

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Колоризація grayscale-зображень

Виконав:
студент гр. КА-31
Якубович В.П.

Науковий керівник:
асистент Кухарєв С.О.

Актуальність теми:

- ▶ Реставрація історичних чорно-білих фотографій
- ▶ Колоризація відеопотоку (зокрема, старих фільмів)
- ▶ Висока затратність (як часова, так і ресурсна) сучасних методів колоризації

Постановка задачі:

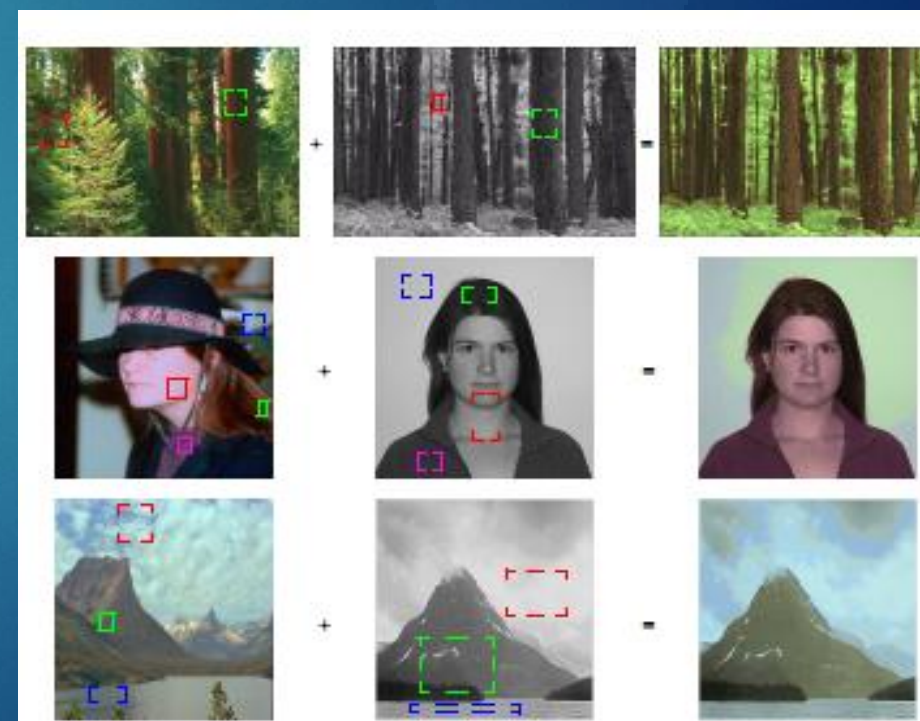
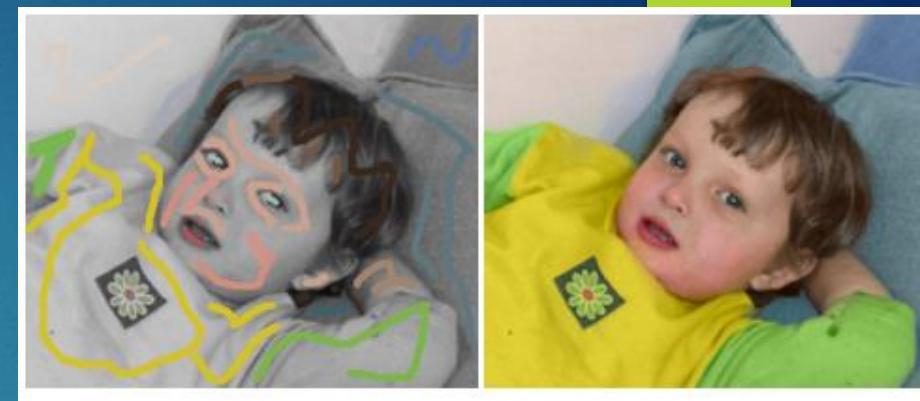
3

- ▶ Провести аналіз існуючих підходів та методів колоризації
- ▶ Створити програмний продукт, що реалізовує алгоритм колоризації.
- ▶ Експериментально перевірити роботу алгоритму при різних значеннях параметрів
- ▶ Порівняти результати роботи алгоритму з іншими підходами

Існуючі підходи:

4

- ▶ 1) Colorization using Optimization – на основі ручного позначення кольорів (верхній рисунок)
- ▶ 2) Transferring Color to Greyscale Images – вибір зон, з яких та в які буде переноситися колір (нижній рисунок)
- ▶ 3) Алгоритм Вієри – детально розглянутий в наступних слайдах
- ▶ 4) Онлайн-сервіс «Algorithmia» - за допомогою нейронних мереж



Використовуваний алгоритм – алгоритм Вієри

5

- ▶ Суть полягає у перенесенні кольору на grayscale-зображення з деякого іншого, кольорового зображення



Алгоритм Вієри

Крок 1: переведення зображень з RGB в Lab

► Відбувається в 3 етапи:

$$1) \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.414253 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

$$2) X \leftarrow X/0.950456 \quad ; \quad Z \leftarrow Z/1.088754.$$

$$3) L = \begin{cases} 116Y^{\frac{1}{3}} - 16, & Y > 0.008856 \\ 903.3Y, & Y \leq 0.008856 \end{cases}$$

$$a = 500(f(X) - f(Y))$$

$$b = 200(f(Y) - f(Z))$$

► Де

$$f(t) = \begin{cases} t^{\frac{1}{3}}, & t > 0.008856 \\ 7.787t + \frac{16}{116}, & t < 0.008856 \end{cases}$$

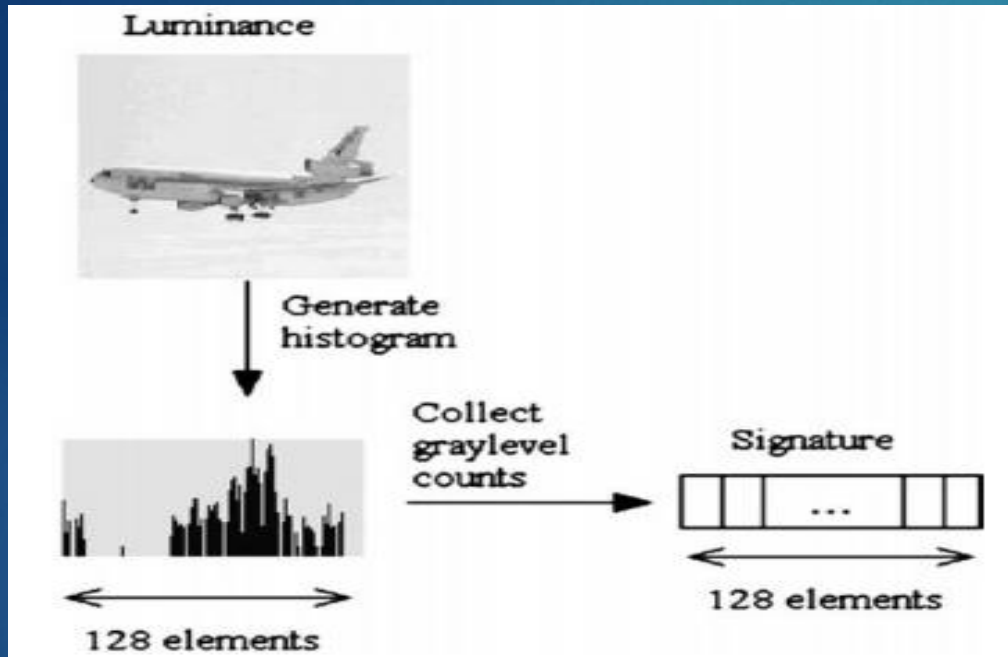


Алгоритм Вієри

7

Крок 2: знаходження зображення-джерела кольору

- Обчислення сигнатур вхідного grayscale-зображення та всіх кольорових зображень-кандидатів:



$$h(r_k) = n_k$$

де r_k – числове значення каналу, n_k – кількість пікселів зображення, що мають таке значення цього каналу.

- Пошук найбільш підходящого зображення: знайти кореляції за формулою:

$$d_{correl}(H_1, H_2) = \frac{(H_1, H_2)}{\|H_1\| \cdot \|H_2\|} = \frac{\sum_{i=1}^k H_{1i} \cdot H_{2i}}{\sqrt{\sum_{i=1}^k H_{1i}^2} \cdot \sqrt{\sum_{i=1}^k H_{2i}^2}}$$

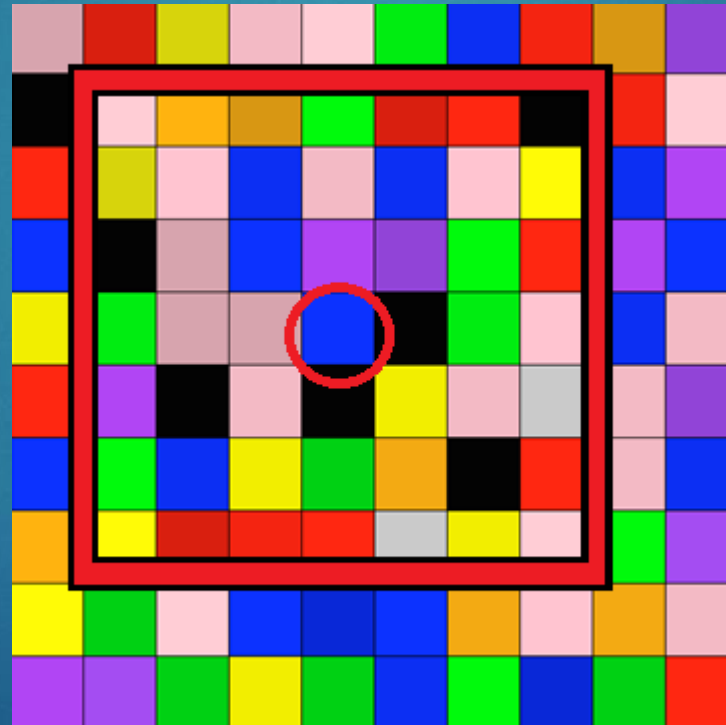
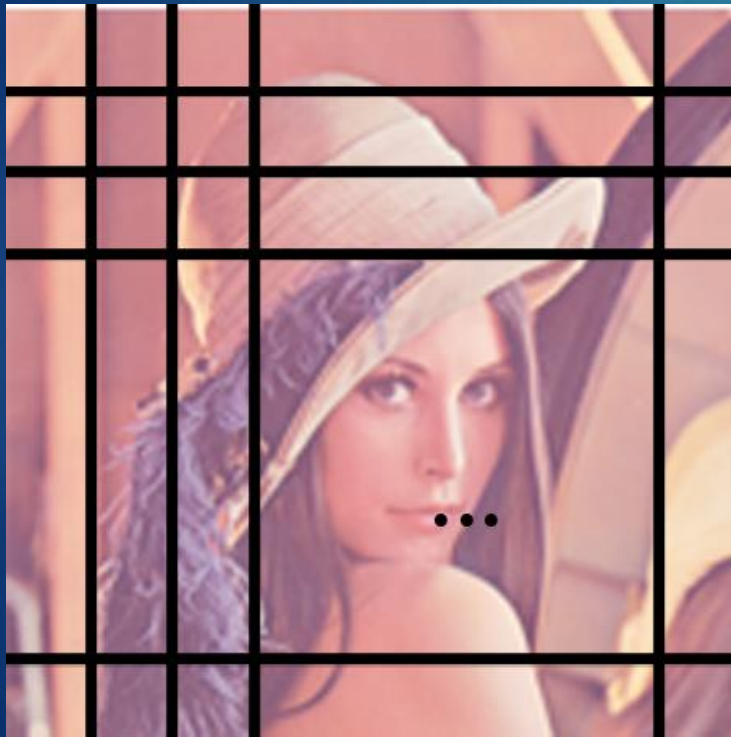
і визначити найбільшу.

Алгоритм Вієри

Крок 3: перенесення кольору з джерела на grayscale-зображення

► 3.1: Знаходження зразків кольору:

Зображення-джерело розбивається на сітку $N \times N$, в кожній комірці випадково обираємо піксель. В околі пікселя розміром $M \times M$ рахуємо мат. сподівання та дисперсію значень каналу L .



$$M = \frac{\sum_{i=1}^n L_i}{n}$$

$$D = \frac{n}{n-1} \left(\frac{1}{n} \sum_{i=1}^n L_i^2 - M^2 \right)$$

В результаті отримали N^2 зразків кольору.

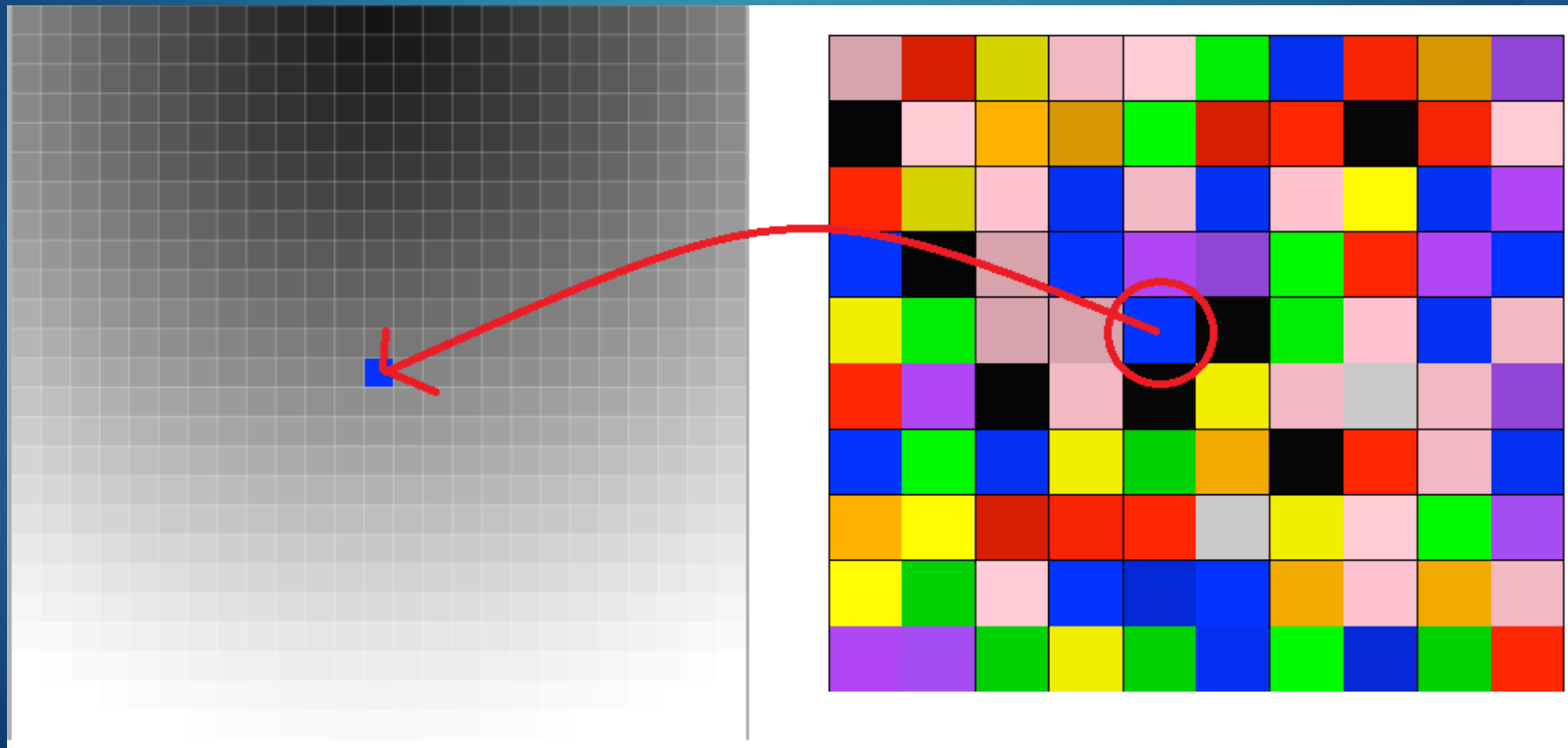
Алгоритм Вієри

9

Крок 3: перенесення кольору з джерела на grayscale-зображення

► 3.2: власне перенесення кольору:

Здійснюється повний попіксельній обхід grayscale-зображення, так само рахується мат. сподівання та дисперсія, порівнюються із зразками. З найбільш підходящого переносяться значення каналів a та b (L залишається незмінним!).



Програмний продукт

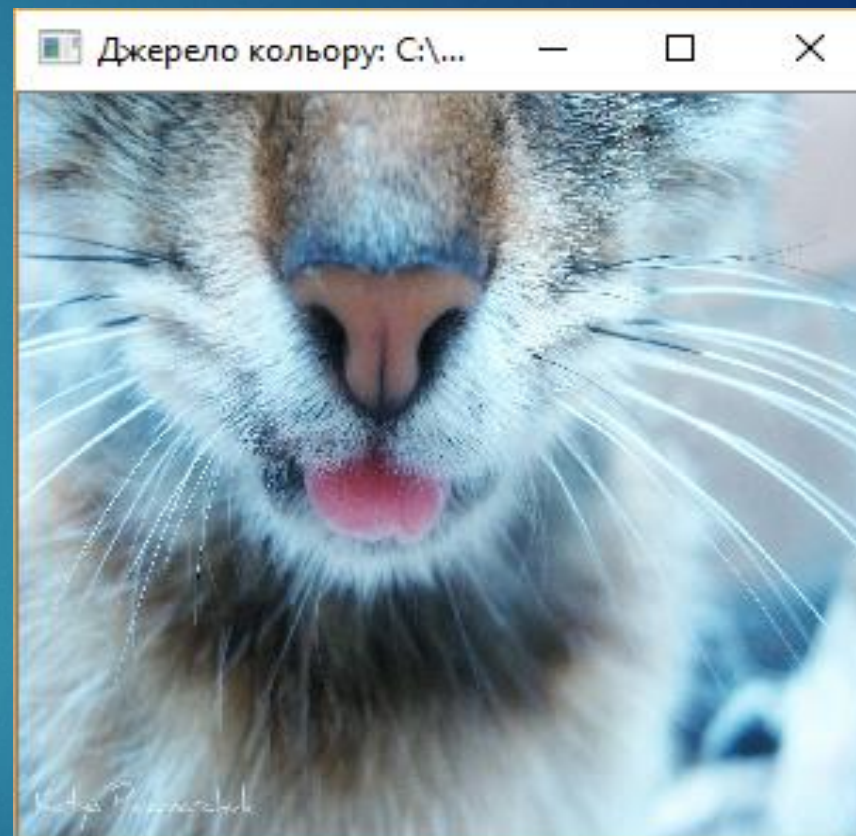
10

Був розроблений програмний продукт, що реалізує алгоритм Вієри. При розробці до майбутнього продукту виставлялися такі вимоги:

- ▶ Якомога більша автоматичність (тобто, бажана відсутність додаткових дій з боку користувача).
- ▶ Висока швидкість роботи алгоритму.
- ▶ Адекватність роботи (суб'єктивна).

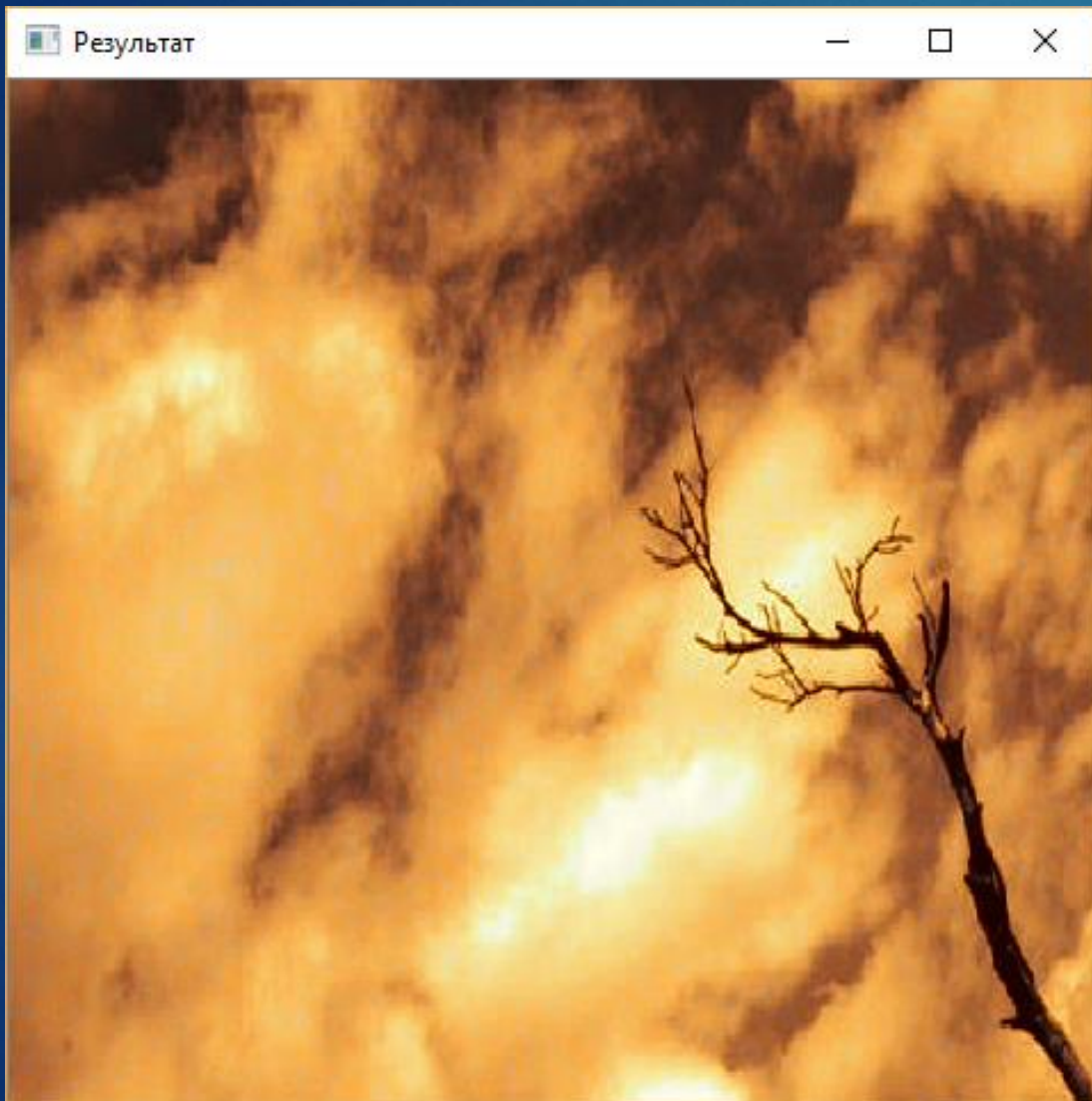
Результати роботи:

11

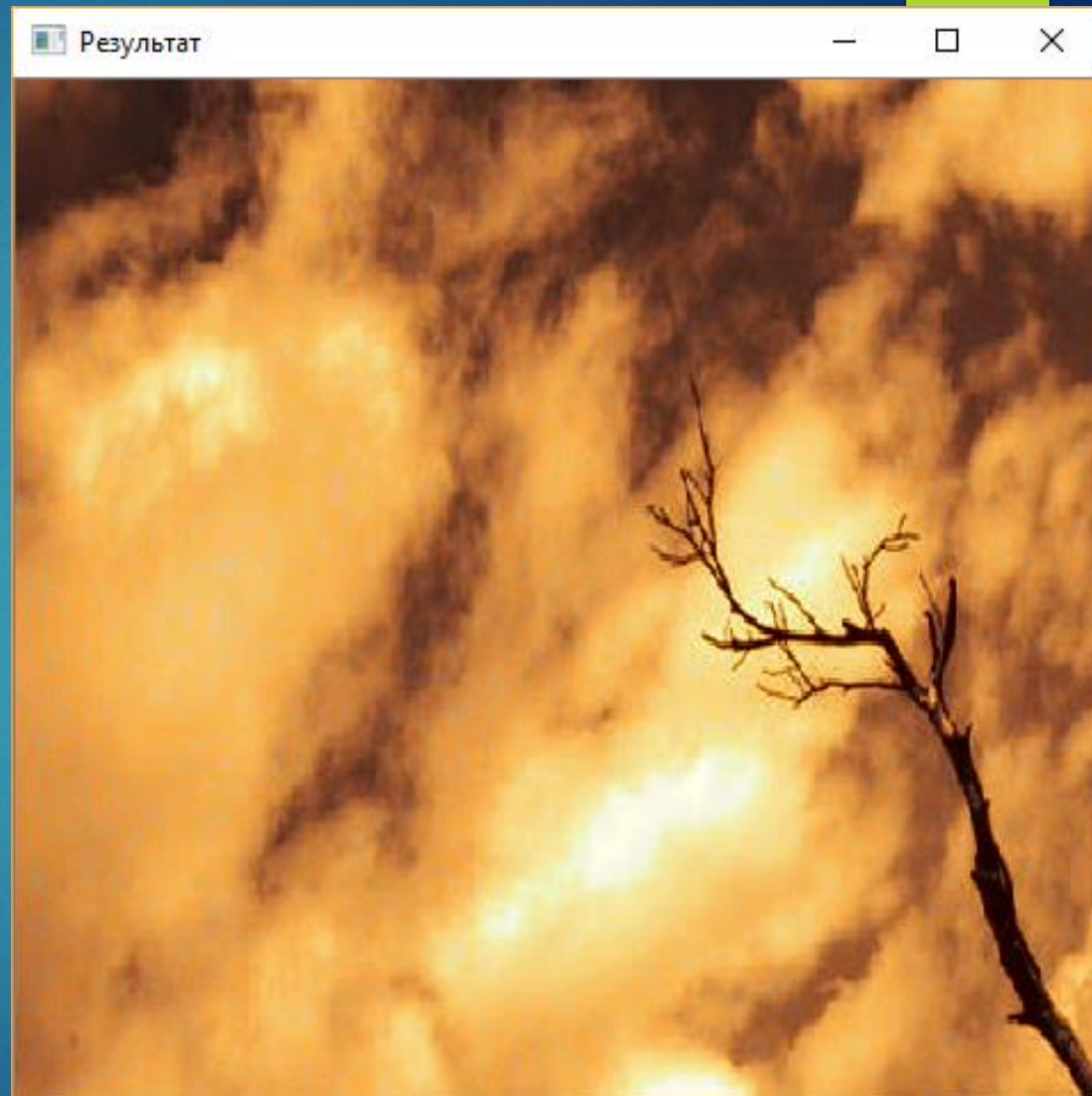


Перевірка роботи при різних параметрах

12



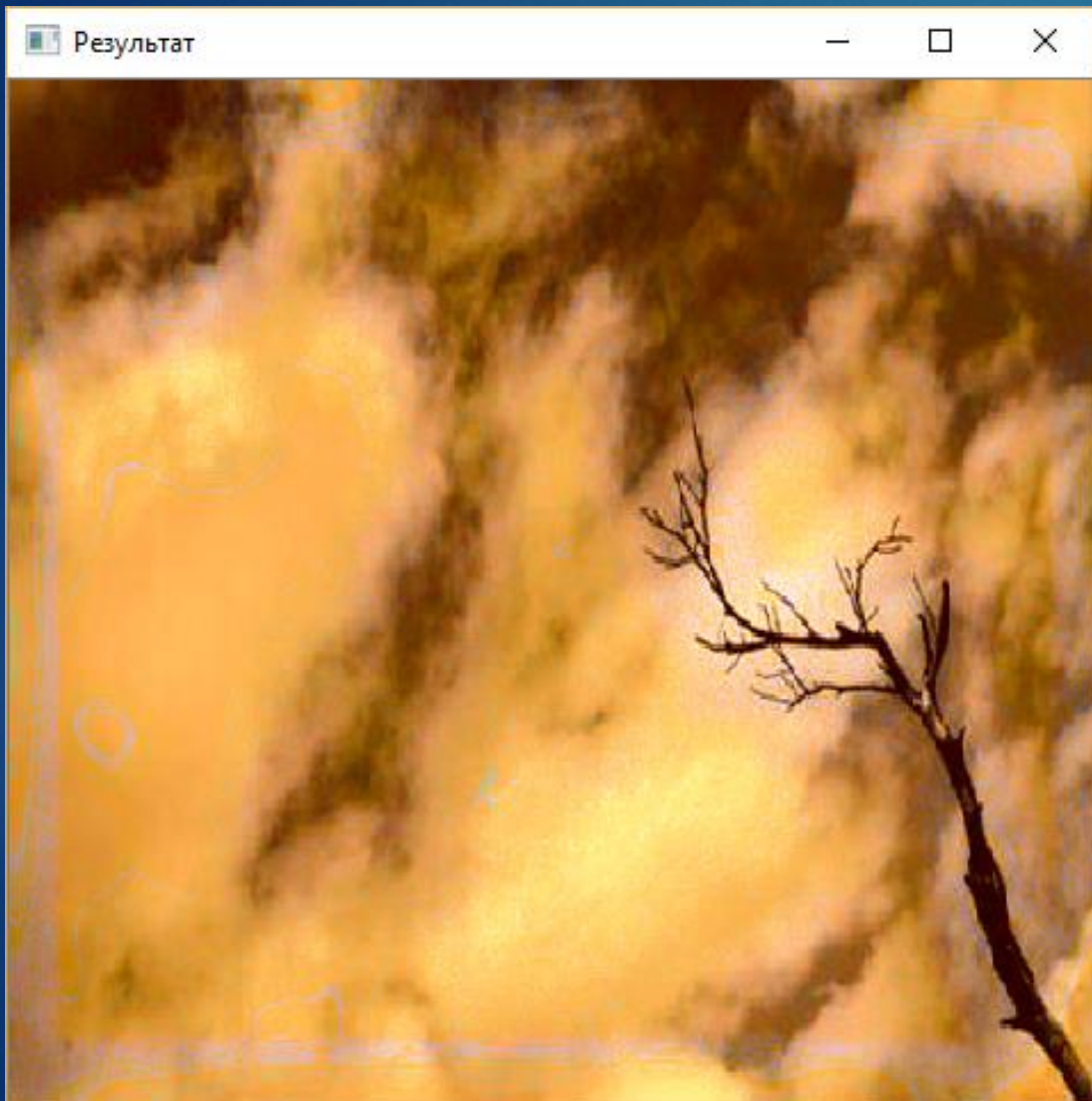
$N=150, M=5$



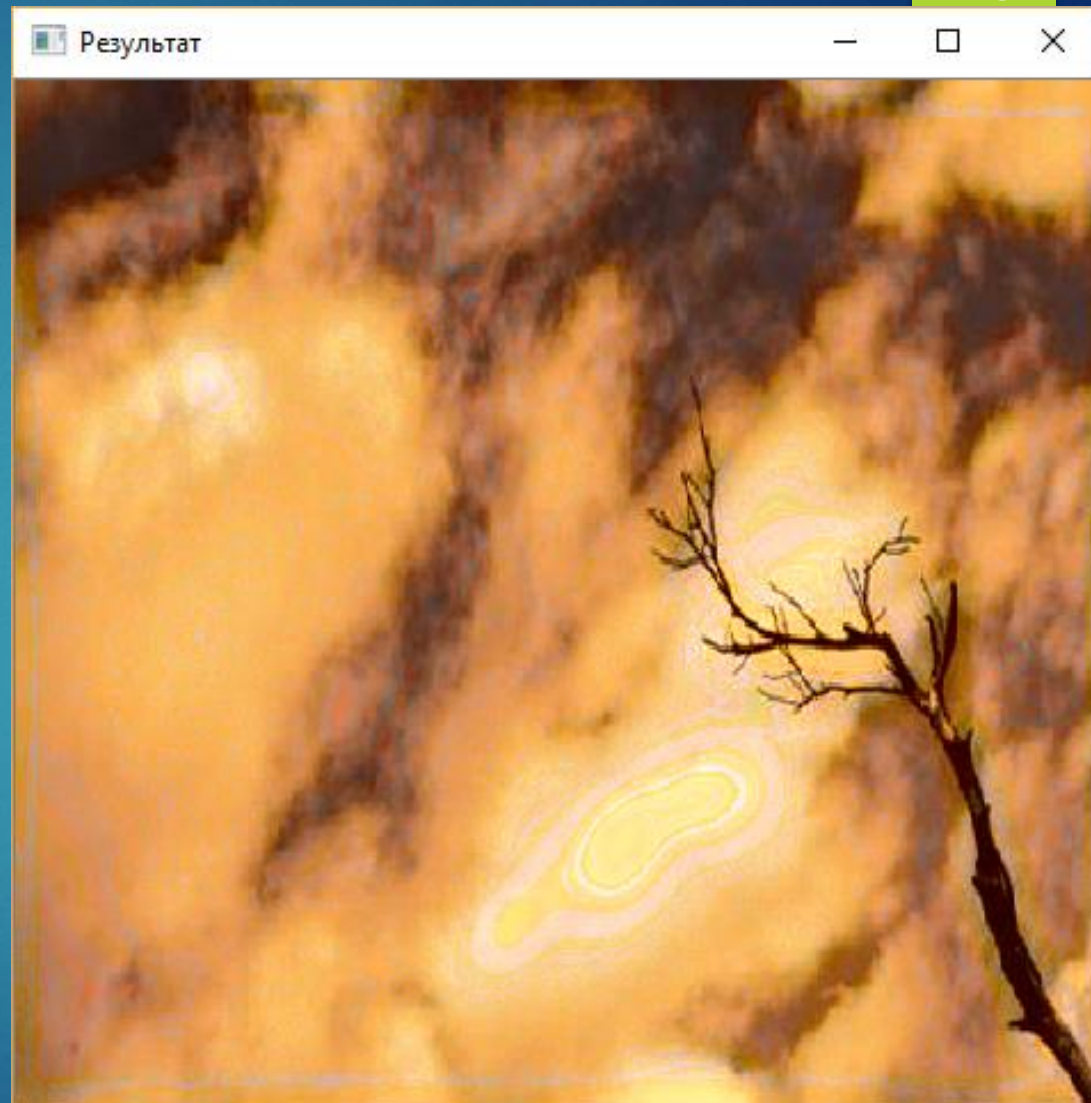
$N=300, M=5$

Перевірка роботи при різних параметрах

13



$N=40, M=50$

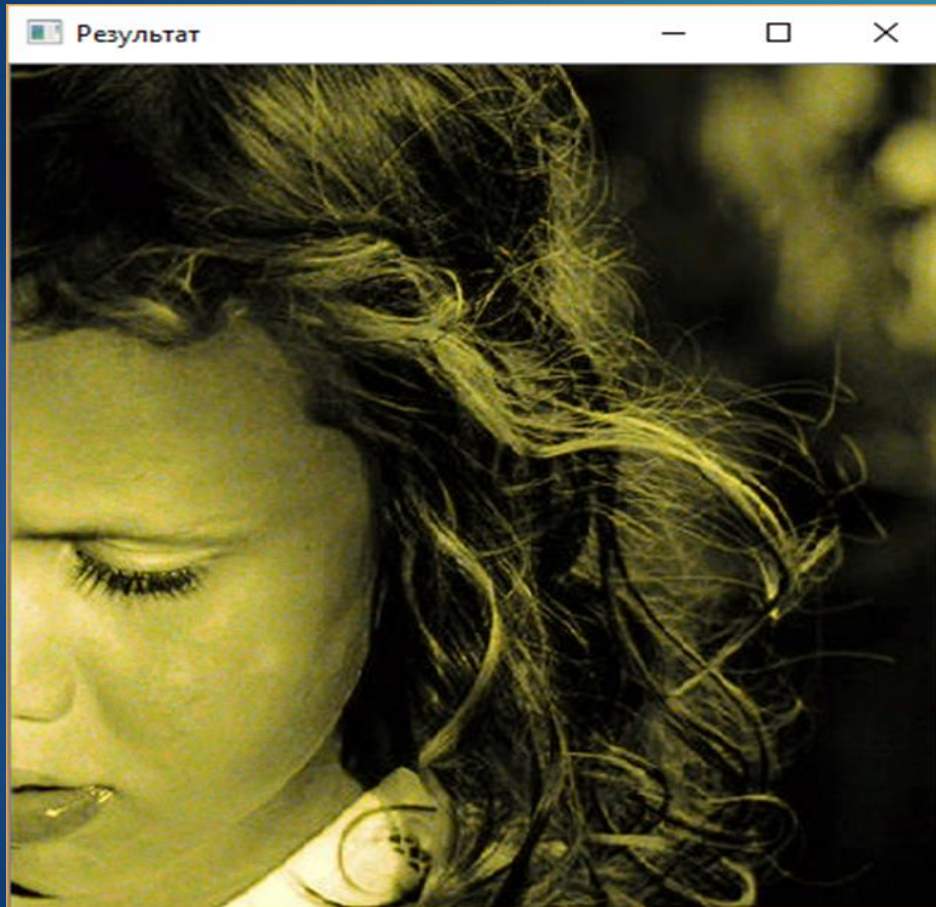


$N=40, M=25$

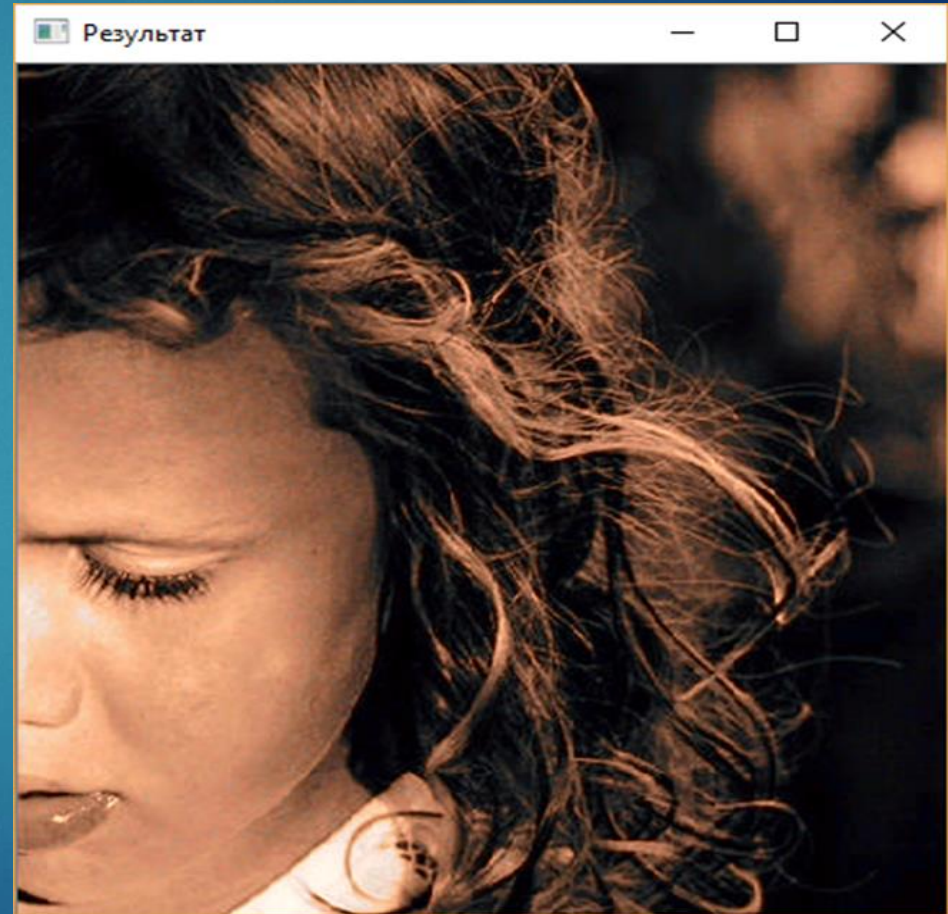
Неадекватні результати та їх покращення

14

- ▶ Якщо джерело кольору в базі даних явно не підходить, можна обрати в базі лише ті зображення, які здаються правильними



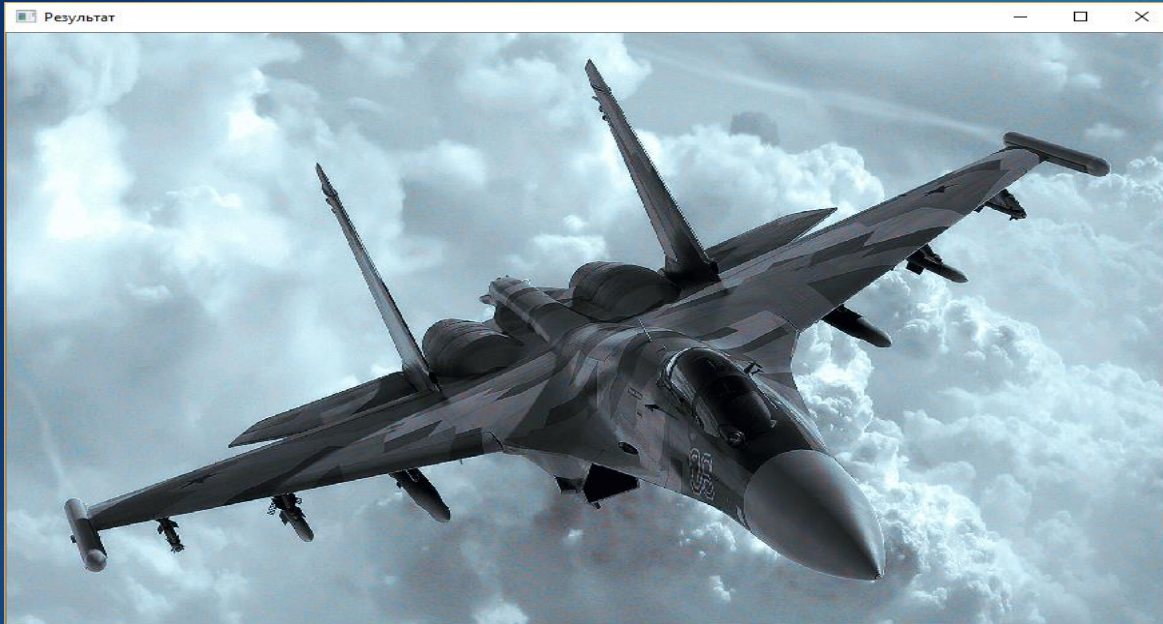
Джерело було вибране серед всіх зображень



Джерело було вибране серед зображень людей

Порівняння з іншим алгоритмом

15

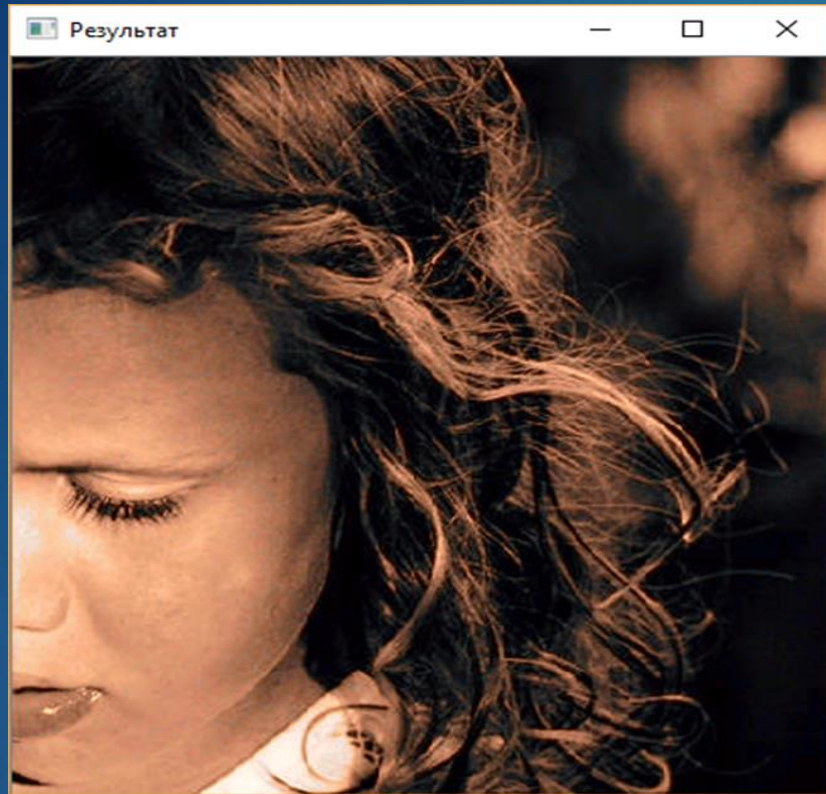


Тут і далі: фотографії зліва – результати роботи розробленого програмного продукту, фотографії справа – результати роботи онлайн-сервісу «Algorithmia» (див. слайд 4, п.4)

Як видно, результат роботи онлайн-сервісу є гіршим через наявність зайвих рудих областей на літаку. Хоча колоризація неба вийшла більш реалістичною.

Порівняння з іншим алгоритмом

16



Колоризація портрету в онлайн-сервісі вийшла трохи більш реалістичною (особливо колір волосся та задній план). Зазначимо також, що фотографія зліва була отримана шляхом ручного відбору зображень-джерел кольору. Тобто, в цілому онлайн-сервіс впорався краще

Порівняння з іншим алгоритмом

17



В обох випадках програми продемонстрували адекватний результат, хоча фото з онлайн-сервісу має більше розмаїття кольорів.

Порівняння з іншим алгоритмом

18



Обидва зображення є адекватними, але зображення зліва, на суб'єктивну думку автора, є більш правдоподібним

► Практична цінність:

Розроблено програмний продукт, що дозволяє колоризувати grayscale-зображення.

► Новизна:

Проведено експерименти щодо роботи алгоритму при різних параметрах та порівняння з існуючим програмним продуктом

Подальші кроки:

20

- ▶ Вдосконалення алгоритму таким чином, щоб покращити підбір джерела кольору
- ▶ Портування на різні операційні системи та створення web-версії
- ▶ Застосування алгоритму для колоризації відеопотоку