

О. В. Гороховатський, О. О. Передрій

Харківський національний економічний університет ім. С. Кузнеця
проспект Науки, 9-А, 61166 Харків, Україна

Багатошаровий перцептрон як інструмент первинної кластеризації зображень

Роботу присвячено дослідженню багатошарового перцептрона як інструмента первинного аналізу зображення, зокрема, для пошуку областей інтересу або розбиття на класи. Розглянуто різні реалізації процедури навчання, отримання мережі мінімальної структури, процедури вибору розміру областей, вибору розмірів кроку між областями. Запропоновано коефіцієнт, що дозволяє порівняти матриці з розмітками зображень, наведено приклади його використання. Проведено експериментальні дослідження, які підтвердили ефективність запропонованих методів.

Ключові слова: зображення, багатошаровий перцептрон, ковзне вікно, області інтересу, навчання, коефіцієнт ідентичності, кластеризація.

Вступ

Розвиток методів та алгоритмів штучного інтелекту для аналізу зображень характеризуються значним якісним стрибком останнім часом, унаслідок чого вони все більше і більше впроваджуються в повсякденне життя. Існує значна кількість нейронних мереж різного типу, одним із найбільш простих і канонічних є перцептрон, який було запропоновано у 50–70-х роках ХХ століття.

Традиційним завданням, що пов'язане з обробкою зображень і може бути вирішене за допомогою нейронних мереж, зокрема, перцептрона, є завдання розпізнавання [1–4]. Проте, можливості використання нейромережових підходів не обмежуються лише розпізнаванням, той самий перцептрон також використовують для боротьби із завадами [5], реконструкції [6] і стиснення зображень [7]. У той самий час перцептрон рідко використовують для пошуку областей інтересу на зображеннях, хоча вирішення подібних задач є надзвичайно цікавим.

Незважаючи на високі якісні показники ефективності, яких можна досягти з використанням нейромережових реалізацій, їхнім основним недоліком є значна кількість операцій обчислення, наслідком чого часто є недостатній рівень швидкодії і необхідність використання апаратних можливостей для швидких паралельних обчислень.

Статтю присвячено розробці методу первинного аналізу зображень за допомогою багат шарового персептрона з метою знаходження областей інтересу. Додатковим завданням є також реалізація запропонованого методу у часі, що є близьким до реального.

Багат шаровий персептрон

Однією з найбільш простих нейромережових архітектур є одно шаровий персептрон, однак з огляду на його обмежені можливості при вирішенні лінійно-нероздільних завдань ми будемо використовувати багат шаровий персептрон [8].

Вхідний шар персептрона складається з матриці яскравостей пікселів розміром $n \times n$, яка є частиною зображення або еталонним патерном. Перед подачею на вхід мережі значення яскравостей початкового кольорового зображення конвертуються у сірі тони з подальшим нормуванням значень у межах $[-0,5; 0,5]$.

Як активаційну функцію для кожного з нейронів будемо використовувати класичний варіант сигмоїда:

$$y(x) = \frac{1}{1 + e^{-z}}, \quad z = wx + b,$$

де x — вхідне значення; w — ваговий коефіцієнт; b — зміщення.

Значення, що є виходом j -го нейрона H_j прихованого шару можна записати у вигляді $H_j = f(\sum_i w_{ij} X_i)$, де X_i — значення вхідного сигналу (або значення з виходу нейрона попереднього шару); вихід k -го нейрона Y_k останнього шару мережі можна записати у вигляді: $Y_k = f(\sum_j w_{jk} H_j)$.

Значення вихідного вектора варіюються від 0 до 1 залежно від подібності вхідного фрагмента зображення до одного з еталонних патернів.

Кількість прихованих шарів мережі та кількість нейронів у кожному шарі підбирається експериментальним шляхом.

Мережа навчається з учителем з використанням алгоритму зворотного поширення помилки методом градієнтного спуску (узагальнене дельта-правило) і включає три етапи:

— пряме поширення вхідного сигналу мережею та отримання значень на виході кожного нейрона;

— зворотне поширення помилки;

— корекція вагових коефіцієнтів і зміщень.

Для нейрона k вихідного шару, що є пов'язаним з нейроном j попереднього шару, помилка вираховується як

$$\delta_k = (y_{desired_k} - y_k) y_k (1 - y_k), \quad (1)$$

де $y_{desired_k}$ — потрібне значення виходу нейрона k .

Для нейрона j прихованого шару, що є пов'язаним з нейроном i попереднього шару, помилка вираховується як

$$\delta_j = h_j(1-h_j) \sum_k w_{jk} \delta_k. \quad (2)$$

Корекція коефіцієнтів за зміщень відбувається згідно

$$w = w + \Delta w, \quad \Delta w = \eta \delta x + m \Delta w, \quad b = b + \eta \delta,$$

де η — коефіцієнт швидкості навчання (при реалізації було використано значення $\eta = 0,2$); δ — значення помилки, що отримане згідно (1) чи (2); x — вихідне значення відповідного нейрона з попереднього шару; m — коефіцієнт інерції, що зменшує ймовірність можливого «застрягання» в точці локального мінімуму (експериментальне значення дорівнювало $m = 0,8$).

Критерієм зупинки навчання служить правильне розпізнавання всіх навчальних патернів (у даному випадку це доцільно з огляду на те, що їхня кількість є незначною) та специфіки використання навченої мережі.

Навчальні патерни

Розглянемо процес аналізу зображення за допомогою ковзного вікна, при цьому зображення розбивається на квадратні області розміру, що обробляються незалежно. Очевидно, що, подаючи яскравості пікселів, які знаходяться всередині області, як вхідні значення деякої абстрактної нейронної мережі, логічно очікувати приблизно однакові вихідні значення для схожих областей зображення. Потрібно обрати параметри нейронної мережі таким чином, щоб забезпечити високу якість знаходження однакових областей зображення зі збереженням швидкодії, близької до реального часу, і слабкої залежності від зображення.

Для проведення експериментів розглянемо використання двох наборів навчальних патернів Z_{27} та Z_{13} , перший з яких містить 27 класів (рис. 1), інший містить 13 класів і був отриманий з першого шляхом відкидання тих патернів, які давали незначні значення на виході мережі за результатами досліджень на усіх зображеннях тестового набору (рис. 1).

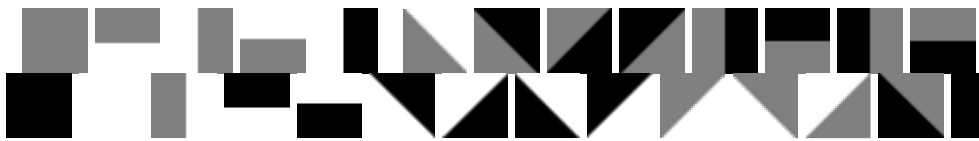


Рис. 1. Навчальні патерни набору Z_{13} (перший рядок) та Z_{27} усі зображення)

Структура мережі та навчання

З практичної точки зору, завжди є доцільним створення нейронної мережі мінімальної структури. Кількість вхідних нейронів визначається розміром $n \times n$ областей, на які розбивається зображення під час обробки методом ковзного вікна. Кількість виходів відповідає кількості навчальних патернів, у експериментальній частині для наборів Z_{27} та Z_{13} ці значення склали 27 і 13 відповідно.

Під час створення структури найбільшою проблемою є підбір кількості нейронів у прихованих шарах.

Розглянемо ітераційний підхід для обрання кількості нейронів в єдиному прихованому шарі мережі, оскільки попередньо виконане моделювання показало, що єдиного прихованого шару достатньо для результативного навчання. Основу цього підходу складає спостереження за процесом навчання та відповідна реакція у вигляді зміни структури мережі у разі, якщо обрані критерії процесу навчання виходять за межі очікуваних.

На початковому етапі створюється мережа з мінімально можливою структурою: кількість вхідних і вихідних нейронів є відомою, прихований шар містить єдиний нейрон.

На кожній ітерації навчання j введемо сумарну помилку навчання:

$$\Delta_j = \sum_L \sum_k (y_{desired_k}^l - y_k^l), \quad (3)$$

де $l \in L$; l — елемент навчального набору L ; y_k^l — вихідне значення нейрона k ; $y_{desired_k}^l$ — потрібне вихідне значення нейрона k .

Критерій $\Delta_j < \Delta_{j+1}$ вказує на можливість невдалого навчання мережі з обраною структурою. Експерименти показують, що поодинокі випадки збільшення відхилення на двох послідовних ітераціях можливі навіть при вдалому виборі структури мережі, відповідно ми вважаємо зміну конфігурації необхідною, якщо помилка збільшується протягом ε ітерацій поспіль, під час експериментального моделювання було використано значення $\varepsilon = 10$.

Потрібно також мати на увазі те, що навчання є більш нестабільним під час перших χ ітерацій, при розмірі області зображення $n = 16; 32; 64$ було використано значення $\chi = 100$, у той час, як процес навчання при $n = 8$ є нестабільним впродовж більшого часу, і це значення було збільшено до $\chi = 1000$.

Розглянемо процедуру навчання мережі для розпізнавання двох наборів патернів Z_{27} та Z_{13} . Для моделювання будемо використовувати дві реалізації персептрона P_{seq} та P_{par} , перша з яких є класичним послідовним навчанням, друга являє собою конкурентне паралельне навчання. Перевагою паралельної реалізації порівняно з послідовною є швидкодія, недоліком — неможливість точного повторення результатів експериментів як наслідок елемента випадковості при навчанні.

Процедура навчання містить обробку всіх еталонних патернів розміру $n \times n$ з набору Z_{27} чи Z_{13} . Критерієм зупинки навчання вважаємо встановлення таких вагових коефіцієнтів нейронної мережі, котрі дозволяють коректно розпізнавати всі еталонні патерни.

Для експериментальних досліджень будемо використовувати два тестових набори зображень D_{SBD} та D_{author} . D_{SBD} являє собою колекцію зображень невеликого розміру (Stanford Background Dataset) [9], для котрих розмітка та розбиття на класи є відомими, D_{author} — набір фотографій з особистої колекції автора великого розміру без попередньої обробки та відомих семантичних класів.

У табл. 1 наведено усереднений за трьома експериментами час навчання та кількість ітерацій навчання для різних умов, різних розмірів областей $n \times n$ та ре-

лізацій методу навчання. Як можна побачити, використання паралельної реалізації дозволяє досягти такого рівня швидкодії, що є значно вищим за традиційне послідовне навчання.

Таблиця 1. Час навчання та кількість ітерацій для різних стратегій

Z_{27}	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$
P_{seq}	50200 / 87 с	1366 / 14 с	975 / 37 с	458 / 189 с	476 / 1200 с
P_{par}	3690 / 7 с	909 / 5 с	856 / 12 с	405 / 55 с	452 / 393 с
Z_{13}	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$
P_{seq}	3969 / 1,24 с	732 / 1,14 с	783 / 3,55 с	282 / 16,54 с	574 / 180,74 с
P_{par}	1941 / 0,7 с	841 / 0,68 с	656 / 1,32 с	306 / 5,78 с	461 / 56 с

Процедура обробки зображення полягає у скануванні методом ковзного вікна розміру $n \times n$ з кроком s , де s — константа. Нормовані значення яскравостей пікселів кожного із вікон подаються на вхід мережі, яка формує вихідний вектор значень. Два вектори одразу групуються в один клас, якщо сумарне абсолютне відхилення є незначним у рамках деякого порогу. У тому випадку, якщо для вектора, що був отриманий, не знайдено близький клас, цей вектор відноситься до нового класу.

У табл. 2 наведено час обробки тестових наборів зображень для різних розмірів областей і реалізації методів навчання. При розбитті зображень на маленькі області ($n = 8$) паралельна реалізація доцільна лише з огляду на швидкість навчання, але не використання навченої мережі.

Безумовно, крім порівняння швидкодії необхідно також порівнювати якість обробки. Таке порівняння буде наведено нижче в табл. 3.

Таблиця 2. Час обробки різних тестових наборів

Z_{27}	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$
P_{seq}	483 с	275 с	98 с	397 с	423 с
P_{par}	2159 с	237 с	97 с	142 с	189 с
Z_{13}	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$
P_{seq}	478 с	128 с	88 с	400 с	408 с
P_{par}	542 с	115 с	57 с	128 с	180 с

Порівняння розміток

Розглянемо процедуру порівняння двох матриць розміток однакового розміру, при цьому кількість класів може мати значні відмінності, і мітки класів можуть не збігатися, наприклад, класу з міткою 1 у першій розмітці може відповідати клас 5 у другій розмітці, і ця відповідність невідома.

Нехай M_1 та M_2 — матриці розміток, а значення на перетині i -ї строки та j -го стовпця вказує на номер класу, якому належить піксель зображення на позиції (i, j) . Нехай також $m_1(i, j)$ та $m_2(i, j)$ — мітки класів матриць M_1 та M_2 у точці (i, j) відповідно. Порівняння еквівалентності двох матриць розміток будемо застосовувати відповідно до введеного коефіцієнта ідентичності $E(M_1, M_2)$.

Розглянемо алгоритм побудови відповідностей класів між двома розмітками (рис. 2). Будемо розглядати заповнення переліку відповідностей, котрий задається у вигляді $\langle m_1, \langle m_2, k \rangle \rangle$, де m_1, m_2 — значення класів, взятих з матриць M_1 та M_2 відповідно; k — кількість пікселів, що відповідають класу із позначкою m_1 у матриці M_1 та m_2 у M_2 одночасно.

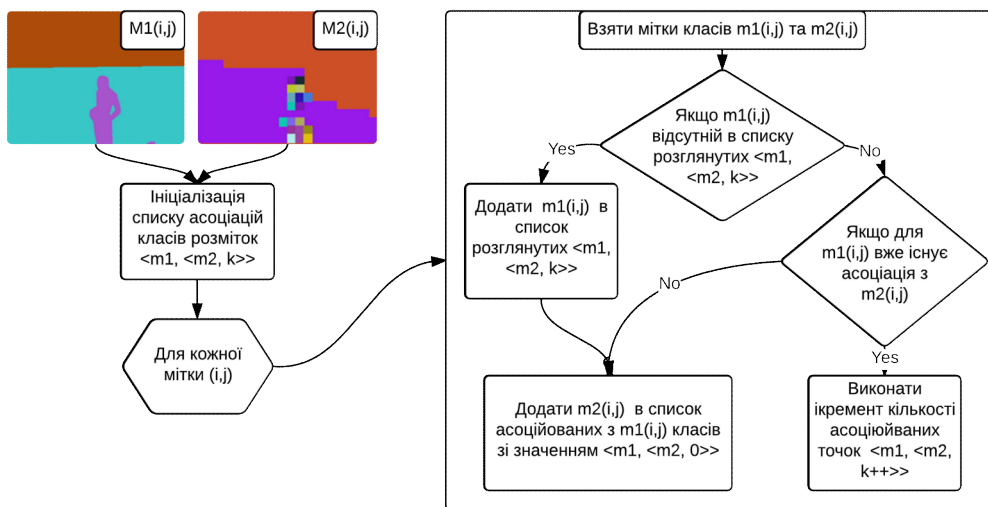


Рис. 2. Опис алгоритму побудови відповідностей класів між матрицями розміток

Оснóву запропонованого алгоритму становить порівняння двох розміток по точках і складання всіх варіацій відповідностей між різними класами для однієї точки.

Підрахунок коефіцієнта ідентичності $E(M_1, M_2)$ після встановлення асоціацій можна описати алгоритмом, що наведений на рис. 3. Результат роботи такого алгоритму буде відображати долю пікселів, які було віднесено до спільного класу (незалежно від мітки класу) в матрицях M_1 та M_2 одночасно. Значення коефіцієнта ідентичності $E(M_1, M_2)$ не є симетричним, тобто $E(M_1, M_2) \neq E(M_2, M_1)$, однак абсолютне відхилення між цими значеннями не є суттєвим.

На рис. 4 наведено графічні порівняння двох розміток на зображеннях тестового набору [9] зі значним показником ідентичності (рис. 4, верхній рядок) і низьким (рис. 4, нижній рядок). Для кожного з прикладів наведено початкове зображення, еталонну (ground-truth) розмітку та результат обробки за допомогою нейронної мережі. Для першого експерименту коефіцієнт ідентичності між еталонною та отриманою розмітками склав $E(M_1, M_2) = 0,74$, у результаті було сформо-

вано 17 класів, для другого прикладу — $E(M_1, M_2) = 0,35$ із 99 класами, що були сформовані нейронною мережею.

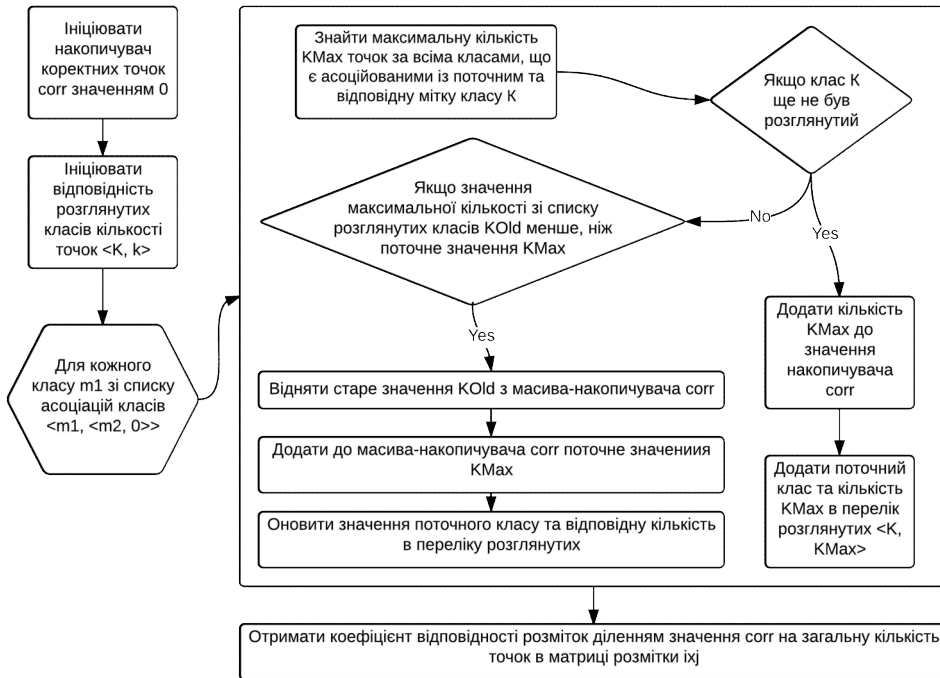


Рис. 3. Опис алгоритму обрахунку коефіцієнта ідентичності $E(M_1, M_2)$



Рис. 4. Порівняння розміток для різних тестових зображень

Обидва експерименти було проведено з використанням нейронної мережі, навченої на областях з розміром $n = 16 : 27$ нейронами на виході та 30 нейронами у прихованому шарі. Як можна побачити, нейронна мережа виокремлює багато класів у присутності складного фону, що не дозволяє отримати високий показник $E(M_1, M_2)$.

Результати експериментів

На рис. 5 наведено початкове зображення з набору [9], розмітку ground-truth (рис. 5, друге зображення) та порівняльні візуалізовані результати розмітки зображення нейронними мережами, що були навчені на наборах патернів Z_{27} (рис. 5, третє зображення) і Z_{13} (рис. 5, останнє зображення) відповідно. В результаті обробки зображення нейронною мережею, що була навчена на наборі Z_{27} , було сформовано 86 класів при коефіцієнті ідентичності з розміткою ground-truth $E = 0,24$. Структуру мережі було побудовано в ручному режимі із 256 нейронами першого шару, 30 прихованого та 27 вихідними.

Використання мережі, що була навчена на множині патернів Z_{13} , дозволило отримати 13 класів з показником $E = 0,47$. Швидкодія обох мереж є однаковою і склала 0,1 с. Структура мережі відрізнялася від попередньої лише кількістю нейронів у вихідному шарі, яких, відповідно, було 13.



Рис. 5. Порівняння розміток, що були оброблені нейронними мережами, навченими на різних наборах патернів

Розглянемо вплив розміру кроку s між областями на показник якості розмітки та швидкодію. В табл. 3 наведено результати, отримані із використанням мереж, що були навчені на наборі Z_{27} . Наведено абсолютний час для обробки всієї колекції зображень D_{SBD} . Як можна побачити, паралельне навчання з кроком $s = 0,9$ дозволяє отримати найкращі результати для всіх проведених експериментів при значенні $n = 16$.

Таблиця 3. Якість розмітки залежно від кроку між областями

D_{SBD}	$n = 8$	$n = 16$	$n = 32$
$s = 1$			
P_{par}	$E = 0,4010$ (87 с)	$E = 0,3626$ (47 с)	$E = 0,2841$ (40 с)
P_{seq}	$E = 0,4281$ (51 с)	$E = 0,3323$ (58 с)	$E = 0,3060$ (57 с)
$s = 0,5$			
P_{par}	$E = 0,3583$ (552 с)	$E = 0,3416$ (130 с)	$E = 0,2517$ (80 с)
P_{seq}	$E = 0,4157$ (173 с)	$E = 0,3157$ (170 с)	$E = 0,2667$ (150 с)
$s = 0,9$			
P_{par}	$E = 0,4058$ (120 с)	$E = 0,3654$ (64 с)	$E = 0,2872$ (45 с)
P_{seq}	$E = 0,4281$ (63 с)	$E = 0,3313$ (71 с)	$E = 0,3114$ (70 с)

Розглянемо якість розмітки та швидкодію двох нейронних мереж P_{par} і P'_{par} , структуру першої з них було встановлено в ручному режимі, структуру другої було отримано за допомогою ітераційної процедури (3). Обидві мережі було навчено в паралельному режимі. Як можна побачити з табл. 4, нейронна мережа з меншою кількістю нейронів може показувати результати, що є співставними з мережею, яка містить надлишкову кількість нейронів.

Таблиця 4. Якість розмітки нейронних мереж різної структури

Z_{27}, D_{SBD}	$n = 8$	$n = 16$	$n = 32$
P_{par}	$E = 0,4058$ (120 с)	$E = 0,3654$ (64 с)	$E = 0,2872$ (45 с)
P'_{par}	$E = 0,3041$ (115 с)	$E = 0,3404$ (45 с)	$E = 0,3183$ (36 с)
Z_{13}, D_{SBD}	$n = 8$	$n = 16$	$n = 32$
P_{par}	$E = 0,4509$ (74 с)	$E = 0,4470$ (50 с)	$E = 0,4233$ (45 с)
P'_{par}	$E = 0,4690$ (68 с)	$E = 0,4581$ (37 с)	$E = 0,4261$ (33 с)

Практичне застосування

Запропонований метод обробки може бути корисним при первинному аналізі зображення, наприклад, для виявлення областей інтересу. Наприклад, на рис. 6 наведено розмітку зображення, де колами виділено області з класами, що мають у своєму складі не більше 5 областей. У цьому випадку було сформовано 66 класів, розмір областей склав 32×32 , розмір зображення — 3264×2448 . Результат отримано із застосуванням нейронної мережі із 1024 нейронами першого шару, 11 нейронами прихованого шару та 27 вихідними нейронами, мережу було навчено паралельним способом, структуру мережі отримано за допомогою ітеративної процедури на основі (3).

На рис. 7 наведено аналогічну розмітку зображення, під час котрої рівно 15 класів повинно бути сформовано. Розмір зображення дорівнював 1872×3328 пікселів, розмір області — 32×32 , була використана та ж сама нейронна мережа, що і в попередньому прикладі. Можна побачити, що використанням цієї мережі можна сегментувати об'єкт на складному фоні, незважаючи на завади типу «блік» у правій нижній частині зображення.

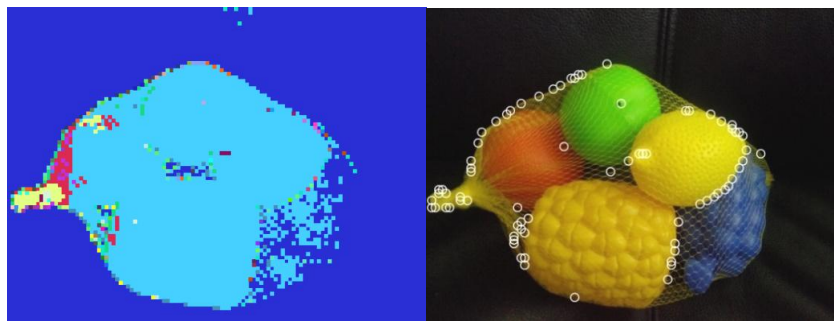


Рис. 6. Візуалізація результатів розмітки користувальницького зображення

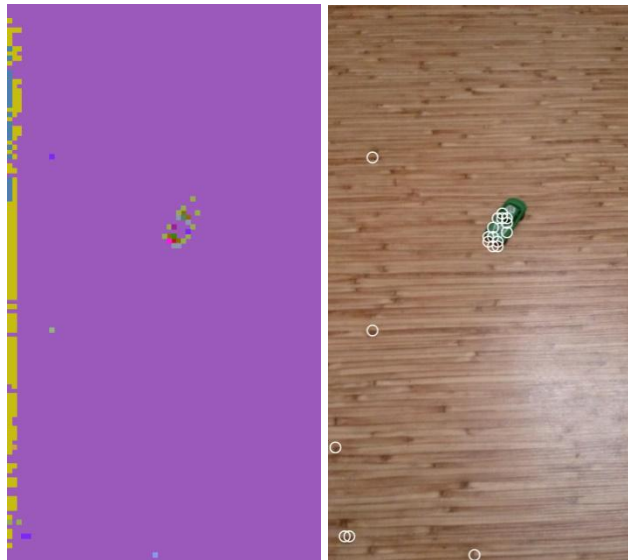


Рис. 7. Візуалізація результатів розмітки користувальницького зображення з обмеженою кількістю класів

Висновки

У результаті дослідження запропоновано метод первинного аналізу зображень за допомогою багатосарового перцептрона, який дозволяє знайти області інтересу та можливі взаємозв'язки між окремими областями зображення. Проведено експериментальні дослідження щодо реалізації процедури навчання послідовним або паралельним способом, вибору розміру областей, вибору розмірів кроку між областями. Введено коефіцієнт, що дозволяє порівняти матриці з розмітками зображень, наведено приклади його використання. Запропоновано алгоритм знаходження мінімальної структури нейронної мережі, експериментальним моделюванням доведено його практичну ефективність. Недоліком роботи є відсутність порівняння запропонованого методу з уже відомими аналогічними методами класифікації, виділення областей і точок інтересу інших авторів.

1. Bhattacharjee Debotosh, Kanti Bhowmik Mrinal, Nasipuri Mita, Basu Dipak Kumar, Kundu Mahantapas. Classification of fused face images using multilayer perceptron neural network. URL: <https://arxiv.org/abs/1007.0633> (Last accessed 13.11.2016).
2. Zhang, Z. Feature-Based Facial Expression Recognition: Sensitivity Analysis and Experiments With a Multi-Layer Perceptron. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*. 1999. Vol. 13(6). P. 893–911.
3. Bhattacharyya S., Ujjwal M. *Soft Computing for Image and Multimedia Data Processing*. Springer Publishing Company, 2013. 267 p. ISBN 978-3-642-40255-5.
4. Sushmita M., Sankar K. Pal, Malay K. Fingerprint Classification using a Fuzzy Multilayer Perceptron. *Neural Computing & Applications*. 1994. Vol. 2. P. 227–233.
5. Harold C. Burger, Christian J. Schuler, Harmeling S. Burger Image denoising: Can plain Neural Networks compete with BM3D? *Proceeding of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 12–21 June 2012.

6. Huang Yu-Len., Chang Ruey-Feng. Error Concealment Using Adaptive Multilayer Perceptrons (MLPs) for Block-Based Image Coding. *Neural Computing & Applications*. 2000. Vol. 9. Issue 2. P. 83–92.

7. Gabriel J., Gomes R.C., Mitra K. Analog multilayer perceptron implementation of Low complexity VQ for image compression. *Proceeding of the 2003 IEEE Conference on Image Processing (ICIP)* (14–17 September 2003). 2003. P. 279–282.

8. Swingler K. Lecture 4: Multi-Layer Perceptrons URL: <http://www.cs.stir.ac.uk/courses/ITNP4B/lectures/kms/4-MLP.pdf> (Last accessed 13.11.2016).

9. Gould, S., Fulton R., Koller D. Decomposing a scene into geometric and semantically consistent regions. *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision* (29 September – 02 October 2009).

Надійшла до редакції 20.11.2016