

wspolne_dla_wszystkich/logo_uczelni.png

Poprawa jakości skanów zdjęć wykonanych techniką analogową

Raport II

projekt realizowany pod opieką prof. dr hab. inż. Artura Przelaskowskiego

wspolne_dla_wszystkich/logo_projektu.png

Streszczenie

Raport 2 projektu poprawy jakości cyfrowych skanów zdjęć wykonanych techniką analogową przez grupę nr 9 (wtorkową z godziny 18) w składzie: Bartosz Wójcik, Katarzyna Szwed, Natalia Szymańska, Patrycja Szałajko, Aleksandra Wójcik, Karol Sęk, Michał Juskiewicz, Filip Sajko.

W tym raporcie zdefiniujemy cel naszego projektu i opiszemy problem z którym się mierzymy. Przedstawimy ponadto wstępną wersję naszego programu i zademonstrujemy jego skuteczność.

Spis treści

1. Korekta do raportu 1	3
2. Cel projektu	3
3. Zdjęcia, zdjęcia!	3
4. Problemy	4
4.1. Niedoświetlenie	4
4.2. Artefakty	7
5. Program i jego działanie	9
5.1. Tworzenie maski	9
5.1.1. Rozmycie gaussowskie	9
5.1.2. Wykonanie różnicy	10
5.1.3. Usunięcie ciemnych pikseli	10
5.1.4. Korekcja gamma	11
5.2. Gotowa maska	12
5.3. Działanie właściwe	13
6. Uwagi co do działania programu	14
7. Program do zmiany kontrastu zdjęcia	14
7.1. Kod	14
7.2. Działanie programu ze względu na parametr gamma	15
8. Dalsze kroki	15
9. Wykorzystywane narzędzia	16
10. Podział obowiązków	16

1. Korekta do raportu 1

- Do skanowania zdjęć w pierwszym etapie projektu użyty był skaner minilab Noritsu HS-1800.
- Zdjęcia były robione przy różnych ustawieniach Exposure Value, za zdjęcia niedoświetlone uznaliśmy te robione przy EV-1, a za prześwietlone przy EV+1.
- Również doprecyzowaliśmy tytuł projektu.


2. Cel projektu

W związku ze słusznymi uwagami i wskazówkami, podjęliśmy decyzję o ukonkretyzowaniu celu naszego projektu. Skupimy się przede wszystkim na poprawianiu defektów cyfrowych skanów zdjęć analogowych. Na ten moment pracowaliśmy nad usuwaniem artefaktów powstałych w procesie wywoływania zdjęć i naprawianiem kontrastu zdjęć. Naszym celem jest zarówno naprawienie jakości starych zdjęć rodzinnych, jak i tych robionych przez współczesnych amatorów fotografii analogowej.

3. Zdjęcia, zdjęcia!

Profilowym zdjęciem dla nas jest portret – tak indywidualny jak i grupowy. Jest to typ zdjęć najbardziej popularny w rodzinnych albumach – mnogość w nich zdjęć z ważnych dla danej rodziny wydarzeń: chrztów, wesel czy pogrzebów... Służą one utrwaleniu wspomnień oraz pamięci po krewnych i bliskich, którzy już odeszli... A więc noszących dużą wartość emocjonalną dla ich posiadacza.

Nasza koleżanka Ola odnalazła stary album rodzinny i zeskanowała znajdujące się w nim zdjęcia. Docelowo zajmiemy się naprawianiem skanów właśnie tych zdjęć. Często pojawiającymi się problemami wśród tych zdjęć są niedoświetlenie oraz zagięcia.



Photos2/STARE/Scan2025-04-14_101906.png

Rysunek 1: Przykładowe zdjęcie z albumu, który można znaleźć tutaj!

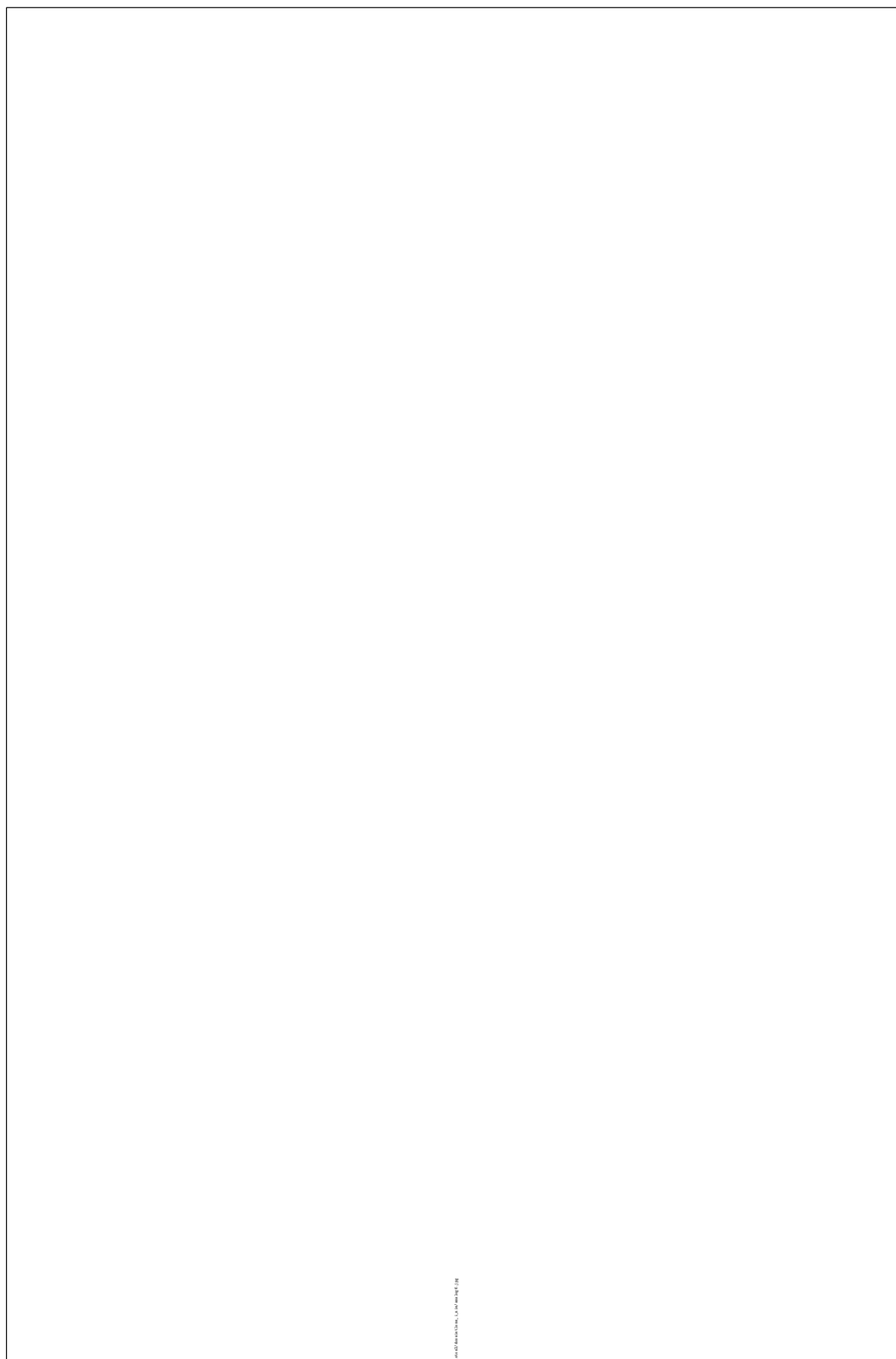
4. Problemy

Wykonywanie, a następnie ‘ucyfrowienie’ zdjęcia w technice analogowej wiąże się z różnymi trudnościami, które mogą znacząco obniżyć jakość zdjęcia – a z tym satysfakcję jego posiadacza. Głównymi problemami, którym będziemy przeciwdziałać, będą niedoświetlenie zdjęcia i zanieczyszczenia powietrza osadzające się na oryginalnym zdjęciu i skanerze podczas procesu zmiany informacji z analogowej na cyfrową.

4.1. Niedoświetlenie

Niedoświetlenie jest problemem trudnym – zwłaszcza dla fanów-amatorów techniki analogowej. Zasadnicza większość klasycznych aparatów nie posiada zaawansowanej mechaniki automatycznie wybierającej odpowiednie ustawienia aparatu, a brak możliwości podglądu tego, jak dane zdjęcie wyszło, często doprowadza do sytuacji, gdzie po wielu dniach okazuje się, że na zdjęciu chwili, którą fotograf chciał uchwycić i utrwalić, niewiele widać, bo przez złe ustawienia większość szczegółów jest niewidoczna. Jest to problem, który szeroko wraz z przykładami i analizą numeryczną opisywaliśmy w raporcie pierwszym.

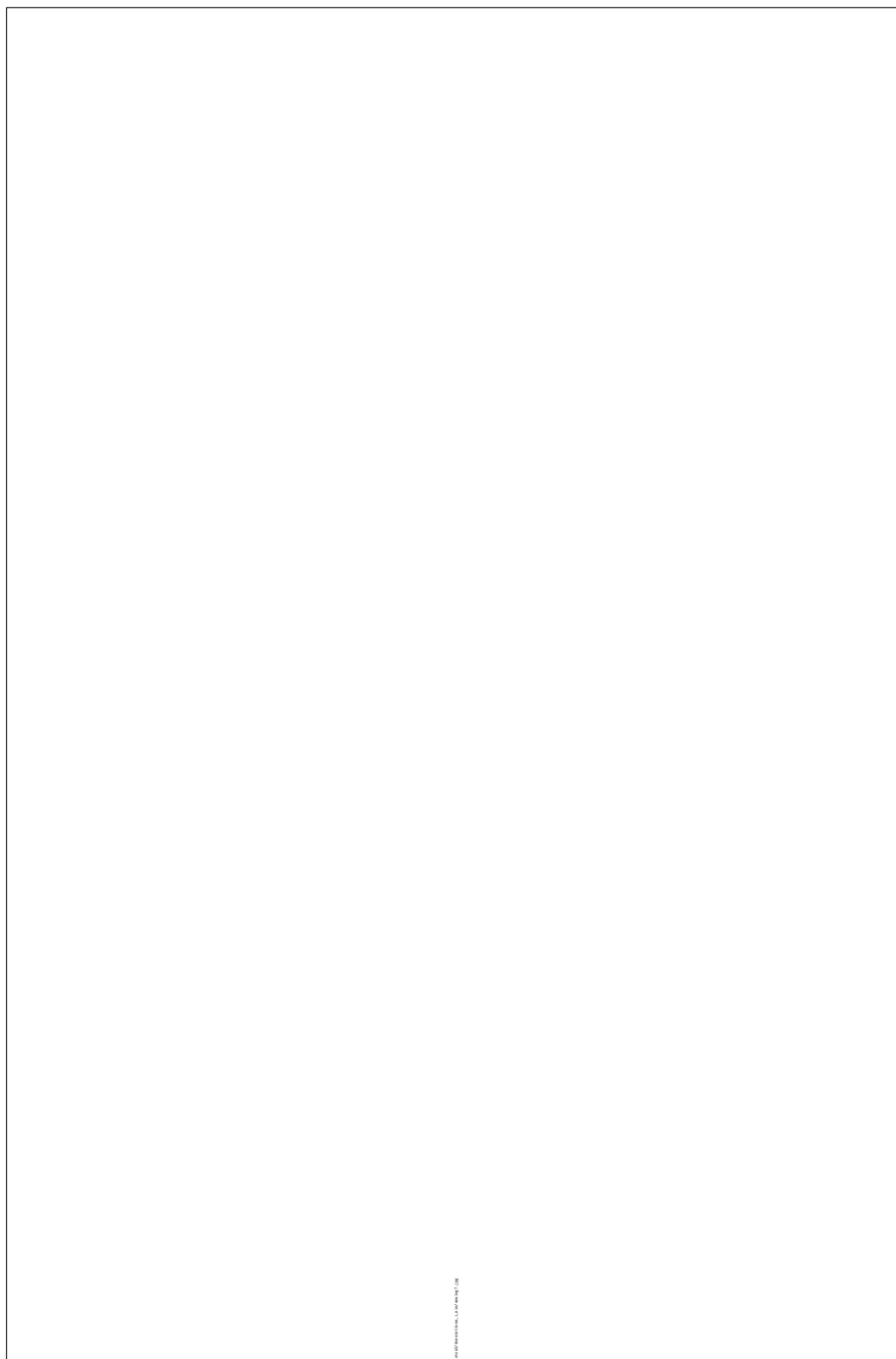
Dla przykładu przypomnijmy:



Rysunek 2: Zdjęcie niedoświetlone.

wspolne_dla_wszystkich/logo_projektu.png
Poprawa jakości zdjęć

wspolne_dla_wszystkich/logo_uczelni.png



Rysunek 3: ...i to w punkt.

4.2. Artefakty

Artefakty na zdjęciach analogowych powstają ze względu na osadzanie się kurzu i innych zanieczyszczeń na wywołanej kliszy, a także przez jej zarysowanie.

Przez to na negatywie powstają ciemne plamki, które w momencie powstawania odbitki z kolei zamieniają się w jasne plamy.

Takie zjawisko można zaobserwować na poniższych zdjęciach:



Rysunek 4: W skrajnych wypadkach może wyglądać to nawet tak.



Rysunek 5: Choć bardziej częstym jest ten przypadek.

wspolne_dla_wszystkich/logo_projektu.png
Poprawa jakości zdjęć

wspolne_dla_wszystkich/logo_uczelni.png

Photos2/przed/new2.jpeg

Rysunek 6: A także taki.

5. Program i jego działanie

Zbrojni w wiedzę co chcemy osiągnąć i zapas zebranych skanów zdjęć do testów wzięliśmy się do pracy nad programem. Stworzyliśmy zaawansowany program, który inteligentnie przeszukuje cały obszar zdjęcia, zaznacza artefakty, a w następnej fazie działania usuwa je. Warto dodać, że algorytm przekształca zdjęcie z formatu RGB do formatu HSV (Hue, Saturation, Value) i działa tylko na Value, która definiuje jasność piksela w skali 0-255.

Przechodząc do szczegółów, działanie programu można opisać za pomocą kilku kolejnych faz działania, na przykładzie poniższego zdjęcia:



Rysunek 7: Przykładowe zdjęcie – to za jego pomocą opiszemy działanie programu.

5.1. Tworzenie maski

5.1.1. Rozmycie gaussowskie

Pierwszym krokiem generowania maski jest stworzenie rozmycia gaussowskiego zdjęcia.

Wygładzanie gaussowskie jest efektem rozmywania obrazu za pomocą funkcji Gaussa, która jest szeroko wykorzystywana w grafice komputerowej w celu uzyskania gładkiego wygładzenia obrazu i wyciszenia szumu informacyjnego.

Za pomocą funkcji danej wzorem dla każdego piksela:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

gdzie x, y to współrzędne danego piksel'a, a σ oznacza odchylenie standardowe.¹

¹ Źródło: https://en.wikipedia.org/wiki/Gaussian_blur

Photos2/gauss_blurr/gauss_blurr_gpt1.png

Rysunek 8: Zdjęcie po wykonaniu rozmycia gaussowskiego.

5.1.2. Wykonanie różnicy

Następnie odejmujemy od oryginału zdjęcia otrzymane w poprzednim etapie rozmycia. Pozwala to na wykrycie najbardziej kontrastowych elementów zdjęcia.

Photos2/difference/difference_gpt1.png

Rysunek 9: Zdjęcie po wykonaniu różnicy.

5.1.3. Usunięcie ciemnych pikseli

Zostawiamy tylko jasne piksele (według standardowych ustawień jest to jasność powyżej 60) i nadajemy im maksymalną wartość 255. Pozostałym pikselom ustawiamy jasność na 0.

5.1.4. Korekcja gamma

Korekcja gamma jest techniką stosowaną w grafice komputerowej, której celem jest dostosowanie jasności obrazu do ludzkiego nielinowego postrzegania światła. Korekcja gamma kompensuje ten nieliniowy sposób widzenia, pozwalając na efektywniejsze wykorzystanie dostępnych poziomów jasności.

Korekcja gamma dokonuje transformacji jasności w następujący sposób:

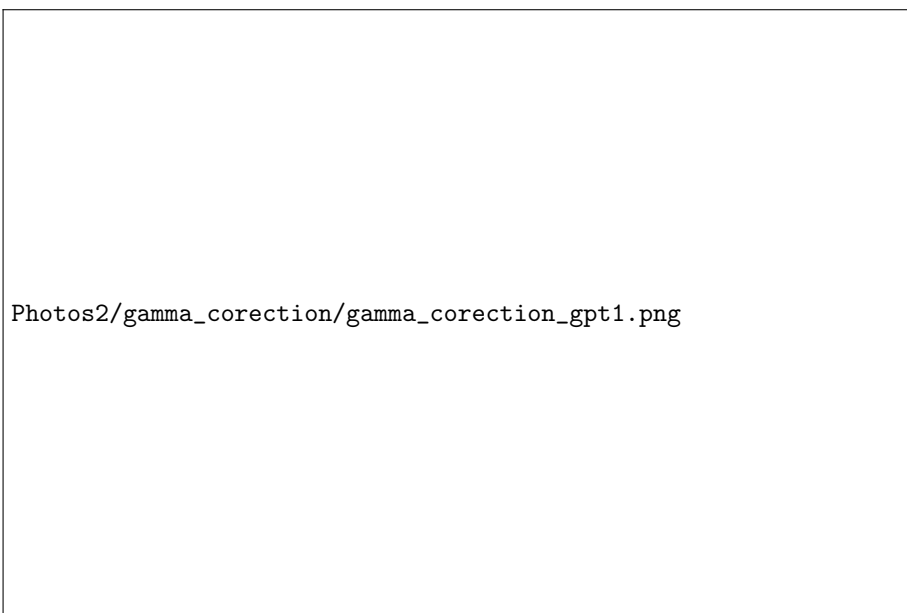
$$L'(x, y) = L(x, y)^\gamma \quad (2)$$

gdzie:

- γ to współczynnik gamma,
- $L(x, y)$ to wartości jasności pikseli.

Wiedząc to, w tym etapie bierzemy ponownie oryginalne zdjęcie, wykonujemy na nim korekcję gamma, a następnie na tym zdjęciu wykonujemy etapy 1-3.

Działanie to pozwala uwzględnić jak najwięcej artefaktów – znajdujących się także na jasnym tle.



Rysunek 10: Po wykonaniu korekcji gamma i etapów 1-3.

5.2. Gotowa maska

Po tym wszystkim otrzymujemy dwie podmaski (jedna robiona na oryginalnym zdjęciu, a druga na jaśniejszym – rozświetloną modulacją gamma). Końcową maskę otrzymujemy biorąc wszystkie znalezione piksele z obu podmasek.



Rysunek 11: Maska wykonana z oganianego zdjęcia



Rysunek 12: Maska wykonana z zdjęcia rozświetlonego korekcją gamma

5.3. Działanie właściwe

Mając wygenerowaną maskę właściwą, przechodzimy po wyznaczonych przez nią pikselach na oryginalnym zdjęciu. Dla każdego piksela w zaznaczonego w masce wyliczamy nową wartość jasności – średnią z jasności wszystkich (nie zaliczają się do tego pikseli wyznaczone wcześniej przez maskę.) pikseli na odległości nie więcej 15 od aktualnie analizowanego.

Wynik finalny:



Rysunek 13: Przykładowe zdjęcie – porównanie efektu przed i po.

6. Uwagi co do działania programu

Przez to, że algorytm działa lokalnie (tylko w punktach wyznaczonych przez maskę) nie wpływa on na ogólny wygląd zdjęcia. Z tego też powodu możemy wykonywać program kilkakrotnie na zdjęciu – uzyskując lepsze efekty.

Końcowy algorytm składa się z pięciu iteracji opisanego wyżej algorytmu, znacząco zwiększając szansę na usunięcie zanieczyszczeń.

Ponadto można uzyskać dodatkowe informacje o działaniu programu i jakości zdjęcia – za przykład funkcja: `.countNoise()` zlicza ilość wyznaczonych przez maskę pikseli – które są uznane za zanieczyszczenia.

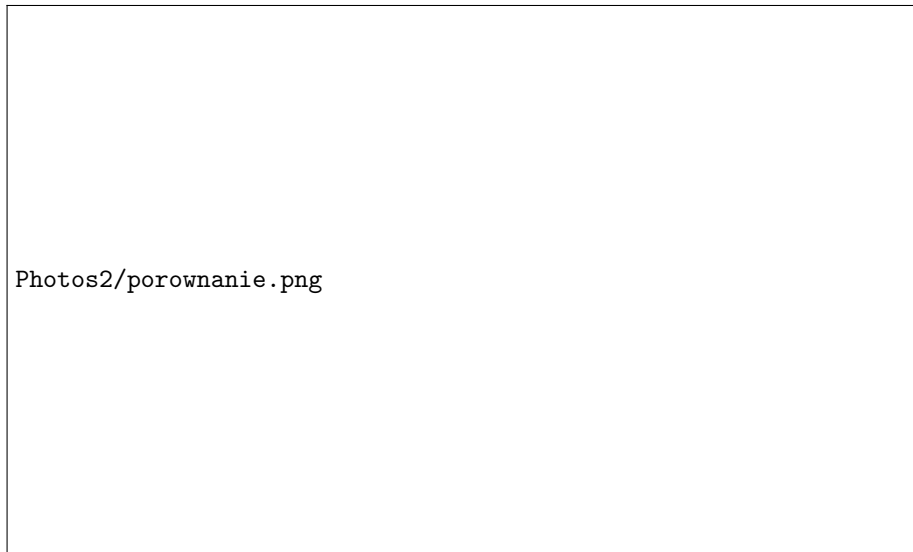
7. Program do zmiany kontrastu zdjęcia

Napisaliśmy też osobny program zmieniający kontrast obrazu. Użyliśmy języka Matlab ze względu na to, że umożliwia on łatwiejsze porównanie wyników ze względu na parametr gamma. Algorytm korzysta z korekcji gamma opisaną wcześniej w punkcie 4.1.4.

7.1. Kod

```
g = imread("test_2.png");
figure(1);
tiledlayout(3,3)
nexttile
imshow(g);
title("zdjęcie niezmienione")
gamma = 0.25;
for k=1:9
    if gamma ~= 1
        gout = double(g)/255;
        gout = gout.^gamma;
        gout = gout*255;
        gout = uint8(gout);
        nexttile
        imshow(gout);
        title(['gamma = ' num2str(gamma)]);
    end
    gamma = gamma + 0.25;
end
```

7.2. Działanie programu ze względu na parametr gamma



Rysunek 14: Działanie programu dla różnych parametrów γ

8. Dalsze kroki

Na chwilę obecną nasze rozwiązanie jest programem terminalowym, działającym na systemie nie starszym niż Windows 10 – choć istnieje plan przeniesienia go także na inne popularne systemy operacyjne.

Tak samo pracujemy obecnie nad stworzeniem bardziej przystępnego interfejsu okienkowego, a przy tym planujemy integrację poszczególnych elementów programu. Zamierzamy również między innymi dodać funkcjonalność usuwającą widoczne zagięcia.

Program dostępny jest na licencji *open source* i jego kod źródłowy można znaleźć na GitHubie pod adresem:

<https://github.com/ssk12o/PTI-Foto-Projekt>.

9. Wykorzystywane narzędzia

W tej części naszego projektu korzystaliśmy z następujących narzędzi:

- Programu i języka Matlab – do analizy zdjęć i kontrastu;
- Języka C++ – do napisania programu usuwającego artefakty;
- Programu VS Code – do tworzenia, edycji i dokumentacji kodu programu i raportów;
- Programu LibreOffice Calc – do analizy części danych numerycznych;
- \LaTeX 2 ϵ – do przygotowania raportu;
- Strony Github i programu Git – do udostępniania, dystrybucji i pracy nad kodem;
- 7zip – do kompresji zdjęć;
- Google Drive – do udostępniania plików;
- Skanera minilab Noritsu HS-1800 – do wykonywania wysokiej jakości cyfrowych skanów zdjęć wykonanych techniką analogową;
- Aparatów:
 - Canon EOS 300 z obiektywem Tamron 28-105mm 1:4-5.6 i kliszą Fomapan 400
 - Fujifilm FinePix L55 Digital Camera – Black (12MP, 3x Optical Zoom)

10. Podział obowiązków

Na tym etapie projektu podzieliśmy się pracą, obowiązkami i zadaniami w następujący sposób:

- Bartosz Wójcik – wykonywanie, skanowanie i analiza zdjęć; research.
- Katarzyna Szwed – korekta raportu; tworzenie, analizowanie i pisanie algorytmu.
- Natalia Szymańska – pisanie raportu.
- Patrycja Szałajko – zarządzanie pracą zespołu, kontakt.
- Aleksandra Wójcik – skanowanie zdjęć rodzinnych w celu polepszenia ich jakości w końcowych etapach projektu.
- Karol Sęk – tworzenie, analizowanie i pisanie algorytmu.
- Michał Juskiewicz – tworzenie, analizowanie i pisanie algorytmu.
- Filip Sajko – pisanie raportu, implementacja w \LaTeX .