

УДК 004.02

## FEATURE DETECTION METHODS IN IMAGE RECOGNITION PROBLEMS ON PYTHON

### МЕТОДИ ВИЯВЛЕННЯ ОЗНАК В ЗАДАЧАХ РОЗПІЗНАВАННЯ ОБРАЗІВ ЗАСОБАМИ МОВИ PYTHON

Yurchenko I.V. / Юрченко І.В.

*c.f.-m.s., as. prof. / к.ф.-м.н., доц.*

ORCID: 0000-0001-9929-5758

*Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Kotsjybynskogo, 12, 58012*

*Чернівецький національний університет, Чернівці, вул.Коцюбинського, 12, 58012*

**Анотація.** У роботі розглядається реалізація алгоритму виявлення ознак HOG; розглянуто побудову класифікаторів для задач машинного навчання; засобами мови Python з використанням бібліотеки Scikit-Learn розроблено покроковий алгоритм розпізнавання образів з використанням методу виявлення ознак HOG та класифікатора SVM; проведено порівняння роботи SVM з найвним Бассовим класифікатором.

**Ключові слова:** гістограма напрямлених градієнтів, алгоритм розпізнавання образів, Scikit-Learn бібліотека.

**Abstract.** It is described an implementation of the “Histogram of Oriented Gradients” (HOG) feature detecting algorithm; the construction of classifiers for machine learning problems is considered and Python language tools (Scikit-Learn library) have developed a step-by-step pattern recognition algorithm using the HOG feature detection method and the SVM-classifier; the work of SVM is compared with the naive Bayesian classifier.

**Key words:** histogram of oriented gradients, pattern recognition algorithm, Scikit-Learn library.

**Вступ.** На етапах постановки задачі машинного навчання [1,2] і формування даних не завжди зрозуміло, які ознаки важливі для побудови оптимального алгоритму, тому часто в даних зустрічається багато надлишкової інформації – шуму. Поява шумових ознак погіршує якість роботи алгоритму й уповільнює його роботу. Тому в більшості випадків перед розв’язанням завдання класифікації, регресії або прогнозування необхідно вибрати ті ознаки, які є найбільш інформативними. Правильний вибір ознак може бути важливішим завданням, ніж зменшення часу обробки даних або поліпшення точності класифікації. Алгоритм виявлення ознак є основним інструментом у задачі розпізнавання образів. У наш час розпізнавання образів активно розвивається у криміналістиці (встановлення особи зловмисника з відео,

отриманого з камер спостереження), медицині (вивчення знімків серця у розрізі, отриманих з МРТ, для діагностики серцевих захворювань), дорожньому русі (визначення марок машин, номерних знаків, встановлення особи водія, дотримання правил дорожнього руху), класифікації документів, розпізнаванні штрих-кодів, розпізнаванні мови тощо.

**Постановка задачі.** Задача розпізнавання образів має дві головні підзадачі: виявлення ознак та класифікація об'єктів. Найбільш поширеними алгоритмами, які використовуються для виявлення ознак з відео чи фотоданих, є гістограма напрямлених градієнтів (HOG), локальні бінарні шаблони (LBP), прискорені стійкі ознаки (SURF). Найкращими класифікаторами для таких задач є метод опорних векторів (SVM), випадковий ліс (RF),  $k$ -найближчих сусідів (KNN) [1,2].

**Опис алгоритму.** Детальніше розглянемо метод виявлення ознак HOG. Гістограма напрямлених градієнтів (англ. Histogram of Oriented Gradients, HOG) – це дескриптори особливих точок, які використовуються в комп'ютерному баченні та обробці зображень з метою розпізнавання об'єктів. Дана техніка заснована на підрахунку кількості напрямків градієнта в локальних областях зображення. У роботі [3] Навніт Далал і Білл Тріггс використовували алгоритм HOG для знаходження пішоходів на статичних зображеннях, хоча згодом розширили область застосування до знаходження людей на відео, а також різних тварин і машин на статичних зображеннях.

Основною ідеєю алгоритму є припущення, що зовнішній вигляд і форма об'єкта на ділянці зображення можуть бути описані розподілом градієнтів інтенсивності або напрямленням країв. Реалізація цих дескрипторів може бути проведена шляхом поділу зображення на маленькі зв'язані області, іменовані осередками, і розрахунком для кожного осередку гістограми напрямлень градієнтів або напрямків країв для пікселів, що знаходяться всередині осередку. Комбінація цих гістограм і є дескриптором. Для збільшення точності локальні гістограми піддаються нормалізації по контрасту. З цією метою обчислюється

міра інтенсивності на великому фрагменті зображення, який називається блоком, отримане значення використовується для нормалізації.

Першим кроком обчислень у багатьох детекторах особливих точок є нормалізація кольору і гамма-корекція. Далал і Тріггс встановили [3], що для дескриптора HOG цей крок можна опустити, оскільки подальша нормалізація дасть той самий результат. Тому на першому етапі розраховуються значення градієнтів. На наступному кроці обчислюються гістограми осередків. Кожен піксель в осередку бере участь у зваженому голосуванні для каналів гістограми напрямків, який ґрунтується на значенні градієнтів. Осередки можуть бути прямокутної або круглої форми, канали гістограми рівномірно розподіляються від 0 до 180 або ж від 0 до 360 градусів, в залежності від того, обчислюється «знаковий» або «беззнаковий градієнт». Далал і Тріггс виявили [3], що беззнаковий градієнт спільно з дев'ятьма каналами гістограми дає кращі результати при розпізнаванні людей. При розподілі ваг у голосуванні вага пікселя може задаватися або абсолютним значенням градієнта, або деякою функцією від нього. Для прийняття до уваги яскравості й контрастності градієнти слід локально нормувати, для чого осередки потрібно згрупувати в більш великі зв'язні блоки. Дескриптор HOG, таким чином, є вектором компонент нормованих гістограм осередків з усіх областей блоку. Кінцевим кроком в розпізнаванні об'єктів з використанням HOG є класифікація дескрипторів за допомогою системи навчання з учителем. Далал і Тріггс використовували метод опорних векторів (SVM, Support Vector Machine) [3].

**Програмна реалізація.** При проведенні досліджень з програмної реалізації алгоритму виявлення обличчя на зображенні у роботі [5] обрано мову Python та бібліотеку Scikit-Learn [4]. Для оцінки якості роботи алгоритму виявлення ознак HOG та класифікатора отримано точність роботи алгоритму, а також встановлено найкращий класифікатор. Для перевірки роботи програми (визначення обличчя на зображенні) використано довільні зображення, отримані з мережі Інтернет. У якості навчальної вибірки вибрано базу даних Wild, яка міститься у Scikit-Learn. Для кращого процесу навчання моделі Wild

доповнено зображеннями з SL бібліотеки, які не містять зображення людських облич. Після виявлення ознак вхідного зображення за допомогою алгоритму HOG у дію вступає класифікатор SVM, який визначає, чи присутнє на зображенні обличчя людини. Класифікатор SVM вибирає вказану кількість пікселів на зображенні, перебираючи таким чином все зображення. На виході отримується те ж вхідне зображення з червоними рамками (у випадку, якщо обличчя на фото присутнє). Також проведено порівняння роботи SVM з наївним Баєсовим класифікатором.

У програмній реалізації було використано класифікатор SVM для класифікації патчів. Патч – це рамка, яка вказує на наявність обличчя на фото. Розглянуто Linear SVM з бібліотеки Scikit-Learn [4], оскільки в порівнянні з SVM він часто має краще масштабування для великої кількості зразків у випадку двовимірних даних. Для початку розглянемо класифікатор NB (наївний алгоритм Байєса [4])

```
In: from sklearn.naive_bayes import GaussianNB
    from sklearn.cross_validation import cross_val_score
    print(cross_val_score(GaussianNB(), X_train, y_train))
Out: array([ 0.9408785 , 0.8752342 , 0.93976823])
```

Як видно з вищенаведеного фрагменту коду, навіть простий наївний алгоритм Байєса дає точність 94%, що є досить хорошим результатом. За допомогою крос-валідації отримано усереднені показники по всій вибірці. Розглянемо далі класифікатор SVM із пошуком сітки за кількома варіантами параметра C (на 1, 2, 4, 8 пікселів):

```
In: from sklearn.svm import LinearSVC
    from sklearn.grid_search import GridSearchCV
    grid = GridSearchCV(LinearSVC(), {'C': [1.0, 2.0, 4.0, 8.0]})
    grid.fit(X_train, y_train)
    print(grid.best_score_)
    print(grid.best_params_)
Out: 0.98667684407744083
     {'C': 4.0}
```

Отже, класифікатор SVM працює з точністю 98,7% з використанням 4 клітинок сітки. Отримаємо далі найкращий класифікатор:

```
In: model = grid.best_estimator_model.fit(X_train, y_train)
Out: LinearSVC(C=4.0, class_weight=None, dual=True,
fit_intercept=True, intercept_scaling=1, loss='squared_hinge',
max_iter=1000, multi_class='ovr', penalty='l2',
random_state=None, tol=0.0001, verbose=0)
```

**Висновки.** У роботі описано покрокову реалізацію алгоритму виявлення ознак HOG; розглянуто побудову класифікаторів для задач машинного навчання; засобами мови Python з використанням бібліотеки Scikit-Learn розроблено покроковий алгоритм розпізнавання образів з використанням методу виявлення ознак HOG та класифікатора SVM; проведено порівняння роботи SVM з наївним Баєсовим класифікатором.

Література:

1. Tuv E., Borisov A., Runger G., Torkkola K.: Feature Selection with Ensembles, Artificial Variables and Redundancy Elimination // The Journal of Machine Learning Research.– 2009.–Vol. 10.– P.1341–1366.
2. Laptin Yu., Likhovid A. P., Vinogradov A.P. Approaches to Construction of Linear Classifiers in the Case of Many Classes // Pattern Recognition and Image Analysis.– 2010.– Vol. 20, No. 2.– P.137–145.
3. Dalal N., Triggs B. Histograms of oriented gradients for human detection // 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA.– 2005.– Vol.1.– P.886–893.– DOI: 10.1109/CVPR.2005.177.
4. Плас Дж. Вандер. Python для сложных задач: наука о данных и машинное обучение. — СПб.: Питер, 2018. — 576 с.
5. Юрченко І.В., Голик Д.Ю. Застосування методів виявлення ознак для машинного навчання засобами мови Python // The I International Science Conference on Multidisciplinary Research (January 19 – 21, 2021, Berlin, Germany). Abstracts of I International Scientific and Practical Conference. Technical Sciences.– PP.1077-1082.

Стаття відправлена: 30.01.2021 г.

© Юрченко І.В.