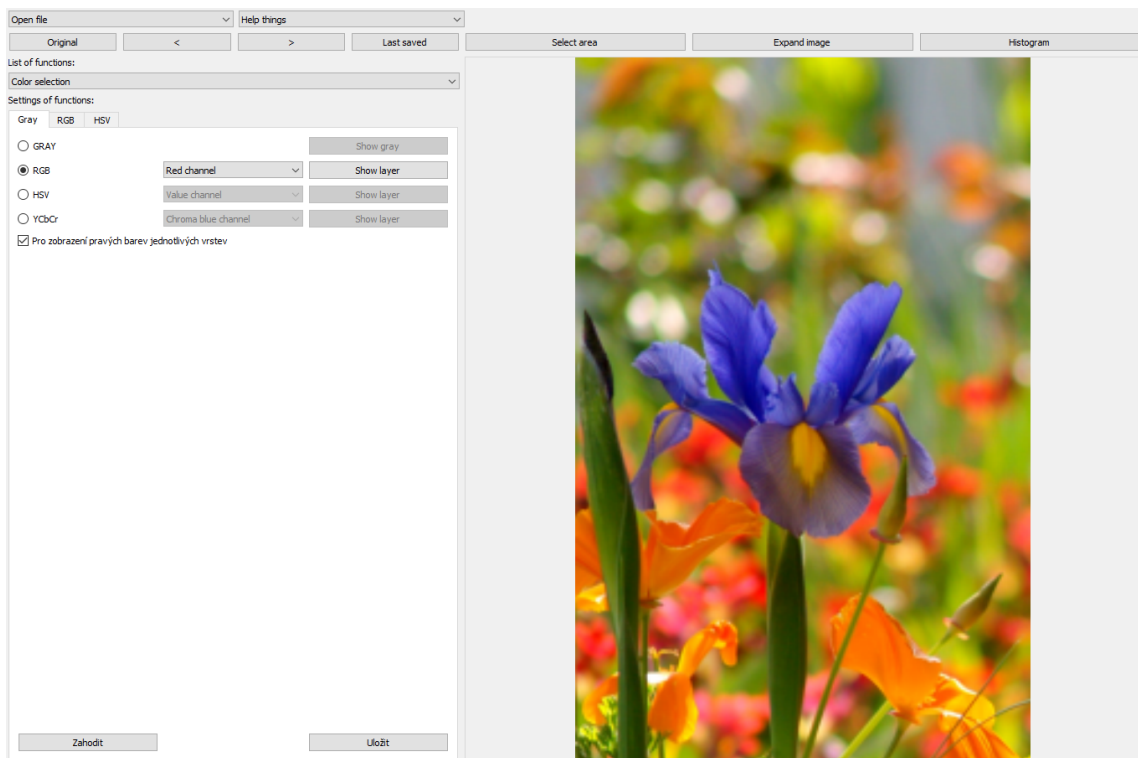


Obrázek 4.1: Výběr části obrazu

Na výstřížek je možné zavolat další funkci, kterou je zvětšení. Pomocí operace může dojít k vytvoření takového zvětšeného výběru, ve kterém je snazší pracovat na jeho detailech nebo zornosti. Ke zvětšení do maximální velikosti obrazu dochází automaticky. Maximální velikost zobrazení je ovlivněna rozlišením zařízení, na kterém proces práce probíhá. Při zvětšení je zachován poměr stran zvoleného obrazu.



Obrázek 4.2: Obraz po výběru



Obrázek 4.3: Zvětšení obrazu

4.1.3 Histogram obrazu

Funkce histogramu slouží k zobrazení jasu, tudíž dále napomáhá pro transformaci hodnot jasu. Jedná se o jasové korekce a modifikace jasové stupnice. Jasová korekce je žádoucí, pokud nedojde při snímání obrazu a jeho následné digitalizaci za optimálních podmínek. Tedy pokud zařízení nemají stejnou citlivost. Často nastává zeslabení světla, čím dále se bod nachází od optické osy. Ke ztrátám může dojít i díky nerovnoměrnému osvětlení snímaného objektu. Vady způsobují další vnější faktory jako jsou drobné částičky či prachové částice, které mohou být pouhým okem nepostřehnutelné. Pokud se nejedná o náhodné vady, pak je možné na ně reagovat prací s jasovou korekcí. Pro každý bod je možné získat hodnotu zkresleného obrazu. Korigovat jas je možné pomocí změny hodnot intervalu získaných hodnot nebo posunutím celé jasové stupnice k žádoucím hodnotám. Rozdílem je transformace jasové stupnice, která nezávisí na konkrétním bodu v obraze, nýbrž je stejná pro všechny pixely. Tato metoda se využívá s cílem poskytnout snazší pohled na vizualizovaný obraz pro oko pozorovatele. Jisté úskalí se zde objevuje, neboť při transformaci jasu dochází ke ztrátě informací, které obraz nese. [3]

Histogram je jednou ze základních metod v oblasti obrazové analýzy. *Histogram jasu je vektor H s počtem složek rovným počtu jasových úrovní.* [4] Jedná se často o diagram sloupcového typu. Tato funkce nám poskytuje poměrně obecnou informaci o zpracovávaném obrazu. Je v něm čitelná jasová stupnice či segmentace na objekty v okolí. Histogram je vyobrazením a výkladem hustoty jasu v obrazu závislých na množství pixelů. Je nutné si uvědomit, že jeden obraz má právě jeden histogram, ale tato úměra neplatí opačně, to znamená, že jeden histogram může odpovídat více obrazům. [5] K úskalím, která mohou nastat při práci s ním se řadí jeho minimální a maximální možnosti. Proto je potřeba provádět vyhlazování histogramu pomocí sousedních prvků, aby byly eliminovány možné lokální nedostatky, chyby. Transformace jasové stupnice neboli ekvalizace histogramu slouží ke zvyšování kontrastu obrazu do optimálního stavu.

Nový histogram $h_g(q_i)$, kde konstanta K uvádí velikost okolí je vypočten ze vztahu:

$$h_g(q_i) = \frac{1}{2K + 1} \sum_{j=-K}^K h_f(p_{i|j}) \quad (4.1)$$

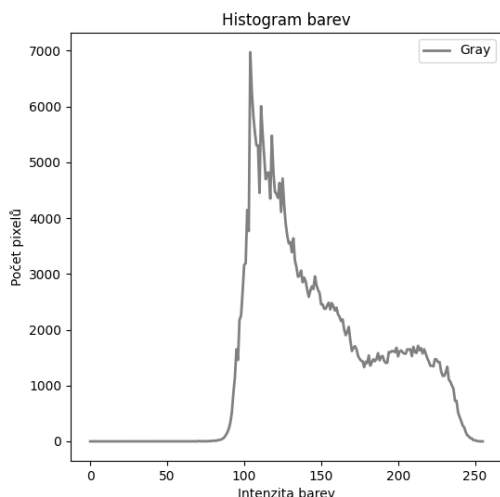
Využívá se pro vytvoření lepšího výstupního snímku a pro následnou digitalizaci. Metoda se využívá například pro obrazy, které mají nízký kontrast. Dochází k jasové transformaci za takových podmínek, aby byly jasové úrovně zastoupeny v podobné četnosti. [5]



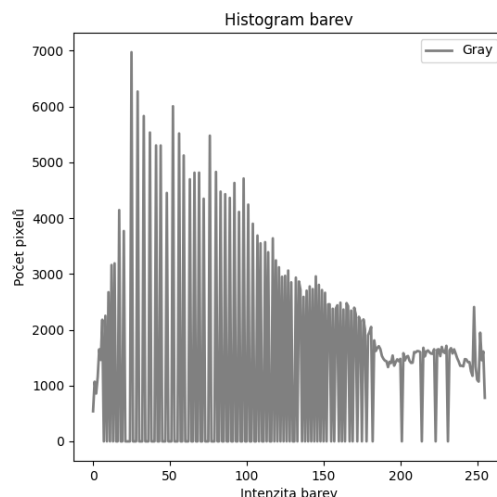
Obrázek 4.4: Originální obraz



Obrázek 4.5: Equalizovaný obraz



Obrázek 4.6: Původní histogram



Obrázek 4.7: Equalizovaný histogram

4.2 Konfigurovatelné funkce

Funkce, které jsou zařazené do tohoto souboru umožňují uživateli přesnější nastavení jednotlivých vstupních dat do funkcí, které manipulují s obrazem. Například velikost matice nebo její vzor a počet provedení eroze. Na rozdíl od statických funkcí, které se ukládají ihned po jejich zavolání, zde je potřeba výslednou manipulaci s obrazem uložit pomocí tlačítka „Uložit“, které se nachází na každé kartě s funkcí.

4.2.1 Barevné prostory

Obraz může být monochromatický, reprezentován jedinou obrazovou funkcí nebo barevný, multispektrální, kde každé vrstvě odpovídá jiný z barevných kanálů. Tyto kanály jsou definované množiny barev. Barevný model je označení souboru z pravidla tří nebo čtyř barevných kanálů. Barevný prostor dává způsob, jakým lze barvy reprezentovat v počítači pomocí čísel. Mezi jednotlivými prostory lze libovolně přecházet, avšak každý barevný prostor má různou množinu barev, které v tomto

barevném prostoru můžeme zobrazit. Tato množina se nazývá gamut, pokud se při transformaci z jednoho barevného prostoru do druhého ocitne barva mimo gamut cílového prostoru, je tato informace ztracena a barva je zobrazena pouze přibližně. Změna barevného prostoru může být užitečná například při zobrazování na různých displejích nebo při tisku daného obrazu. Ve výsledném programu lze manipulovat se čtyřmi barevnými prostory, modely RGB, HSV, YCbCr a Šedotónové zobrazení neboli Gray. [7]

RGB

Barevný model RGB se skládá ze tří samostatných složek barvy světla. Jedná se o barvy červená, red o vlnové délce 700nm, zelená, green 546,1nm a modrá, blue 435,8nm. V reálném použití se používá velice podobný model RGBA, kde složka A zastupuje kanál Alpha, který je brán jako průhlednost. Smícháním těchto tří, v případě RGBA čtyř, složek vzniká barevný obraz. Každá barva je tvořena jedním vektorem sytosti od 0 do 255. Hodnota 0 znamená, že barva není zastoupena. Naopak hodnota 255 znamená, že je zastoupena plně sytosti. Obraz RGB je reprezentován třírozměrnou maticí kde x a y je velikost obrazu v pixelech a třetí osa z reprezentuje počet barevných vrstev. (obr. 4.8)

Pokud barva není zastoupena v žádné vrstvě, výsledkem je barva černá, v opačném případě, kdy je barva zastoupena v plném rozsahu, vzniká barva bílá. Barevný model RGB se používá v technologii monitorů, při tvorbě a úpravě fotografií nebo v počítačové grafice tedy je vhodnější v digitální grafice.

Výslednou barvu v programu lze definovat pomocí tří posuvníků. Hodnoty zastoupení jednotlivých složek se zapisují přímo do příslušného pole, které je omezeno rozsahem zastoupení (0-255) nebo barvu definovat pomocí hexadecimální soustavy, kdy každou barvu reprezentují hodnoty od 00 do FF. U barev, které mají v hodnotách jednotlivých intenzit stejný dvojnásobek, je možné zápis zkrátit uvedením pouze jednoho znaku - #FFFFFF > #FFF. Podle druhu zadávání jsou dynamicky přepisovány ostatní druhy zadávání, aby nedocházelo k chybám. Ke každé barvě lze zavolat toleranci, v rozsahu od 0 - 100%, to znamená, že se z obrazu vyfiltruje zadaná barva a barvy sousedící s touto barvou.

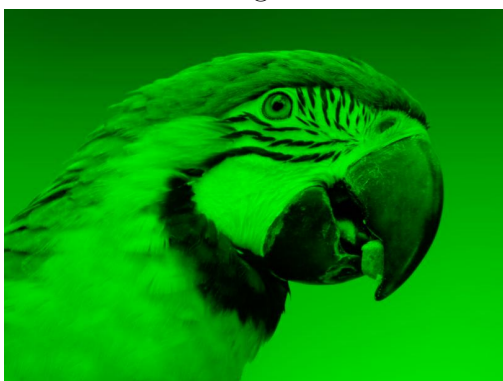
Vyfiltrovaných barev se dá uložit více a následně je v rámci jedné manipulace v kartě RGB zpětně zobrazovat, případně odstranit z výběru. Pro uložení vybraných barev do paměti pro další manipulaci je potřeba vybrané položky sloučit kliknutím na tlačítko „Sloučit barvy“. Ve výsledném obrazu jsou zastoupeny pouze vyfiltrované barvy zadané uživatelem. (obr. 4.12)



Obrázek 4.8: Originální zobrazení



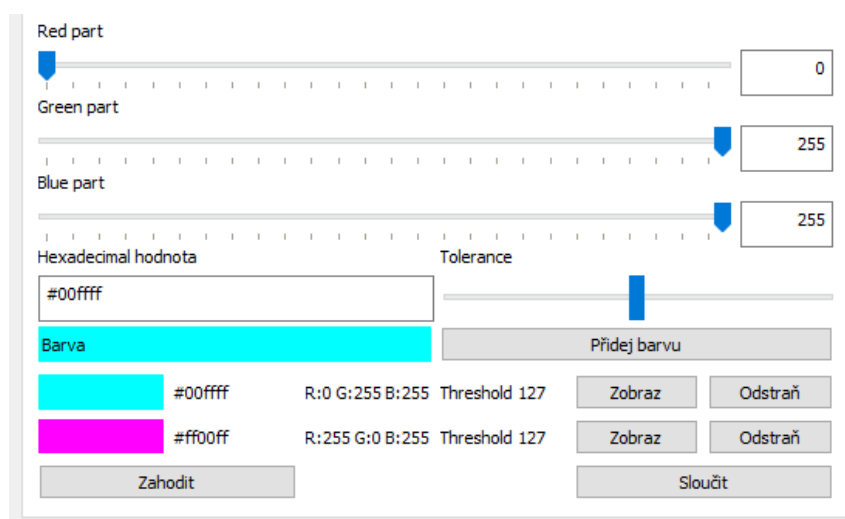
Obrázek 4.9: Červený komponent



Obrázek 4.10: Zelený komponent



Obrázek 4.11: Modrý komponent



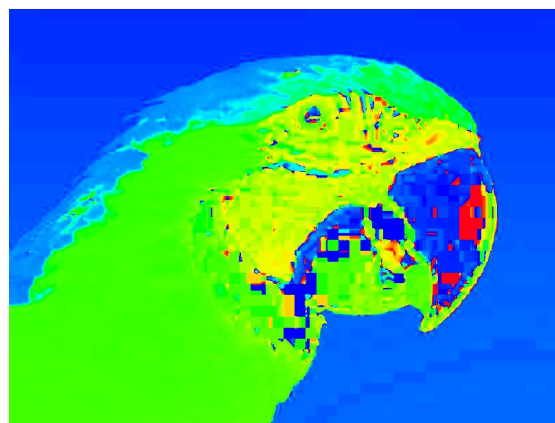
Obrázek 4.12: Vybírání jednotlivých barev

HSV

Barevný prostor HSV popisuje barvy způsobem, který je nejbližší lidskému vnímání barev. V definování jednotlivých barev se výrazně podobá modelu RGB. Jeho složky tvoří odstín/barevná složka (hue - H), sytost (saturation - S) a intenzita jasu (value - V). Model může být nazýván jako HSB, modely se liší jen v názvu poslední složky brightness - B. Odstín se typicky vyjadřuje ve stupních ($0^\circ - 360^\circ$) nebo je hodnota normovaná. Ostatní složky jsou potom popisovány procentuálně od 0 do 100%, kdy nulové zastoupení u saturace znamená šedou naopak stoprocentní zastoupení barvu v plné sytosti. Podobně pro value to je relativní světlost nebo tmavost neboli hodnota jasu, množství bílého světla.



Obrázek 4.13: Originální zobrazení



Obrázek 4.14: Zobrazení barev



Obrázek 4.15: Zobrazení sytosti



Obrázek 4.16: Zobrazení intenzity jasu

Podobně jako pro výběr v modelu RGB, jsou i zde k dispozici tři posuvníky pro nastavení konkrétní složky. Změna však nastává v limitech, kde první zastává barevnou složku, druhý saturaci a třetí jasovou hodnotu. Jsou tedy omezeny na maximální hodnoty zastoupení jednotlivých složek. Pro odstín barvy od 0 do 360 a pro ostatní dvě hodnoty od 0 do 100. Tyto hodnoty jsou při manipulaci s posuvníkem vypisovány jako číselná hodnota v příslušném poli. Naopak při vložení hodnoty do pole je automaticky nastaven i příslušný posuvník. S každou manipulací se barva

ihned zobrazuje i s aktuálním hexadecimálním zápisem. Funkce umožňuje zavolat toleranci k dané barvě a tím k vyfiltrovanému obrazu přidat i sousedící barvy. Obraz s vyfiltrovanou barvou lze opět uložit do seznamu barev, zobrazit ji nebo odstranit z výběru. Pro další manipulaci je potřeba vybrané barevné složky sloučit, pomocí tlačítka v pravém dolním rohu karty.

YCbCr

YCbCr, známý také jako Y'CbCr, barevný model využívaný v digitálním světě, zejména v oblasti videa a fotografie. Byl vytvořen lineární transformací složek RGB.

$$\begin{bmatrix} Y \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4.2)$$

Jas a chromatická složka jsou odděleny pro efektivnější kompresi obrazu. Složka Y byla vytvořena z původních složek RGB a reprezentuje jas obrazu. Složky Cb a Cr byly vytvořeny z rozdílu mezi modrou nebo červenou a luminancí. Díky oddělení jasu od chromatické složky mohla být obrazová data komprimována s lepší účinností, neboť jas se obvykle mění méně než samotné barvy. Jedná se standardní model pro televizní vysílání a je kompatibilní se současnými systémy. Převod YCbCr zpět na RGB není obtížný, což umožňuje zobrazení obrazu na našich televizích a monitorech.

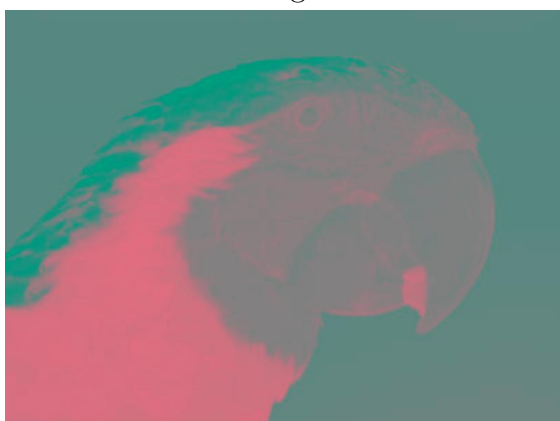
Nevýhodou tohoto modelu je menší intuitivnost pro člověka oproti RGB. V některých případech, jako je počítačová grafika, zůstává preferován model RGB. YCbCr hraje důležitou roli v televizním vysílání, digitálním videu, fotografiích, kompresi videa a zpracování obrazu. Zadávání a následná manipulace s jednotlivými vrstvami tohoto modelu je totožná jako při zvolení výběru RGB nebo HSV. Tři posuvníky, omezené maximálním rozsahem dané vrstvy.



Obrázek 4.17: Originální zobrazení



Obrázek 4.18: Jasová složka



Obrázek 4.19: Chroma red



Obrázek 4.20: Chroma blue

Gray

Šedotónové zobrazení, synonymum pro stupně šedi neboli gray, představuje základní metodu popisu obrazové informace bez využití barevného spektra. Nachází uplatnění v černobílé fotografii, tisku a digitálních obrazech. Princip spočívá v lineární škále hodnot od černé tedy absence světla až po bílou, maximum světla. Mezi těmito krajními body se nachází paleta šedých odstínů reprezentující různé stupně intenzity světla odraženého od jednotlivých pixelů v obraze. V digitálním prostředí se tyto hodnoty obvykle ukládají s využitím 8 bitů na pixel, což umožňuje rozlišit 256 odstínů šedi (0 pro černou, 255 pro bílou). Zvýšení počtu bitů na pixel vede k jemnější gradaci odstínů a lepšímu zobrazení detailů.

Rovnice převodu RGB na stupně šedi:

$$GRAY = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B. \quad (4.3)$$

Pro tento prostor neexistuje zpětná transformace. Po převodu RGB na stupně šedi ho nelze zpět rekonstruovat na RGB. Zobrazení může být uplatněno pro zjednodušení procesu analýzy obrazu. Přístup se využívá například při detekci hran objektů nebo rozpoznávání tvarů a při editaci fotografií, kdy je manipulováno s jed-

notlivými barevnými složkami. Každý barevný kanál v modelu RGB je samostatně převeden do černobílého.

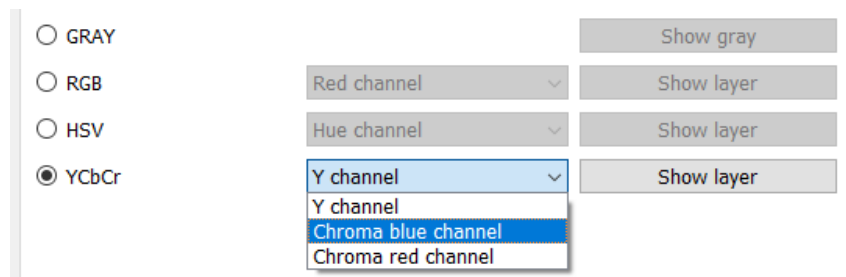


Obrázek 4.21: Originální zobrazení



Obrázek 4.22: Šedotónové zobrazení

Při výběru možnosti v souboru funkcí „Barevné prostory“ je uživateli zobrazena jiná nabídka, než u předchozích tří funkcí. Uživatel si může vybrat kteroukoliv vrstvu z výše zmíněných barevných modelů, tu nejdříve zobrazit a následně převést do šedotónového zobrazení. Pro další manipulaci s výsledným obrazem je potřeba všechny změny uložit.



Obrázek 4.23: Ovládání výběru šedotónového spektra

4.2.2 Geometrická transformace

Při snímání obrazu často dochází ke geometrickému zkreslení. Nejedná se o ojedinělý jev, neboť může nastat kdykoliv není-li snímáný objekt pod pravým úhlem nebo je objekt jiný než plošný. Geometrická transformace představuje obrácený proces ke geometrickému zkreslení. [4]

Geometrická transformace a korekce se často využívá při procesu předzpracování obrazu v prostoru. Je potřeba brát v úvahu souřadnice jako nosiče funkce. Tato funkce se využívá ve 2D obrazu aplikaci zvětšování, posouvání, rotace, zkosení či při odstraňování geometrických zkreslení. [3] Jedná se o vektorovou funkci, která mění bod v rovině do jiného bodu. V případě rotace, posunu nebo zvětšení je vztah transformovaných předem znám. Při geometrické transformaci dochází ke dvěma

krokům. Plošná transformace nejprve dohledá ke vstupnímu bodu bod výstupní. Dále dochází k nalezení úrovně jasu většinou pomocí metody nejbližšího souseda. Ta přiřadí transformovanému bodu hodnoty nejbližšího vzorku.

Rotace obrazu

Geometrická transformace rotace obrazu umožňuje otočení obrazu o určitý úhel ve směru nebo proti směru hodinových ručiček. Použití a aplikace funkce je následující. Uživatel vloží obraz, který chce otáčet, zároveň zadá úhel, o kolik chce, aby se obraz otočil. Zadání je zpravidla číselné či pomocí rozhraní programu. Dále zvolí bod, kolem kterého má rotace probíhat. Po zadání těchto proměnných dojde k vypočtení transformační matice, která určí samotnou transformaci obrazu, tedy přiřadí každému bodu v obrazu hodnoty, které má získat po rotaci. Proběhne samotný proces geometrické transformace pomocí rotace, dojde k zachování obrazu a snížení zkreslení. Výstupní obraz je poskytnut uživateli, který ho může uložit a nadále ho využívat.

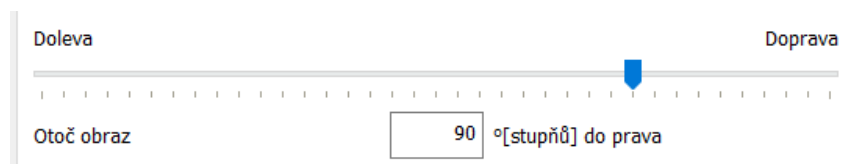
Příklad rotace kolem počátečního bodu o úhel α_r proti směru hodinových ručiček pomocí rovnic

$$\begin{aligned}x' &= x \cos \alpha_r - y \sin \alpha_r, \\y' &= x \sin \alpha_r + y \cos \alpha_r\end{aligned}\tag{4.4}$$

nebo pomocí rotační matice

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha_r & -\sin \alpha_r & 0 \\ \sin \alpha_r & \cos \alpha_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.\tag{4.5}$$

Funkce rotace je k nalezení v listu funkcí pod položkou geometrické operace. Při práci s funkcí program umožňuje otáčet obraz až o 180° doleva i doprava. Z toho vyplývá, že je umožněno otočit obrazem až o 360° . Rotace probíhá kolem středového bodu obrazu. Praktická manipulace se zprostředkovává pomocí pohybu, posunu na škále s krajními body -180° a 180° . Velikost rotace lze zadat i číselným záznamem na stejné škále. V průběhu rotace dochází ke změně velikosti obrazu v rozsahu výšece prostoru. Původní velikost prostoru se nemění, ale samotný používaný obraz svou velikost přizpůsobuje k podmínkám rotace. Z pravidla krajní body obrazu bývají skryty. Po provedení funkce rotace je potřeba manipulaci uložit do paměti pro volání dalších, jiných funkcí.



Obrázek 4.24: Nastavení otočení obrazu

Posunutí obrazu

Při použití funkce posunutí jako další možnosti geometrické transformace je proces práce funkce podobný. Po vložení obrazu uživatel zvolí směr posunu. Vybírá z možnosti horizontálního či vertikálního posunutí. Dále určuje o kolik pixelů, bodů či procent se má obraz posunout. Zadává číselně nebo pomocí rozhraní programu. Maximální hodnota posunutí v těchto osách se vypočítá pomocí dvojnásobné velikosti obrazu. Umožňuje tedy posunout obrázek mimo zorné pole v plném jeho rozsahu, a to v obou osách.

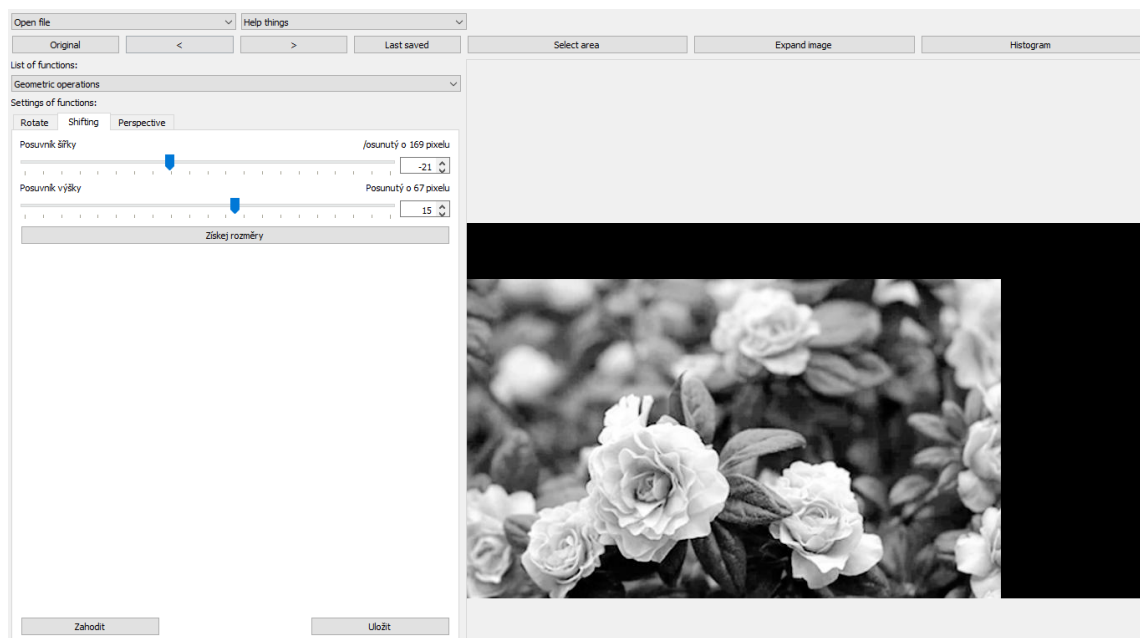
Posun lze pak vyjádřit jednoduchou soustavou rovnic

$$\begin{aligned}x' &= a_0 + x, \\y' &= b_0 + y\end{aligned}\tag{4.6}$$

nebo maticí posunu, kde a_0 je posunutí v ose x a b_0 v ose y ,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a_0 \\ 0 & 1 & b_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.\tag{4.7}$$

Funkce stanoví matici pro posun každého bodu. Při aplikaci je každý bod posunut na nové souřadnice pomocí vypočtené matice. Hodnoty jsou vyrovnány tak, aby nedocházelo ke zkreslení a ztrátě informací. Nakonec je nový obraz promítnut uživateli, který ho může uložit a dále s ním pracovat. Nevyužité zorné pole se stává součástí nově uloženého obrazu.



Obrázek 4.25: Ukázka posunutí obrazu

Změna perspektivy

Objekty, na které cílí snímání a později jsou převáděny do obrazu jsou součástí trojrozměrného světa, proto aby mohly být převáděny do dvourozměrného obrazu se využívá geometrická transformace v podobě perspektivního zobrazení. Reálný trojrozměrný předmět je stlačen do dvou rozměrů. Pro vypočtení transformace se uvažují souřadnice tří bodů a ohnisková vzdálenost. Často se využívá pravoúhlé zobrazení, díky kterému dochází ke zjednodušení. Při tomto procesu dochází k vysoké ztrátě informací. Rekonstrukce původního zobrazení není bez získání dalších informací zpětně možná. Usnadnění se objevuje při pořizování materiálů, které zaznamenávají objekty dvourozměrného charakteru. [4]

Ve funkci změna perspektivy se otevírá, již více možností, jak manipulovat s obrazem v rámci této kategorie. Jedná se o kombinaci oříznutí obrazu, jeho zvětšení a rotace. Díky prvkům je možné získat lepší obrazová data. Změna perspektivy probíhá vytyčením hledaného objektu pomocí čtyř přímk, jejichž případ použití je zobrazen na příkladovém obrázku pod posuvníky, které jsou určeny k nastavení jednotlivých přímk.



Obrázek 4.26: Proces vytyčení hledaného obrazu

Podle tohoto vzoru je následně hledaný, vytyčený objekt transformován do výsledného tvaru. Zjištění vrcholů výsledného útvaru je provedeno díky výpočtu průsečíků jednotlivých přímk. Velikost tohoto útvaru je vypočítána pomocí Pythagorovy věty, tedy jednotlivé délky stran jsou zjištěny díky její aplikaci.



Obrázek 4.27: Znázornění výpočtu stran

4.2.3 Filtrace obrazu

Funkce filtrace obrazu jsou mimo jiné využívány v oblastech, které pracují s různými typy fotografií. Cílem práce často je odstranění šumů a vylepšení obrazů, které poslouží k lepšímu vnímání lidského oka. Kromě potlačení šumů se filtrace často používá k získání potřebných informací. Mimo zmíněného potlačení či odstranění šumů lze filtraci používat se záměrem vyhlazování, zvýrazňování či objevení rozhraní. Je nutné si uvědomit, že šumy se nachází v každém obrazovém materiálu, nicméně ne vždy je nezbytně nutné či žádoucí jejich odstranění. Rozhodnutí závisí na vlivu využití dalších procesů práce. Pokud je filtrace použita může docházet k negativním výsledkům v dalších krocích. Tento krok by měl pramenit z hodnocení samotného procesu, ve kterém bude brána v úvahu i skutečnost, že díky filtraci dojde ke ztrátě některých informací. Každý pixel každého objektu obsahuje náhodný šum. Ten je pro funkci obrazu nezávislý. Společně s průměrem bodu a šumu vzniká veličina, ve které se patrně odchylka zmenšuje. Tento opakovaný proces napomáhá k vyhlazení obrazu za absence rozmazání. Opakovaným průměrováním postupně dochází k vyhlazování a získání odpovídajících hodnot jasové funkce. Příkladem využití může být potlačování tepelného šumu kamery, kdy odhadovaný počet obrazů pro získání správné hodnoty se dostává až k 50 obrazům. [5]

Konvoluční filtr

Metoda lokální filtrace pro předzpracování obrazu využívá princip okolí reprezentativního bodu, který slouží k vyhledání hodnoty daného pixelu. K hodnotě reprezentativního pixelu dochází výpočtem a zkoumáním malého okolí, často obdélníku daného bodu, výsledek je pak zapsán do výstupního obrazu. Mezi lokální předzpracování obrazu se řadí vyhlazování a detekce hran. [3]

Průměrovací filtr

Zvolit lokální transformace je výhodné v závislosti velikosti objektu v daném obraze. Pokud je objekt větší, využívá se ke zlepšení vyhlazení nespojitostí. Je-li tomu naopak filtrují se velké nespojitosti v jasových charakterech. [4] Vyhlazování a detekce hran jsou však protikladné, proto se doporučuje používat nelineární metody, u kterých nedochází k takové ztrátě informací při vzájemné spolupráci. To nás zavádí k druhému třídění metod filtrace a jimi jsou lineární a nelineární metody, které využívají matematické vlastnosti transformací. V jejich případě se často využívá obdélníkové okolí, které obsahuje lichý počet řádků a sloupců proto, aby reprezentativní bod mohl ležet uprostřed. Vzhledem ke skutečnosti, že filtrace pracuje s malým prostorem v obraze, je nutné si uvědomit, že nezná další informace o něm samotném. Je výhodné je zadat do proměnných v algoritmech, pokud nám jsou známé. V průběhu procesu předzpracování obrazu nedochází k objevení či získání žádné nové informace. Informace mohou být pouze potlačeny nebo zvýrazněny. Při lokálním vyhlazování obrazu je výhodné, pokud disponujeme více stejnými obrazy, které se

liší právě šumem. V tom případě vyhlazování probíhá pomocí průměrování hodnot pixelu z více stejných obrazů m .

$$f(i, j) = \frac{1}{m} \sum_{k=1}^m g_k(i, j) \quad (4.8)$$

Pokud je poskytnut pouze jeden vstupní obraz g s váhovými koeficienty h , dochází k využití sousedních pixelů pro odstranění šumů reprezentativního pixelu. V okolí O se analyzují hodnoty jasu, tedy na okolích pixelech.

$$f(i, j) = \sum_{(m,n) \in O} h(i-m, j-n)g(m, n) \quad (4.9)$$

Může dojít k jejich průměrování, a tak vytvoření hodnoty vybraného, upraveného pixelu v bodě (i, j) . [3]

Filtry lineární jsou značně používanou skupinou filtrů. Jedná se o prostorově invariantní či homogenní filtry. Jejich chování se nemění v závislosti na změnu polohy obrazu. Reálně se však tato teorie u mnoha obrazů neobjevuje, neboť předpoklad je oslabený jasem v obraze. Konvoluční maska, s rozměry 3×3 , pak určuje výběr a průměrování okolí, pro získání hodnot.

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.10)$$

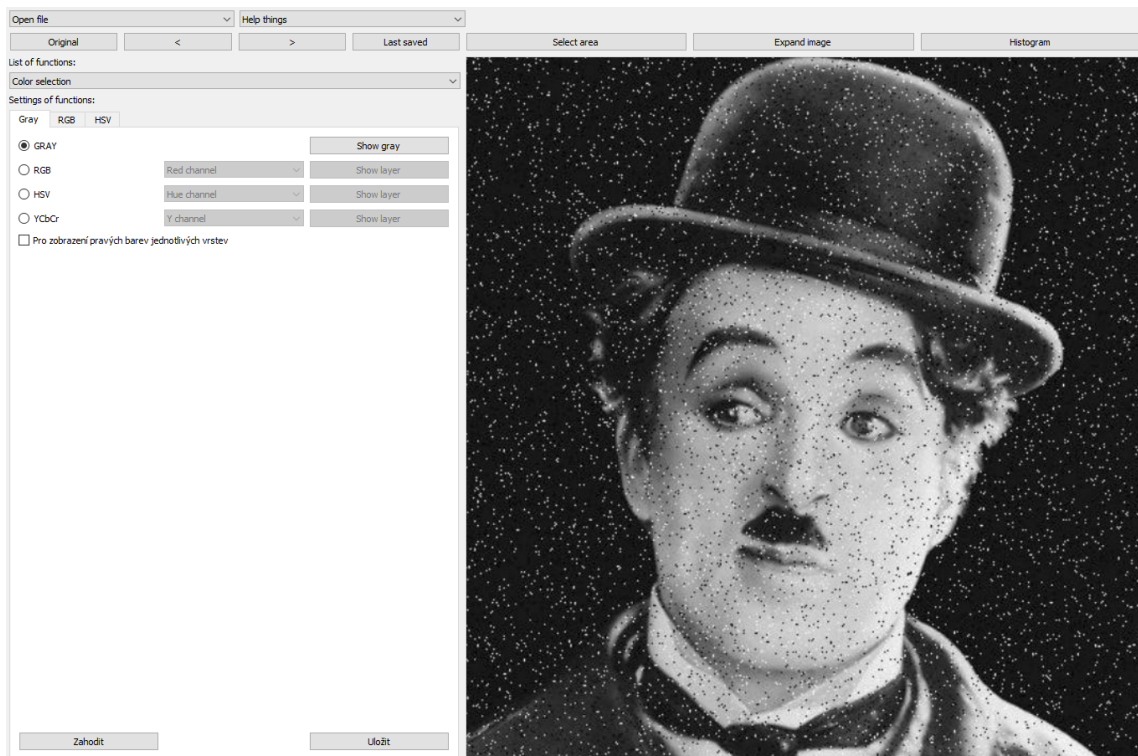
Možné problémy s rozmazáním mohou být potlačeny, pokud se pro výpočet hodnot nepoužívá celé okolí reprezentativního bodu, ale pouze ty pixely, které mají úroveň jasu takové, do kterých daný bod náleží. Při procesu uplatnění výše zmíněného je využíváno průměrování s omezením změn jasu. Je nalezen pixel s vyšší hodnotou zadaného maximálního rozdílu jasu, než byla zadána, pak se nevyužívá a hodnota není změněna. Dále je možné aplikovat metodu rotující masky. V tomto případě je kolem reprezentativního bodu vytvořena maska, která mění svou polohu a v každé fázi je vypočítán rozptýl jasu, poté je zjištěn aritmetický průměr bodů.

Gaussův filtr

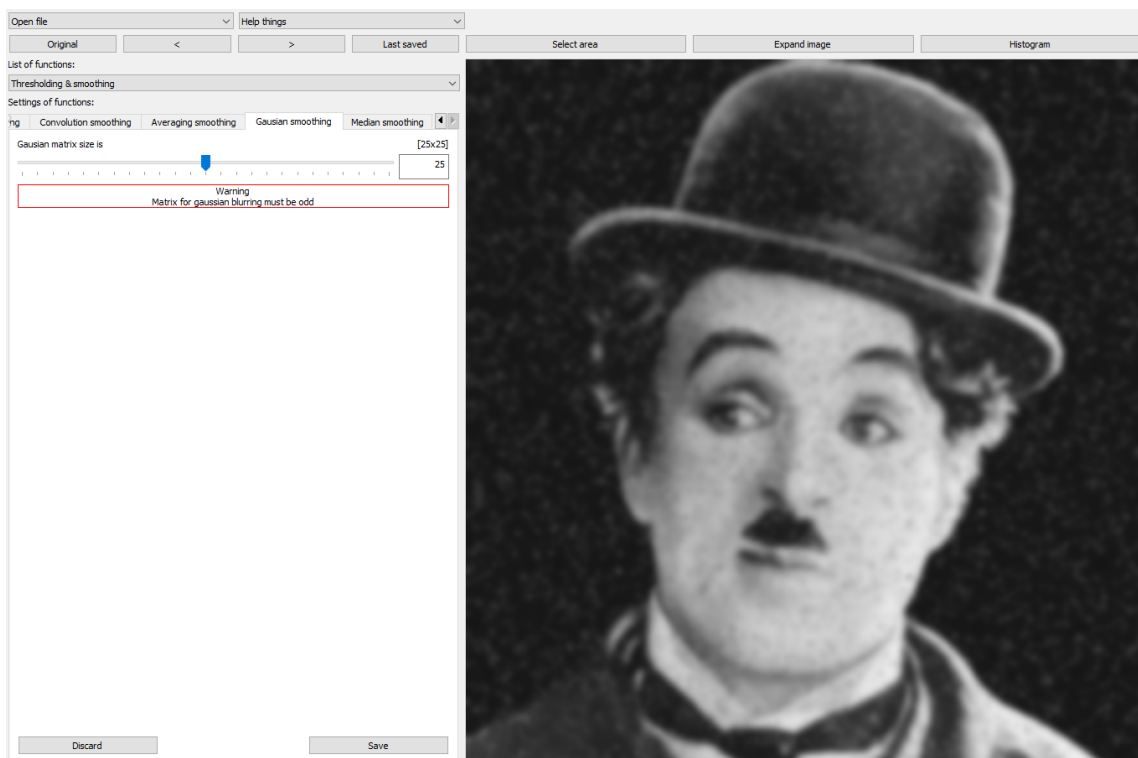
Gaussovy filtry, podobně jako průměrovací filtry, rozmazávají hrany v obraze. Rozmazání je však méně patrné, jelikož středový pixel má v Gaussově masce větší váhu než okolní pixely. Výsledný efekt je ekvivalentní váženému průměru pixelů v okolí.

Rovnice ukazuje příklad Gaussova konvolučního jádra o rozměrech 3×3 . V jádru můžeme pozorovat, že středový pixel má hodnotu 4, zatímco okolní pixely mají hodnoty 2. Toto rozdělení vah dává středovému pixelu dvakrát větší váhu než okolním pixelům, čímž se redukuje efekt rozmazání ve srovnání s průměrovacím filtrem. [2]

$$h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (4.11)$$



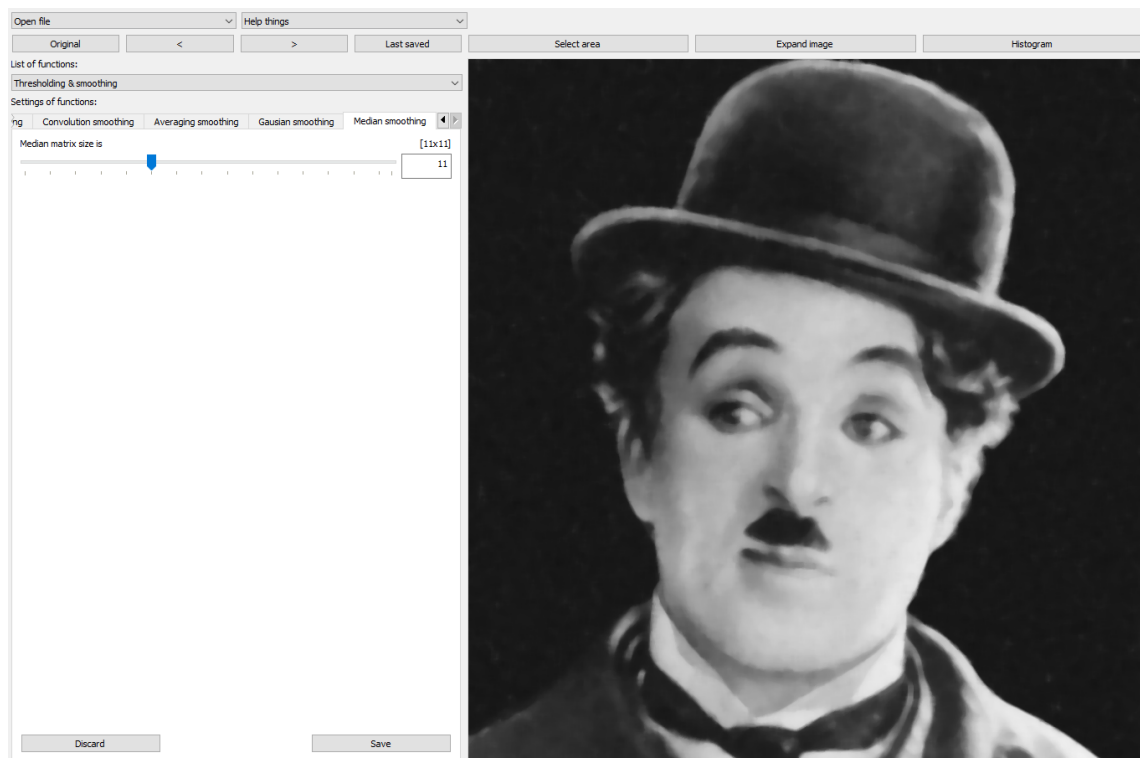
Obrázek 4.28: Originální obraz s přidaným šumem



Obrázek 4.29: Obraz po aplikaci Gaussova filtru s velikostí matice 25×25

Mediánový filtr

Vedle filtrů lineárních se vyskytují i filtry nelineární. Například pro potlačení šumu v podobě bílých a černých teček se využívá mediánový filtr. [5] Medián je veličina, která určuje náhodné proměnné. Pro výpočet mediánu jsou uspořádány hodnoty jasu v okolí bodu a to vzestupně, medián je pak prvek, který se nachází uprostřed této řady. Metoda stanový jas reprezentativního bodu jako vypočítaný medián hodnot jasu jednotlivých bodů a v okolí. Metoda je dobře využitelná pro potlačení impulzivních šumů a nižšího rozmazání hran. Jeho nevýhoda, však spočívá v porušování tenkých čar a rohů v obraze, k tomu dochází, pokud je využit obdélníkové okolí. [4]



Obrázek 4.30: Obraz po aplikaci mediánového filtru s velikostí matice 11×11

4.3 Funkce segmentace obrazu

Při práci s filtrací je dále podstatný pojem hrany. Hrana pro oko člověka znamená místo, kde se nenadále mění hodnota jasu. Jednotlivé pixely s odlišnou hodnotou jasu se v tomto případě označují jako hrany. Taková místa je schopná najít metoda lokálního předzpracování. Hrana je určována svým okolím. Jedná se o bod v objektu, kde se náhle mění hodnota funkce $f(x, y)$ v závislosti na ostatních bodech. Matematicky je možné tuto problematiku analyzovat pomocí parciální derivace. Je zde uvažován *gradient*, jako vektorová veličina ∇ , která udává směr a strmost růstu funkce. Pixely, které charakterizuje velká míra gradientu, jsou zmíněné hrany.

Pro spojitou funkci $f(x, y)$ jsou velikost gradientu $|\nabla f(x, y)|$ a směr gradientu ψ dány vztahy

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}, \psi = \arg\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right). \quad (4.12)$$

Teoretický postup je uplatnitelný při hledání hranic objektu. Může docházet k tomu, že pixely, hrany se spojují do hranic. Definujeme nejběžnější hrany. Mezi ně patří idealizovaná skoková, střešková a hrana v podobě linie, ve skutečném obraze je možné najít hranu zašuměnou. Model, který pracuje s gradientem se uplatňuje pro ostření obrazu. V tomto případě dochází k předzpracování obrazu tak, aby byly jednotlivé hrany strmější. [3] Při rozboru obrazu je klíčové definovat jednotlivé části obrazu, hovoříme o segmentaci. Rozlišuje segmentaci kompletní, při které jednotlivé segmenty jsou v souladu s objekty v původním obraze. Druhý druh segmentace je segmentace částečná, kde jednotlivé segmenty nemusí korespondovat s jednotlivými objekty. Segmentace se jeví jako nápomocná při rozložení dat pro jejich oddělené zpracovávání. Proces často stěžuje nejednoznačnost dat a obrazové šумы. Metody využívají různé proměnné. Jednou z nich jsou globální znalosti o obraze. Ty jsou často interpretovány pomocí histogramu. Dále může probíhat pomocí určování hranic mezi jednotlivými oblastmi, a nakonec může docházet k vytváření daných oblastí.

4.3.1 Prahování

Základní metodou využívanou pro segmentaci je tzn. prahování. Základní princip použití vychází ze skutečnosti, že každý objekt obrazu je charakterizován svým povrchem. V tom případě je možné využívat jasové konstanty, tedy prahu T k oddělení objektů od jejich pozadí. „Prahování je transformace vstupního obrazu f na výstupní (segmentovaný) binární obraz g podle vztahu.“ [4]

$$g(i, j) = \begin{cases} 1 & \text{prof}(i, j) \geq T \\ 0 & \text{prof}(i, j) < T \end{cases} \quad (4.13)$$

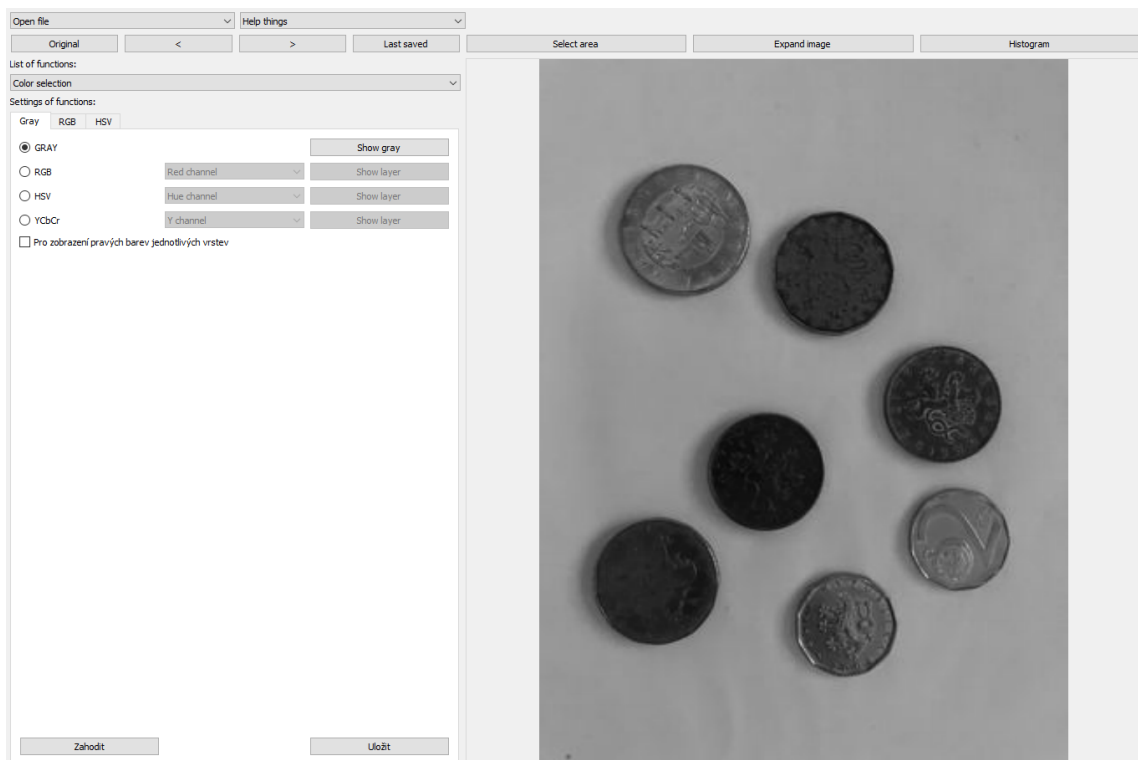
Pro úspěšný proces je klíčové zvolit správný práh, který však většinou není využitelný pro celou plochu obrazu, neboť se v něm často objevují odchylky, či nerovnoměrnosti osvětlení. Oproti tomu lokální práh je uplatňován při těchto patrných nerovnostech. Obraz je rozložen na jednotlivé podobrazy, který se liší svou polohou, a tedy i charakteristickými vlastnostmi. Prah je třeba určit v každém podobrazu nezávisle na ostatních.

$$g(i, j) = \begin{cases} 0 & \text{prof}(i, j) \in D \\ 1 & \text{jinak} \end{cases} \quad (4.14)$$

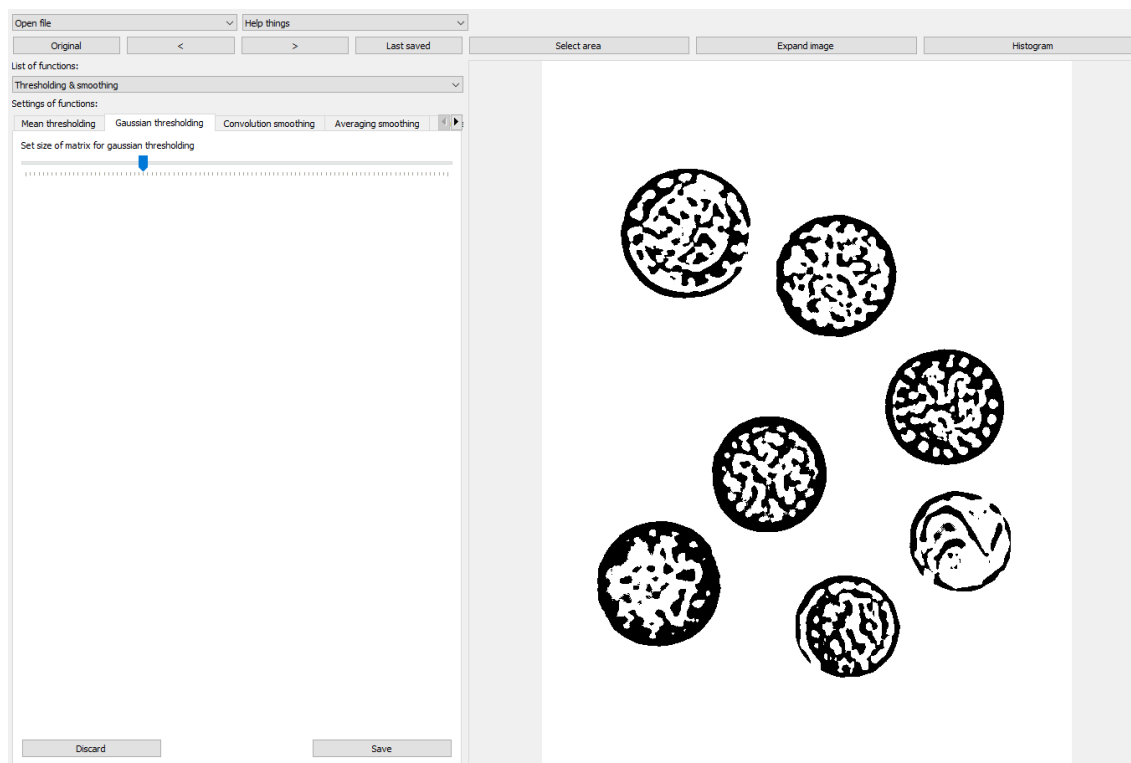
Další možností je prahování s více prahy n . Výsledkem této funkce je obraz s omezeným počtem jasových úrovní.

$$g(i, j) = \begin{matrix} 1 & prof(i, j) \in D_1 \\ 2 & prof(i, j) \in D_2 \\ \vdots & \vdots \\ n & prof(i, j) \in D_n \\ 0 & jinak \end{matrix} \quad (4.15)$$

Výše zmíněná prahování využívají jasové vlastnosti obrazů. Aplikovatelné jsou i v dalších charakteristikách jako je hodnota gradientu, texturní vlastnosti nebo další kritéria při možném rozložení obrazu. Mezi metody určování prahu se řadí procentní prahování, kde je potřeba znát vlastnost, kterou má obraz po prahování mít, či metoda, která pracuje s výkladem histogramu. Z toho může vyplývat skutečnost, že některé hodnoty jsou v obraze zastoupeny více a jiné méně, či vůbec, tak mohou být stanoveny hranice mezi objekty a pozadím. Přesnost tohoto postupu nemusí být vždy zaručena nebo někdy není možné jasně určit vrcholy histogramu, proto je nutné srovnávat a vyhodnocovat správnost segmentace prahováním.



Obrázek 4.31: Originální obraz, převedený do šedotónového spektra



Obrázek 4.32: Použití Gaussova prahování

4.3.2 Srovnávání se vzorem

Dalším metodou segmentace je segmentace pomocí nalezení hran. Jak již bylo zmíněno výše i zde se objevuje předpoklad, že jednotlivé objekty obrazu se skládají z hran a ty jsou spojovány do hranic. Úskalí se jeví v přítomnosti šumů, které mohou způsobit, že hrany nevznikají tam, kde by měly být nebo naopak se objeví tam, kde hranice objektu není. Pokud hrany nejsou patrné je nejprve potřeba sledovat, tedy postupně krok po kroku procházet obrazem a stanovovat oblasti. Vyhledávání hranice je možné například pomocí Houghovy transformace. Segmentace narůstání okolí se nejčastěji aplikuje v případech obrazů se šumem. Základní myšlenkou této metody je rozdělení oblastí obrazu do takových oblastí, které budou mít co největší vlastní míru homogenity, shodnosti v různých kritériích. Metoda nadále využívá spojování oblastí, ve které od stanovené oblasti dochází ke spojování za pomoci datové struktury tedy supermřížky. Spojování nastává, dokud není porušena zadaná platnost podmínek. Opačným procesem je štěpení oblastí, kdy jedna výchozí oblast je štěpena na dílčí, ve kterých neplatí stanovené podmínky. Tyto dva procesy pak mohou být pro větší efektivitu spojovány. Tedy štěpení a spojování. Segmentace srovnávání se vzorem předpokládá existenci vzoru h , který obsahuje známé objekty v obraze f , ty jsou pomocí srovnávání nacházeny. Z určitého obrazu je vyčleněn objekt, pomocí kterého se pak v dalších obrazech vyhledávají stejné nebo podobné objekty. Vzhledem k šumům či geometrickým transformacím obrazů je nepravděpodobné, že hledaný objekt bude naprosto totožný se vzorem, proto dochází ke hledání

maximálního dosažený zadaného kritéria.

$$C_1(u, v) = \frac{1}{\max_{(i,j) \in V} |f(i+u, j+v) - h(i, j)|} \quad (4.16)$$

$$C_2(u, v) = \frac{1}{\sum_{(i,j) \in V} |f(i+u, j+v) - h(i, j)|} \quad (4.17)$$

$$C_3(u, v) = \frac{1}{\sum_{(i,j) \in V} [f(i+u, j+v) - h(i, j)]^2} \quad (4.18)$$

V je množina všech obrazových elementů vzoru a (u, v) poloha vzoru v obrazu.

4.4 Funkce matematické morfologie

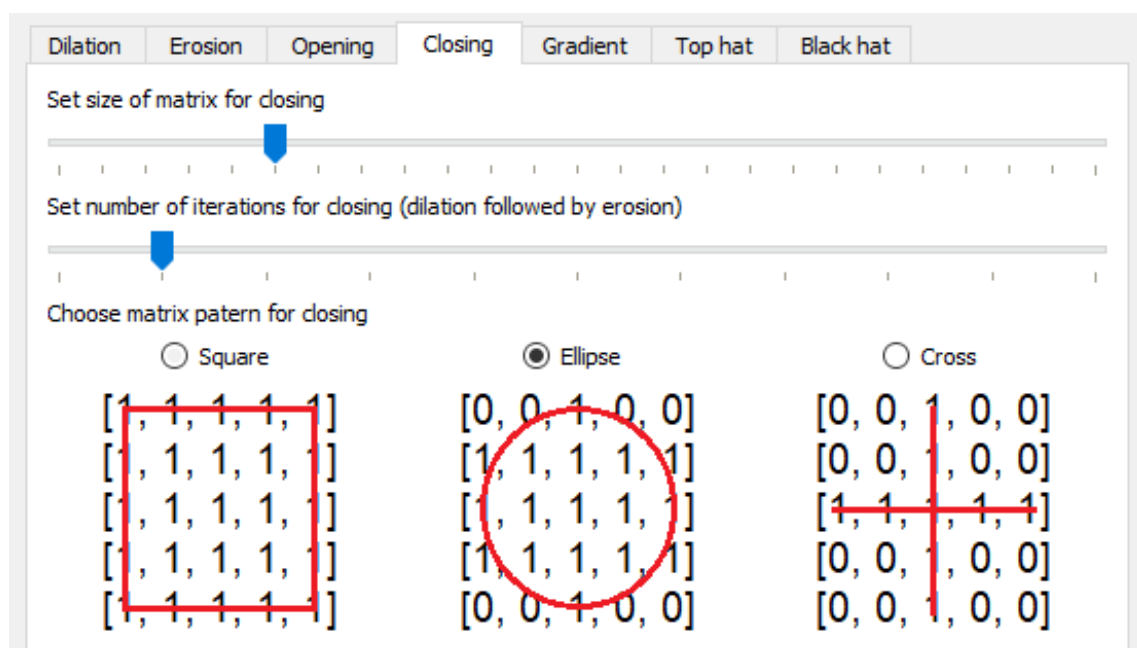
Morfologii můžeme pro naše potřeby chápat jako nauku, která se zabývá tvary. Při zpracovávání obrazu využíváme matematickou transformaci, která aplikuje matematické prvky pro získání vyžadovaných částí obrazového materiálu. Matematická morfologie je uplatňována v oblastech, kde se objevuje absence dostatečného množství času. Jedná se například o biologii, výzkum materiálů, inspekci obrazu v různých průmyslových odvětvích, geologii, kriminalistice či pro rozpoznávání dokumentů a jednotlivých znaků. Aplikace, které využívají matematickou morfologii se uplatňují jak při předzpracování obrazu, tak i při jeho konečných úpravách. V těchto případech se jedná hlavně o detekci hran či segmentaci obrazu. Oblast můžeme rozdělovat na binární a šedotónovou. Obor využívá nelineárních operací spolu s použitím teorie množin. Zaměřuje se na vlastnosti bodových množin a výsledků z integrální geometrie a topologie. Prvky a nástroje matematické morfologie jsou využívány v mnoha software pro práci s obrazem.

Obor matematická morfologie operuje se skutečností, ve které jsou skutečné obrazy tvořeny díky bodovým množinám, a to ve dvojrozměrném euklidovském prostoru. Ve zpracovávání binárních obrazů je využívána binární matematická morfologie, tedy předpoklad, že popisný nástroj je tvořen množinou dvojic čísel. V tom případě můžeme obrazy popisovat jako podmnožiny ve 2D prostoru. Obrazy jsou tvořeny z pixelů. Kdy odlišnou hodnotu v jejím oboru mají body objektu a odlišnou body pozadí. Každý pixel je definován dvěma čísly, číslo popisuje polohu v diskrétní mřížce v dané souřadné ose. Z toho vyplývá, že dva body určují polohu pixelu v dané mřížce, samozřejmě v závislosti na vytyčeném počátku. Morfologická transformace je poté dána vztahem mezi obrazem a další jinou bodovou množinou. Transformace se provádí posouváním strukturálního elementu neboli reprezentativního bodu po vybraném původním obraze. Ke každé morfologické transformaci existuje duální transformace, což vyplývá z duality morfologických operací. Může docházet k posouvání i radiusvektor, vektor či k zrcadlení neboli transponování bodové množiny vůči reprezentativnímu bodu.

Mezi základní transformace matematickou morfologie řadíme dilataci, erozi, otevření, uzavření, zjištění gradientu nebo vrchní a spodní část klobouku. Všechny tyto funkce jsou zastoupeny v knihovně OpenCV a mají několik parametrů, které mohou

být nastaveny. Všechny implementované funkce mají totožné vstupní parametry jejichž přesnější nastavení je velmi podobné. Vstupní obraz je poslední známý uložený obraz při zvolení souboru funkcí „Matematická morfologie“. Uživatel si může vybrat, kterou z výše zmíněných funkcí aplikuje.

Zvolit může styl násobící matice. Knihovna OpenCV nabízí tři základní typy, jako základní styl je zvolen typ čtverec, dále je na výběr elipsa nebo kříž. Je možné mezi nimi libovolně přepínat a sledovat výsledky na aktuálním obrazu v reálném čase. (obr. A.1) Další nastavením je velikost násobící matice. Vzor matice může být zvětšen pomocí příslušného posuvníku pro zvětšení matice, základní hodnota je 1 a maximální 50. Podobným způsobem lze nastavit i počet iterací zvolené morfologické operace. To znamená, kolikrát je funkce na obrazu provedena, zde jsou limitní hodnoty nastaveny od 0 do 20. Každá změna posuvníku pro velikost matice nebo pro počet iterací je ihned zobrazena na aktuálním obrazu.



Obrázek 4.33: Možnosti nastavení morfologických operací

4.4.1 Dilatace a eroze

Dilatace a eroze jsou základní operace, které můžeme označit za převrácené. Pracujeme zde s Minkovského množinovým součtem \oplus a rozdílem \ominus . Dilatace je chápána jako skládání dvou množin vektorovým součtem. Můžeme ji vyjádřit jako součet dvou posunutých množin. Po nastavení matice a počtu dilatací dochází k posunutí objektu o zadaný počet. Dilataci definujeme jako kumulativní, asociativní, rostoucí transformaci, která je invariantní vůči posunu. Funkce dilatace nachází využití při zaplňování malých děr či zvětšování objektů atd.

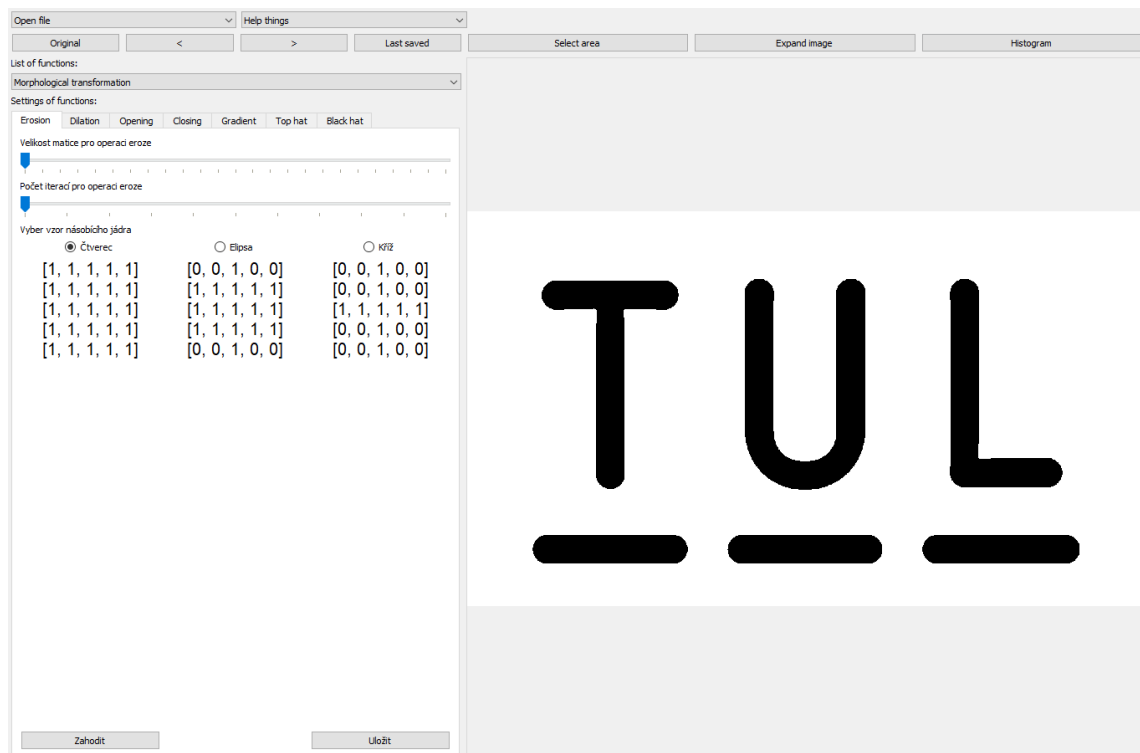
Dilataci \oplus lze vyjádřit jako sjednocení posunutých bodových množin, kde X_b představuje obraz X posunutý vektor b . (obr. 4.35)

$$X \oplus B = \bigcup_{B \in b} X_b \quad (4.19)$$

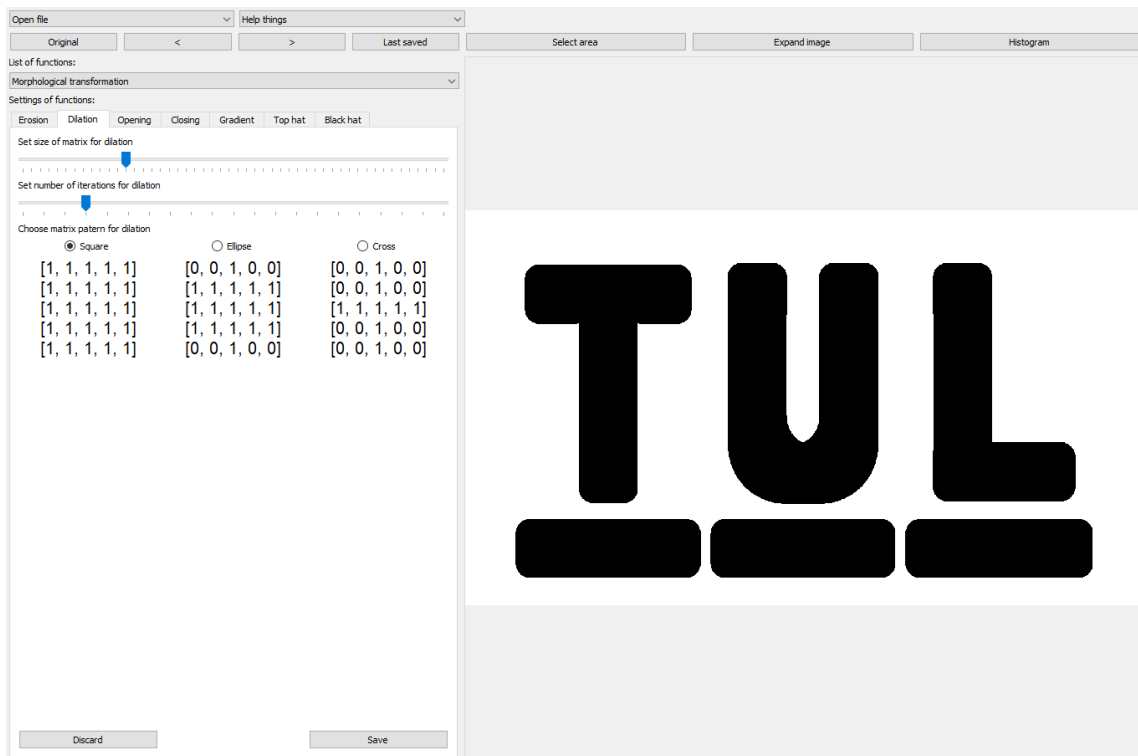
Dilatace se kombinuje s erozí, která je její duální morfologickou operací. V tomto případě se nejedná o množinový součet, nýbrž o množinový rozdíl. Při implementaci se zjišťuje, zda bod leží v objektu či nikoliv. Eroze charakterizuje průnik všech posunů v obrazu. Posunutý bod je součástí obrazu, odpovídá reprezentativnímu bodu, a tedy patří do procesu eroze. Při využití může docházet k jevu, při kterém z obrazu zmizí objekty, které mají pouze jednobodovou tloušťku. Eroze se využívá, s cílem zjednodušit struktury. Pokud dojde k odečtení obrazu po erozi od původního obrazu, můžeme získat samotný obrys objektu. Erozi definujeme jako invariantní vůči posunu, rostoucí a za jistých podmínek antiextenzivní transformaci. S dilatací je duální, avšak na rozdíl od ní není komutativní. Výsledkem obou operací je zjednodušený obraz, který obsahuje méně detailů, se kterým se nadále dá pracovat v otevření a uzavření.

Erozi \ominus lze vyjádřit jako průnik všech posunů X o vektoru $-b \in B$. (obr. 4.36)

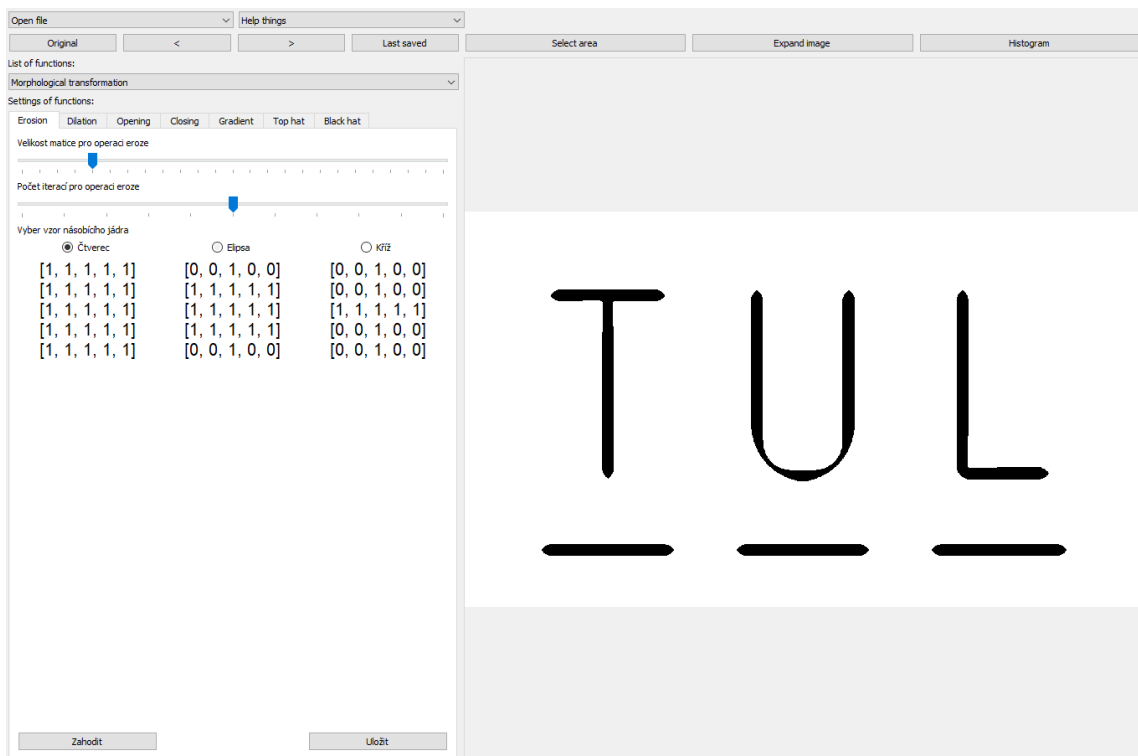
$$X \ominus B = \bigcap_{B \in b} X_{-b} \quad (4.20)$$



Obrázek 4.34: Originální šedotónový obraz pro dilataci a erozi



Obrázek 4.35: Dilatace provedena 12×, s velikostí jádra 3×3



Obrázek 4.36: Eroze provedena 5×, s velikostí jádra 6×6

4.4.2 Otevření a uzavření

Ve funkci otevření dochází ke zjednodušování struktur. Dokáže odstraňovat spojnice a malé objekty. Umí rozdělit objekty spojené jen tenkou linií. V následujících rovnicích X představuje obraz a B je násobící element. (obr. 4.37)

$$X \circ B = (X \ominus B) \oplus B \quad (4.21)$$

Oproti tomu uzavření dokáže zaplnit malé díry nebo vyhladí obrysy. K tomu dochází díky zaplnění a spojení objektů, které se nacházejí ve své blízkosti. (obr. 4.39)

$$X \bullet B = (X \oplus B) \ominus B \quad (4.22)$$

Uzavření a otevření jsou stejně jako dilatace a eroze duální morfologické operace. Jedná se o rostoucí transformace. Otevření je antiextenzivní

$$X \circ B \subseteq X, \quad (4.23)$$

nýbrž uzavření je extenzivní

$$X \subseteq X \bullet B. \quad (4.24)$$

U těchto operací se objevuje vlastnost idempotence.

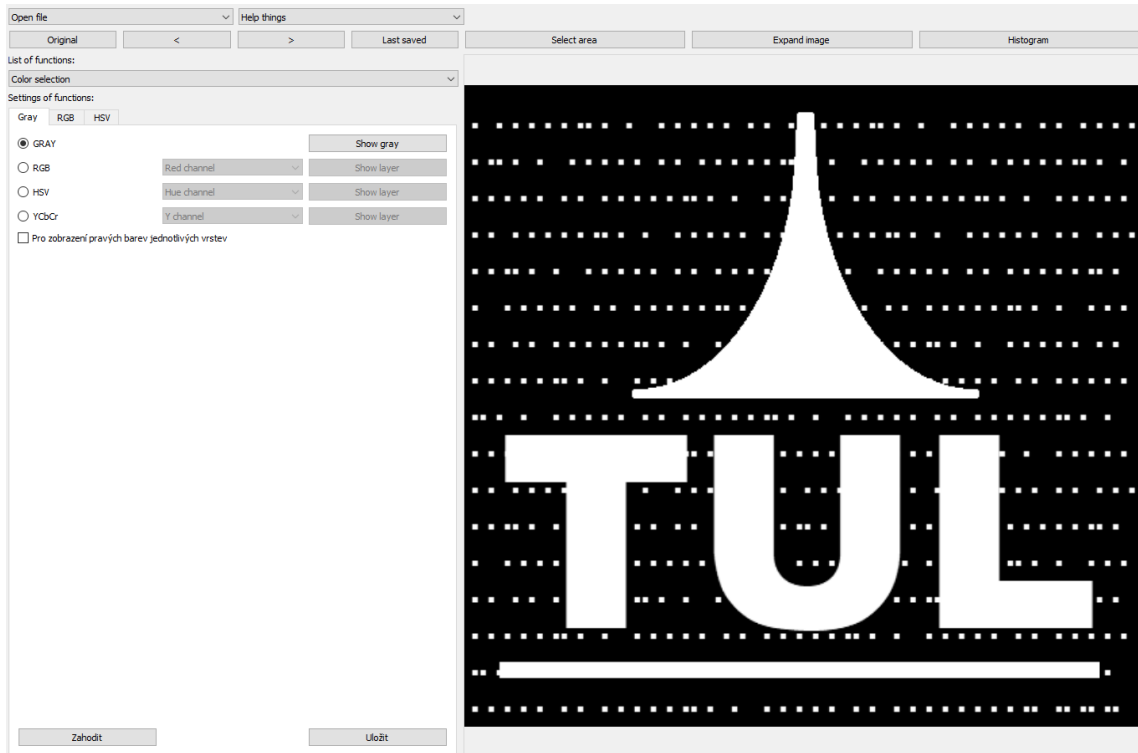
Což můžeme vykládat jako skutečnost, že které dojde po uzavření, či otevření. Dochází k tomu, že po použití těchto operací se objekt již nemění. Tedy otevření nebo uzavření pak probíhá vzhledem k jinému elementu.

4.4.3 Gradient

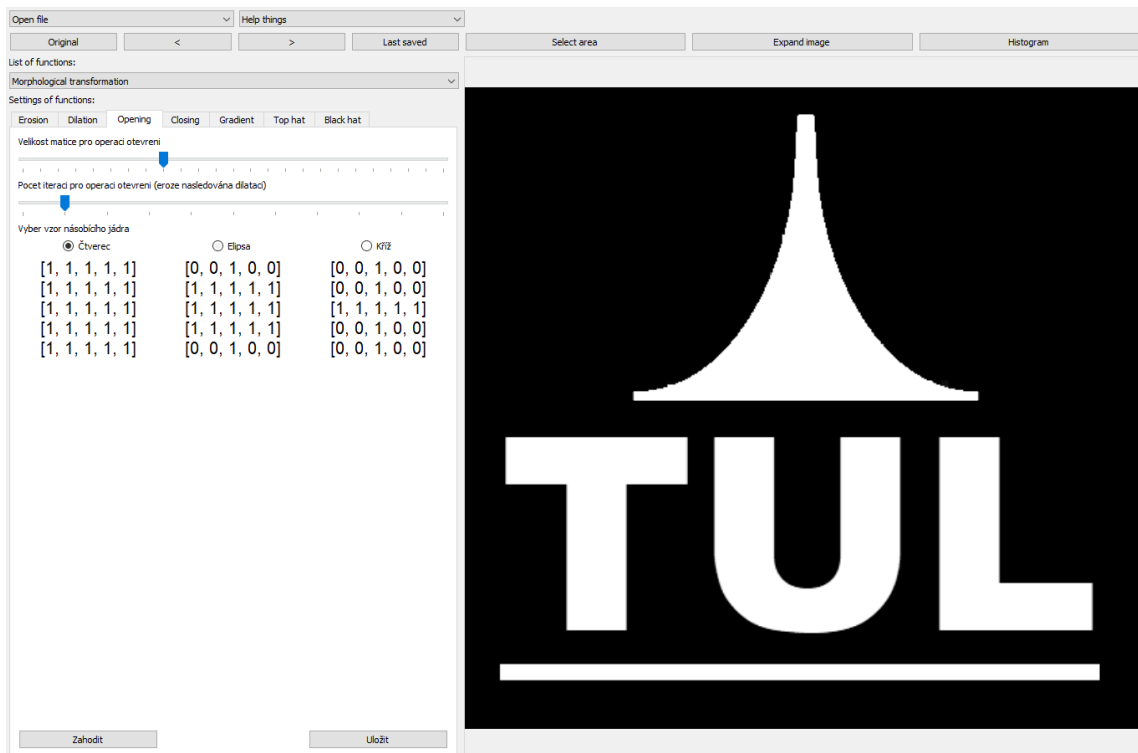
V matematické morfologii a ve zpracování digitálního obrazu je morfologický gradient vyjádřený jako rozdíl mezi dilatací a erozí vstupního obrazu. (obr. 4.41)

$$X_g = (X \oplus B) - (X \ominus B) \quad (4.25)$$

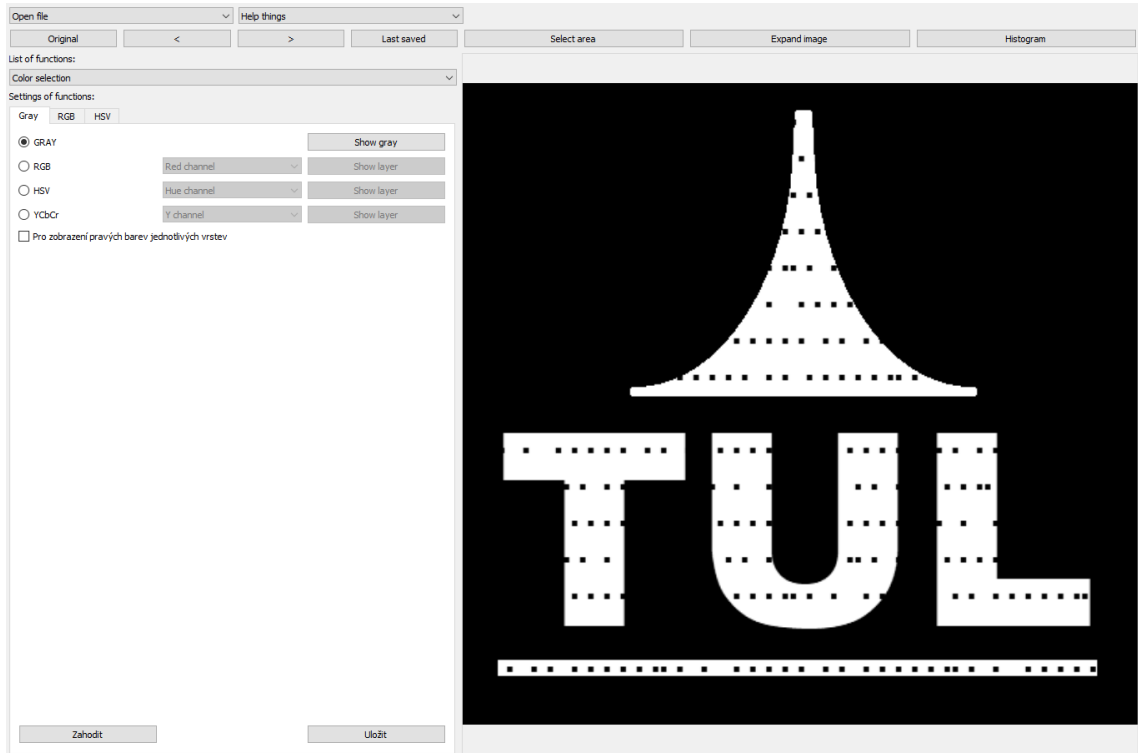
Je to obraz, kde každá hodnota pixelu zobrazuje rozdíl intenzity v těsném okolí tohoto pixelu. Morfologický gradient je užitečný pro detekci hran v obrazu nebo segmentaci obrazu. [1]



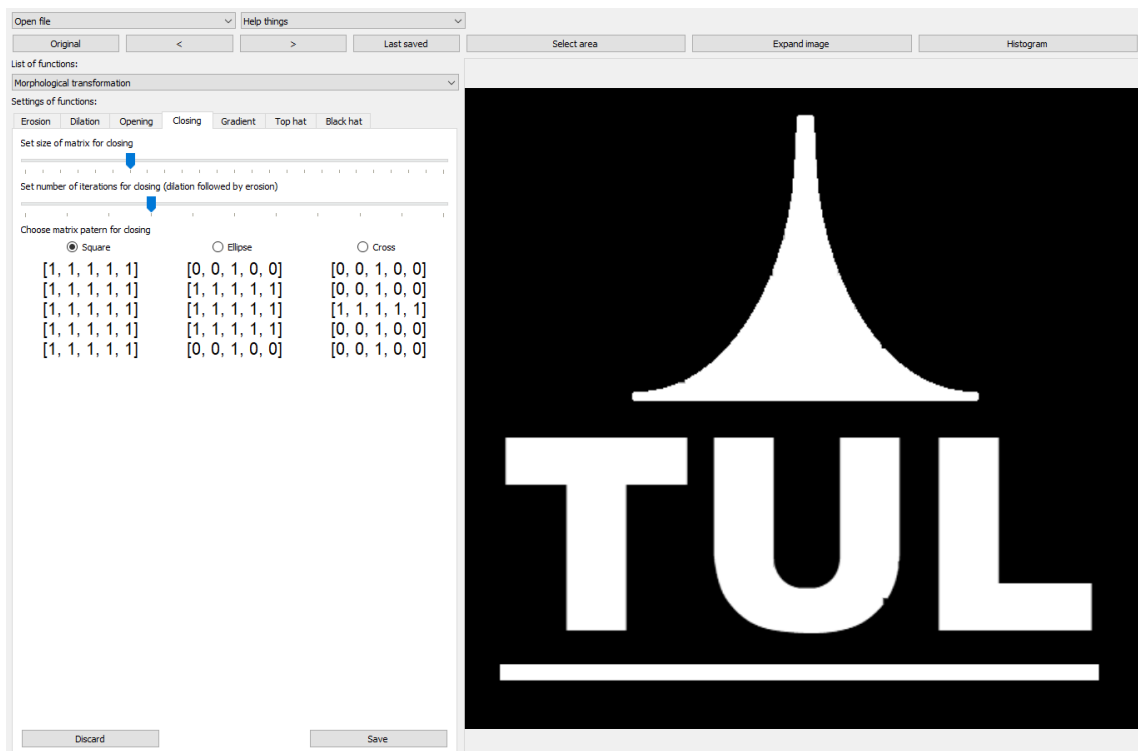
Obrázek 4.37: Původní obraz pro operaci otevření



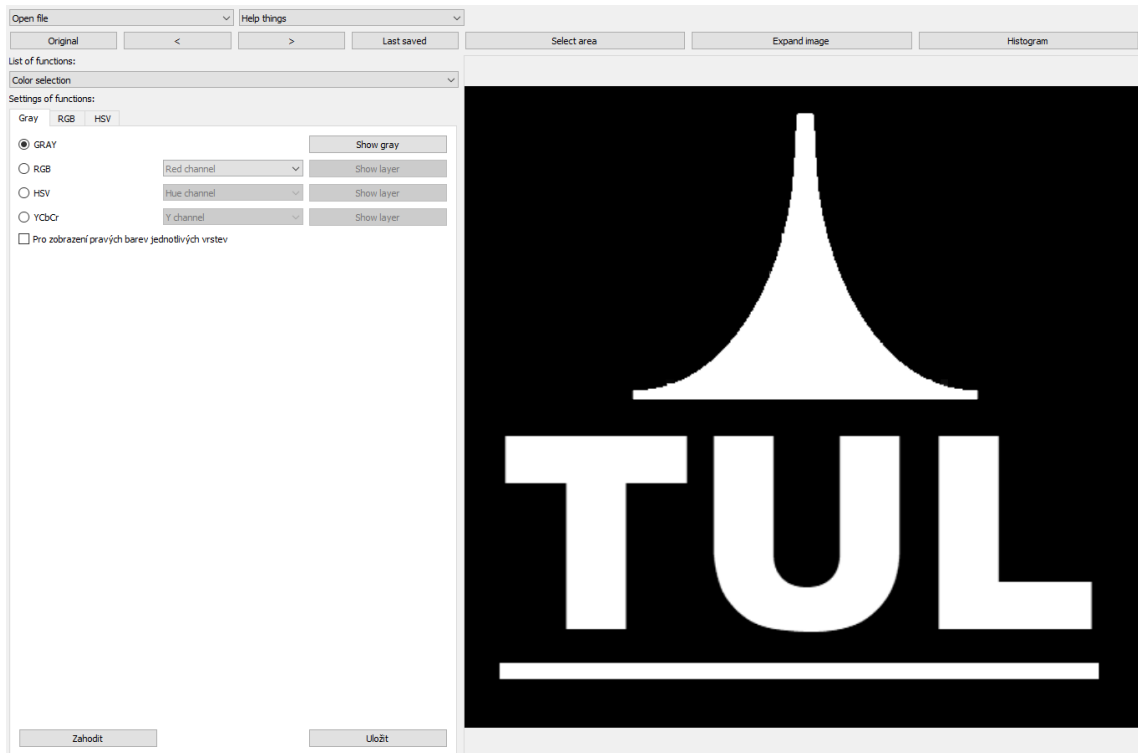
Obrázek 4.38: Obraz po operaci otevření



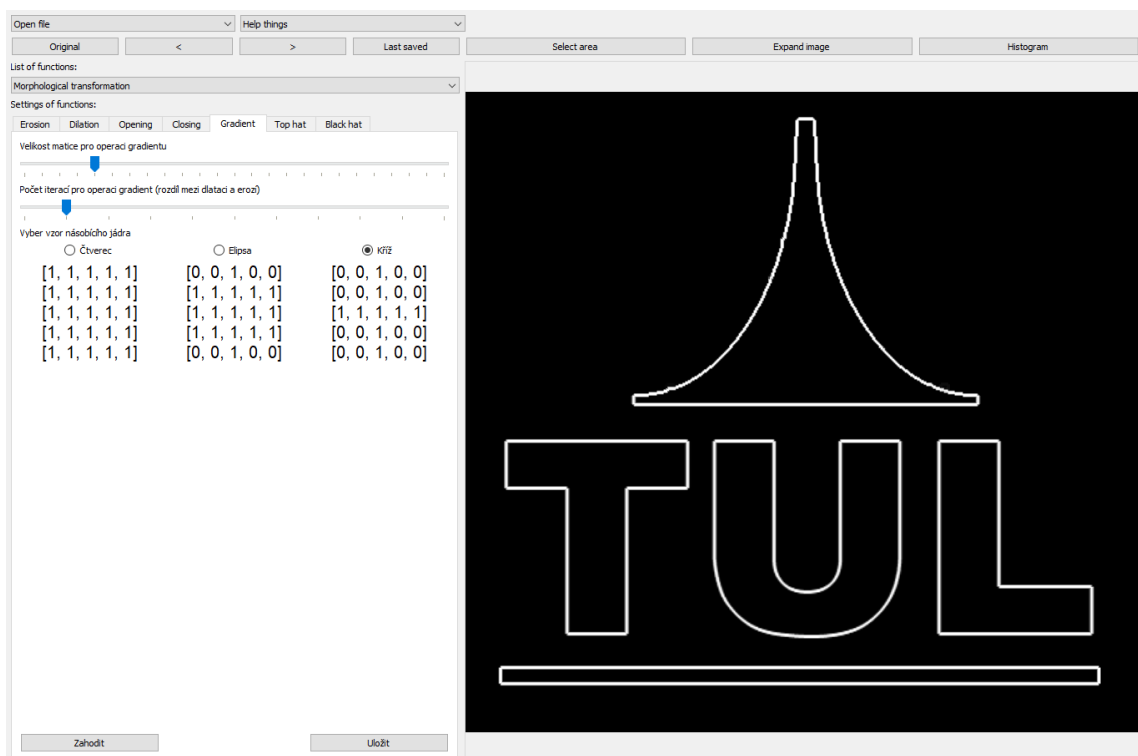
Obrázek 4.39: Původní obraz pro operaci uzavření



Obrázek 4.40: Obraz po operaci uzavření



Obrázek 4.41: Původní obraz pro operaci gradient



Obrázek 4.42: Obraz po výpočtu gradientu

4.4.4 Vrchní a spodní část klobouku

V případě obrazu s více úrovněmi jasu se využívá šedotónová matematická morfologie, neboť uvažuje trojici celých čísel. Výše zmíněné operace lze zobecnit i pro objekty s více jasovými úrovněmi. Jasová hodnota je pak interpretována jako výška ve 3D zobrazení. Stejně jako výše je objekt popisován jako body v množině pomocí souřadnic. Množina bodů se nachází ve trojrozměrném euklidovském prostoru a stejně jako předešlá je definována dvěma souřadnicemi. Zde se však objevuje třetí souřadnice, která udává výšku. Spolu s tím se objevují termíny supremum, tedy nejmenší horní odhad a infimum, nejmenší odhad. Tyto termíny popisují v operacích min a max. Pokud zde hovoříme o dilataci a erozi, pak je zaváděn termín stín a vršek bodové množiny. Pro každou dvojici souřadnic je vršek nejvyšší hodnotou zbylé souřadnice množiny. Stínem je pak rozuměna oblast stínu způsobena neprůhledným objektem. V matematické morfologii je stín funkce množiny, která se skládá z vršku a celého prostoru pod ním.

Při dilataci \oplus a erozi \ominus , kde X je obraz a B je strukturální element za pomocí vzorců,

$$X \oplus B = T\{V[X] \oplus V[B]\} \quad X \ominus B = T\{V[X] \ominus V[B]\} \quad (4.26)$$

dochází k operacím, které pracují právě s vrškem a stínem. Nejprve dochází k získání stínu v 1D funkci, který je nadále dilatován nebo erodován. Výsledek je vypočten jako vršek množiny. Operace šedotónové dilatace a eroze jsou používány ve vzájemné kombinaci, mají význam a výsledky jsou aplikovatelné pro další operace jako uzavření, otevření a transformace vrchní a spodní části klobouku. Vlastnosti, které byly popsány v binární morfologii platí i pro šedotónovou. Výše zmíněné procesy lze použít například při snižování nebo odstraňování šumů.

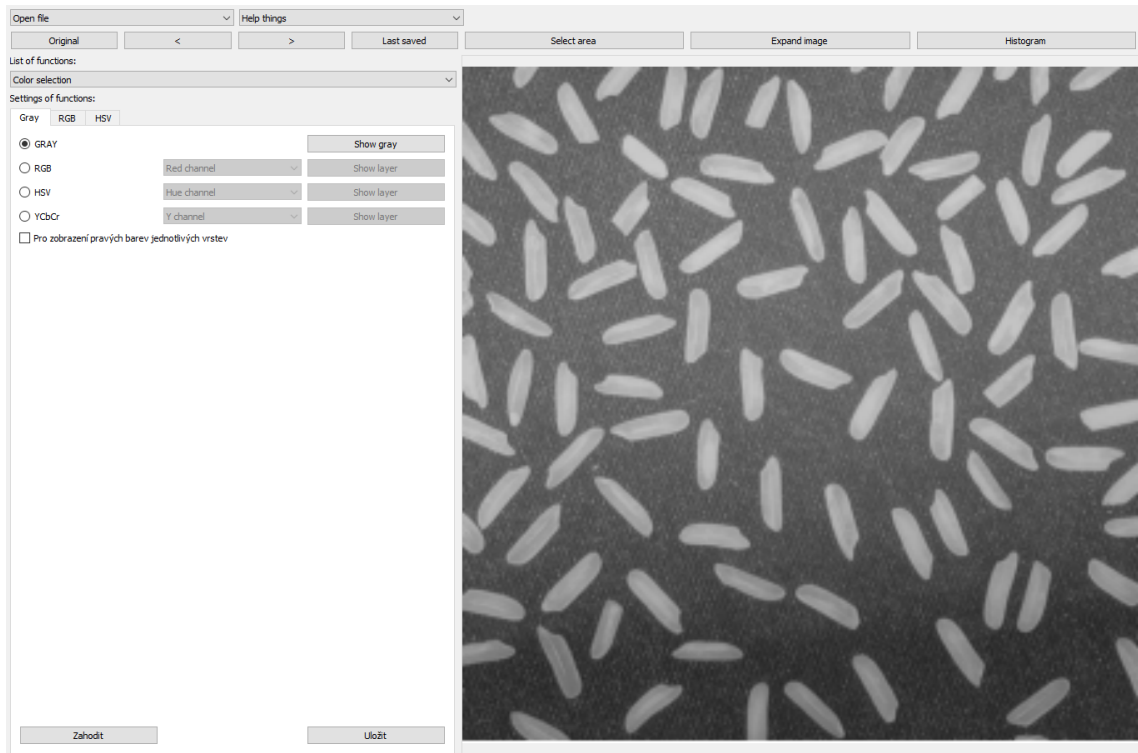
Transformace vrchní a spodní části klobouku je další z operací, které se objevují při práci s šedotónovou variantou. Jedná se o množinový rozdíl mezi otevřením a původním obrazem

$$X_{th} = X - (X \circ B), \quad (4.27)$$

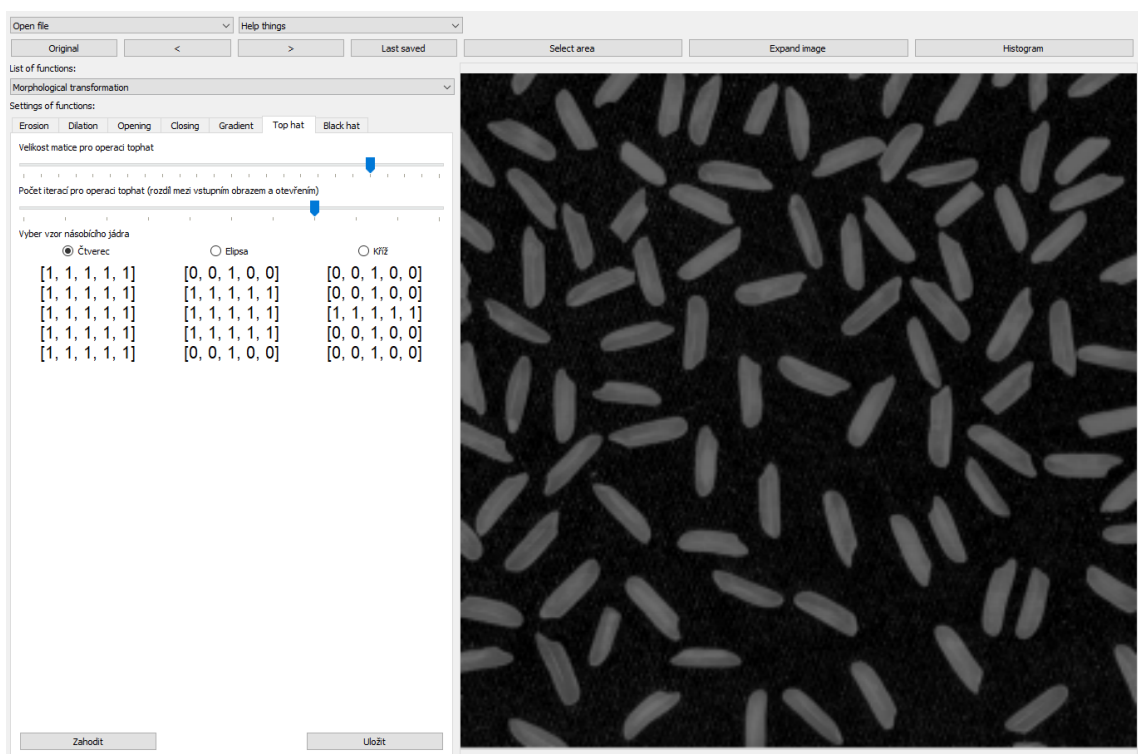
kde X odpovídá množině představující šedotónový obraz a B strukturálnímu elementu. Funkci je vhodné používat pro členění objektů, které se liší jasem, tedy když dochází k průběžnému měnění se jasu v objektu. Definuje se strukturální element a pokud je část obrazu vyšší než hodnota elementu, pak dojde k jeho odstranění. Obdobnou funkcí představuje transformace spodní části klobouku (obr. 4.45), která však nevyužívá otevření, nýbrž uzavření

$$X_{bh} = (X \bullet B) - X. \quad (4.28)$$

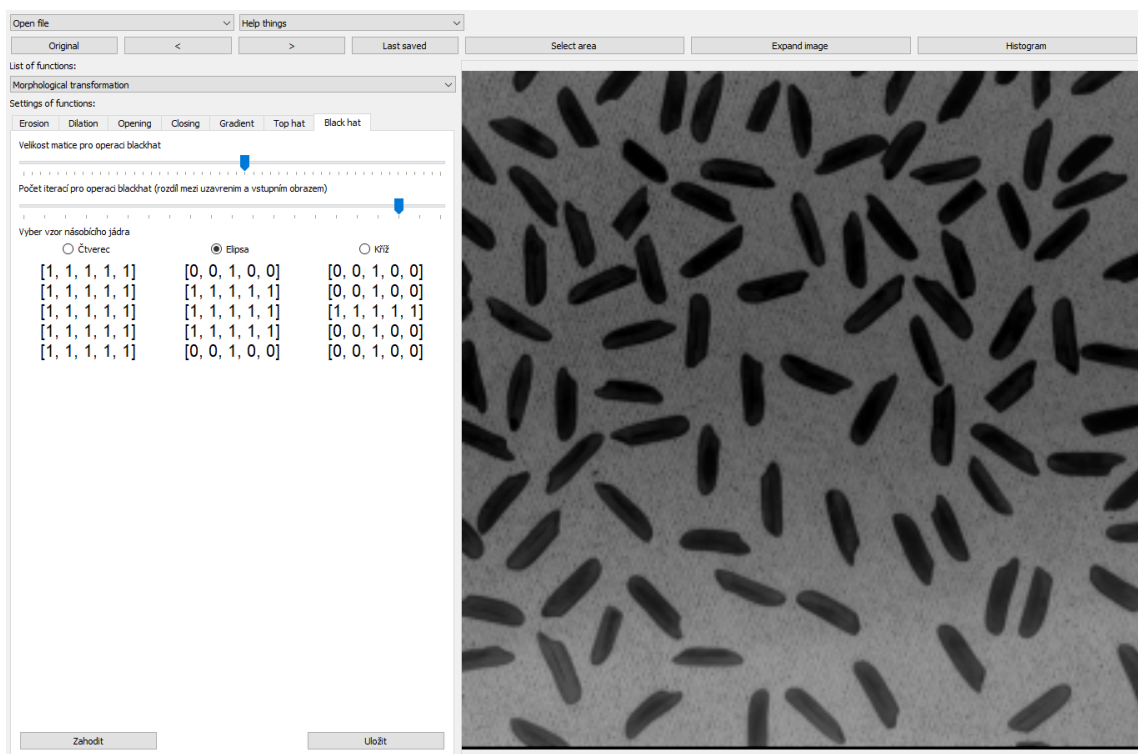
Šedotónová transformace vrchní a spodní části klobouku nachází uplatnění při hledání vad. [5]



Obrázek 4.43: Původní obraz pro operace vrchní a spodní část klobouku

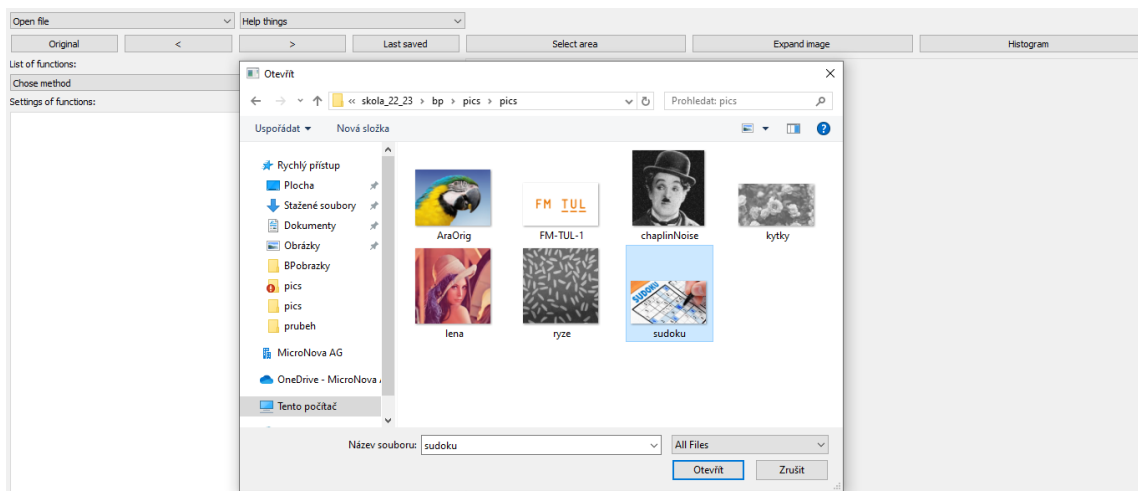


Obrázek 4.44: Aplikace operace vrchní část klobouku



Obrázek 4.45: Aplikace operace spodní část klobouku

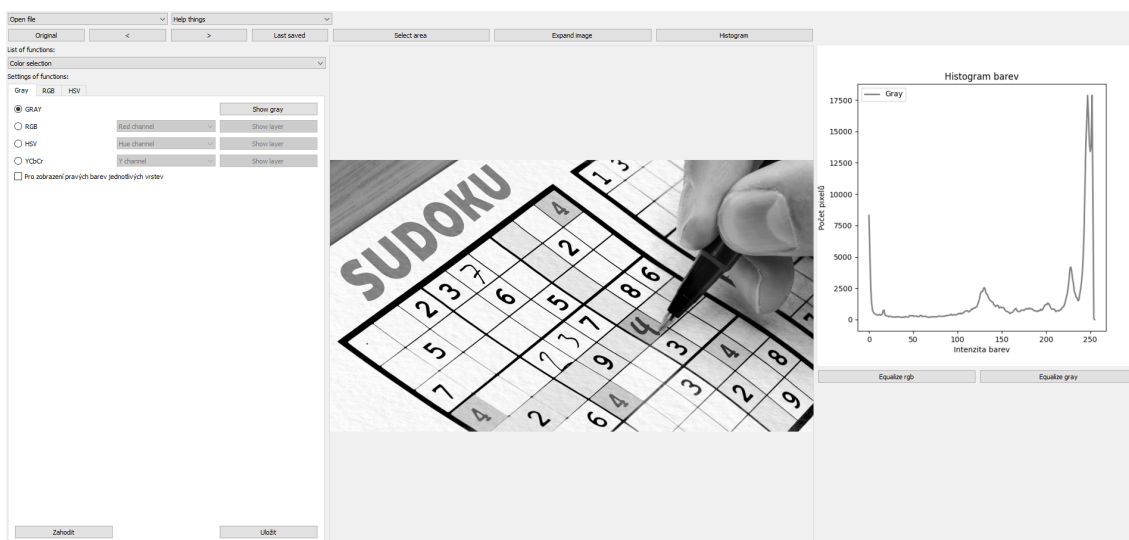
5 Příklad použití programu



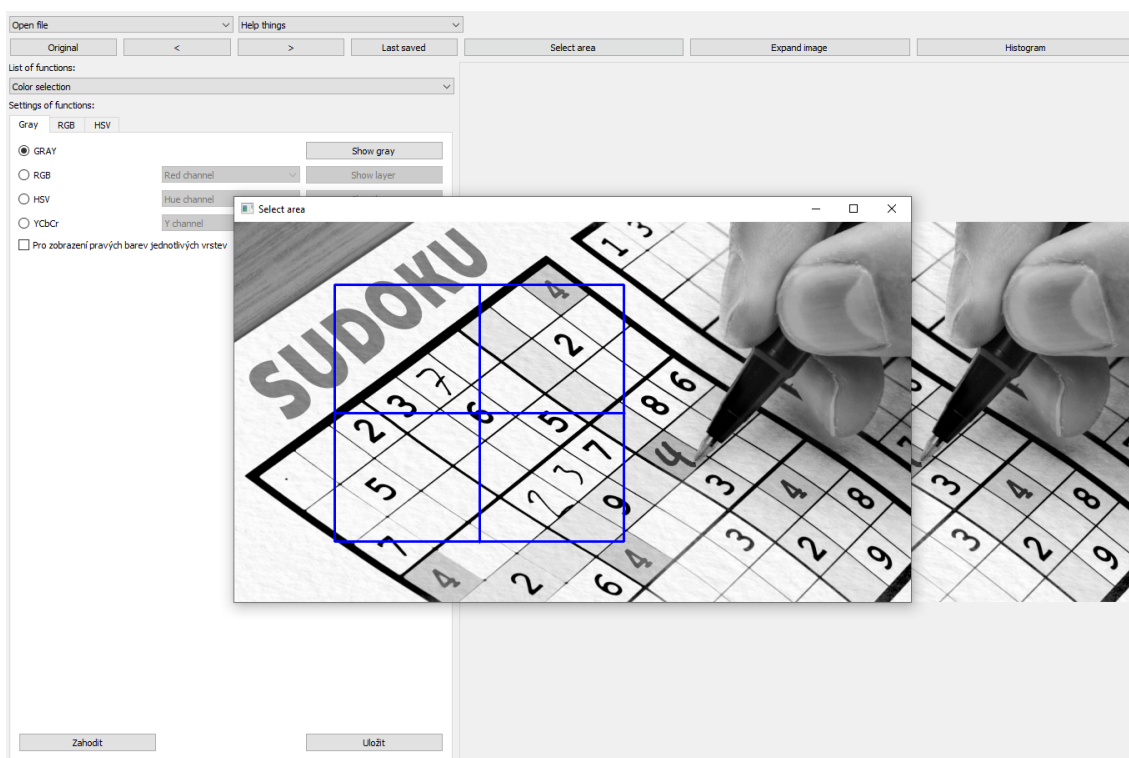
Obrázek 5.1: Načtení obrazu ze souboru



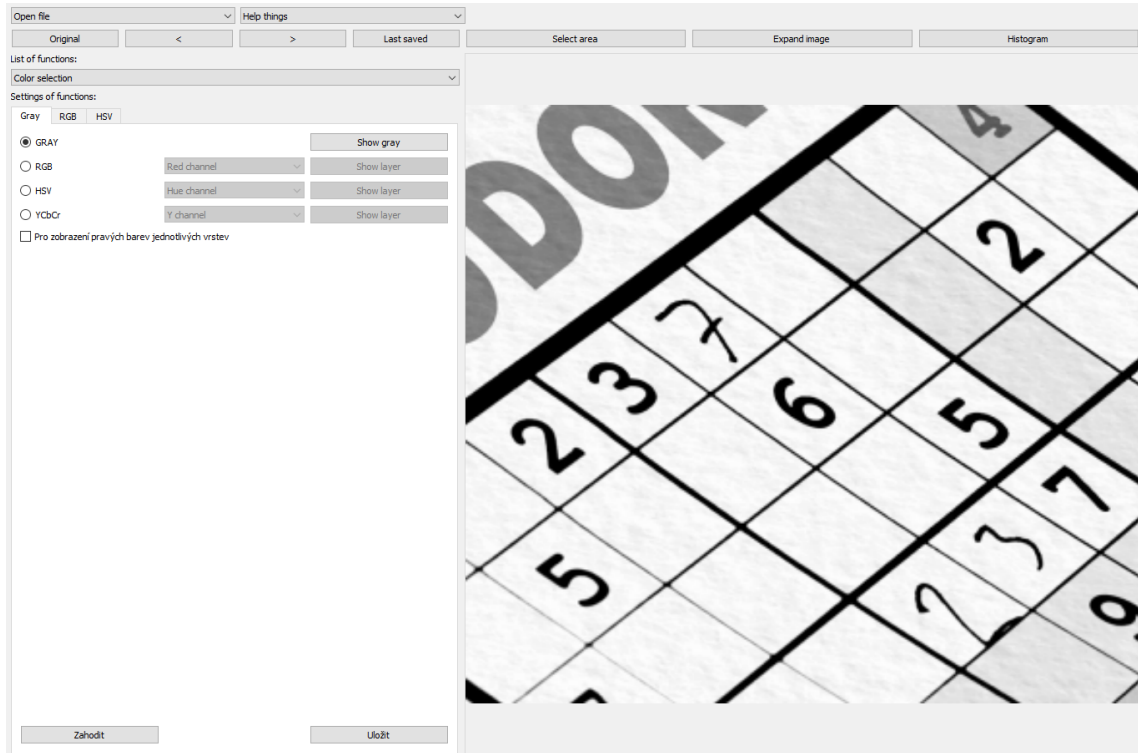
Obrázek 5.2: Zobrazení histogramu obrazu



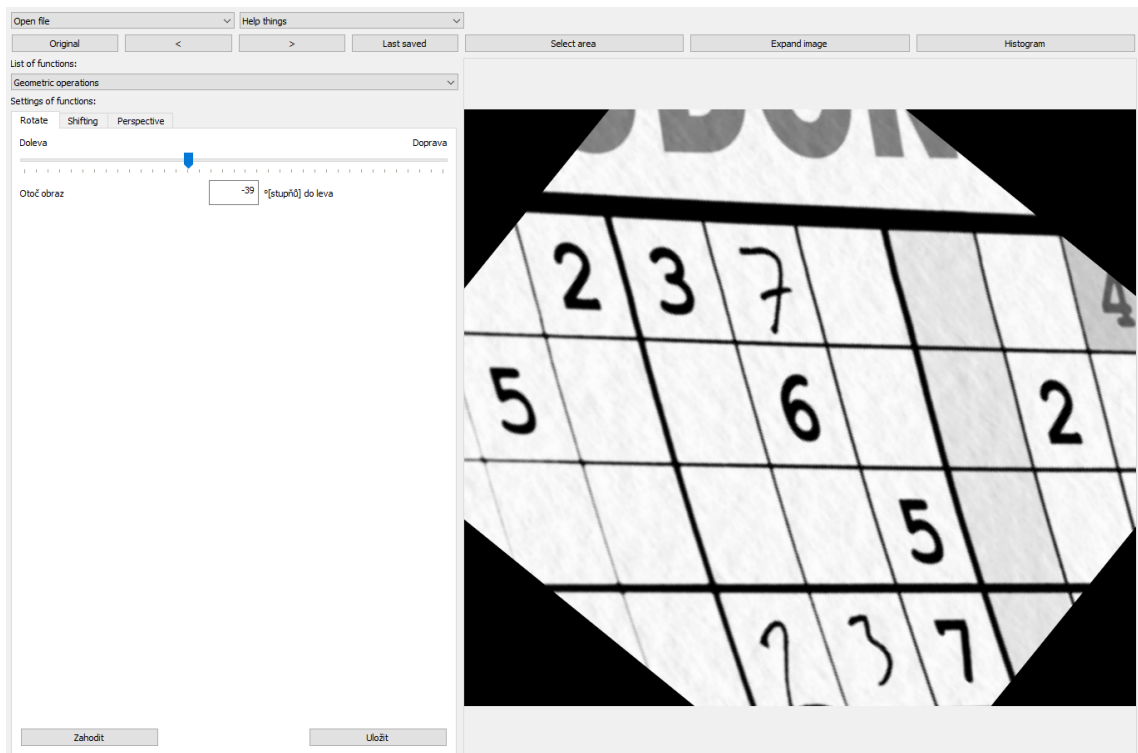
Obrázek 5.3: Převedení obrazu na monochromatický + zobrazení histogramu



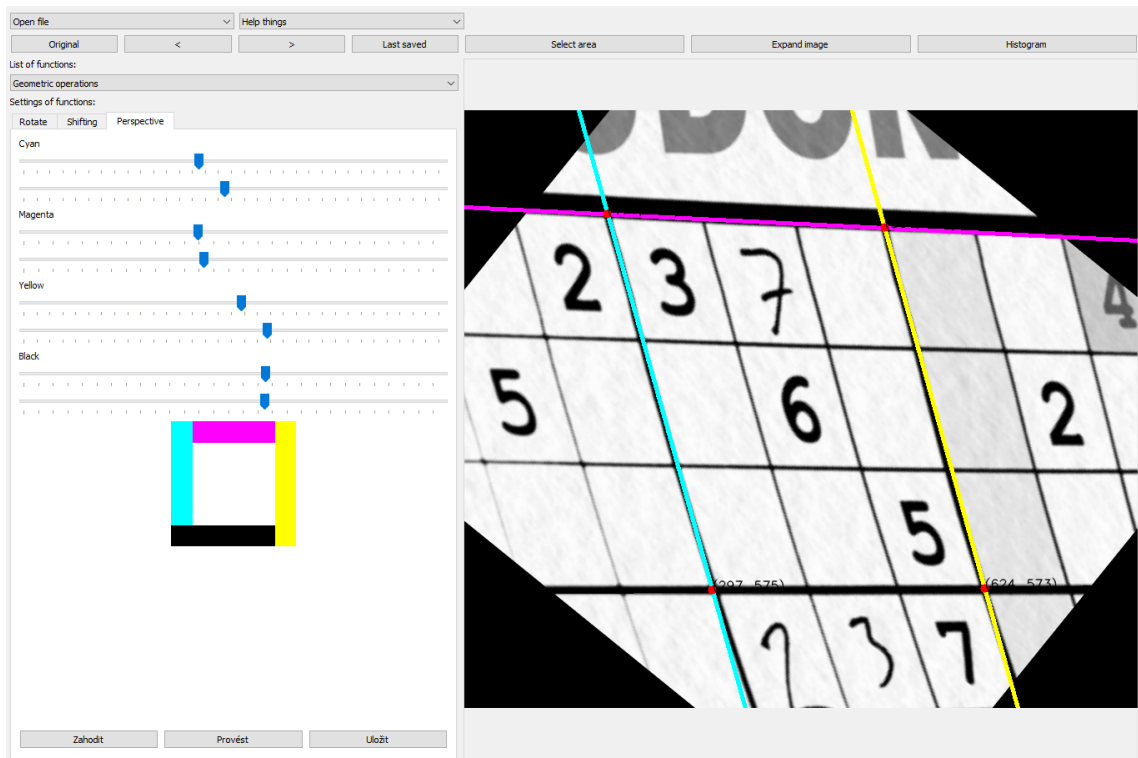
Obrázek 5.4: Výběr části obrazu pro další manipulaci



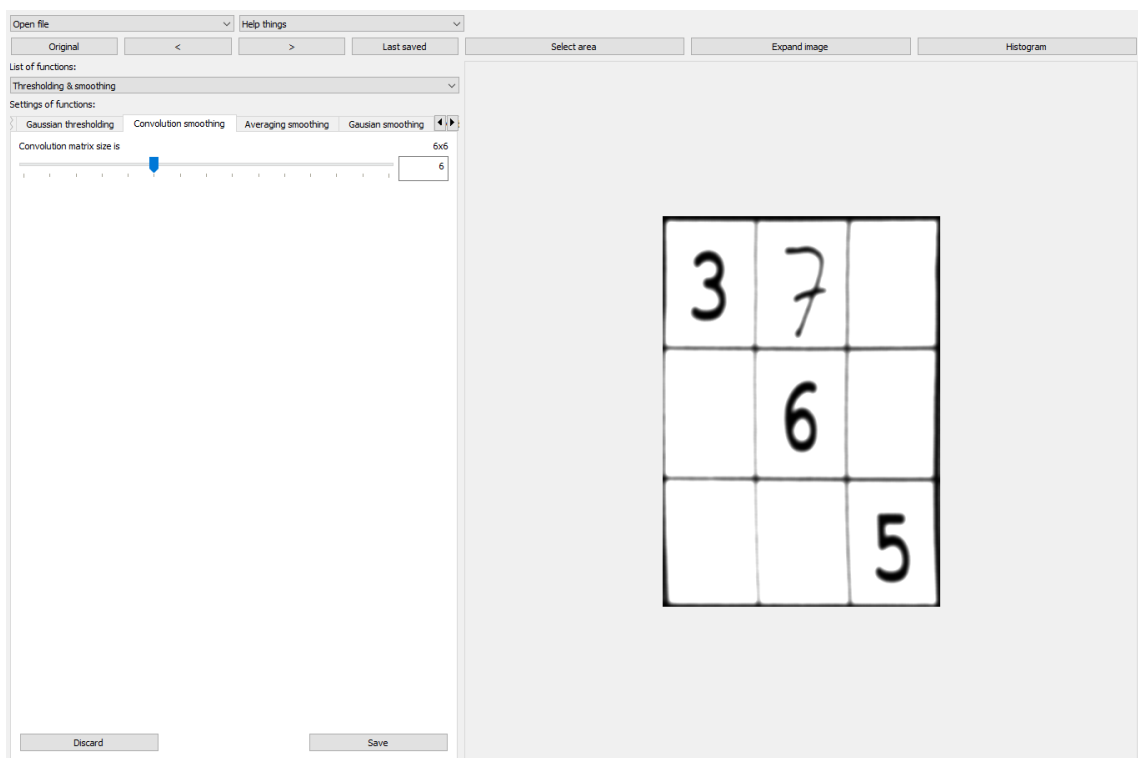
Obrázek 5.5: Zvětšení obrazu do maximální velikosti pracovní plochy



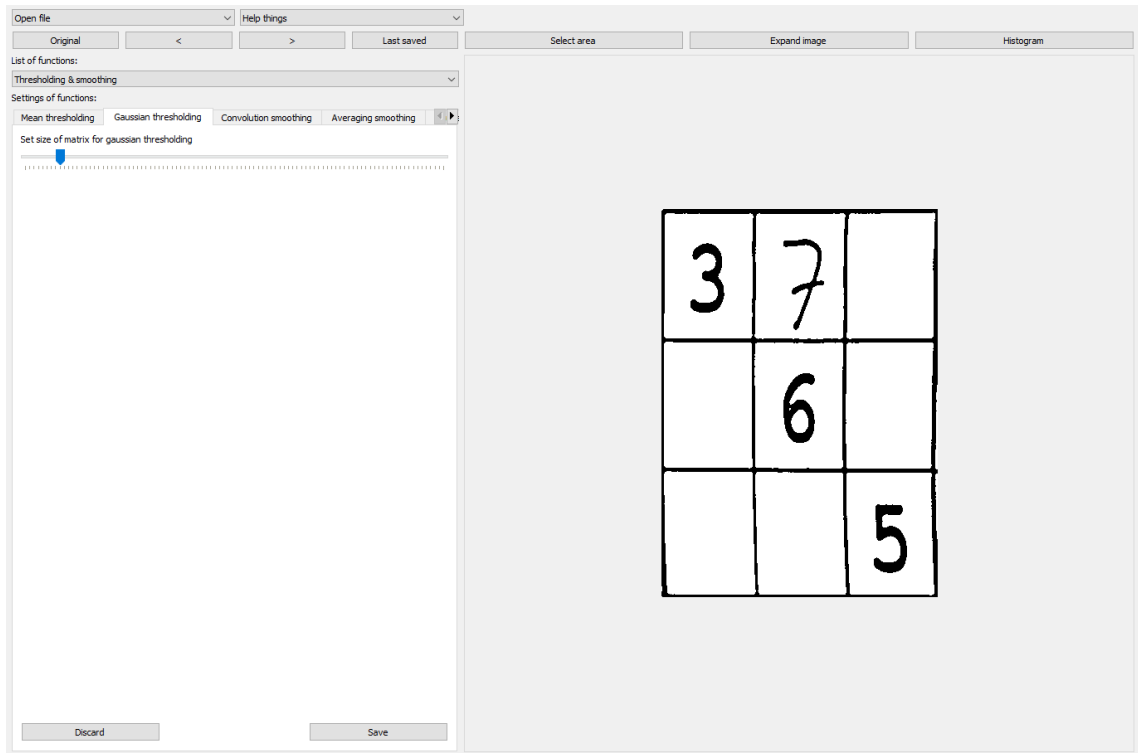
Obrázek 5.6: Rotace obrazu



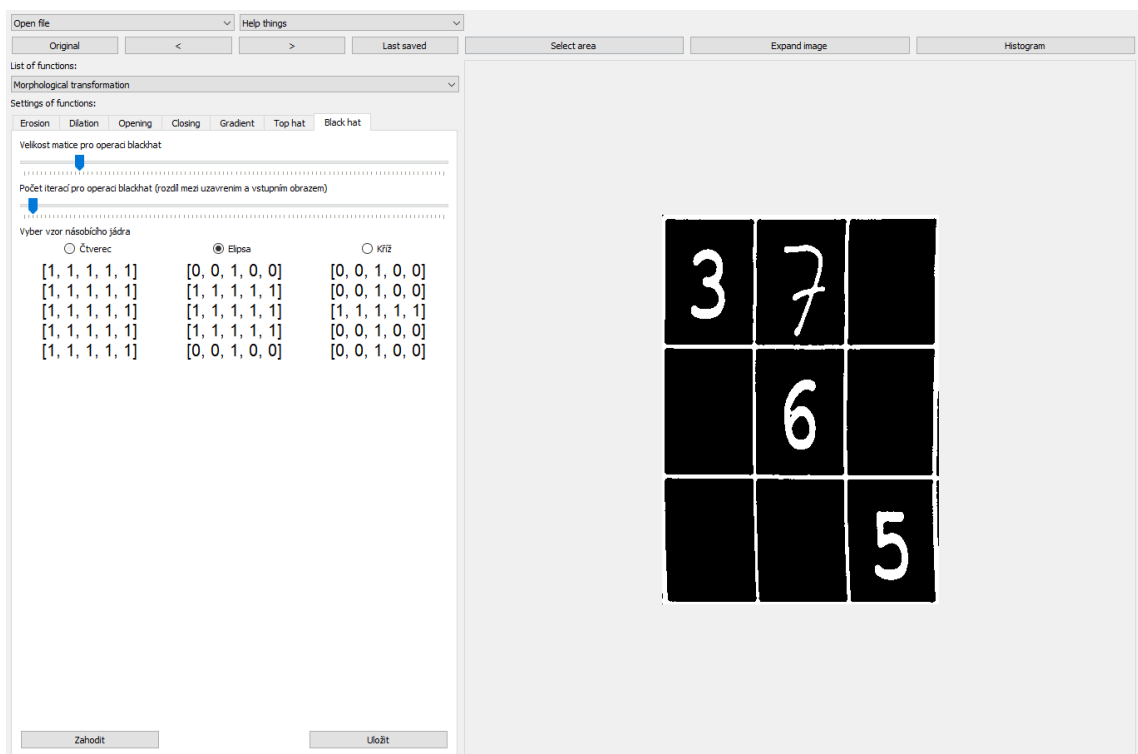
Obrázek 5.7: Vytyčení obrazu pro změnu perspektivy, na dalším obrázku je vidět provedená změna



Obrázek 5.8: Použití konvolučního filtru



Obrázek 5.9: Zdůraznění hran pomocí Gaussova prahování



Obrázek 5.10: Využití morfologické operace spodní části klobouku

Závěr

V rámci této bakalářské práce byl navržen a vytvořen program pro základní poloautomatickou analýzu obrazu s využitím prostředí Python a knihovny OpenCV. Tento systém splňuje požadavky na modularitu a otevřenost, což umožňuje snadné rozšíření o nové funkční moduly v budoucnosti.

Práce prezentuje základní principy zpracování a rozpoznávání obrazu a uvádí přehled současných algoritmů a metod v této oblasti. Z oblasti barevných prostorů jsou využívány konverze mezi modely RGB, HSV a YCbCr. Geometrické transformace obrazu, jako rotace, posunutí a změna perspektivy, jsou realizovány pomocí souboru geometrických funkcí. Pro odstranění šumu nebo negativních vlivů jsou implementovány filtry, především Gaussův nebo mediánový filtr. Pro finální úpravu obrazu jsou zahrnuty funkce z matematické morfologie, tedy operace eroze, dilatace a jejich kombinace jako otevření, uzavření a výpočet gradientu. Systém se zaměřuje na uživatelskou přívětivost a intuitivnost uživatelského rozhraní, což zajišťuje jeho snadné použití pro širokou škálu uživatelů.

Vzhledem k rychlému rozvoji technologií zpracování obrazu a rozpoznávání vzorů je možné očekávat další rozšíření a zdokonalení implementovaného systému v budoucnosti. Tato práce tak poskytuje solidní základ pro vzdělávání v této problematice nebo pro další výzkum a aplikace v oblasti zpracování obrazu.

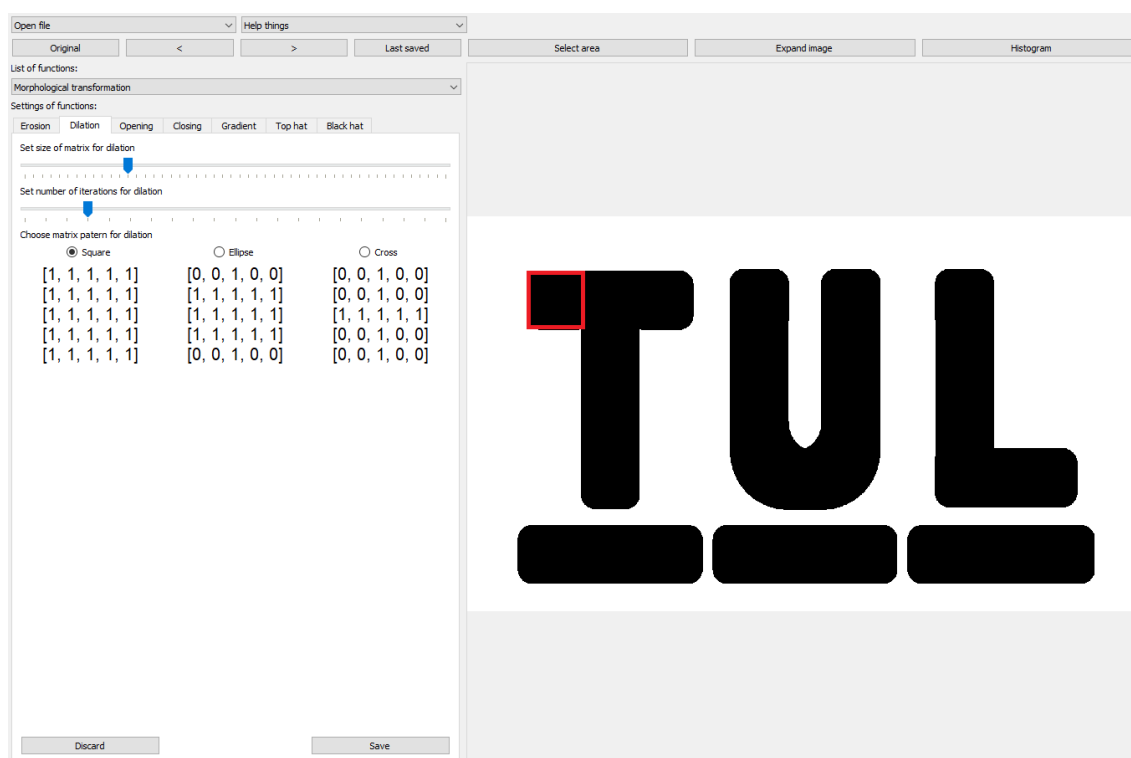
Během tvorby práce jsem si zvýšil své znalosti v problematice zpracování obrazu, v programování a navrhování uživatelských rozhraní. Tyto znalosti bych chtěl dále využívat a zdokonalovat, ať už ve stávajícím nebo v budoucím povolání. Mé zkušenosti z této práce mě motivují k hledání a zapojení se do projektů, které se zabývají pokročilým zpracováním obrazu a jeho aplikacemi v různých odvětvích, především průmysl nebo autonomní řízení.

Použitá literatura

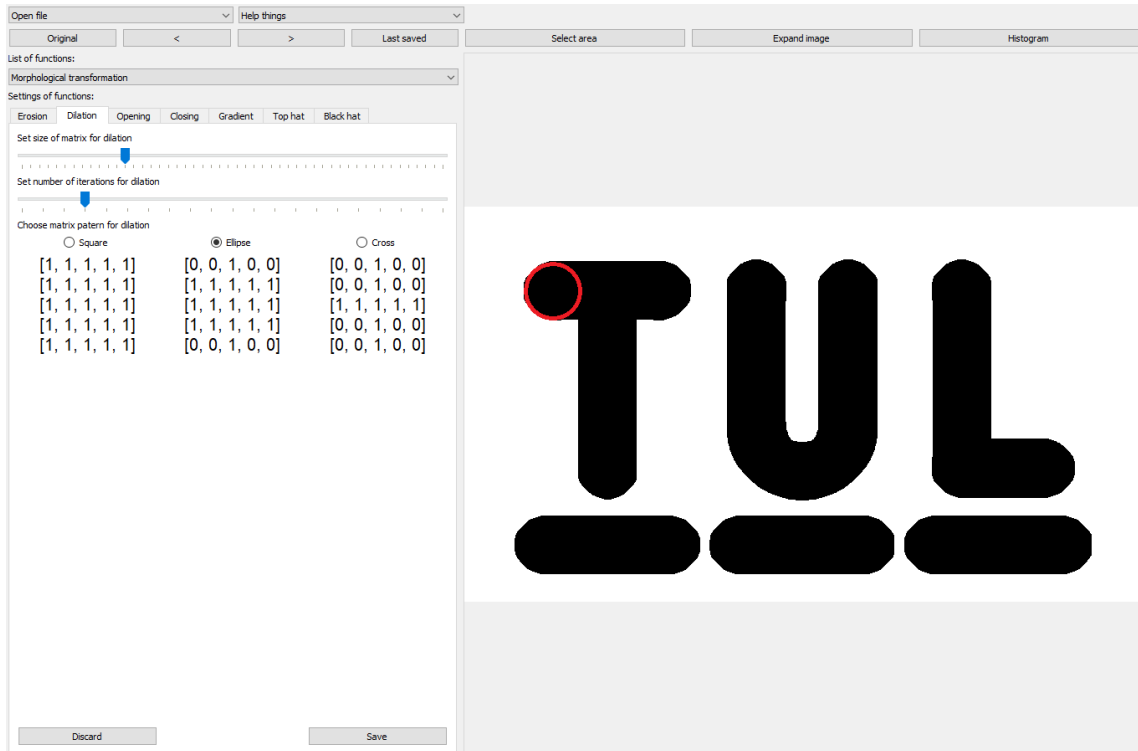
- [1] GONZALEZ, Rafael C. a Richard E. WOODS. *Digital image processing*. Fourth edition. New York: Pearson, 2018. ISBN 978-1-292-22304-9.
- [2] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. Vyd. 2., přeprac. Praha: Vydavatelství ČVUT, 2005. ISBN 978-80-01-03110-0.
- [3] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. Vyd. 3. V Praze: České vysoké učení technické, 2009. ISBN 978-80-01-04442-1.
- [4] HLAVÁČ, Václav a Milan ŠONKA. *Počítačové vidění*. Praha: Grada, 1992. ISBN 80-854-2467-3.
- [5] HOTAŘ, Vlastimil. *Úvod do problematiky strojového vidění*. 1. vyd. V Liberci: Technická univerzita, 2015. ISBN 978-80-7494-202-0.
- [6] HUNTER, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. 2007, roč. 9, č. 3, s. 90–95. Dostupné z DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [7] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. *Image processing, analysis, and machine vision*. Fourth edition. Stamford, CT: Cengage Learning, 2015. ISBN 978-1-133-59369-0.

A Přílohy

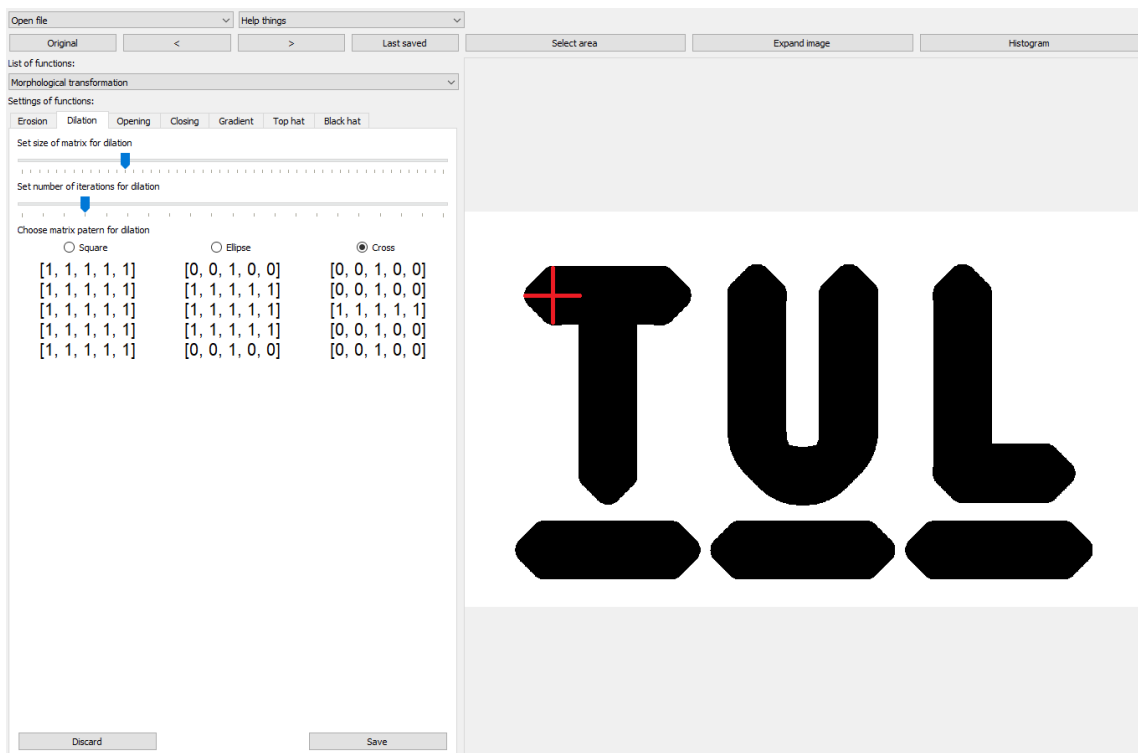
A.1 Příklady změny násobícího jádra



Obrázek A.1: Dilatace za použití jádra čtverec



Obrázek A.2: Dilatace za použití jádra kruh



Obrázek A.3: Dilatace za použití jádra kříž