

Spis treści

Wstęp	3
Cel i zakres pracy	4
1 Rozdział z dziedziny pracy - część teoretyczna	5
1.1 Rys historyczny	5
1.1.1 Wywoływanie zdjęć	5
1.1.2 Cyfrowe przetwarzanie obrazów	6
1.2 Przegląd istniejących rozwiązań	7
1.2.1 Photoshop	7
1.2.2 ImageJ	8
1.2.3 Fiji	9
1.2.4 GIMP	10
1.2.5 Adobe Substance Designer	11
2 Zakres użytych technologii i opis wykorzystywanych narzędzi	12
2.1 OpenCV	12
2.2 Visual Studio Code	12
2.3 Git	12
2.4 GitHub	12
3 Realizacja projektu	13
3.1 Zastosowania	13
3.2 Wymagania funkcjonalne i нефункционалне	13
3.2.1 Wymagania funkcjonalne	13
3.2.2 Wymagania нефункционалне	13

4 Podsumowanie	14
Bibliografia	15
Spis rysunków	16
Spis tabel	17
A. Załączniki	18

Wstęp

Przetwarzanie obrazów to dziedzina informatyki zajmująca się automatyzacją zadań związanych z obrazami. Obejmuje ona szeroki zakres zagadnień, takich jak sztuczna inteligencja, uczenie maszynowe, analiza danych i grafika komputerowa.

Oprogramowanie do przetwarzania obrazów jest wykorzystywane w wielu różnych dziedzinach, takich jak medycyna, przemysł, bezpieczeństwo i rozrywka. W medycynie jest wykorzystywane do diagnostyki, obrazowania medycznego i chirurgii. W przemyśle jest wykorzystywane do kontroli jakości, automatyzacji i inspekcji. W bezpieczeństwie jest wykorzystywane do monitorowania, rozpoznawania twarzy i identyfikacji. W rozrywce jest wykorzystywane do tworzenia efektów specjalnych, animacji i gier wideo.

Jedną z popularnych bibliotek do przetwarzania obrazów jest OpenCV. OpenCV to darmowy, otwartoźródłowy pakiet oprogramowania dla systemów operacyjnych Windows, macOS i Linux. Zapewnia szeroki zakres funkcji do przetwarzania obrazów, w tym: podstawowe operacje takie jak filtrowanie, skalowanie i konwersja formatów, wykrywania i rozpoznawania obiektów czy obsługę algorytmów do analizy wideo.

W ramach czasu, który został poświęcony na niniejszy projekt zakres możliwości ograniczono do prostych operacji przetwarzania obrazów bez uwzględniania wideo oraz wszelkiego rodzaju danych oprócz samych obrazów.

OpenCV jest często wykorzystywany do tworzenia oprogramowania do przetwarzania obrazów. Jednak jego interfejs użytkownika jest dostosowany do programistów, a nie do użytkowników końcowych. Oznacza to, że użytkownicy końcowi muszą mieć podstawowe umiejętności programowania, aby korzystać z oprogramowania stworzonego w oparciu o OpenCV.

W tej pracy naukowej proponujemy nowe oprogramowanie do przetwarzania obrazów oparte na bibliotece OpenCV. Oprogramowanie to posiada funkcjonalny interfejs graficzny, który ułatwia użytkownikom końcowym korzystanie z niego.

Cel i zakres pracy

Celem tej pracy jest zaprojektowanie i implementacja oprogramowania do przetwarzania obrazów opartego na bibliotece OpenCV z funkcjonalnym interfejsem graficznym.

Zakres pracy objął: analizę istniejących rozwiązań na podstawie której stwierdzono, że nie ma aktualnego i gotowego produktu który realizuje dokładnie założenie zastąpienia programistycznego podejścia do obsługi biblioteki OpenCV. Jest wiele rozwiązań które realizuje część biblioteki lub jej obsługa nie pozwala na przetwarzanie obrazów w niedestrukcyjny sposób bez dodatkowego wysiłku od użytkownika jak używanie warstw czy inkrementalnych zapisów.

Jako platformę wybrano komputer osobisty z powodu znaczącej mocy obliczeniowej, dostępu do dużej ilości pamięci RAM oraz dostępu do operacji procesorów x86 [1]. Jako docelowy system operacyjny wybrano Microsoft Windows 11 ponieważ jest to system używany przez autora tej pracy. Do tworzenia aplikacji skorzystano z języka C# oraz Windows Presentation Framework.

Pomysł na ten projekt powstał dzięki doświadczeniu autora pracy w używaniu różnych systemów opartych o edytory typu węzłowego jak Adobe Substance Designer, edytor Blueprintów w Unreal Engine oraz edytor shaderów czy geometry nodes w Blenderze. Ten typ interfejsu pozwala na interakcję użytkownika z wizualnym przedstawieniem jakichś metod, funkcji czy bloków logicznych pozwalających na przesyłanie wartości w podobny sposób jak można robić to w kodzie, ale zdecydowanie bardziej przystępne dla osób które nie mają odpowiedniego doświadczenia w programowaniu.

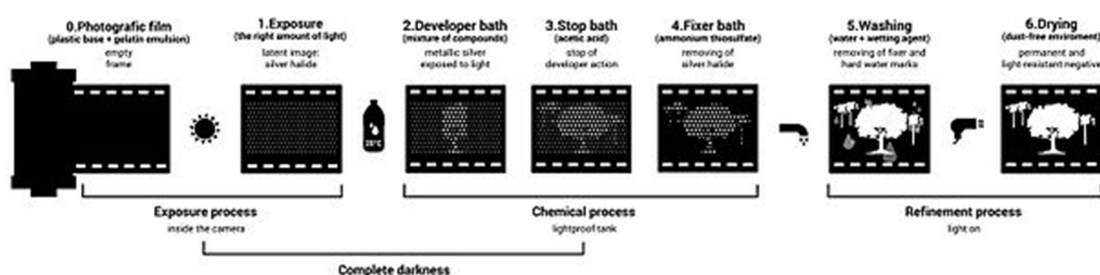
1 Rozdział z dziedziny pracy - część teoretyczna

1.1 Rys historyczny

Dziedzina tej pracy jaką jest przetwarzanie obrazów ma początki swojej historii wcześniej niż urządzenia które obecnie nazywamy komputerami.

1.1.1 Wywoływanie zdjęć

Przetwarzanie obrazów to nie jest technologia, która mogła zacząć istnieć po wynalezieniu komputerów. Używając aparatów korzystających z kliszy filmowej, po ekspozycji należy go poddać procesowi wywoływania Rys. 1. Polega on na wyciągnięciu pierwotnego efektu naświetlenia do obrazu, który oddaje scenę uchwyconą przez aparat [2]. Proces w przypadku zdjęć aparatem na kliszę polega na zanurzaniu jej w odpowiednich związkach chemicznych na określone ilości czasu by otrzymać zamierzony efekt.



Rys. 1: Wywoływanie białą czarnej kliszy

Istnieją wariacje na temat takiego przetwarzania, różni się ono trochę w zależności od technologii kliszy. W niektórych przypadkach należy wywołać pozytywny zamiast negatywu [3]. Następnie po wywoływaniu można poddać obróbkę dalszymi chemikaliami jak np. siarczek sodu dla uzyskania efektu sepii [4].

1.1.2 Cyfrowe przetwarzanie obrazów

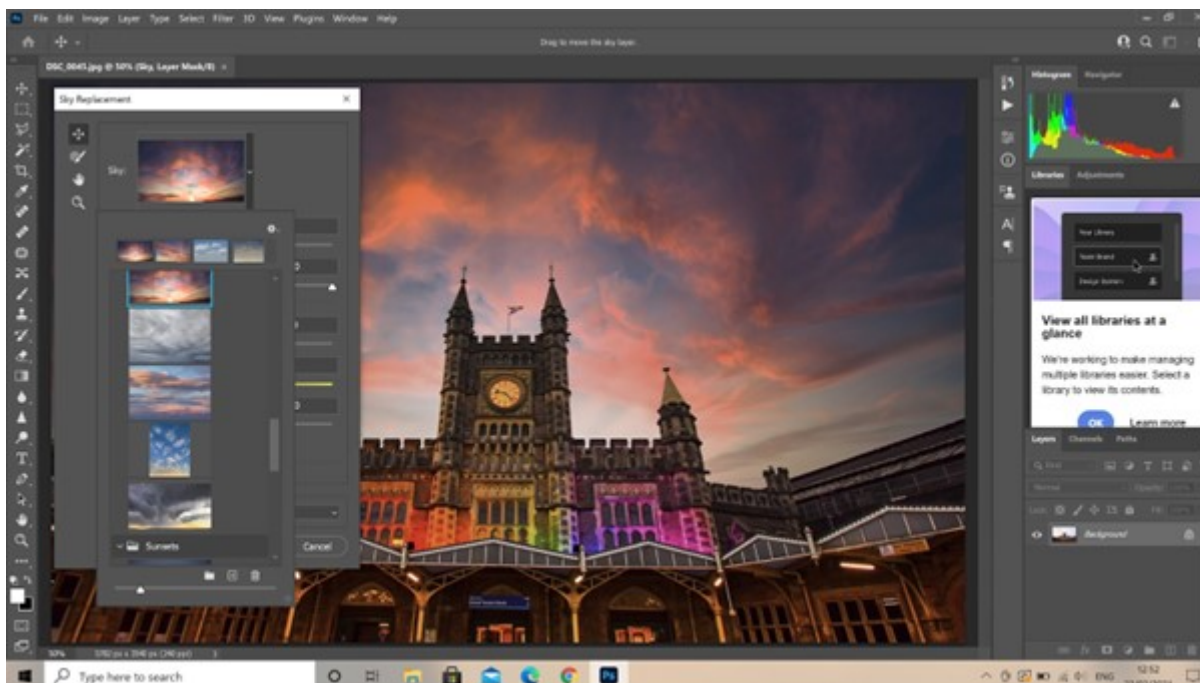
Początki nowoczesnego przetwarzania obrazów zostały stworzone w latach 60 XX w. w Bell Laboratories, Jet Propulsion Laboratory, Massachusetts Institute of Technology i University of Maryland [5]. Początkowe obszary zastosowania to obrazy satelitarne, przesyłanie obrazów przez linie telefoniczne, diagnostyka obrazowa, wideofony, rozpoznawanie znaków i ulepszanie fotografii.

Początkowo dużo skupiano się na ulepszeniu jakości obrazu. Pierwszym znaczącym użyciem tych technologii było mapowanie powierzchni księżyca za pomocą zdjęć z sondy kosmicznej w 1964 roku, gdzie naukowcy z Jet Propulsion Laboratory na podstawie tysięcy zdjęć odtworzyli powierzchnię księżyca. Przy następnej okazji mieli dostęp do 100000 zdjęć na podstawie których mogli stworzyć mapę topograficzną, mapę kolorową oraz panoramiczną mozaikę księżyca które przyczyniły się do pierwszego lądowania człowieka na księżycu [6].

Technologia ta jednak była ograniczona przez bardzo małe możliwości oraz trudną dostępność komputerów tamtych czasów. Wraz z ich rozwojem i wzrostem popularności coraz większą ilość operacji można było przeprowadzić w czasie rzeczywistym pozwalając np. na konwersję standardów telewizyjnych.

1.2 Przegląd istniejących rozwiązań

1.2.1 Photoshop

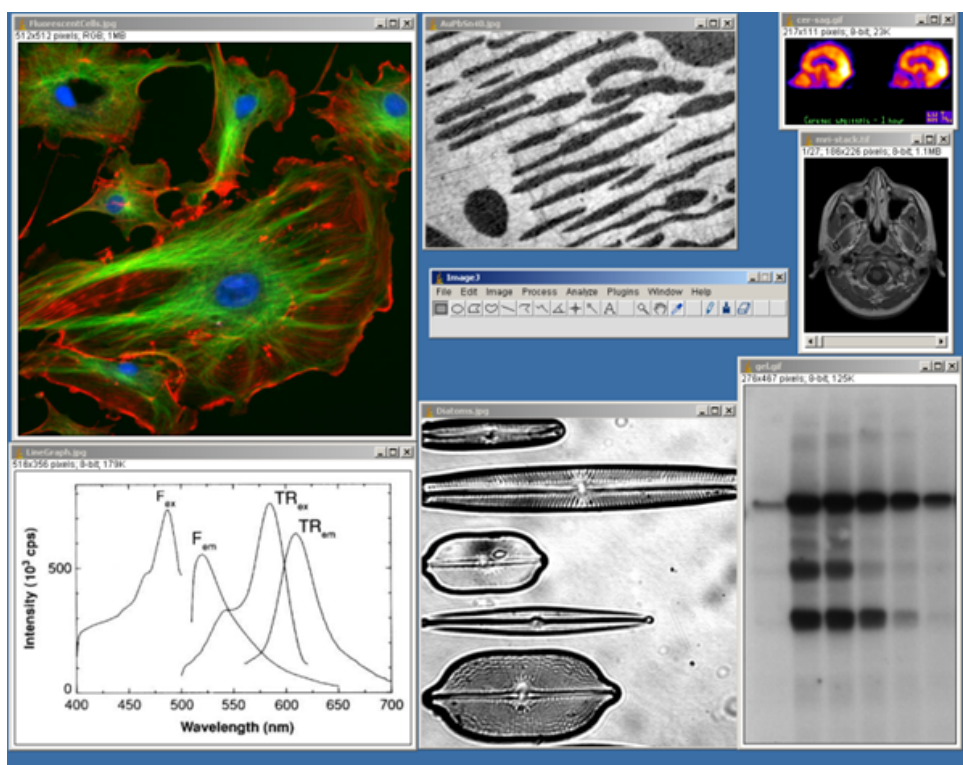


Rys. 2: Adobe Photoshop

Wydany w 1990 roku do tworzenia i edytowania obrazów rastrowych jest jednym z najbardziej popularnych programów, a jego nazwa przedarła się do języka potocznego jako zamiennik dla fotomontażu. Użytkownikami tego oprogramowania są przeważnie artyści, fotografowie i twórcy internetowych memów. Narzędzia są przystosowane do łatwości obsługi przez mniej zaawansowanych użytkowników i wszystko posiada przyjazny graficzny interfejs użytkownika.

Istnieją tam operacje, które pozwalają na przetwarzanie obrazów, ale często nie odnoszą się one bezpośrednio do operacji znanych z bibliotek do przetwarzania obrazów, ich obsługa jest uproszczona i przystosowana do bardziej artystycznych zastosowań. Bez zastosowania szczególnej ostrożności - jak tworzenie nowych warstw po każdej operacji lub częste zapisywanie kopii zapasowej obrazu – przetwarzanie obrazów często jest destruktywne, nie możemy odnieść się do dowolnego momentu w ciągu naszych operacji w celu dopasowania jej parametrów. Przetwarzanie innego obrazu za pomocą tego samego procesu jest problematyczne, gdy niektóre operacje są destruktywne.

1.2.2 ImageJ

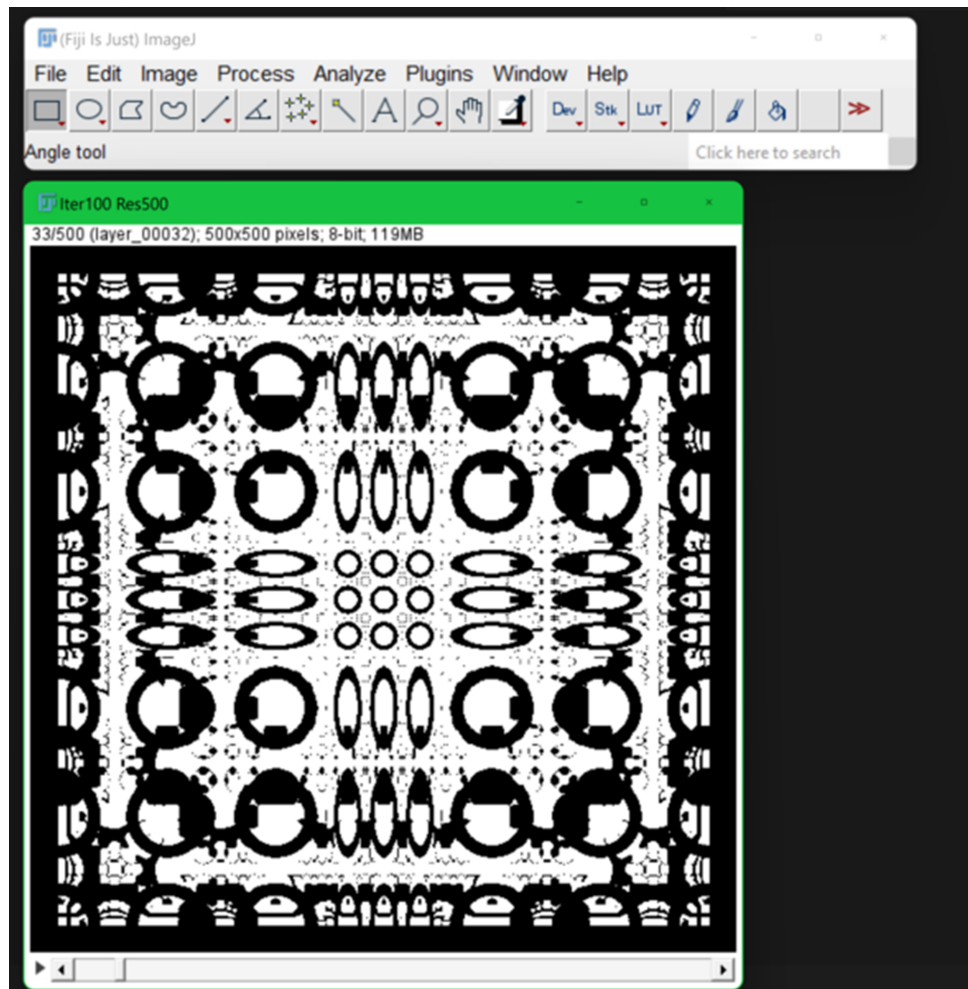


Rys. 3: ImageJ

Wydany w 1997 roku program ImageJ został wykonany przez National Institutes of Health, został tworzony podstawowo z zamiarem analizy obrazów w zastosowaniach medycznych. Odtwarza dokładnie wiele operacji z bibliotek do przetwarzania obrazów i kod aplikacji jest otwarty. Działa to w oparciu o Javę i ma bardzo niskie wymagania sprzętowe jak na obecne czasy.

Program jest jednak przestarzały, mogą pojawić się problemy z uruchomieniem na nowszych systemach. Interfejs użytkownika jest przestarzały oraz jego układ jest niepraktyczny biorąc pod uwagę przetwarzanie obrazów. Wszystkie operacje są schowane pod 1-2 poziomami menu. Operacje są destrukcyjne oraz nie da się przygotować ciągów operacji wcześniej. Od dawna nie jest rozwijany, został zastąpiony przez ImageJ2 który pozwala na przetwarzanie obrazów wielowymiarowych z dodatkowymi danymi np z mikroskopów elektronowych, skanerów itp.

1.2.3 Fiji

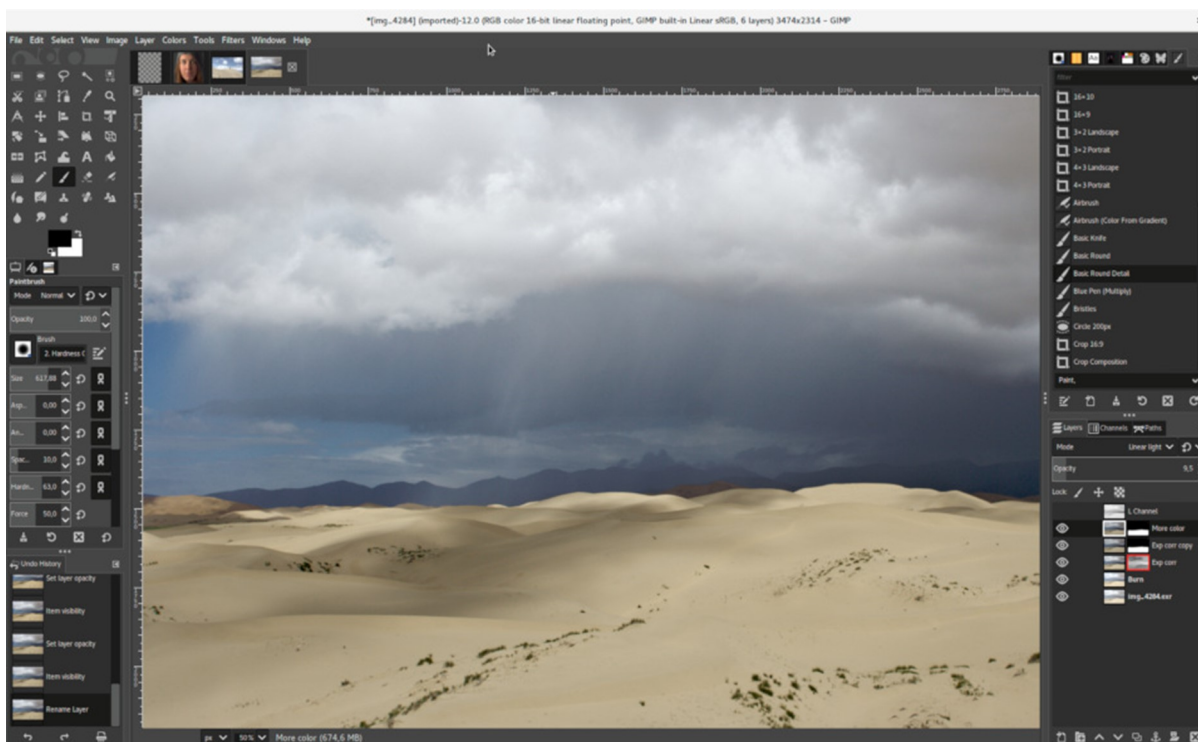


Rys. 4: Fiji - sukcesor ImageJ

Projekt open source oparty o ImageJ2. Oprócz podstawowych operacji wbudowanych w ImageJ posiada też wiele pluginów znacząco rozszerzających możliwości programu. Są one skupione na wspomaganiu przetwarzania obrazów skupionych na dziedzinie neurobiologii, ale możliwości są na tyle szerokie, że wiele innych dziedzin nauki z niego korzysta.

Podstawowe działanie programu nie różni się od ImageJ więc wszystkie jego problemy są też tutaj obecne. Poprawa istnieje tu pod względem uruchamiania tego oprogramowania na nowszych systemach.

1.2.4 GIMP

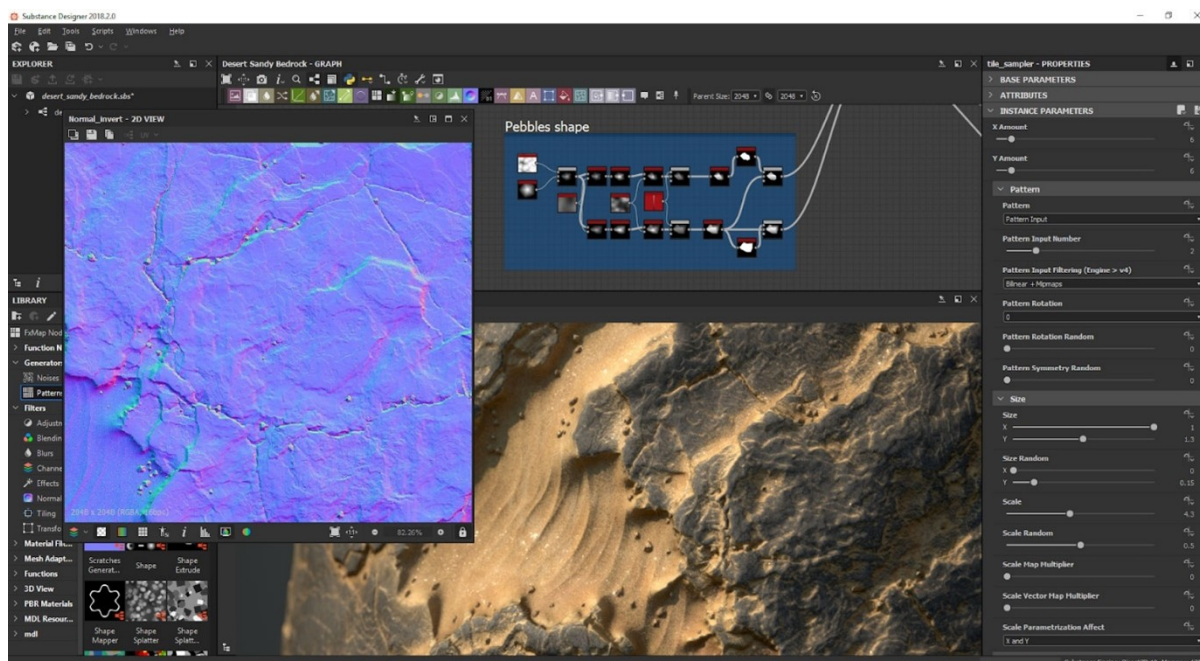


Rys. 5: GIMP wersja 2.10

Wydany w 1998 roku GNU Image Manipulation Program jest tworzony jako konkurencja dla Photoshopa. Program też w pierwszej kolejności jest tworzony z myślą o artystach, ale posiada więcej zaawansowanych opcji jak np. konwolucje macierzowe które można dowolnie edytować, dając duże możliwości filtrowania obrazów.

Pomimo odwzorowaniu większej ilości operacji ze standardowych bibliotek do przetwarzania obrazów i większej kontroli nad niektórymi z nich nadal występuje problem destruktywnego przetwarzania obrazów który wynika z pracy bezpośrednio na obrazie jak w Photoshopie i wymaga starań przy warstwach aby nie tracić bezpowrotnie swoich pośrednich operacji w dłuższym ciągu.

1.2.5 Adobe Substance Designer



Rys. 6: Adobe Substance Designer

Wydany w 2007 roku program był dla mnie główną inspiracją dla tego projektu. Potężną zaletą jest to, że operują na nodach – układamy z „błoczków” serię operacji, które są wykonywane na obrazie, który importujemy do programu lub generujemy go od początku w nim. Każdy dowolny node można kliknąć i zmienić wszystkie parametry jego operacji niezależnie w którym miejscu ciągu znajduje się, a jego wynik i wszystkie następne operacje zależne od niego są obliczane ponownie na podstawie zmienionego wyniku. Zmiana przetwarzanego obrazu polega na przeciągnięciu połączenia z obecnego node'a z naszym plikiem wejściowym na nowego node'a a wszystkie kolejne operacje zaktualizują się samodzielnie.

Program ten niestety ma wiele ograniczeń związanych z tym, że nie jest stworzony do ogólnego przetwarzania obrazów tylko do tworzenia materiałów/tekstur do grafiki komputerowej. Obrazy są ograniczone do boków o długości 2^n , najlepiej kwadratowych. Parametry operacji są często uproszczone, ponieważ mimo większej złożoności aplikacji, jest ona nadal skierowana do artystów. Rodzaje operacji i wspierane formaty pikseli w pliku są przystosowane do wymagań grafiki komputerowej.

2 Zakres użytych technologii i opis wykorzystywanych narzędzi

Zastosowane technologie:

- **Otwarty kod źródłowy:** OpenSource jest spoko

2.1 OpenCV

2.2 Visual Studio Code

2.3 Git

2.4 GitHub

3 Realizacja projektu

3.1 Zastosowania

- **Tworzenie gotowych ciągów operacji:** Czasem potrzebne jest przetworzenie małej ilości obrazów, lub problem jest na tyle mało skomplikowany, że nie ma sensu szukać programisty, który stworzy dla nas program, ale bez wiedzy o programowaniu zrobienie tego w kodzie samodzielnie może być zbyt ciężkie. Program z interfejsem graficznym pozwoli na stworzenie procesów nawet przez osoby mniej techniczne.
- **Uczenie się:** Osoby chcące poznać techniki przetwarzania obrazów będą mogły w prosty sposób bez znajomości programowania zobaczyć na własne oczy działanie funkcji, ich interakcję ze sobą oraz łatwo dopasowywać ich parametry.
- **Komunikacja:** Dzięki temu oprogramowaniu programista może łatwiej i efektywniej porozumieć się z osobami mniej zaznajomionymi z programowaniem i dużo szybciej wprowadzać zmiany w porównaniu z pisanem kodu i jego kompilacją przy każdej zmianie.

3.2 Wymagania funkcjonalne i нефункционалне

Oprogramowanie opisane w tej pracy to **NoodleCV**. Następne podrozdziały przedstawia wymagania z jakimi trzeba było się zmierzyć w trakcie tworzenia aplikacji.

3.2.1 Wymagania funkcjonalne

Wymagania funkcjonalne?

- Przetwarzanie obrazów

3.2.2 Wymagania нефункционалне

Нефункционалне wymagania?

- Łatwość użytkowania

4 Podsumowanie

Na koniec należy podsumować, co było celem pracy. Czy cel ten został osiągnięty. Jakie są możliwe wnioski. Czy są dalsze możliwości rozwoju.

Bibliografia

1. OpenCV. *OpenCV CPU optimizations* <https://github.com/opencv/opencv/wiki/CPU-optimizations-build-options>. (dostęp: 07.01.2024).
2. Keller, K. *i in. w Ullmann's Encyclopedia of Industrial Chemistry* (John Wiley & Sons, Ltd, 2000). ISBN: 9783527306732.
3. Dalladay, A. J. *The British Journal Photographic Almanac* 149–155 (Henry Greenwood & Co Ltd, London, 1956).
4. Blaker, A. *Photography: Art and Technique* ISBN: 9780716711162. <https://books.google.pl/books?id=bWdgQgAACAAJ> (W. H. Freeman, 1980).
5. Rosenfeld, A. Picture Processing by Computer. *ACM Comput. Surv.* **1**, 147–176. ISSN: 0360-0300. <https://doi.org/10.1145/356551.356554> (1969).
6. Gonzalez, R. C. & Woods, R. E. *Digital image processing* 3rd, 23–28. ISBN: 978-0-13-168728-8 (Prentice Hall, Upper Saddle River, N.J., 2008).

Spis rysunków

1	Wywoływanie białą czarnej kliszy	5
2	Adobe Photoshop	7
3	ImageJ	8
4	Fiji - sukcesor ImageJ	9
5	GIMP wersja 2.10	10
6	Adobe Substance Designer	11

Spis tabel

A. Załączniki