**ПОШУК ОБ’ЄКТІВ НА ЗОБРАЖЕННЯХ ЗА ДОПОМОГОЮ ВИПАДКОВИХ ЛІСІВ**

**Онищенко О.Ф.**, [elenaonishhenk0@yandex.ru](mailto:elenaonishhenk0@yandex.ru)

*Дніпропетровський національний університет ім. О. Гончара*

Задача розпізнавання об’єктів на зображеннях є однією з основних задач комп’ютерного зору. Є декілька різновидів цієї задачі: класифікація зображень, локалізація об’єктів шуканого класу, знаходження конкретного об’єкту, знаходження «схожих» зображень та ін. Для тренування класифікаторів використовують алгоритми, що базуються на: методі опорних векторів, нейронних мережах, деревах рішень. Широко застосовуються можливості графічних процесорів. Метою цієї роботи є дослідження алгоритму пошуку об’єктів заданого класу за допомогою випадкових лісів.

Усі зображення, тренувальні та тестові, подаються у вигляді масиву каналів I = (I1, I2, ..IС), де перші три – канали RGB, інші – модулі перших і других похідні по x та y, ще 9 каналів – гістограми напрямлених градієнтів[2]

Кожне зображення розбивається на певну кількість шматків (для зображень розміром від 74х35 – 50 шматків) певного розміру (16х16) , де Ii – масив каналів, ci = 1, якщо шматок повністю лежить у межах рамки об’єкту, що належить до шуканого класу, сi = 0, якщо належить фону. di  - вектор, що описує зсув центру шматка відносно центру об’єкту і має зміст коли ci = 1.

Випадковий ліс[4] – множина дерев рішень, що мають наступні особливості:

1. Дерева будуються на різних наборах даних. Якщо вибірка обмежена, наприклад, потрібно m вибірок по n елементів у кожній, але є усього n елементів, деякі екземпляри можна брати декілька разів, або брати дзеркальні відображення. Для побудови дерев використовуються підмножини випадкових ознак для того, щоб дерева не корелювали між собою.
2. Для кожного листа дерева зберігається СL (СL=0 – усі шматки з фону) – пропорція шматків, що належить об’єкту та DL – масив зсувів.
3. Для побудови чергового правила формується набір випадкових піксельних тестів вигляду

Для оцінки якості тестів випадково обирається один з двох критеріїв, що перевіряються на множині А шматків даного вузла дерева:

1. невизначеність класу,
2. невизначеність зсуву, , де dA – середній зсув

Побудова дерева закінчується, якщо число примітивів менше деякого числа (Nmin= 20), або якщо досягнуто максимальна глибина (dmax = 15).

Під час тестування набір шматків Pi тестового зображення пропускається через кожне дерево, на виході маємо вирогідності Сi,j, що центр об’єкту шуканого класу знаходиться у . На практиці будується аналог Хаф-простору V(x), в якому кожен кінцевий лист «голосує» за певний піксель, що відповідає центру об’єкту.

Центри об’єктів відповідають максимумам відображення V(x). Для врахування масштабу об’єктів тестові зображення масштабуються з різними коефіцієнтами s1, s2… і відображення Хафа Vi(x) будуються окремо для кожного si. Результати, таким чином, постають у вигляді векторів (x, s, Vs(x)), центр та границі об’єкту визначаються як .

Програмне забезпечення для дослідження написане мовою С++ з використанням фреймворка Qt та бібліотеки OpenCV.

1. **Р. Гонсалес, Р. Вудс** Цифровая обработка зображений – М.: Техносфера, 2005 – ISBN 5-94836-028-8
2. **N. Dalal, B. Triggs** Histograms of oriented gradients for human detection // *IEEE Conf. Computer Vision and Pattern Recognition* – 2005 – pp 886–893.
3. **J. Gall, V. Lempitsky** Class-specific hough forests for object detection // *IEEE Conf. Computer Vision and Pattern Recognition* – 2009
4. **L. Breiman**. Random forests. *Machine Learning*, 45(1):5–32, 2001.