

# Analiza Obrazów - dokumentacja projektu

Michał Kacprzak  
Paweł Korytowski  
Mikołaj Marchewa

13 stycznia 2021

## Spis treści

<b>1</b>	<b>Opis projektu</b>	<b>2</b>
<b>2</b>	<b>Założenia wstępne przyjęte w realizacji projektu</b>	<b>2</b>
<b>3</b>	<b>Wykorzystane technologie</b>	<b>2</b>
<b>4</b>	<b>Podział pracy</b>	<b>2</b>
<b>5</b>	<b>Analiza projektu</b>	<b>2</b>
5.1	Dataset . . . . .	2
5.2	GUI . . . . .	3
5.3	Sieć neuronowa . . . . .	5
<b>6</b>	<b>Testowanie</b>	<b>5</b>
<b>7</b>	<b>Podsumowanie</b>	<b>5</b>

# 1 Opis projektu

Projekt zakłada stworzenie aplikacji dzięki, której użytkownik będzie mógł dowiedzieć się czy wybranego przez niego zdjęcie zostało kiedykolwiek zmodyfikowane **poprzez wklejenie jakiegoś dodatkowego elementu???**

## 2 Założenia wstępne przyjęte w realizacji projektu

1. umożliwienie użytkownikowi wczytania zdjęcia w formacie PNG lub JPG o maksymalnym rozmiarze(**przetestować jakiś maksymalny rozmiar openCV/pamięć sieci**);
2. udostępnienie użytkownikowi możliwości edycji wczytanego zdjęcia;
3. Stworzenie datasetu, który będzie potrzebny do treningu sieci neuronowej;
4. **COS O SIECI NEURONOWEJ**
5. graficzne przedstawienie rezultatów otrzymanych dzięki wytrenowanej sieci neuronowej;

## 3 Wykorzystane technologie

Projekt został napisany w całości w Pythonie 3. Dodatkowo wykorzystaliśmy następujące biblioteki:

- PyQt5 - stworzenie prostego GUI aplikacji
- tensorflow - stworzenie sieci neuronowej
- Pillow - stworzenie datasetu, który był używany do treningu sieci neuronowej.

## 4 Podział pracy

Dokonany został następujący podział zadań:

- Michał Kacprzak - stworzenie graficznej strony GUI za pomocą Qt Designera, implementacja funkcjonalności głównego menu oraz (**JAK NAZWAĆ TO PIERWSZE OKNO**), stworzenie datasetu używanego do treningu sieci neuronowej, stworzenie dokumentacji,
- Paweł Korytowski - stworzenie sieci neuronowej, stworzenie dokumentacji
- Mikołaj Marchewa - implementacja funkcjonalności w edytorze wczytanego obrazu oraz w **JAKIES LADNE OKRESLENIE NA WYNIKOWE OKNO**, stworzenie dokumentacji

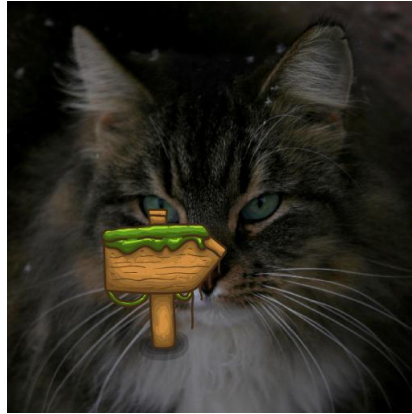
## 5 Analiza projektu

### 5.1 Dataset

Realizowanie projektu rozpoczęliśmy od stworzenia datasetu, który będzie użyty do treningu sieci neuronowej. Bazą naszego datasetu został **zbiór obrazów dostępny na stronie kaggle**. Obrazy z tego zbioru zgrupowaliśmy w jeden folder, ponieważ w treningu sieci miały być sklasyfikowane jako autentyczne. Następnie za pomocą biblioteki Pillow stworzyliśmy przerobioną kopię każdego obrazu. Dla każdego obrazu losowaliśmy miejsce modyfikacji oraz jaki obrazek zostanie wklejony. Przykładowy rezultat modyfikacji obrazu znajduje się poniżej.



Rysunek 1: Oryginalny obraz



Rysunek 2: zmodyfikowany obraz



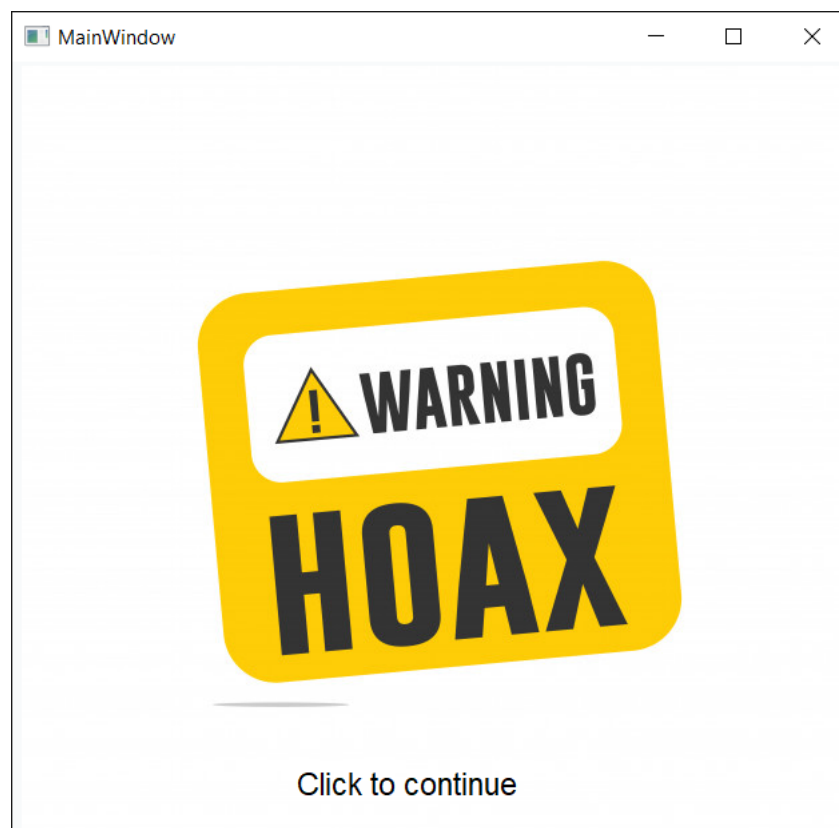
Rysunek 3: maska użyta do modyfikacji obrazu

Obrazy, które zostały sztucznie zmodyfikowane posłużyły do treningu sieci neuronowej, ponieważ zostały sklasyfikowane jako fałszyfikaty.

## 5.2 GUI

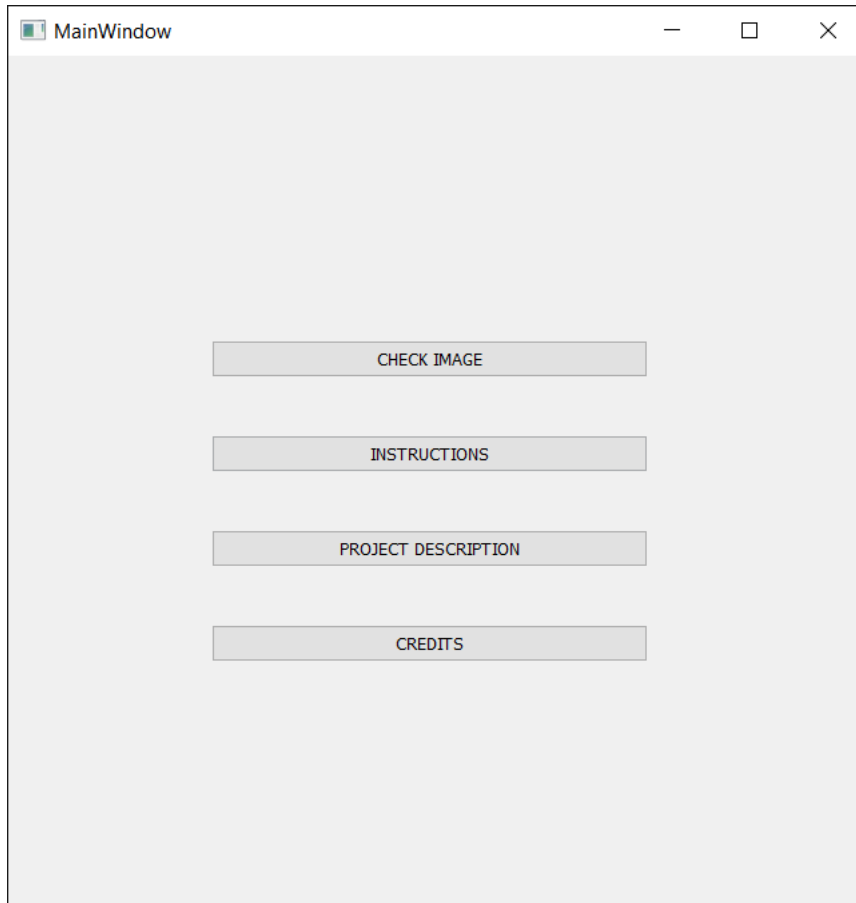
GUI naszej aplikacji zostało stworzone za pomocą frameworka Qt a dokładniej biblioteki PyQt, która umożliwia tworzenie aplikacji w Pythonie za pomocą tego frameworka. Layout poszczególnych okien został stworzony za pomocą Qt Designera. Następnie została zaimplementowana warstwa logiczna aplikacji, czyli np przełączanie pomiędzy poszczególnymi oknami aplikacjami. W naszej aplikacji można wyróżnić następujące okna:

1. IntroWindow - jest to okno, które użytkownik widzi jako pierwsze w momencie gdy uruchamia aplikację. Zawiera tylko logo aplikacji. Jeśli użytkownik chce przejść dalej musi kliknąć na okno, wtedy zostanie przekierowany do StartWindow.



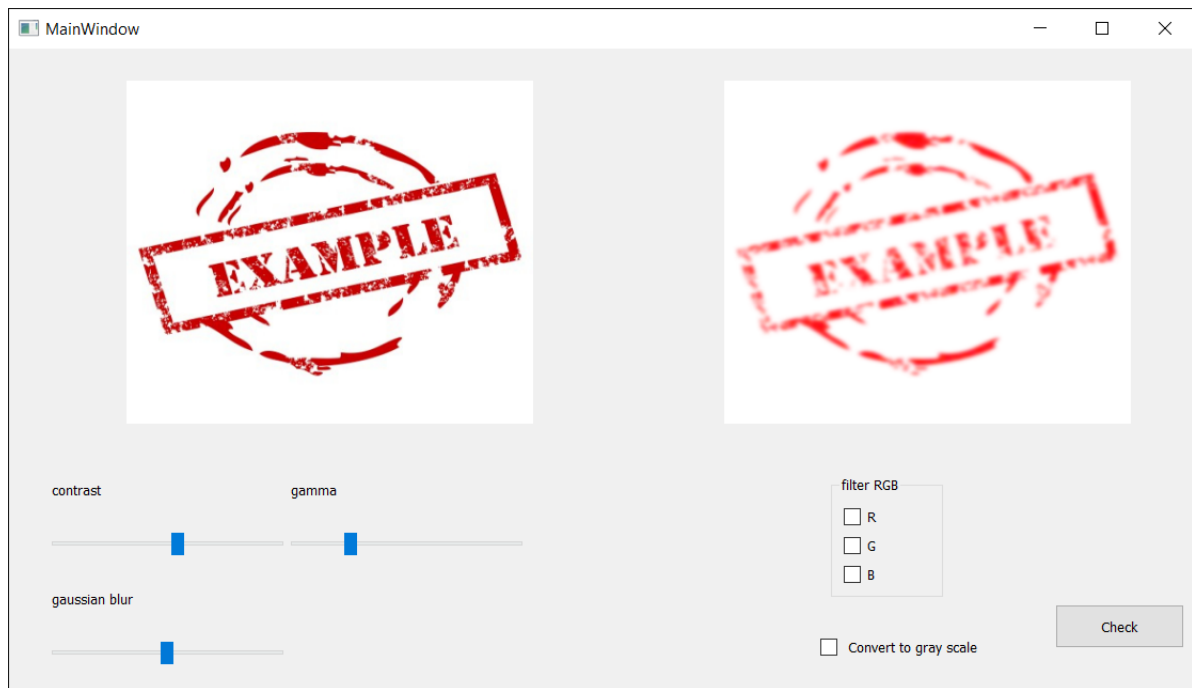
Rysunek 4: Wygląd okna IntroWindow

2. StartWindow - jest to główne okno aplikacji. Udostępnia następujące funkcjonalności użytkownikowi:
- (a) sprawdzenie zdjęcia - główna funkcjonalność programu. Po kliknięciu na przycisk "CHECK IMAGE" użytkownik zostanie poproszony o wybranie dowolnego zdjęcia które aktualnie znajduje się na komputerze użytkownika a następnie zostanie przeniesiony do ImageEditorWindow.
  - (b) przeczytanie dokumentacji projektu
  - (c) instrukcje
  - (d) credits



Rysunek 5: Wygląd okna StartWindow

3. ImageEditorWindow - okno, w którym użytkownik może modyfikować wybrane wcześniej przez siebie zdjęcie. Użytkownikowi może modyfikować zdjęcie w następujący sposób:
- (a) zmiana kontrastu zdjęcia;
  - (b) zmiana parametru gamma zdjęcia;
  - (c) nałożenie filtru Gaussa na zdjęcie;
  - (d) przekonwertowanie zdjęcia do skali szarości
  - (e) wyzerowanie poszczególnych kanałów RGB (jak to napisać ?????)



Rysunek 6: Wygląd okna ImageEditorWindow wraz z zastosowanymi pewnymi modyfikacjami(po lewej oryginalny obraz, po prawej zmodyfikowany)

4. ResultWindow - okno, w którym użytkownikowi zostają przedstawione wyniki działania sieci neuronowej, która sprawdza czy i z jaką pewnością dany obraz był modyfikowany poprzez wklejenie dodatkowego obiektu do zdjęcia(**ładniej opisać**) Udostępnia następujące funkcjonalności użytkownikowi:
  - (a) powrót do MenuWindow - poprzez kliknięcie przycisku (**Dokończyć jak Mikołaj skończy**)
  - (b) powrót do ImageEditorWindow - poprzez kliknięcie przycisku (**Dokończyć jak Mikołaj skończy**)

### 5.3 Sieć neuronowa

Paweł

## 6 Testowanie

Testowanie aplikacji przebiegało równoległe wraz z implementacją kolejnych funkcjonalności. Takie postępowanie zagwarantowało nam bieżące wykrywanie i usuwanie błędów co pozwoliło uniknąć nakładania się błędów na siebie. W rezultacie finalna wersja programu jest w naszej ocenie pozbawiona błędów. Działanie sieci było testowane na własnoręcznie stworzonym zbiorze obrazów. W folderze "**tu scieżka**" znajdują się zdjęcia, które są nie modyfikowane i za każdym razem zostaną uznane przez aplikację za niemodyfikowane. Natomiast w folderze "**tu scieżka**" znajdują się zdjęcia modyfikowane i każdorazowo przy ich sprawdzeniu wynik będzie potwierdzał że te zdjęcia były modyfikowane.

## 7 Podsumowanie

Efektom końcowym realizacji projektu jest aplikacja, w której użytkownik ma możliwość wczytania wybranego przez siebie zdjęcia, jego modyfikacji oraz sprawdzenia czy obraz był modyfikowany poprzez wklejenie elementu. W ramach dalszego rozwijania projektu należałoby na pewno poprawić całościowy wygląd aplikacji, aby uczynić ją atrakcyjniejszą dla użytkownika. Dodatkową funkcjonalnością którą w następnej kolejności należałoby zaimplementować z pewnością jest wykrywanie obszarów zdjęcia w których doszło do modyfikacji i ich zaznaczanie. Reasumując aplikacja spełnia wszystkie konieczne wymogi i implementuje wszystkie wstępne założenia projektu.