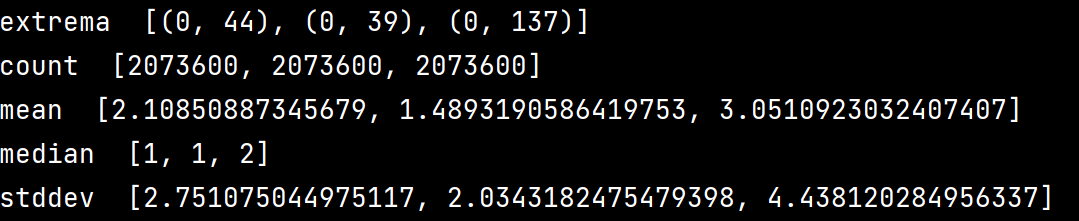


Pomimo, że na pierwszy rzut oka nie widać, to efekt porównania dwóch tych obrazów w różnych formatach (jpg, bmp), nie jest czarny. W takim przypadku wystarczy tylko przybliżyć, i widać różnice, jednak nie zawsze się tak da, czasem różnice są tak małe, że warto zastosować inne metody. Poniżej widzimy statystyki powyższego obrazu.



Extrema (wartości maksymalne i minimalne):

* Extrema to zakres wartości pikseli w obrazie RGB w poszczególnych kanałach. Na przykład, w pierwszym kanale (czerwonym) wartości wahają się od 0 (najniższa jasność) do 44 (najwyższa jasność). W drugim kanale (zielonym) wartości wynoszą od 0 do 39, a w trzecim kanale (niebieskim) od 0 do 137.

Count (ilość pikseli):

* Liczba pikseli w każdym z kanałów wynosi 2073600. To jest całkowita liczba pikseli w obrazie w danym kanale.

Mean (średnia):

* Średnia wartość pikseli w każdym z kanałów wynosi około 2.1 (w pierwszym kanale), 1.49 (w drugim kanale) i 3.05 (w trzecim kanale). To daje ogólny pogląd na jasność kanałów, gdzie wyższa średnia oznacza jaśniejszy kanał.

Median (mediana):

* Mediana w pierwszym i drugim kanale wynosi 1, a w trzecim kanale wynosi 2. Mediana jest wartością, która dzieli dane na pół, więc to informacja o centralnej jasności kanałów.

Stddev (odchylenie standardowe):

* Odchylenie standardowe wskazuje, jak rozproszone są wartości pikseli w danym kanale. Wartości odchylenia standardowego wskazują na to, że piksele w pierwszym i drugim kanale są mniej rozproszone (blisko średniej), podczas gdy w trzecim kanale są bardziej rozproszone (większa zmienność).

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, diagram

Opis wygenerowany automatycznie

Histogram jest graficznym narzędziem do analizy rozkładu danych, które pomaga zrozumieć, jak często dane występują w określonych przedziałach. W kontekście przetwarzania obrazów, histogram jest często używany do analizy rozkładu jasności lub kolorów pikseli na obrazie. Powyżej mamy histogram stworzony z obrazu diff.png, a poniżej funkcje wykonujące powyższe czynności.

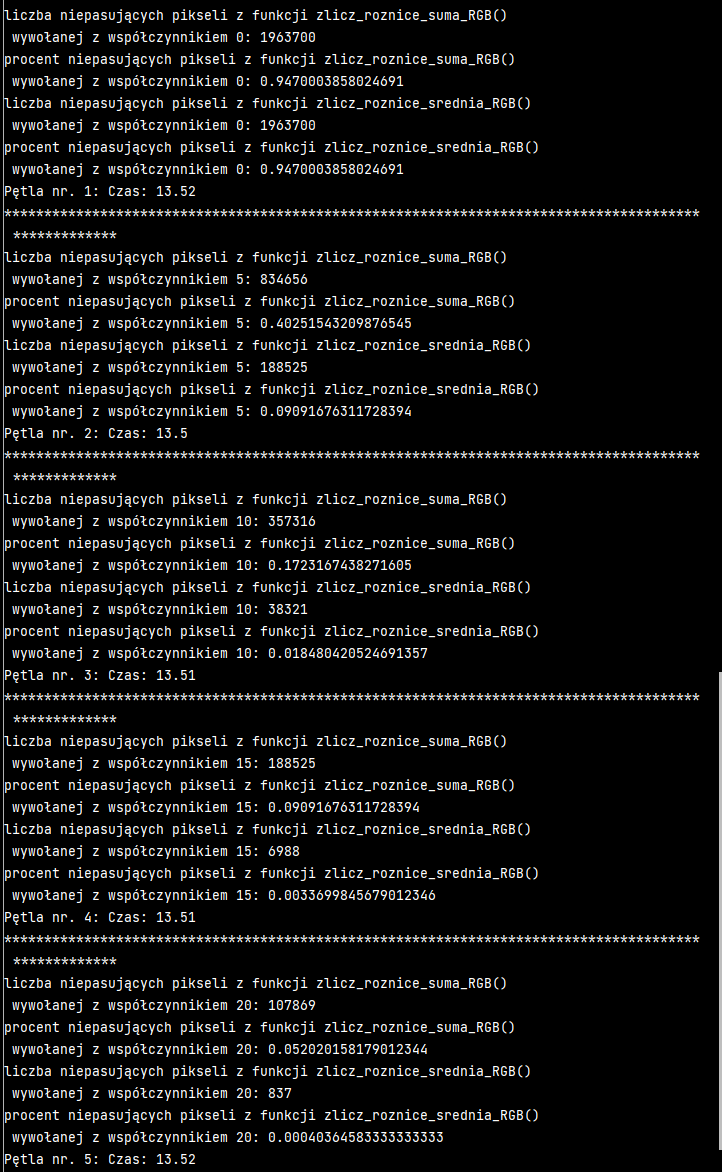
Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Powyższe funkcje pozwolą zobaczyć jak obraz się od obrazu czarnego. Nie wiem w jakim zakresie widzi ludzkie oko, ale np. zmiana wartości o 1-2 na pojedynczych pikselach sprawi, że powstały obraz będzie inny, chociaż na „oko” oba mogą być takie same. Powyższe funkcje wywołałem w pętli z różnymi współczynnikami, zaczynając od 0 i zwiększając o 5, aż do 20. Poniżej kod wywołujący te funkcje, a także wyniki jakie wyszły w zależności od współczynników. Jako, że w „print” opisałem dokładnie co aktualnie się wyświetla, pozwolę sobie nie komentować tego ponownie. Jedynie dodałem metodę time() z biblioteki time, aby można było sprawdzić czas działania w sekundach.





Obraz w formacie JPG jest kompresją stratną, co oznacza, że część danych jest trwale usuwana w procesie kompresji. Mimo że po wielu kompresjach może się wydawać, że obraz jest już wystarczająco "dobry", wciąż zachodzi utrata informacji, co oznacza, że nie można przywrócić pełnej jakości oryginalnego obrazu.

Jeśli będziesz porównywać nowy obraz JPG z obrazem JPG, który był wielokrotnie kompresowany i otwierany, to nawet jeśli różnica w jakości będzie trudna do zauważenia wizualnie, nadal będzie istniała różnica w jakości i zawartości pikseli. Obrazy te będą różniły się od siebie, choć może być to trudne do wykrycia. Różnica między obrazem oryginalnym i 5 jest łatwiejsza do zauważenia, niż miedzy 4 i 5. Z każdym zapisem strata jest mniejsza, ale nadal występuje.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Korzystając z funkcji z poprzednich lab, porównałem różnice w pojedynczych pikselach najpierw obraz oryginalny z piątym:

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

Następnie obraz 4 i 5:

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

Na poniższym obrazie zakodowano inny obraz. Gołym okiem jest całkowicie nie widoczny, bo różni się o 1-2 wartości na pikselu.

Obraz zawierający drzewo, na wolnym powietrzu, upadek, krajobraz

Opis wygenerowany automatycznie

Poniższy kod korzysta z dwóch obrazów, oryginały i zakodowanego. W pierwszej kolejności zamienia oba obrazy na tablicę RGB. Następnie tworzymy pustą tablicę w formacie uint8 w tym przypadku wypełnionej jedynkami, wypełnienie jej zerami nie zrobi różnicy. Następnie przechodzimy przez cała tablicę, granice range() ustalamy za pomocą metody shape(). W tym przypadku zakładamy, że obie tablice są równe. Jeśli w danych współrzędnych piksele są równe, w odpowiadających współrzędnych na tablicy odkodowanej zamienia piksel na wartość 0, a kiedy znajdzie różnice, zamienia go na 255. Następnie zamienia tablicę na obraz, i zwraca go jako wynik działania funkcji.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Teraz wystarczy już tylko wywołać funkcję i zapisać rezultat jako kod2.bmp

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

A o to rezultat odkodowania:

Obraz zawierający Czcionka, zrzut ekranu, Grafika, czarne

Opis wygenerowany automatycznie