ЗЕРТХАНАЛЫҚ ЖҰМЫС № 1.1

NumPy кітапханасымен жұмыс жасау. Кескіндерді матрица ретінде ұсыну

1. Жұмыстың мақсаты

NumPy массивтерімен негізгі операцияларды меңгеру және цифрлық кескіннің бір немесе бірнеше сандық матрица түрінде ұсынылуын тәжірибе жүзінде зерттеу.

2. Қажетті құралдар

- Python 3.10 немесе одан жоғары нұсқа;
- Jupyter Notebook, Google Colab немесе PyCharm;
- NumPy, Matplotlib және Pillow кітапханалары.

pip install numpy matplotlib pillow

Қысқаша теориялық мәлімет

Цифрлық кескін — пиксельдерден тұратын деректер жиыны. Компьютерде кескін сандық массив ретінде сақталады. NumPy кітапханасындағы ndarray типі пиксель мәндерін тиімді сақтауға және өңдеуге мүмкіндік береді.

1 Кескін матрицасының өлшемдері

Кескін түрі	Массив пішіні	Пиксель мәні	Түсіндірме
Сұр реңкті	(H, W)	0–255	0 — қара, 255 — ақ
RGB	(H, W, 3)	[R, G, B]	Үш түс арнасы
RGBA	(H, W, 4)	[R, G, B, A]	A — мөлдірлік арнасы

Мұнда H — кескін биіктігі, W — ені, ал арналар саны түсті ақпараттың құрылымын көрсетеді.

2 NumPy-дің негізгі қасиеттері мен амалдары

- shape — массив өлшемдері;
- ndim — өлшемдер саны;
- dtype — элементтердің деректер типі;
- min(), max(), mean() — ең кіші, ең үлкен және орташа мәндер;
- массивті индекстеу image[y, x] тәртібімен орындалады.

Жұмыстың орындалу барысы

1-тапсырма. NumPy массивін құру және зерттеу

3 × 4 өлшемді массив құрып, оның қасиеттерін экранға шығарыңыз.

```
import numpy as np

A = np.array([[10, 20, 30, 40],
              [50, 60, 70, 80],
              [90, 100, 110, 120]], dtype=np.uint8)

print(A)
print("Пішіні:", A.shape)
print("Өлшем саны:", A.ndim)
print("Деректер типі:", A.dtype)
print("Min, max, mean:", A.min(), A.max(), A.mean())
```

Талдау: массивте неше элемент бар? A[1, 2] элементінің мәнін анықтаңыз.

2-тапсырма. Сұр реңкті кескін матрицасын құру

```
import matplotlib.pyplot as plt

gray = np.array([
   [0,   40,  80, 120, 160],
   [40,  80, 120, 160, 200],
   [80, 120, 160, 200, 240],
   [120,160, 200, 240, 255],
   [160,200, 240, 255, 255]
], dtype=np.uint8)

plt.imshow(gray, cmap="gray", vmin=0, vmax=255)
plt.colorbar(label="Пиксель қарқындылығы")
plt.title("Сұр реңкті матрица")
plt.show()
```

Талдау: 0, 128 және 255 мәндері қандай реңкке сәйкес келеді? Матрицаның бір элементін өзгертіп, кескінге әсерін сипаттаңыз.

3-тапсырма. RGB кескінін матрица түрінде ұсыну

```
rgb = np.zeros((200, 300, 3), dtype=np.uint8)
rgb[:, :100] = [255, 0, 0]       # қызыл
rgb[:, 100:200] = [0, 255, 0]    # жасыл
rgb[:, 200:] = [0, 0, 255]       # көк

plt.imshow(rgb)
plt.title(f"RGB кескіні: {rgb.shape}")
plt.axis("off")
plt.show()
```

Талдау: rgb[50, 150] пикселінің мәнін шығарыңыз. Ол қай түске сәйкес келеді?

4-тапсырма. Дайын кескінді NumPy массивіне айналдыру

Жұмыс қалтасына image.jpg атты кескін орналастырыңыз немесе файл атауын өз кескініңізге сәйкестендіріңіз.

```
from PIL import Image

img = Image.open("image.jpg").convert("RGB")
img_array = np.array(img)

print("Пішіні:", img_array.shape)
print("Типі:", img_array.dtype)
print("Ең кіші/үлкен мән:", img_array.min(), img_array.max())
print("Орташа мән:", img_array.mean())

plt.imshow(img_array)
plt.axis("off")
plt.show()
```

Талдау: кескіннің ені, биіктігі және арналар санын shape қасиеті арқылы анықтаңыз.

5-тапсырма. Кескін фрагментін қиып алу

```
h, w, _ = img_array.shape
fragment = img_array[h//4:3*h//4, w//4:3*w//4]

plt.imshow(fragment)
plt.title("Орталық фрагмент")
plt.axis("off")
plt.show()
```

Талдау: кесу кезінде массив индекстерінің [жол, баған] ретімен берілетінін түсіндіріңіз.

6-тапсырма. Түс арналарын бөлу

```
R = img_array[:, :, 0]
G = img_array[:, :, 1]
B = img_array[:, :, 2]

fig, ax = plt.subplots(1, 3, figsize=(12, 4))
for a, channel, title in zip(ax, [R, G, B], ["Red", "Green", "Blue"]):
    a.imshow(channel, cmap="gray", vmin=0, vmax=255)
    a.set_title(title)
    a.axis("off")
plt.tight_layout()
plt.show()
```

Талдау: қай арнада нысандар айқынырақ көрінеді? Себебін кескіндегі түстермен байланыстырыңыз.

7-тапсырма. Жарықтық пен контраст өзгерту

```
bright = np.clip(img_array.astype(np.int16) + 50, 0, 255).astype(np.uint8)
contrast = np.clip((img_array.astype(np.float32) - 128) * 1.5 + 128,
                   0, 255).astype(np.uint8)

fig, ax = plt.subplots(1, 3, figsize=(13, 4))
for a, im, title in zip(ax, [img_array, bright, contrast],
                        ["Бастапқы", "Жарық +50", "Контраст ×1.5"]):
    a.imshow(im); a.set_title(title); a.axis("off")
plt.tight_layout(); plt.show()
```

Маңызды: uint8 типіне 50 санын тікелей қосқанда 255-тен асқан мәндер айналып кетуі мүмкін. Сондықтан алдымен кеңірек типке түрлендіріп, np.clip() қолданамыз.

8-тапсырма. Жеке практикалық жұмыс

Өзіңіз таңдаған бір кескінге төмендегі әрекеттерді орындаңыз:

- кескін туралы shape, dtype, min, max және mean мәліметтерін шығару;
- кескіннің орталық фрагментін қиып алу;
- бір түс арнасын нөлге теңестіріп, жаңа кескін құру;
- кескінді көлденең бағытта аудару (np.flip немесе slicing қолдану);
- бастапқы және өңделген нәтижелерді бір терезеде көрсету;
- әр өзгерістің матрицаға және визуалды нәтижеге әсерін 3–5 сөйлеммен түсіндіру.

БАҚЫЛАУ СҰРАҚТАРЫ:

1. NumPy массиві мен Python тізімінің айырмашылығы қандай?
2. Сұр реңкті кескін неге екі өлшемді массивпен беріледі?
3. RGB кескінінің shape қасиетіндегі үшінші сан нені білдіреді?
4. Пиксельдің координатасы NumPy массивінде қандай ретпен жазылады?
5. uint8 деректер типінің мәндер диапазоны қандай?
6. np.clip() функциясы не үшін қолданылады?
7. Кескіннің түс арнасын қалай бөліп алуға болады?
8. Массивті кесу операциясы кескін өңдеуде қандай міндет атқарады?