

УДК 004.925.5

**Ботвін М.М.,**  
(аспірант, Державний університет інфраструктури та технологій)  
**Герцій О.А., к.т.н., доц.**  
(завідувач кафедри «Автоматизація та комп'ютерно-інтегровані технології транспорту», Державний університет інфраструктури та технологій)

## ПОРІВНЯННЯ АЛГОРИТМІВ КЛАСТЕРНОГО АНАЛІЗУ ПРИ РОЗПІЗНАВАННІ ОБ'ЄКТІВ

*В статті проаналізовано перспективи застосування алгоритмів кластерного аналізу в цифровій обробці зображень, зокрема, при сегментації та стисненні графічних зображень, а також при розпізнаванні образів у транспортній сфері діяльності. Проведено порівняльне моделювання алгоритмів кластеризації K-means, Mean-shift та DBSCAN на різних типах даних та зроблено висновки щодо доцільності застосування певного алгоритму в тих чи інших умовах.*

**Ключові слова:** розпізнавання об'єктів, кластерний аналіз, алгоритм K-means, Means-shift, DBSCAN.

**Вступ.** Розпізнавання об'єктів – технологія в області комп'ютерного зору для виявлення та ідентифікації образів на зображенні або відеопослідовності. Люди розпізнають безліч об'єктів на зображеннях без особливих зусиль, незважаючи на те, що зображення об'єктів може дещо відрізнятися в різних точках зору, в багатьох різних розмірах і масштабах або навіть коли вони переводяться або обертаються. Об'єкти можуть бути навіть розпізнані, коли вони частково закриті від очей. Це завдання все ще залишається проблемою для систем комп'ютерного зору. Багато підходів для вирішення цього завдання були реалізовані протягом декількох десятиліть [1]. Сучасне визначення розпізнавання об'єктів таке: область розпізнавання образів пов'язана з автоматичним виявленням закономірностей в даних за допомогою комп'ютерних алгоритмів і за допомогою цих закономірностей для виконання таких дій, як класифікація даних по різних категоріях [2].

В комп'ютерній науці для виявлення меж або розпізнавання об'єктів для поділу цифрового зображення на окремі регіони може використовуватися кластерний аналіз [3].

Кластерний аналіз або кластеризація в задачах ідентифікації – це процес розбиття заданої вибірки об'єктів на непересічні підмножини (кластери) так, щоб кожен кластер складався з схожих об'єктів. Подібність об'єктів визначається на основі метрики, обраної відповідно до критерію кластеризації [4].

При побудові системи розпізнавання кластерний аналіз найбільш часто використовується на етапі попереднього аналізу даних. Його основна мета – визначити, які класи можна виділити на заданій множині образів за наявними ознаками. Саме тому процедури кластерного аналізу називають розпізнаванням без навчання або неконтрольованою класифікацією [5].

**Аналіз останніх досліджень і постановка проблеми.** В цифровій обробці зображень (ЦОЗ) кластерний аналіз знайшов широкий круг застосування: він може бути використаний для сегментації зображень з метою спрощення і/або зміни представлення зображення для полегшення його аналізу [6-8]; для стиснення графічних зображень, що дозволяє зменшити їх інформаційний об'єм з метою збільшення швидкості передачі даних по цифрових каналах зв'язку [9]; при виявленні обличчя на основі алгоритмів кластеризації [10].

**DOI:**10.32703/2617-9040-2020-36-12

В транспортній сфері діяльності кластерний аналіз може бути застосований при розпізнаванні образів для підтримки управління, операцій і моделювання транспортної системи [11], для визначення сценарію в системах підтримки прийняття рішень на основі симуляції трафіку, де процедура кластеризації сценаріїв може підтримувати управління трафіком в реальному часі, що реагує на погоду (WRTM), швидко класифікуючи поточні або прогнозовані погодні умови по заздалегідь визначеним категоріям і пропонуючи відповідні WRTM стратегії, які можна протестувати за допомогою симуляції трафіку в реальному часі перед розгортанням [12], а також, як варіант, алгоритми кластеризації можуть бути використані в системах обробки та візуалізації зображень для оцінки безпеки руху поїздів [13].

Проте якість очікуваних результатів розпізнавання образів в певній мірі залежить від коректного вибору алгоритму кластерного аналізу та його параметрів. А, оскільки, алгоритмів кластеризації на сьогоднішній день відомо дуже багато [14], тому ми розглянемо лише деякі з них.

**Мета і завдання дослідження.** Провести порівняльне моделювання таких алгоритмів кластерного аналізу як K-means (K-середніх), Mean-Shift (кластеризація середнього зсуву) та DBSCAN (основана на щільності просторова кластеризація для додатків з шумами) на різних типах даних та зробити висновки щодо доцільності застосування певного алгоритму в тих чи інших умовах.

**Матеріали та методи дослідження.** Моделювання будемо проводити в середовищі Jupyter Notebook з використанням бібліотеки Scikit-learn.

Алгоритм K-means. Кластеризація K-середніх – один з найпростіших і найбільш часто використовуваних алгоритмів кластеризації. Спочатку вибирається число кластерів  $k$ . Після вибору значення  $k$  алгоритм K-середніх відбирає точки, які будуть представляти центри кластерів. Потім для кожної точки даних обчислюється його евклідова відстань до кожного центру кластера. Кожна точка призначається найближчому центру кластера. Алгоритм обчислює центроїди – центри тяжкості кластерів. Кожен центр ваги – це вектор, елементи якого являють собою середні значення характеристик, обчислені по всіх точках кластера. Центр кластера зміщується в його центр ваги. Точки заново призначаються найближчому центру кластера. Етапи зміни центрів кластерів і перепризначення точок ітеративно повторюються до тих пір, поки межі кластерів і розташування центроїдів не перестануть змінюватися, тобто на кожній ітерації в кожен кластер будуть потрапляти одні і ті ж точки даних [15].

Моделювання (табл. 1, 2) ілюструє роботу алгоритму на синтетичних наборах даних. Суть моделювання така:


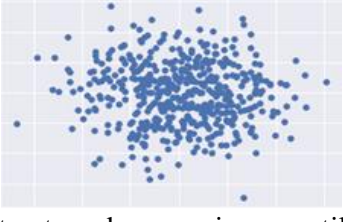
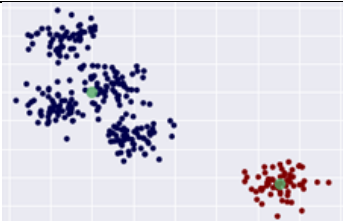
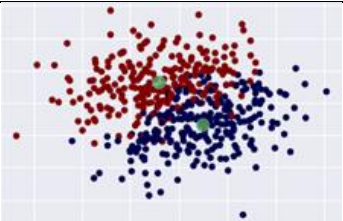
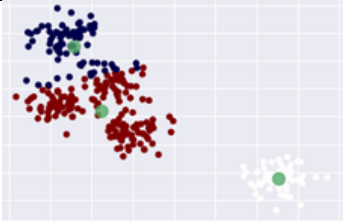
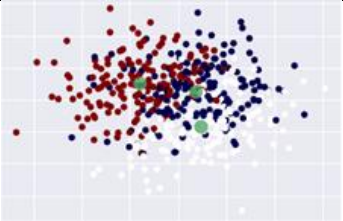
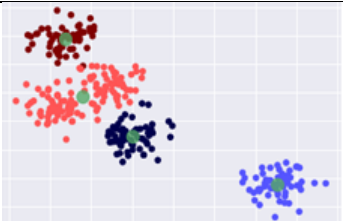
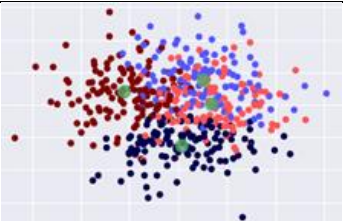
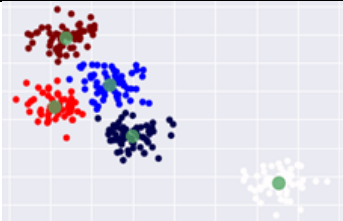
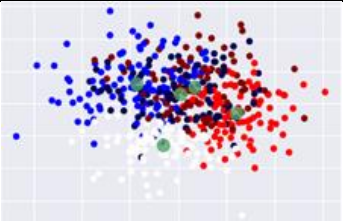
в середовищі Jupyter Notebook за допомогою пакета scikit-learn генеруються чотири набори даних, а саме: `datasets.make_blobs`, `datasets.make_gaussian_quantiles`, `datasets.make_circles` та `datasets.make_moons`;

до кожного типу даних застосовуємо алгоритм K-means, який використовує метрику евклідової відстані, та поетапно збільшуємо число кластерів з 2 до 5, щоб побачити як поводить себе алгоритм на різних типах даних при різному значенні параметра  $k$ .

Примітка. Кількість кластерів не перевищуємо 5, оскільки ми заздалегідь знаємо, що число «правильних» кластерів, наприклад, для даних типу `datasets.make_blobs` дорівнює 5, в інших випадках не більше 2.

З таблиці 1 можна зробити висновок, що алгоритм K-середніх найкраще показав себе до даних типу `datasets.make_blobs` при  $K = 5$ . Кожен кластер визначається виключно його центром, це означає, що кожен кластер має опуклу форму. В результаті цього алгоритм K-середніх може описати відносно прості форми. Крім того, алгоритм K-середніх передбачає, що всі кластери в певному сенсі мають однаковий «діаметр», він завжди проводить межу між кластерами так, щоб вона проходила точно посередині між центрами кластерів. Це іноді може призвести до несподіваних результатів, як показано в табл. 1 на даних типу `datasets.make_gaussian_quantiles`.

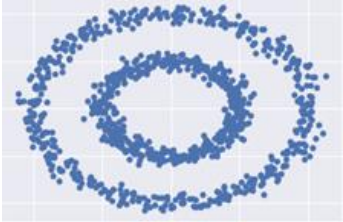
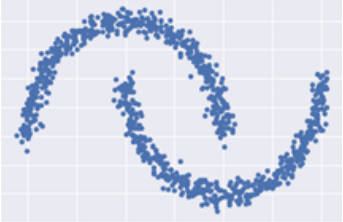
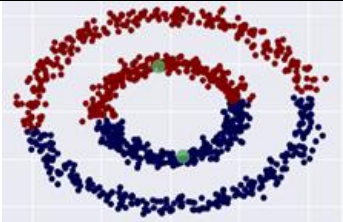
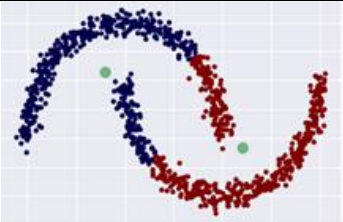
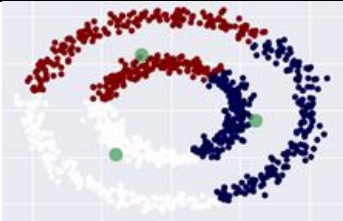
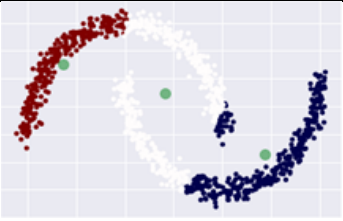
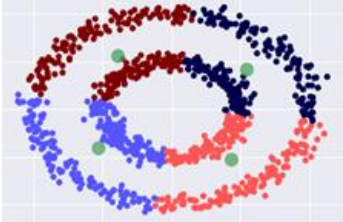
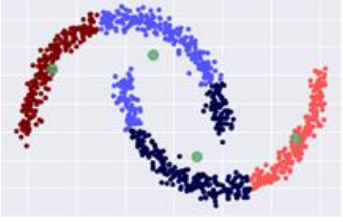
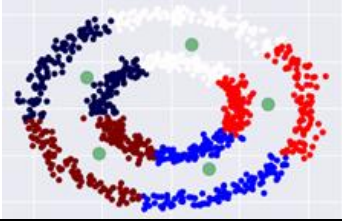
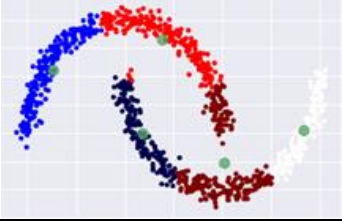
Таблиця 1. Ілюстрація роботи алгоритму К-середніх до штучно-згенерованих даних

Різновид вхідних даних К-сть кластерів	 datasets.make_blobs	 datasets.make_gaussian_quantiles
К = 2		
К = 3		
К = 4		
К = 5		

Також недоліком даного алгоритму є те, що коли ми знаємо «правильну» кількість кластерів для конкретного типу даних, алгоритм не завжди може виділити їх, як це показано в таблиці 2 при  $K = 2$ . Алгоритм К-середніх погано працює, коли кластери мають більш складну форму, як у випадку з даними `datasets.make_circles` та `datasets.make_moons`. В даному випадку ми сподівалися на те, що алгоритм кластеризації зможе виявити два кластери в формі кіл та півмісяців. Однак визначити їх за допомогою алгоритму К-середніх не представляється можливим.

Алгоритм Mean-shift. Алгоритм середнього зсуву вирішує основну проблему градієнтних методів. Основна ідея середнього зсуву полягає в тому, щоб розглядати точки в  $D$ -вимірному просторі ознак як емпіричну функцію щільності ймовірності, де щільні області відповідають локальним максимумам, що лежать в основі розподілу. Градієнтне сходження виконується в просторі ознак при оцінці локальної щільності до збіжності. Після процедури стаціонарні точки відповідають модам розподілу, і одні і ті ж стаціонарні точки вважаються членами одного кластера.

Таблиця 2. Ілюстрація роботи алгоритму К-середніх до штучно-згенерованих даних

Різновид вхідних даних К-сть кластерів	 datasets.make_circles	 datasets.make_moons
К = 2		
К = 3		
К = 4		
К = 5		

Розмір кроку середнього зсуву є адаптивним і залежить від градієнта щільності ймовірності. Градієнт не розраховується, замість цього розраховується більш ефективний вектор середнього зміщення. Вектор середнього зсуву вказує в тому ж напрямку, що і градієнт в градієнтних методах.

Процедура середнього зміщення складається з двох етапів:

- 1) побудова щільності ймовірності в деякому просторі ознак;
- 2) відображення кожної точки на максимум щільності, яка найближче до неї.


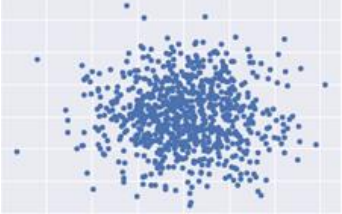
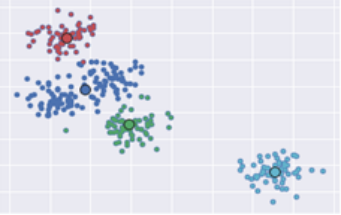
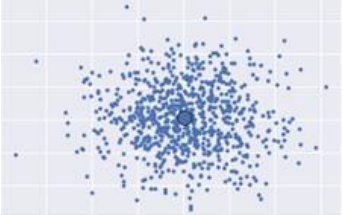
Кожна точка даних зсувається до середньозваженої набору даних. Алгоритм середнього зсуву намагається знайти стаціонарні точки оціненої функції щільності ймовірності [16].

Моделювання проводимо генеруючи ті ж самі чотири типи даних та застосовуємо до них алгоритм Mean-shift.

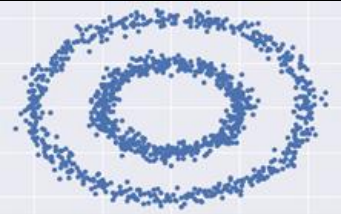
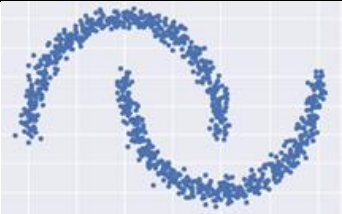
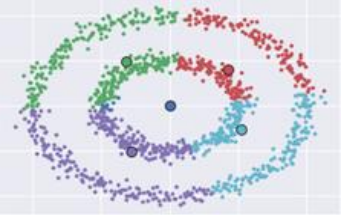
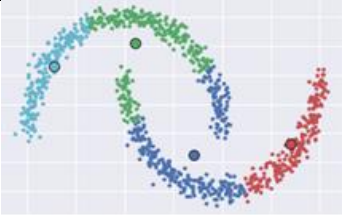
З таблиці 3 бачимо, що на відміну від алгоритму K-means, середнє зміщення Mean-shift не вимагає припущень про кількість кластерів і форми розподілу, але його продуктивність залежить від вибору параметрів масштабу. Смуга пропускання є єдиним параметром для

налаштування, тому для одновимірного випадку це відносно проста процедура, але в багатовимірному випадку це може бути складно. Середнє зміщення може не спрацювати в більш високих вимірах, так як це представлено в таблиці 4.

Таблиця 3. Ілюстрація роботи алгоритму Mean-shift

Різновид вхідних даних		
К-сть кластерів	<code>datasets.make_blobs</code>	<code>datasets.make_gaussian_quantiles</code>
$K_1 = 4$ $K_2 = 1$		
	Розрахункова кількість кластерів $K_1 = 4$	Розрахункова кількість кластерів $K_2 = 1$

Таблиця 4. Ілюстрація роботи алгоритму Mean-shift

Різновид вхідних даних		
К-сть кластерів	<code>datasets.make_circles</code>	<code>datasets.make_moons</code>
$K_1 = 5$ $K_2 = 4$		
	Розрахункова кількість кластерів $K_1 = 5$	Розрахункова кількість кластерів $K_2 = 4$

Алгоритм DBSCAN. Основні переваги алгоритму DBSCAN полягають в тому, що нам не потрібно заздалегідь задавати кількість кластерів, алгоритм може виділити кластери складної форми і здатний визначити точки, які не належать якомусь кластеру. DBSCAN працює трохи повільніше, ніж алгоритм K-means та Mean-shift, але також може масштабуватися на відносно великі набори даних [17].

У DBSCAN кожному зразку (точці) призначається спеціальна мітка, при цьому використовуються такі критерії:

точка розглядається як коренева, якщо, по меншій мірі, вказане число навколишніх точок (MinPts) потрапляє в межі вказаного радіусу  $\epsilon$ ;

гранична точка – це точка, яка має сусідів менше, ніж MinPts в межах  $\epsilon$ , але лежить в межах радіусу  $\epsilon$  кореневої точки;

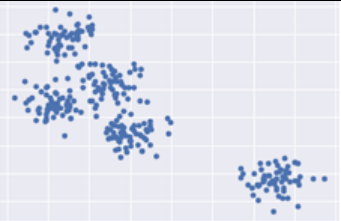
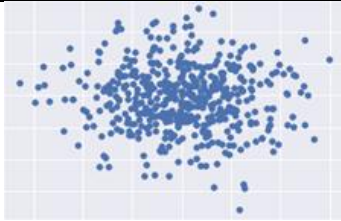
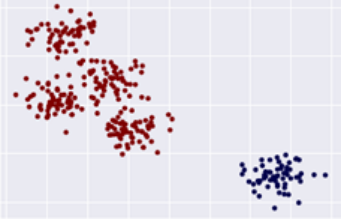
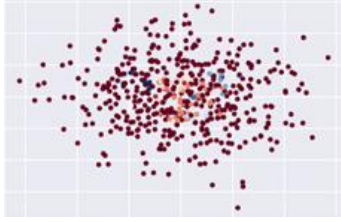
всі інші точки, які не є ні корневими, ні граничними точками, розглядаються як шумові точки.

Після маркування точок як корневих, граничних або шумових алгоритм DBSCAN можна резюмувати в два прості кроки:

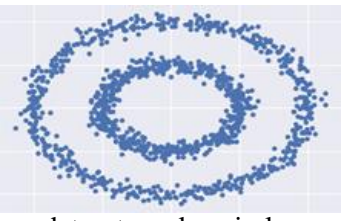
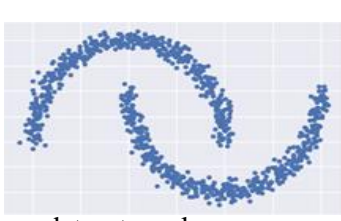
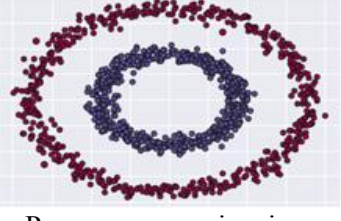
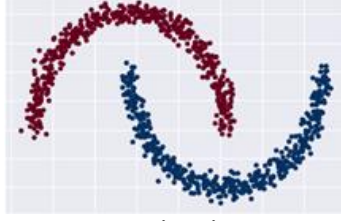
1) сформувати для кожної кореневої точки окремий кластер або зв'язну групу корневих точок (кореневі точки є зв'язковими, в разі якщо вони розташовані не далі, ніж  $\epsilon$ );

2) призначити кожен граничну точку кластеру відповідної кореневої точки.

Таблиця 5. Ілюстрація роботи алгоритму DBSCAN

Різновид вхідних даних	 datasets.make_blobs	 datasets.make_gaussian_quantiles
К-сть кластерів		
$K_1 = 2$ $K_2 = 4$	 Розрахункова кількість кластерів $K_1 = 2$	 Розрахункова кількість кластерів $K_2 = 4$

Таблиця 6. Ілюстрація роботи алгоритму DBSCAN

Різновид вхідних даних	 datasets.make_circles	 datasets.make_moons
К-сть кластерів		
$K_1 = 2$ $K_2 = 2$	 Розрахункова кількість кластерів $K_1 = 2$	 Розрахункова кількість кластерів $K_2 = 2$

Одна з основних переваг використання алгоритму DBSCAN полягає в тому, що він не робить припущення про сферичну форму кластерів, як в алгоритмі K-means. Крім того, алгоритм DBSCAN відрізняється від кластеризації по методу K-means тим, що він з необхідністю не призначить кожному кластеру і одночасно здатний видаляти шумові точки.

Моделювання проводимо по тому ж принципу, що і Mean-shift.

З таблиці 6 можна зробити висновок, що алгоритм DBSCAN може успішно виявляти більш складні фігури, такі як два кільця та півмісяці, що підкреслює одну з сильних сторін алгоритму DBSCAN (кластеризація даних довільної форми). Проте відмітимо деякі недоліки алгоритму DBSCAN. Із зростаючим числом ознак в наборі даних – при заданому фіксованому розмірі тренувального набору – негативний ефект «прокляття розмірності» збільшується. Ця проблема особливо проявляється, якщо ми використовуємо евклідову метрику відстані. Однак проблема «прокляття розмірності» не унікальна для DBSCAN; вона впливає і на інші алгоритми кластеризації, які використовують евклідову метрику відстані, наприклад, алгоритм кластеризації K-means. До того ж в DBSCAN є два гіперпараметра (MinPts і  $\epsilon$ ), які повинні бути оптимізовані з метою отримання хороших результатів кластеризації.

У разі, якщо в наборі даних різниці в щільності відносно великі, знаходження хорошої комбінації параметрів MinPts і  $\epsilon$  може бути проблематичним. Такий негативний результат відображений в таблиці 5.

**Висновки.** Результати моделювання показали, що алгоритм K-means може ефективно описати відносно прості форми. На відміну від нього середнє зміщення Mean-shift не вимагає припущень про кількість кластерів і форми розподілу, але його продуктивність залежить від вибору параметрів масштабу. Алгоритм DBSCAN може успішно виявляти більш складні фігури, що підкреслює одну з сильних сторін цього алгоритму – кластеризація даних довільної форми. На практиці не завжди очевидно, що алгоритм на заданому наборі даних буде виконуватися найкраще, особливо якщо дані надходять в численних розмірностях, ускладнюючи візуалізацію або роблячи її неможливою. Більш того, важливо підкреслити, що успішна кластеризація не залежить тільки від алгоритму і його гіперпараметрів. Швидше, більш важливими можуть бути вибір належної метрики відстані і використання знань в предметній області, які допомагають направити експериментальний пошук найкращої конфігурації в потрібне русло.

### ЛІТЕРАТУРА

1. Giovanni Maria, Farinella Sebastiano, BattiattoRoberto Cipolla. Advanced Topics in Computer Vision and Pattern Recognition. Italy, 2013. 437p.
2. Amita Pal, Sankar K Pal. Pattern recognition and big data. New Jersey, 2017. 862p.
3. Tal Hassner, Ce Liu. Dense Image Correspondences for Computer Vision. Switzerland, 2016. 302p.
4. Селянкі В. В., Скороход С. В. Анализ и обработка изображений в задачах компьютерного зрения: учебное пособие. Таганрог: Изд-во ЮФУ, 2015. 82 с.
5. Чабан. Л.Н. Методы и алгоритмы распознавания образов в автоматизированном дешифрировании данных дистанционного зондирования: учебное пособие. – М.: МИИГАиК, 2016, – 94 с.
6. Neelambike S. Color image segmentation by clustering. *International journal of advanced research in computer science & technology*. 2014. Vol. 2. № 1. P. 95-97.
7. Aqil Burney S.M. K-means cluster analysis for image segmentation. *International Journal of Computer Applications*. 2014. Vol. 96. № 4. P. 8.
8. Nameirakpam Dh. Image segmentation using k-means clustering algorithm and subtractive clustering algorithm. Eleventh international multi-conference on information processing-2015 (IMCIP-2015). India. P. 764 – 771.
9. Xing Wan. Application of K-means Algorithm in Image Compression // AEMCME. 2019. Vol. 5: IOP Conf. Series: Materials Science and Engineering – China 2019.
10. Yousef Farhang. Face Extraction from Image based on K-Means Clustering Algorithms. (IJACSA) *International Journal of Advanced Computer Science and Applications*. 2017. Vol. 8, No. 9.
11. Rajib Saha, Mosammah Tahnin Tariq, Mohammed Hadi, and Yan Xiao. Pattern Recognition Using Clustering Analysis to Support Transportation System Management, Operations, and Modeling. *Journal of Advanced Transportation*. Volume 2019, Article ID 1628417, 12 pages.
12. Ying Chen, Jiwon Kim, Hani S. Mahmassani. Pattern Recognition Using Clustering Algorithm for Scenario Definition in Traffic Simulation-based Decision Support Systems. 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC) October 8-11, 2014. Qingdao, China.

13. Karaman S., Pezzatini D., Bimbo A. A multi-camera image processing and visualization system for train safety assessment. *Multimedia Tools and Applications*. 2018. Vol. 77. № 2. P. 1583–1604.
14. Sergios Theodoridis, Koutroumbas K. An introduction to pattern recognition: a MATLAB approach. / Academic Press, 2010. – 240 p.
15. An Introduction to Machine Learning with Python (O'Reilly) by Andreas C. Mueller and Sarah Guido. Copyright 2017 Sarah Guido and Andreas Mueller, 978-1-449-36941-5.
16. Damir Demirović. An Implementation of the Mean Shift Algorithm // ISSN 2105–1232 с 2019 IPOL & the authors CC–BY–NC–SA.
17. Рашка С. Python и машинное обучение / пер. с англ. А. В. Логунова. - М.: ДМК Пресс, 2017. - 418 с.: ил.

### REFERENCES

1. Giovanni Maria, Farinella Sebastiano, BattiatoRoberto Cipolla. (2013). *Advanced Topics in Computer Vision and Pattern Recognition*. Italy, 437. (in English).
2. Amita Pal, Sankar K Pal. (2017). *Pattern recognition and big data*. New Jersey, 862. (in English).
3. Tal Hassner, Ce Liu. (2016). *Dense Image Correspondences for Computer Vision*. Switzerland, 302. (in English).
4. Selyankin V.V., Skorokhod S.V. (2015). *Analyze i obrabotka izobrajenie v zadachah computernogo zreniya [Analysis and processing of images in computer vision problems]*. Taganrog, SFedU Publ. (in Russian).
5. Chaban L.N. (2016). *Metody i algoritmy raspoznavaniya obrazov v avtomatizirovanom deshifrovanii danyh distancionnogo zonirovaniya: uchebnoe posobie. [Methods and algorithms for pattern recognition in automated decoding of remote sensing data: A study guide]*. Moscow: MIIGAiK. (in Russian).
6. Neelambike S. (2014). Color image segmentation by clustering. *International journal of advanced research in computer science & technology*. Vol. 2, 1, 95-97 (in English).
7. Aqil Burney S.M. (2014). K-means cluster analysis for image segmentation. *International Journal of Computer Applications*. Vol. 96, 4, 8 (in English).
8. Nameirakpam Dh. (2015). Image segmentation using k-means clustering algorithm and subtractive clustering algorithm. *Eleventh international multi-conference on information processing-2015 (IMCIP-2015)*. (pp. 764 – 771). India. (in English).
9. Xing Wan. (2019). Application of K-means Algorithm in Image Compression. *IOP Conference Series Materials Science and Engineerin (AEMCME)*. Vol. 5. (pp. 1-4). China. (in English).
10. Yousef Farhang. (2017). Face Extraction from Image based on K-Means Clustering Algorithms. *International Journal of Advanced Computer Science and Applications*. Vol. 8, 9, 96-107. (in English).
11. Rajib Saha, Mosammat Tahnin Tariq, Mohammed Hadi, and Yan Xiao. (2019). Pattern Recognition Using Clustering Analysis to Support Transportation System Management, Operations, and Modeling. *Journal of Advanced Transportation*. Volume, 2019, Article ID 1628417, 12 pages. (in English).
12. Ying Chen, Jiwon Kim, Hani S. Mahmassani. (2014). Pattern Recognition Using Clustering Algorithm for Scenario Definition in Traffic Simulation-based Decision Support Systems. *IEEE, 17th International Conference on Intelligent Transportation Systems (ITSC) October 8-11, 2014*. (pp. 798-803). Qingdao, China. (in English).
13. Karaman S., Pezzatini D., Bimbo A. (2018). A multi-camera image processing and visualization system for train safety assessment. *Multimedia Tools and Applications*. Vol. 77, 2, 1583–1604. (in English).
14. Sergios Theodoridis, Koutroumbas K. (2010). *An introduction to pattern recognition: a MATLAB approach*. Academic Press, 240. (in English).
15. Andreas C. Mueller and Sarah Guido. (2016). *An Introduction to Machine Learning with Python*. O'Reilly Media, Inc. 978-1-449-36941-5. (in English).
16. Damir Demirovic. (2019). An Implementation of the Mean Shift Algorithm. *IPOL Journal Image Processing On Line*, 9 (2019), 251–268. (in English).
17. Raschka Sebastian. (2015). *Python Machine Learning*. Packt Publishing Ltd: 1<sup>st</sup> edition, 456. (in English).

**Ботвин М.М.,**  
(аспирант, Государственный университет инфраструктуры и технологий)  
**Герций А.А., к.т.н., доц.**  
(заведующий кафедры «Автоматизация и компьютерно-интегрированные технологии транспорта», Государственный университет инфраструктуры и технологий)

### СРАВНЕНИЕ АЛГОРИТМОВ КЛАСТЕРНОГО АНАЛИЗА ПРИ РАСПОЗНАВАНИИ ОБЪЕКТОВ

В статье проанализированы перспективы применения алгоритмов кластерного анализа в цифровой обработке изображений, в частности, при сегментации и сжатии графических изображений, а также при распознавании образов в транспортной сфере деятельности.



*Проведено сравнительное моделирование алгоритмов кластеризации K-means, Mean-shift и DBSCAN на разных типах данных и сделаны выводы относительно целесообразности применения определенного алгоритма в тех или иных условиях.*

**Ключевые слова:** *распознавание объектов, кластерный анализ, алгоритм K-means, Means-shift, DBSCAN.*

**Maksym Botvin,**  
*(PhD-student, State University of Infrastructure and Technology)*  
**Aleksander Gertsy, PhD (Technical Sciences), Associate Professor,**  
*(Head of the Department of Automation and Computer-Integrated Transport Technologies, State University for Infrastructure and Technology)*

### COMPARISON OF CLUSTER ANALYSIS ALGORITHMS IN OBJECT RECOGNITION

*The article is an overview of the direction of graphic image processing based on clustering algorithms. The analysis of prospects of application of algorithms of cluster analysis in digital image processing, in particular, at segmentation and compression of graphic images, and also at recognition of images in transport sphere of activity is carried out. Comparative modeling of such algorithms of cluster analysis as K-means, Mean-Shift (clustering of average shift) and DBSCAN (based on density of spatial clustering for applications with noise) on various types of data is carried out. The simulation was performed on synthetic datasets in a Jupyter Notebook environment using the Scikit-learn library. In particular, four data sets were generated in this environment, to which these clustering algorithms were applied. The simulation results showed that the K-means algorithm can effectively describe relatively simple shapes. In contrast, the mean shift does not require assumptions about the number of clusters and the shape of the distribution, but its performance depends on the choice of scale parameters. The DBSCAN algorithm can successfully detect more complex shapes, which emphasizes one of the strengths of this algorithm - the clustering of arbitrary data. The disadvantages of the selected algorithms are also given and it is indicated on which types of images they effectively work with the estimation of computational speed.*

**Keywords:** *object recognition, cluster analysis, algorithms K-means, Means-shift, DBSCAN.*