**Лабораторна робота № 8**

ДОСЛІДЖЕННЯ МЕТОДІВ ВИЯВЛЕННЯ ТА КЛАСТЕРИЗАЦІЇ ЗОБРАЖЕНЬ

***Мета роботи:*** дослідити можливості методів виявлення та кластеризації зображень за допомогою функцій ППП MATLAB *.*

ТЕОРЕТИЧНІ ВІДОМОСТІ

Теоретичні відомості подані на лекціях.

**ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ**

**Завдання 1. Виявлення і вимір круглих об'єктів на зображенні**

***Виконайте приклад***

Цей приклад демонструє, як використовувати imfindcircles для автоматичного виявлення кіл або круглих об'єктів на зображенні. Він також показує використання viscircles для візуалізації виявлених кіл. Використовуйте Command Window ППП MATLAB.

*Крок 1: Завантажити зображення*

У цьому прикладі використовується зображення круглих пластикових фішок різних кольорів.

rgb = imread('coloredChips.png');

imshow(rgb)

Див рис.1.



Рис.1

Окрім наявності множини фішок для виявлення, на цьому зображенні відбувається кілька цікавих речей з точки зору виявлення кіл:

Є фішки різних кольорів, які мають різні контрасти по відношенню до фону. З одного боку, синій і червоний мають сильний контраст на цьому фоні. З іншого боку, деякі з жовтих фішок погано контрастують з фоном.

Зверніть увагу, що деякі фішки лежать одна на одній, а деякі - близько одна до одної і майже стикаються. Перекриття меж об'єктів і оклюзія об'єкта зазвичай є складними сценаріями для виявлення об'єкта.

***Отримане зображення подібне до рис.1 занесіть у звіт (рис.1 звіту)***

*Крок 2: Визначте довжину радіусу для пошуку кіл*

Функції imfindcircles потрібен діапазон радіусу для пошуку кіл. Швидкий спосіб знайти відповідний діапазон радіусів - використовувати інтерактивний інструмент imdistline для отримання приблизної оцінки радіусів різних об'єктів (рис.2).

d = imdistline;

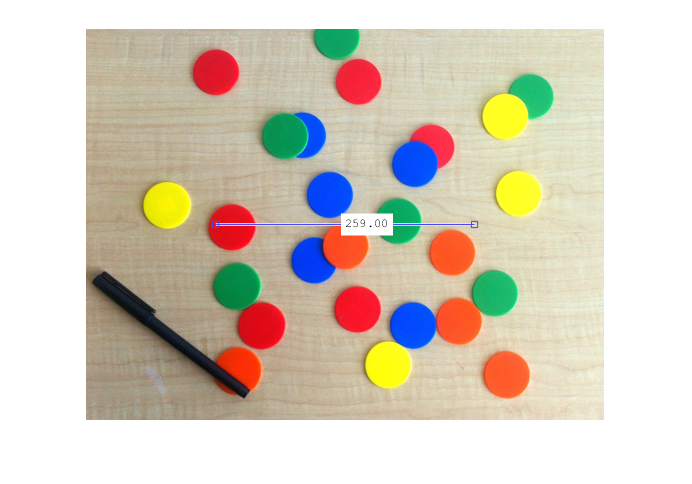


Рис.2

Функція imdistline створює інструмент, що перетаскується, який можна переміщати за розміром фішки, а числа можна читати, щоб отримати приблизну оцінку його радіуса. Більшість фішок мають радіус в діапазоні 21-23 пікселя. Для більшої впевненості використовуйте трохи більший радіус в 20-25 пікселів. Перед цим видаліть imdistline інструмент.

delete(d)

*Крок 3: Початкова спроба знайти кола*

Викличте функцію imfindcircles для цього зображення з радіусом пошуку [20 25] пікселів. Перед цим рекомендується визначити, чи є об'єкти яскравіше або темніше фону. Щоб відповісти на це питання, подивіться на версію зображення в градаціях сірого (рис.3).

gray\_image = rgb2gray(rgb);

imshow(gray\_image)

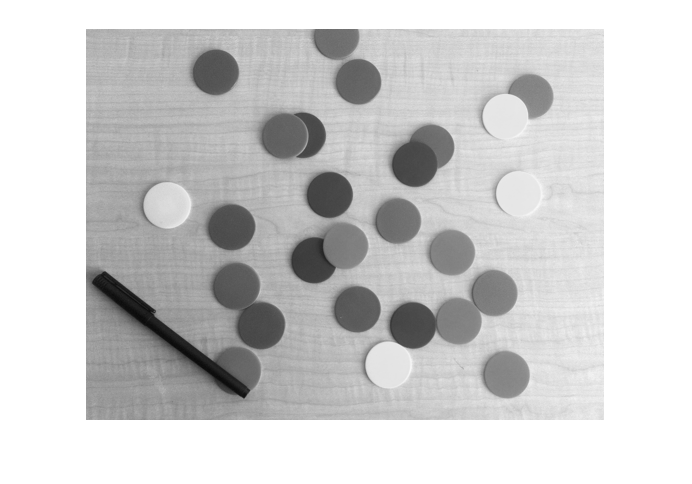


Рис.3

Фон досить яскравий, і більшість фішок темніше, ніж фон. Але, за замовчуванням, imfindcircles находить круглі об'єкти, які яскравіше фону. Отже, встановіть параметр «ObjectPolarity» в «dark» imfindcircles для пошуку темних кіл.

[centers,radii] = imfindcircles(rgb,[20 25],'ObjectPolarity','dark')

centers =

[]

radii =

[]

Зверніть увагу, що вихідні дані centersі radii є порожніми, а це означає, що кола не були знайдені. Це часто відбувається тому, що imfindcircles круговий детектор, як і більшість детекторів, imfindcircles має внутрішній поріг виявлення, який визначає його чутливість. Простіше кажучи, це означає, що впевненість детектора в певному (круговому) виявленні повинна бути вище певного рівня, перш ніж він буде вважати дійсно виявленим. Функція imfindcircles має параметр «Чутливість», який можна використовувати для управління цим внутрішнім порогом, а отже, і чутливістю алгоритму. Більш високе значення параметра «Чутливість» встановлює поріг виявлення нижче і призводить до виявлення більшої кількості кіл. Це схоже на управління чутливістю на детекторах руху, використовуваних в домашніх системах безпеки.

*Крок 4: Збільшіть чутливість виявлення*

Повертаючись до зображення фішки, можливо, що при рівні чутливості за замовчуванням всі гуртки нижче внутрішнього порога, тому окружності не були виявлені. За замовчуванням для параметра «Чутливість» (число від 0 до 1) встановлено значення 0,85. Збільште «Чутливість» до 0,9.

[centers,radii] = imfindcircles(rgb,[20 25],'ObjectPolarity','dark', ...

'Sensitivity',0.9)

centers = 8×2

146.1895 198.5824

328.8132 135.5883

130.3134 43.8039

175.2698 297.0583

312.2831 192.3709

327.1316 297.0077

243.9893 166.4538

271.5873 280.8920

radii = 8×1

23.1604

22.5710

22.9576

23.7356

22.9551

22.9995

22.9055

23.0298

На цей раз imfindcircles знайшов кілька кіл - вісім, якщо бути точним. Параметр centers містить розташування центрів кіл і параметр radii містить передбачувані радіуси цих кіл.

*Крок 5: Намалюйте кола на зображенні*

Функція viscircles може бути використана для малювання кіл на зображенні. Вихідні змінні centersі radiifrom imfindcircles можуть бути передані безпосередньо в viscircles (рис.4).

imshow(rgb)

h = viscircles(centers,radii);



Рис.4

Центри кіл здаються правильно розташованими, і їх відповідні радіуси, здається, добре відповідають реальним фішкам. Але все ж було пропущено досить багато фішок. Спробуйте збільшити «Чутливість» ще більше, до 0,92.

[centers,radii] = imfindcircles(rgb,[20 25],'ObjectPolarity','dark', ...

'Sensitivity',0.92);

length(centers)

ans = 16

Так що збільшення «Чутливості» дає нам ще більше виявлених кіл. Знову намалюйте ці кола на зображенні (рис.5).

delete(h) % Delete previously drawn circles

h = viscircles(centers,radii);



Рис.5

Цей результат виглядає краще.

***Отримане зображення подібне до рис.2 занесіть у звіт (рис.2 звіту)***

***Зробіть висновок, скільки виявлено фішок (кілець) та який параметр чутливості у вас виявився найкращим. Висновок занесіть у звіт.***

**Завдання 2. Виявлення і вимір круглих об'єктів на зображенні по варіантах**

***Виконайте ті ж самі дослідження для зображення обраного згідно вашого варіанту по таблиці 1.*** Використовуйте Command Window ППП MATLAB.

***Отримані зображення подібні до рис.1 та 2 занесіть у звіт (рис. 3-4 звіту).***

***Зробіть висновки та запишіть їх у звіт.***

Таблиця 1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № за списком у журналі | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Вхідні дані | R1.jpg | R2.jpg | R3.jpg | R4.jpg | R5.jpg | R6.jpg | R7.jpg | R8.jpg | R9.jpg | R10.jpg |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № за списком у журналі | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Вхідні дані | R1.jpg | R2.jpg | R3.jpg | R4.jpg | R5.jpg | R6.jpg | R7.jpg | R8.jpg | R9.jpg | R10.jpg |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № за списком у журналі | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| Вхідні дані | R1.jpg | R2.jpg | R3.jpg | R4.jpg | R5.jpg | R6.jpg | R7.jpg | R8.jpg | R9.jpg | R10.jpg |

**Завдання 3. Виявлення і вимір круглих об'єктів на зображенні (метод ІІ)**

Очистіть п’янять змінних. Завантажте зображення прикладу із завдання 1. Використовуйте Command Window ППП MATLAB.

rgb = imread('coloredChips.png');

imshow(rgb)

h = viscircles(centers,radii);

*Крок 1: Використайте другий метод (двоетапний) для пошуку кіл*

У функції imfindcircles є два різних методи для пошуку кіл. До сих пір для виявлення кіл використовувався метод за замовчуванням, званий методом фазового кодування. Є ще один метод, зазвичай званий двоетапним методом, який доступний в imfindcircles. Використайте двоетапний метод і покажіть результати (рис.6).

[centers,radii] = imfindcircles(rgb,[20 25],'ObjectPolarity','dark', ...

'Sensitivity',0.92,'Method','twostage');

delete(h)

h = viscircles(centers,radii);



Рис.6

Двоетапний метод виявляє більше кіл, при Чутливості 0,92. В цілому, ці два методи доповнюють один одного тим, що мають різні сильні сторони. Метод фазового кодування зазвичай швидше і трохи більш стійкий до шуму, ніж двоетапний метод. Але може також знадобитися більш високий рівень «Чутливості», щоб отримати ту саму кількість виявлень, що і в двоетапному методі. Наприклад, метод фазового кодування також знаходить ті ж фішки, якщо рівень «Чутливість» піднімається вище, скажімо, до 0,95 (рис.7).

[centers,radii] = imfindcircles(rgb,[20 25],'ObjectPolarity','dark', ...

'Sensitivity',0.95);

delete(h)

viscircles(centers,radii);

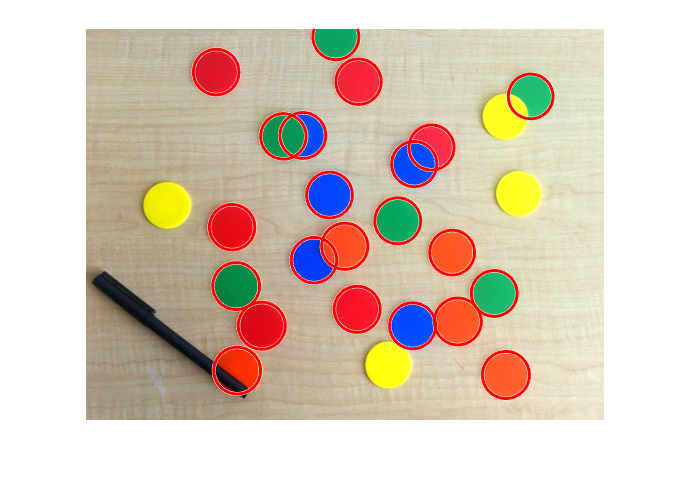


Рис.7

Зверніть увагу, що обидва методи дозволяють imfindcircles точно знайти центри та радіуси частково видимих (оклюзованих) фішок.

*Крок 2: Чому деякі кола все ще пропускаються?*

Дивлячись на останній результат, цікаво, що imfindcircles на зображенні не виявляються жовті фішки. Жовті фішки не мають сильного контрасту з фоном. Насправді вони здаються дуже схожими за інтенсивністю на фон. Чи можливо, що жовті фішки насправді не «темніше» фону, як передбачалося? Для підтвердження покажіть версію зображення в градаціях сірого ще раз (рис.8).

imshow(gray\_image)

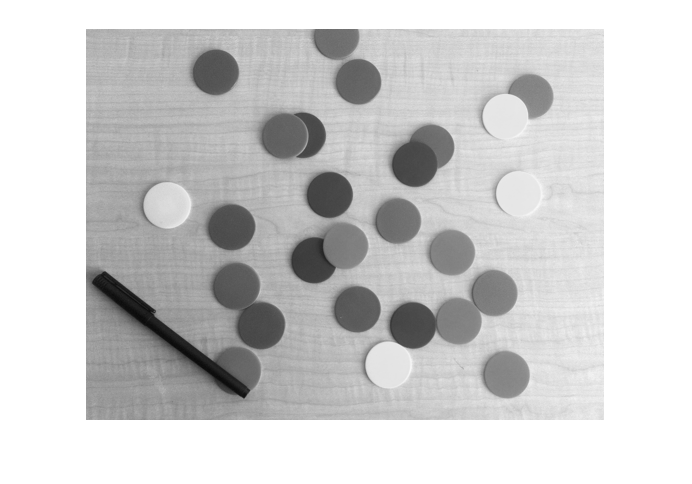


Рис.8

*Крок 3: Знайдіть «яскраві» кола на зображенні*

Жовті фішки мають майже таку ж інтенсивність, можливо, навіть яскравіше, в порівнянні з фоном. Тому, щоб виявити жовті фішки, змініть «ObjectPolarity» на «bright».

[centersBright,radiiBright] = imfindcircles(rgb,[20 25], ...

'ObjectPolarity','bright','Sensitivity',0.92);

*Крок 4: Намалюйте «яскраві» кола з іншим кольором*

Намалюйте яскраві кола іншим кольором, змінивши параметр «Колір» в viscircles (рис.9).

imshow(rgb)

hBright = viscircles(centersBright, radiiBright,'Color','b');

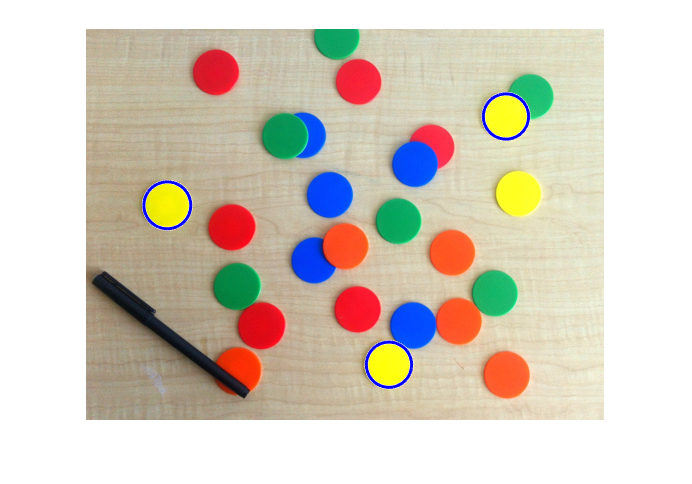


Рис.9

***Отримане зображення подібне до рис.9 занесіть у звіт (рис.5 звіту)***

Зверніть увагу, що три з відсутніх жовтих фішок були знайдені, але одна жовта фішка все ще відсутня. Ці жовті фішки важко знайти, тому що вони не виділяються на тлі інших.

*Крок 5: Понизьте значення «EdgeThreshold»*

Тут є ще один параметр, imfindcircles який може бути корисний, а саме EdgeThreshold. Щоб знайти кола, imfindcirclesіс використовує тільки крайові пікселі на зображенні. Ці крайові пікселі по суті є пікселями з високим значенням градієнта. Параметр EdgeThreshold визначає, наскільки високе значення градієнта в пікселі має бути до того, як воно буде вважатися крайовим пікселем і включене в розрахунок. Високе значення (ближче до 1) для цього параметра дозволить включати тільки сильні краї (більш високі значення градієнта), тоді як низьке значення (ближче до 0) є більш допустимим і включає навіть слабші краю (нижчі значення градієнта) в обчислення. У разі відсутності жовтої фішки, оскільки контрастність низька, очікується, що деякі з граничних пікселів (на периферії фішки) матимуть низькі значення градієнта. Тому зменшите параметр EdgeThreshold, щоб переконатися, що більшість крайових пікселів для жовтої фішки включені в обчислення (рис.10).

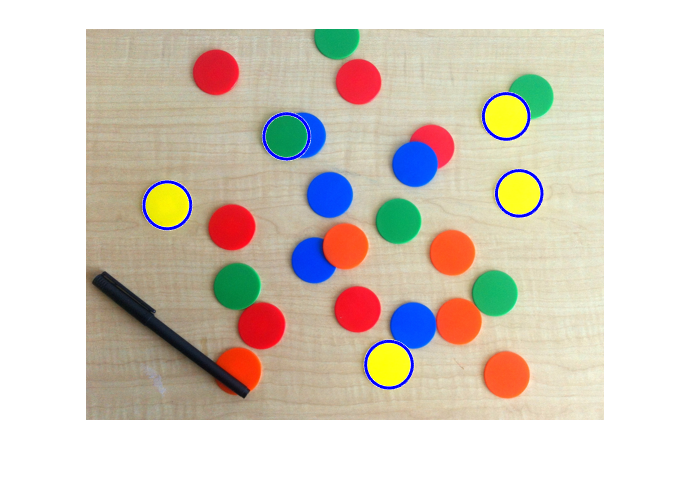
[centersBright,radiiBright,metricBright] = imfindcircles(rgb,[20 25], ...

'ObjectPolarity','bright','Sensitivity',0.92,'EdgeThreshold',0.1);

delete(hBright)

hBright = viscircles(centersBright, radiiBright,'Color','b');

Рис.10



*Крок 6: Намалюйте «темні» і «яскраві» кола разом*

Тепер imfindcircles находить всі жовті фішки, і зелені теж. Намалюйте ці фішки синім кольором, разом з іншими фішками, які були знайдені раніше (з «ObjectPolarity» встановленим на «темний»), червоним кольором (рис.11).

h = viscircles(centers,radii);

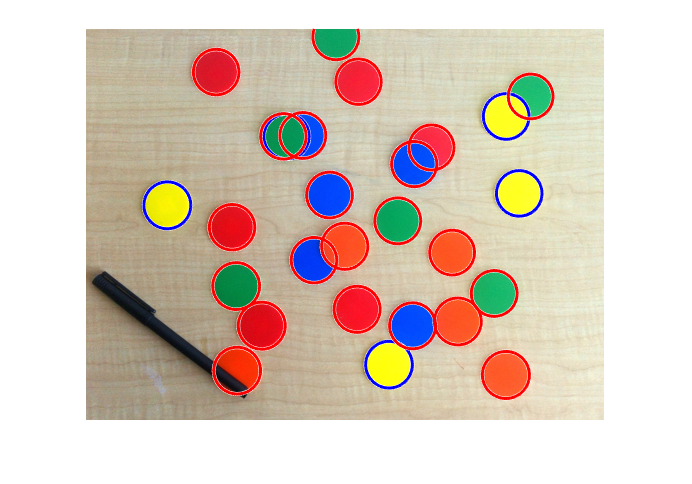


Рис.11

Всі кола виявлені. Слід зазначити, що зміна параметрів для підвищення ймовірності виявлення може призвести до появи більшої кількості кіл, але це також збільшує ймовірність виявлення помилкових кіл. Існує компроміс між кількістю справжніх кіл, які можуть бути знайдені (частота виявлення), і кількістю помилкових кіл, які знайдені за ними (частота помилкових тривог).

***Отримане зображення подібне до рис.11 занесіть у звіт (рис.6 звіту)***

***Зробіть висновок, скільки виявлено фішок (кілець) та який параметр чутливості у вас виявився найкращим. Висновок занесіть у звіт.***

**Завдання 4. Виявлення і вимір круглих об'єктів на зображенні по варіантах двоетапним методом**

***Виконайте ті ж самі дослідження другого методу для зображення обраного згідно вашого варіанту по таблиці 1.***

***Отримані зображення подібні до рис.1 та 2 занесіть у звіт (рис. 7-8 звіту).***

***Зробіть висновки та запишіть їх у звіт.***

-----------------------------------------------------------------------------

**Завдання 5. Сегментація кольорових зображень на основі кластеризації за методом k-середніх (приклад)**

Розглянемо задачу, основною метою якої є автоматична сегментація на основі кластеризації (метод k-середніх) кольорових зображень, представлених в колірному просторі L \* a \* b \*.

Алгоритм роботи:

Крок 1: Зчитування зображення.

Крок 2: Перетворення зображення з колірною системи RGB в колірну систему L\*a\*b\*.

Крок 3: Класифікація кольорів в просторі 'a \* b \*' з використанням кластеризації (метод k-середніх).

Крок 4: Присвоєння міток кожному пікселю зображення на основі методу k-середніх.

Крок 5: Створення сегментованого зображення на основі кольорового.

Крок 6: Сегментація ядер на підставі окремого зображення.

Використовуйте Command Window ППП MATLAB

*Крок 1: Зчитування зображення.*

Завантажимо файл hestain.png (рис.12), який містить медичне зображення гемотоксину і еозину (H & E). Тут застосований метод фарбування для детального аналізу патологій.

%Сегментація зображень.

he=imread('hestain.png');

imshow(he), title('H&E изображение');

text(size(he, 2),size(he, 1)+15,'Image courtesy of Alan Partin, Johns Hopkins University', 'FontSize',7,'HorizontalAlignment','right');

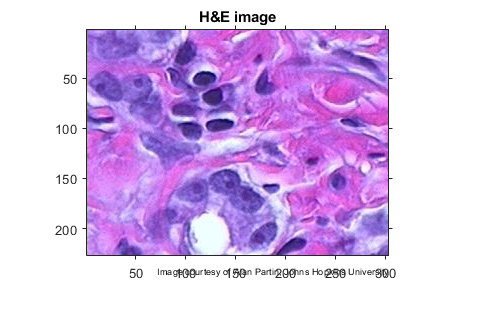


Рис.12

***Отримане зображення подібне до рис.12 занесіть у звіт (рис.9 звіту)***

*Крок 2: Перетворення зображення з колірною системи RGB в колірну систему L\*a\*b\*.*

Яку кількість кольорів видно на зображенні, коли не брати до уваги можливість комбінації яскравостей? Насправді їх три: білий, блакитний і рожевий. Слід зазначити відмінності цих кольорів між собою. Кольорова палітра L\*a\*b\* (вона ще відома як CIELAB або CIE L\*a\*b\*) дає можливість розрізняти ці візуальні відмінності.

Кольорова палітра L\*a\*b\* отримана на основі триколірних значень CIE XYZ. Простір L\*a\*b\* включає інформацію про значення інтенсивності 'L\*', значення кольоровості 'a\*', яке показує який колір вибраний на червоно-зеленій осі і значення кольоровості 'b\*' показує який колір вибраний на блакитно-жовтої осі. Вся інформація про кольори міститься в значеннях 'a\*' і 'b\*'. Оцінити різницю між двома кольорами можна з використанням евклідової відстані.

Перетворимо зображення в колірний простір L\*a\*b\* з використанням функцій makecform і applycform.

cform=makecform('srgb2lab');

lab\_he=applycform(he, cform);

*Крок 3: Класифікація кольорів в просторі 'a\*b\*' з використанням кластеризації (метод k-середніх).*

Кластеризація призводить до поділу об'єктів на групи. Кластеризація методом k-середніх призводить також до локалізації об'єктів в просторі. Пошук поділу, тобто який об'єкт до якого класу належить, відбувається на основі аналізу метричної відстані між об'єктами.

Далі на підставі інформації про кольори в просторі 'a\*b\*', кожного пікселя об'єкта присвоюється значення 'a\*' і 'b\*'. Використаємо кластеризацію методом k-середніх для поділу об'єктів на три кластери. Для цього використовуємо Евклідовому метрику.

ab=double(lab\_he(:, :, 2:3));

nrows=size(ab, 1);

ncols=size(ab, 2);

ab=reshape(ab, nrows\*ncols, 2);

nColors=3;

[cluster\_idx cluster\_center]=kmeans(ab, nColors, 'distance', 'sqEuclidean','Replicates',3);

*Крок 4: Присвоєння міток кожному пікселю зображення на основі методу k-середніх.*

Для кожного об'єкта на оригінальному документі метод k-середніх повертає індекс відповідного кластера. Значення параметра cluster\_center, яке отримано в результаті застосування методу k-середніх буде використано при подальшій демонстрації методу. Відзначимо пікселі, які містяться в cluster\_index (рис.13).

pixel\_labels=reshape(cluster\_idx, nrows,ncols);

imshow(pixel\_labels, []), title('изображение, отмеченное кластерными индексами');



Рис.13

*Крок 5: Створення сегментованого зображення на основі кольорового.*

Використовуючи параметр pixel\_labels, можна розділити об'єкти на зображенні hestain.png по кольорах.

segmented\_images=cell(1, 3);

rgb\_label=repmat(pixel\_labels, [1 1 3]);

for k=1 : nColors

color=he;

color(rgb\_label~=k)=0;

segmented\_images{k}=color;

end

imshow(segmented\_images{1}), title('объекты в кластере 1');

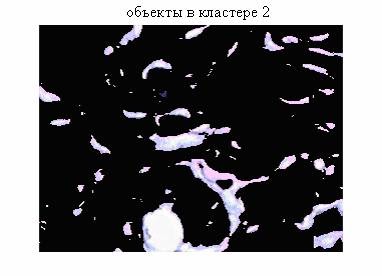
Рис.14



***Отримане зображення подібне до рис.14 занесіть у звіт (рис.10 звіту)***

imshow(segmented\_images{2}), title('объекты в кластере 2');

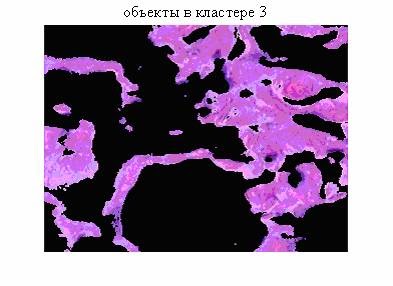
Рис.15



***Отримане зображення подібне до рис.15 занесіть у звіт (рис.11 звіту)***

imshow(segmented\_images{3}), title('объекты в кластере 3');

Рис.16



***Отримане зображення подібне до рис.16 занесіть у звіт (рис.12 звіту)***

*Крок 6: Сегментація ядер на підставі окремого зображення.*

Розглянемо зображення, яке містить сині об'єкти. Відзначимо, що вони є темно-синіми і світло-синіми. Використовуючи значення 'L\*' в колірному просторі L\*a\*b\* можна відокремити темно-сині об'єкти від світло-синіх.

Нагадаємо, що параметр 'L\*' містить значення інтенсивностей для кожного кольору. Знайдемо кластери, які містять сині об'єкти. Отримаємо значення інтенсивностей об'єктів в цьому кластері і опрацюємо їх пороговим методом з використанням функції im2bw.

Параметр cluster\_center містить середнє значення 'a\*' і 'b\*' для кожного кластера. Синій кластер має друге найбільше значення cluster\_center.

mean\_cluster\_val=zeros(3, 1);

for k=1:nColors

mean\_cluster\_val(k)=mean(cluster\_center(k));

end

[mean\_cluster\_val,idx]=sort(mean\_cluster\_val);

blue\_cluster\_num=idx(2);

L=lab\_he(:, :, 1);

blue\_idx=find(pixel\_labels==blue\_cluster\_num);

L\_blue=L(blue\_idx);

is\_light\_blue=im2bw(L\_blue,graythresh(L\_blue));

Використовуємо маску is\_light\_blue для того, щоб позначити ті пікселі, які є частиною синього ядра. Відобразимо сині ядра на розділеному зображенні (рис.17).

nuclei\_labels=repmat(uint8(0), [nrows ncols]);

nuclei\_labels(blue\_idx(is\_light\_blue==false))=1;

nuclei\_labels=repmat(nuclei\_labels, [1 1 3]);

blue\_nuclei=he;

blue\_nuclei(nuclei\_labels~=1)=0;

imshow(blue\_nuclei), title('синие ядра');

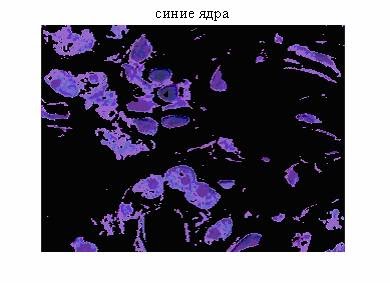


Рис.17

Таким чином підсвічують патології на медичних зображеннях

------------------------------------

**Завдання 6. Сегментація кольорових зображень на основі кластеризації за методом k-середніх**

***Виконайте сегментацію на три кластери зображення обраного згідно вашого варіанту по таблиці 2.*** Використовуйте Command Window ППП MATLAB.

***Отримане початкове зображення занесіть у звіт (рис. 13 звіту).***

***Отримані зображення кластерів подібні до рис.14-16 занесіть у звіт (рис. 14-16 звіту).***

***Зробіть висновки та запишіть їх у звіт.***

Таблиця 2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № за списком у журналі | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Вхідні дані | С1.jpg | С2.jpg | С3.jpg | С4.jpg | С5.jpg | С6.jpg | С7.jpg | С8.jpg | С9.jpg | С10.jpg |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № за списком у журналі | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Вхідні дані | С1.jpg | С2.jpg | С3.jpg | С4.jpg | С5.jpg | С6.jpg | С7.jpg | С8.jpg | С9.jpg | С10.jpg |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № за списком у журналі | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| Вхідні дані | С1.jpg | С2.jpg | С3.jpg | С4.jpg | С5.jpg | С6.jpg | С7.jpg | С8.jpg | С9.jpg | С10.jpg |

Назвіть бланк звіту СШІ-ЛР-9-NNN-XXXXX.doc

де NNN – номер групи

XXXXX – позначення прізвища студента.

Переконвертуйте файл звіту в СШІ-ЛР-9-NNN-XXXXX.pdf

**ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ**

1. За якими ознаками знаходяться фішки на зображенні.

2. До чого приводить кластеризація методом k-середніх.

3. Що включає кольорова палітра L\*a\*b\*.

4. В чому полягає основна мета сегментації.