

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА

Навчально-науковий інститут фізико-технічних та комп'ютерних наук  
Кафедра комп'ютерних програмного забезпечення комп'ютерних систем

**ЗВІТ**  
**про виконання лабораторної роботи № 6**  
**з дисципліни**  
**«Прикладна інтелектуальна обробка сигналів та зображень»**  
**на тему: «“Блокова обробка. Реалізація алгоритму JPEG”»**

Виконала  
студентка 5-го курсу  
544 групи  
Турецька Даяна Вікторівна  
(П.І.Б)

Чернівці, 2025

## 1. Мета

Метою даної лабораторної роботи є набуття знань про існуючі методи стиснення зображень та ознайомитися з основними з них.

## 2. Хід роботи

### 2.1. Завантаження зображень

- `imgCol1 = imread('peppers.png');` % Кольорове зображення
- `imgGray1 = imread('cameraman.tif');` % Чорно-біле зображення

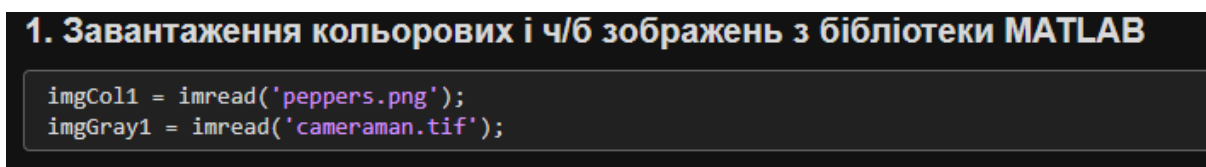


Рисунок 1 - завантаження зображень

На цьому кроці імпортуються два вихідні файли зображень із бібліотеки MATLAB: `peppers.png` (кольорове, 512×512 пікселів) та `cameraman.tif` (грампіксельне зображення 256×256). Це дозволяє продемонструвати застосування ДКП до різномірних типів даних.

### 2.2. Перетворення кольорового зображення в чорно-біле

- `gray1 = im2double(rgb2gray(imgCol1));` % Перехід у відтінки сірого та нормалізація
- `gray3 = im2double(imgGray1);` % Конвертування до подвійної точності

```

gray1 = im2double(rgb2gray(imgCol1));
gray3 = im2double(imgGray1);

figure;
subplot(2,2,1), imshow(imgCol1), title('Color: peppers');
subplot(2,2,2), imshow(gray1), title('Gray from peppers');
subplot(2,2,3), imshow(imgGray1), title('Original cameraman');
subplot(2,2,4), imshow(gray3), title('Double-scaled cameraman');
truesize;

```

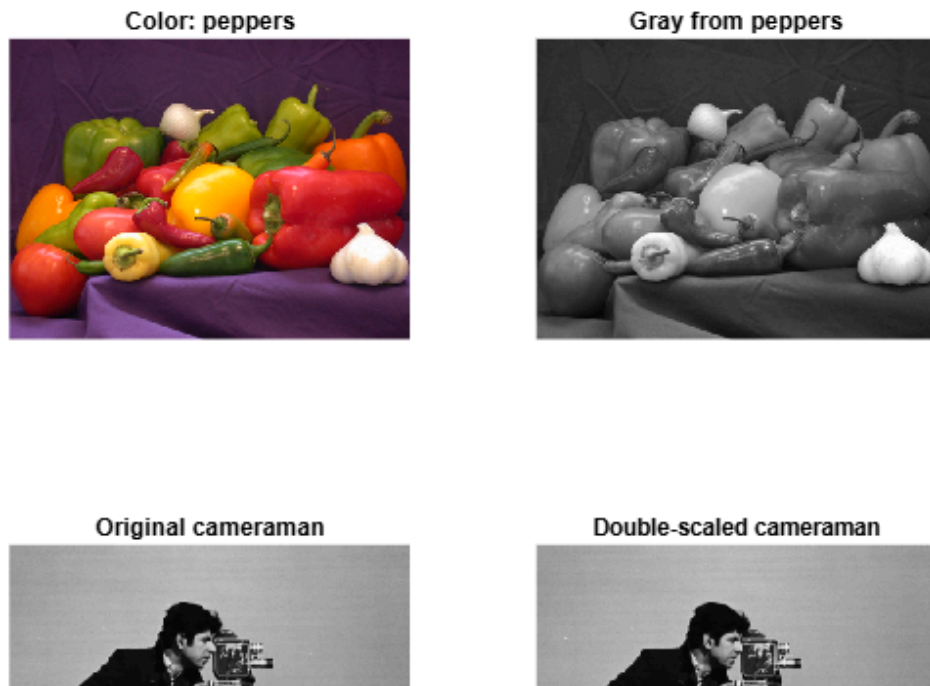


Рисунок 2 - перетворення в чорно-білі зображення

Функція `rgb2gray` усуває колірну інформацію, обчислюючи зважену суму каналів RGB, що спрощує обчислення ДКП. Переведення до `double` дає рівномірний діапазон  $[0,1]$  для точності обчислень.

### 2.3. Поблочне ДКП розміром $8 \times 8$

```
blkSize = 8;
```

```
dctM = dctmtx(blkSize);
```

```
dctFun = @(bs) dctM * bs.data * dctM';
```

```
coeffsA = blockproc(gray1, [blkSize blkSize], dctFun);
```

```
coeffsB = blockproc(gray3, [blkSize blkSize], dctFun);
```

```
3. Поблочне ДКП (розмір блока — 8×8)

% розмір блока та матриця коефіцієнтів ДКП
blkSize = 8;
dctM = dctmtx(blkSize);

% процедура
dctFun = @(bs) dctM * bs.data * dctM';    % пряме ДКП

% поблочна обробка зображень
coeffsA = blockproc(gray1, [blkSize blkSize], dctFun);
coeffsB = blockproc(gray3, [blkSize blkSize], dctFun);
```

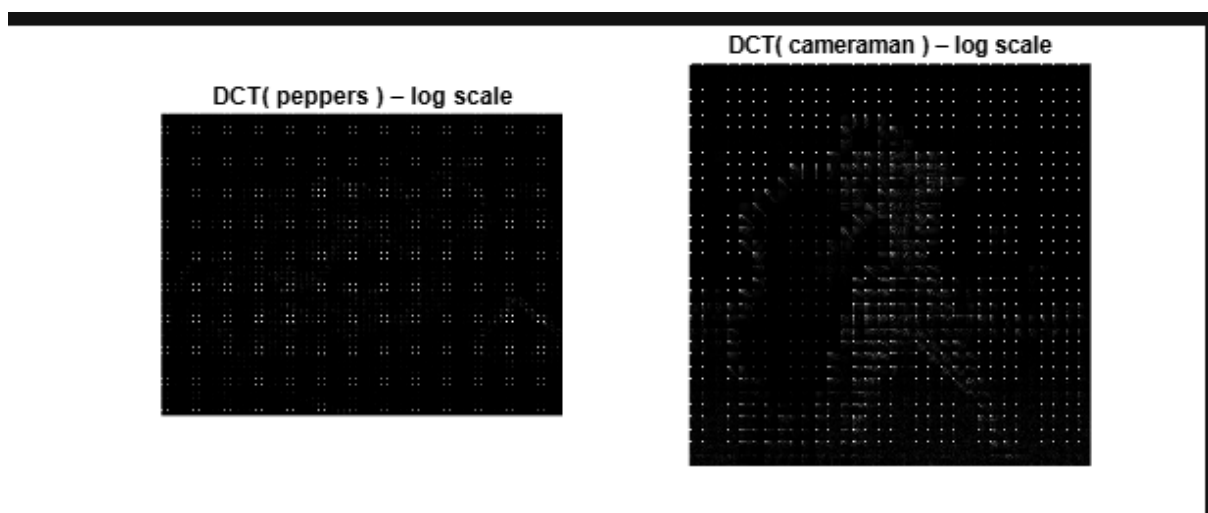
Рисунок 3 - реалізація поблочного ДКП

ДКП використовується для перетворення кожного  $8 \times 8$  блоку з просторового представлення в частотне. Матриця `dctM` містить базисні функції ДКП. Основа методу — виділення низьких (значущих) та високих частот для подальшого стиснення.

## 2.4. Відображення спектру ДКП в логарифмічному масштабі

- `figure;`
- `imshow(log(abs(coeffsA)+1), []), title('DCT( peppers ) – log scale');`

```
imshow(log(abs(coeffsB)+1), []), title('DCT( cameraman ) – log scale');
```



## Рисунок 4 - Відображення спектрів ДКП

Пояснення: Логарифмічне відображення дозволяє «підняти» слабкі (високочастотні) коефіцієнти, які при лінійному масштабі практично «згасають» на фоні сильніших низьких частот.

Спектр ДКП містить сильні низькочастотні коефіцієнти в кутах блоків та слабкі високочастотні по діагоналі. Логарифм відображає широкий діапазон амплітуд, що допомагає візуалізувати й дрібні компоненти.

### 2.5. Відновлення зображення зі спектру ДКП

- $\text{idctFun} = @(bs) \text{dctM}' * bs.\text{data} * \text{dctM};$
  - $\text{rec1} = \text{blockproc}(\text{coeffsA}, [\text{blkSize} \text{blkSize}], \text{idctFun});$
- $\text{rec2} = \text{blockproc}(\text{coeffsB}, [\text{blkSize} \text{blkSize}], \text{idctFun});$

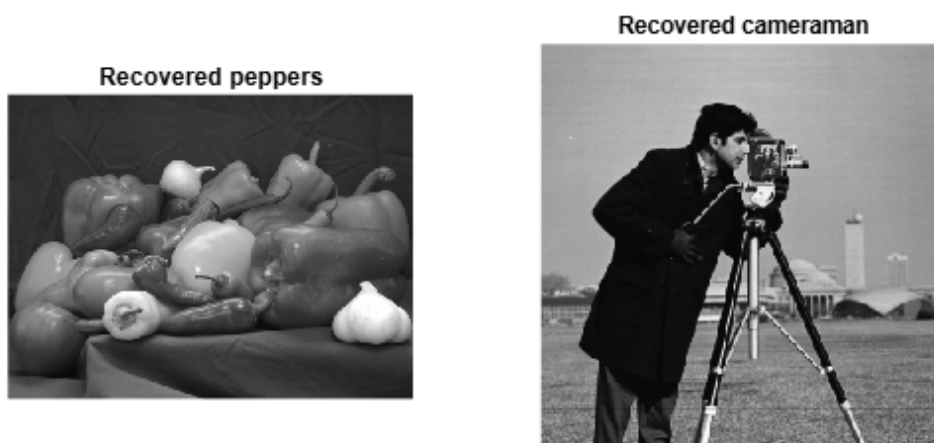


Рисунок 5 - Результат відновлення зображень

Обернена ДКП точно відновлює початкове зображення, оскільки блокове перетворення є без втрат до квантування. Порівняння  $\text{rec1}$  та  $\text{rec2}$  з оригіналами підтверджує коректність реалізації.

### 2.6. Квантування результатів ДКП з різними кроками

Як працює квантування: Для кожного коефіцієнта  $J$  виконується  $\text{round}(J/s)$  — отримуємо ціле число, а потім множимо назад на  $s$ . Тобто значення «зкруглюється» до найближчого кратного  $s$ .

```
stepSizes = [0.1, 0.5, 1, 5];  
  
for s = stepSizes  
  
    qA = s * round(coeffsA / s);  
  
    qB = s * round(coeffsB / s);  
  
    rq1 = blockproc(qA, [blkSize blkSize], idctFun);  
  
    rq2 = blockproc(qB, [blkSize blkSize], idctFun);  
  
    % відображення rq1 та rq2...  
  
end
```

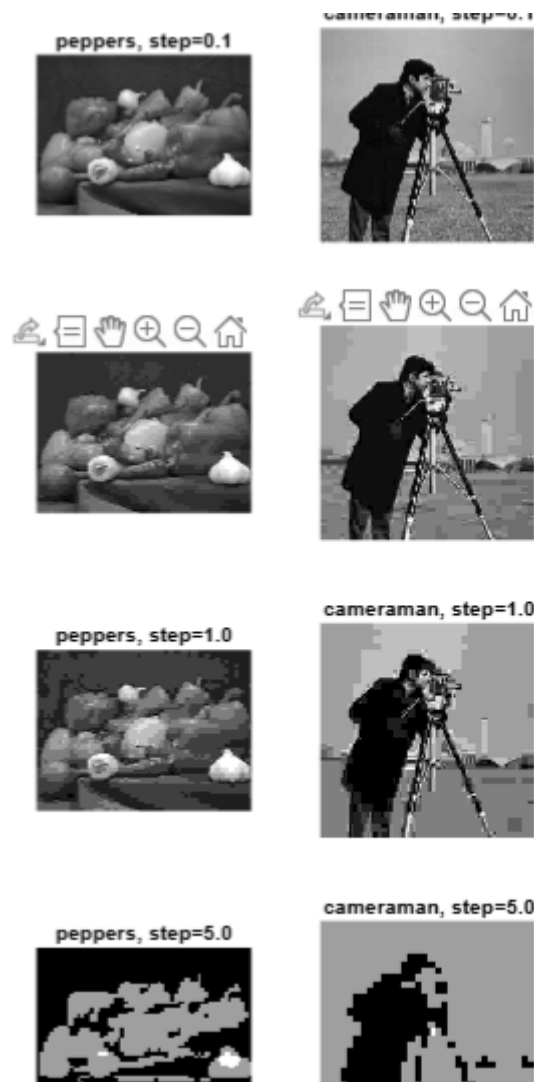


Рисунок 6 - Відображення квантування результатів ДКП з різними кроками

Зростання кроку  $s$  імітує сильніше округлення коефіцієнтів. При  $s=0.1$  якість майже не погіршується, а при  $s=5$  зображення втрачає дрібні деталі та набуває пікселізованого вигляду через значну деградацію високих частот.

## 2.7. Квантування коефіцієнтів ДКП за маскою

```
quantMask = [
    1 1 1 1 0 0 0 0;
    1 1 1 0 0 0 0 0;
```

```

1 1 0 0 0 0 0 0;

1 0 0 0 0 0 0 0; zeros(4,8) ];

maskedA = blockproc(coeffsA, [blkSize blkSize], @(bs)
quantMask .* bs.data);

maskedB = blockproc(coeffsB, [blkSize blkSize], @(bs) quantMask .*
bs.data);

```

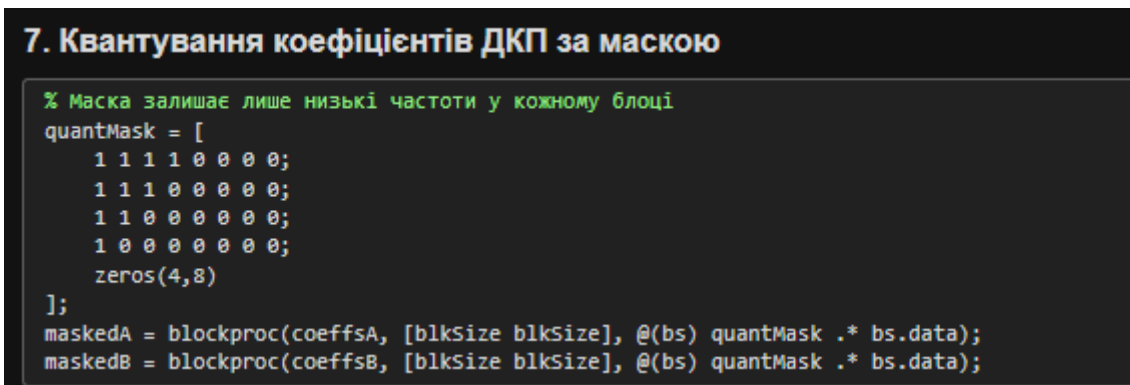


Рисунок 7 - Реалізація квантування за маскою

Маска залишає лише перші 10 коефіцієнтів (низькі частоти) кожного блоку. Це ілюстрація жорсткого відсічення високочастотних компонент для максимальної компресії.

## 2.8. Відновлення зображень після маскування спектру

```

recM_A = blockproc(maskedA, [blkSize blkSize], idctFun);

recM_B = blockproc(maskedB, [blkSize blkSize], idctFun);

```





Рисунок 8 - Результат відновлення зображень

При жорсткому відсіченні високих частот результати виражаються в розмитості та втраті текстур. Великі форми та контури зберігаються, але деталі (тіні і структури) суттєво згладжуються.

## 2.9. Обговорення результатів та мета квантування

Просте квантування (раз на  $N$ ) дає поступову втрату деталей у міру збільшення кроку  $s$ : чим більше  $s$ , тим грубіша «приблизна» картина.

Від первинного завантаження до зворотного ДКП демонструє можливість точного перетворення між просторовою та частотною областями без втрат перед квантуванням. Необхідна для аналізу розподілу частотних компонент по амплітуді.

Квантування з кроком: забезпечує контрольований компроміс між ступенем стиснення та втратою якості. Малий крок — слабка втрата; великий — виражені артефакти.

Квантування за маскою повністю відкидає високочастотні компоненти, залишаючи лише загальні форми (низькі частоти). Це дає сильний ступінь стиснення з втратами (як у JPEG), при цьому зберігається % візуально важлива інформація (силуети, крупні деталі).

Маскування ілюструє принцип JPEG, коли масивні блоки високих частот можуть бути повністю усічені без великого сприймаємо деградації.

### **3. Висновки**

У роботі показано, що квантування є ключовим етапом стиснення зображень з втратами, що дозволяє значно скоротити обсяг даних, зберігаючи візуально важливу інформацію низьких частот.