

# Aplikacja z graficznym interfejsem użytkownika do przetwarzania obrazów

Bartosz Dobija

Uniwersytet Bielsko-Bialski  
Wydział Budowy Maszyn i Informatyki

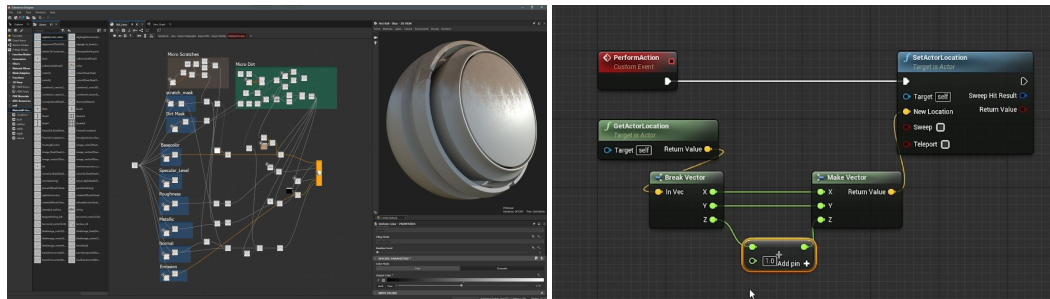
Obrona pracy inżynierskiej  
Bielsko-Biała, 5 lutego 2024

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

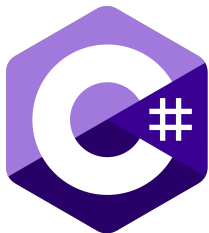
Wykonane prace:

- Analiza istniejących rozwiązań,
- Wybór technologii,
- Zaimplementowanie metod przetwarzania obrazów,
- Stworzenie interfejsu graficznego.

Aplikacja NoodleCV pozwala na obsługę części biblioteki OpenCV za pomocą edytora węzłowego.



Rysunek: Przykłady edytorów węzłowych.



Rysunek: Język programowania - C#.



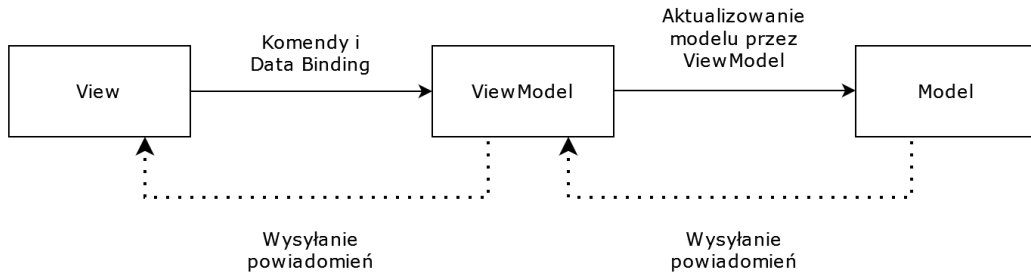
Rysunek: Biblioteka komponentów - WPFUI.



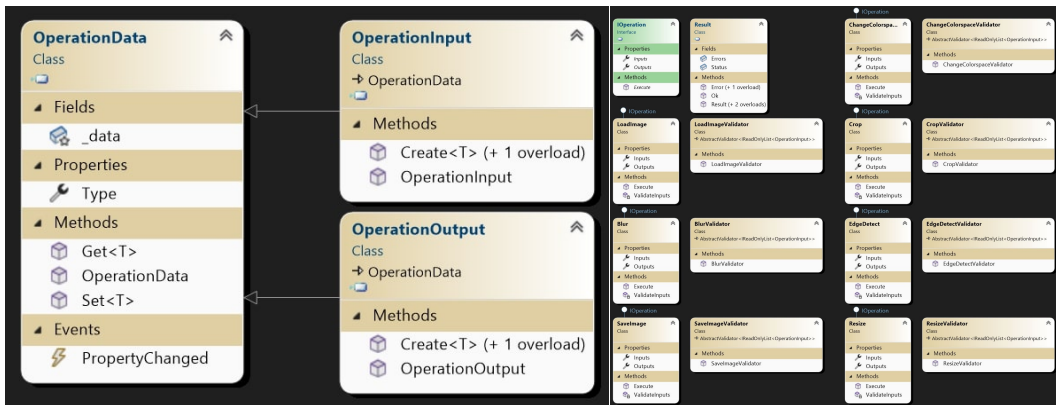
Rysunek: Edytor węzłowy - Nodify.



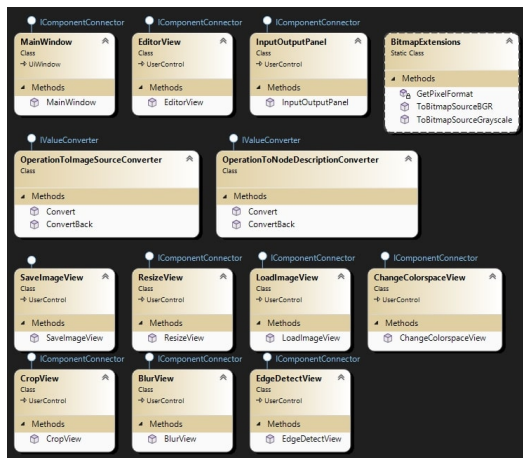
Rysunek: Przetwarzanie obrazów - OpenCV.



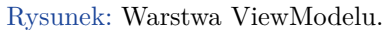
Rysunek: Architektura MVVM.



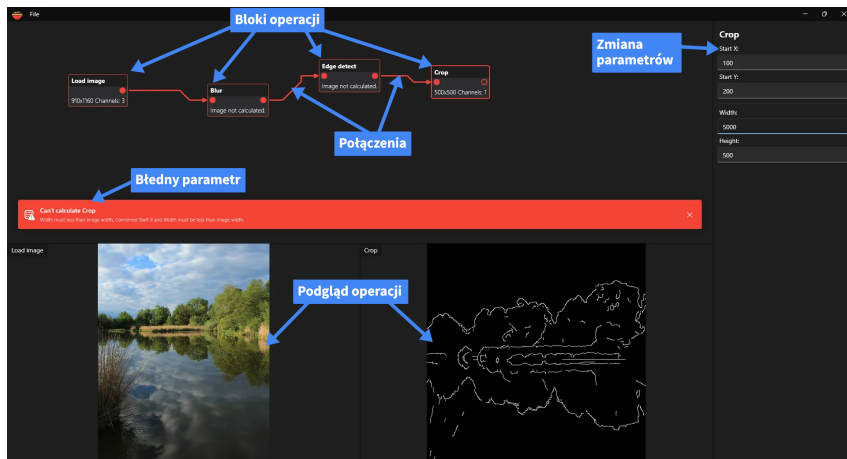
Rysunek: Warstwa Model.



Rysunek: Warstwa View.







Rysunek: Wygląd przykładowej operacji.

Prezentacja działania:

- Działanie programu
- Zapis i odczyt stanu
- Interfejs

Celem pracy było stworzenie aplikacji do przetwarzania obrazów z interfejsem graficznym. Interakcja użytkownika z programem odbywa się przez edytor węzłowy, a parametry odpowiadają tym używanym w bibliotece OpenCV. Wyniki obliczeń zostają wyświetlane natychmiastowo, a parametry które można wprowadzić są sprawdzane pod kątem poprawności. Cel został zrealizowany.

- Odpowiednia separacja logiki biznesowej i widoku pozwala na lepszą organizację kodu oraz na szybsze wprowadzanie zmian,
- Wymagane jest, aby wynik operacji przetwarzającej obraz był wyświetlany natychmiastowo. Końcowy użytkownik chce widzieć rezultat zmian w czasie rzeczywistym,
- Tworzenie narzędzi do rozwijania wymaga stworzenia przemyślanej architektury,
- Na rynku, mimo popularności edytorów węzłowych do przetwarzania tekstur, nie istnieją aplikacje ściśle związane z przetwarzaniem obrazów,
- Język C# pozwala na szybkie tworzenie multiplatformowych aplikacji. Wybór biblioteki OpenCV też umożliwił uruchomienie aplikacji na różnych systemach.

Pomysły z innych edytorów węzłowych:

- tworzenie i zapisywanie parametrów,
- łączenie kilku bloków w jeden przez użytkownika,
- grupowanie, układanie czy komentarzne,
- miniaturki z podglądem,
- eksportowanie pliku do edycji w zewnętrznych programach.

Rozbudowa wsparcia biblioteki OpenCV:

- zwiększenie liczby metod dostępnych w aplikacji,
- wsparcie dla innych rodzajów danych jak wideo, chmury punktów itd.,
- użycie funkcji uczenia maszynowego lub sieci neuronowych,
- wybór metody gdy więcej niż jedna realizuje ten sam cel.