

I Міжнародна наукова конференція з міждисциплінарних досліджень,  
19 – 21 січня 2021 р., Берлін, Німеччина

Секція – Технічні науки

## **ЗАСТОСУВАННЯ МЕТОДІВ ВИЯВЛЕННЯ ОЗНАК ДЛЯ МАШИННОГО НАВЧАННЯ ЗАСОБАМИ МОВИ PYTHON**

**Юрченко Ігор Валерійович**

кандидат фіз.-мат.наук, доцент  
Чернівецький національний університет імені Юрія Федьковича  
[i.yurchenko@chnu.edu.ua](mailto:i.yurchenko@chnu.edu.ua)

**Голик Дарія Юріївна**

магістр системного аналізу  
Чернівецький національний університет імені Юрія Федьковича  
[d.holyk@chnu.edu.ua](mailto:d.holyk@chnu.edu.ua)

На етапах постановки задачі машинного навчання [1–3] 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 і 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. Наприклад, у медицині знаходження мінімального набору ознак, який є оптимальним для задачі класифікації, може бути корисним для розробки діагностичного тесту. Виявлення важливих ознак (наприклад, відбір генів, що відповідають певному типу раку) може допомогти розшифрувати механізми, що лежать в основі проблеми, яка представляє інтерес для дослідження.

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.

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) [1-3].

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. Цей метод схожий на гістограми напрямлення країв, дескриптори SIFT і контексти форми, але відрізняється тим, що обчислюється на щільній сітці рівномірно розподілених осередків і використовує нормалізацію перекриваючого локального контрасту для збільшення точності.

Навніт Далал і Білл Тріггс, дослідники INRIA, вперше описали гістограму напрямлених градієнтів у своїй роботі на CVPR в червні 2005 року. У роботі [4] вони використовували алгоритм 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. Нормалізовані дескриптори мають кращу інваріантність по відношенню до висвітлення.

Дескриптор HOG має кілька переваг над іншими дескрипторами. Оскільки HOG працює локально, метод підтримує інваріантність геометричних і фотометричних перетворень, за винятком орієнтації об'єкта. Подібні зміни з'являться тільки у великих фрагментах зображення. Більше того, як виявили Далал і Тріггс [4], грубе розбиття простору, точне обчислення напрямків і сильна локальна фотометрична нормалізація дозволяють ігнорувати рух пішоходів, якщо вони підтримують вертикальне положення тіла. Дескриптор HOG, таким чином, є хорошим засобом знаходження людей на зображеннях.

Першим кроком обчислень у багатьох детекторах особливих точок є нормалізація кольору і гамма-корекція. Далал і Тріггс встановили [4,5], що для дескриптора HOG цей крок можна опустити, оскільки подальша нормалізація дасть той самий результат. Тому на першому етапі розраховуються значення градієнтів. Найпоширенішим методом є застосування одновимірної диференціюючої маски в горизонтальному і/або вертикальному напрямку. Цей метод вимагає фільтрації (колірної або складової яскравості) за допомогою фільтруючих ядер  $[-1,0,1]$  та  $[-1,0,1]^T$ .

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 прямокутною або круглою 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 [4,5], 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; в реальних тестах абсолютне значення градієнта дає кращі результати. Іншими можливими варіантами можуть бути квадратний корінь, квадрат або урізане абсолютне значення градієнта.

Для прийняття до уваги яскравості й контрастності градієнти слід локально нормувати, для чого осередки потрібно згрупувати в більш великі зв'язні блоки. Дескриптор НОГ, таким чином, є вектором компонент нормованих гістограм осередків з усіх областей блоку. Як правило, блоки перекриваються, тобто кожен осередок входить більш ніж в один кінцевий дескриптор. Використовуються дві основні геометрії блоку: прямокутні R-НОГ і круглі С-НОГ. Блоки R-НОГ зазвичай є квадратними сітками, що характеризуються трьома параметрами: кількістю осередків на блок, кількістю пікселів на осередок і кількістю каналів на гістограму осередка. В експерименті Далала й Тріггса [4] оптимальними параметрами є блоки 16x16, осередки 8x8 і 9 каналів на гістограму. Більше того, вони виявили, що можна трохи підвищити швидкість обчислень, застосовуючи гаусів фільтр всередині кожного блоку до процедури голосування, що, в свою чергу, знижує вагу пікселів на границях блоків. Блоки R-НОГ виявляються дуже схожими на SIFT-дескриптори; однак, незважаючи на їх схожу структуру, блоки R-НОГ обчислюються на щільних сітках фіксованого масштабу без фіксованого напрямку, в той час як SIFT-дескриптори обчислюються в розріджених, що не є чутливими до масштабу, ключових точках зображення і використовують поворот для вирівнювання напрямку. Крім того, для кодування інформації про форму об'єктів блоки R-НОГ використовуються спільно, в той час як SIFT-дескриптори використовуються окремо.

Блоки С-НОГ мають 2 різновиди: з цілим центральним осередком і розділеною на сектори. Ці блоки можуть бути описані 4 параметрами: кількість секторів і кілець, радіус центрального кільця і коефіцієнт розширення для радіусів інших кілець. Далал і Тріггс дослідили, зокрема, наступні методи нормалізації блоків. Нехай  $v$  – ненормований вектор, який містить всі гістограми даного блоку,  $\|v\|_k$  – його  $k$ -норма при  $k = 1, 2$  і  $e$  – деяка мала константа (точне значення не є важливим). Тоді нормований множник можна отримати одним із наступних способів:

$$L_2 \text{ – норма: } f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}$$

$L_2$  – норма обмежується зверху (значення  $v$  більші за 0,2 вважаються рівними 0,2) і перенормується;

$$L_1 \text{ – норма: } f = \frac{v}{(\|v\|_1 + e)}$$

Далал і Тріггс [4] встановили, що  $L_1$ -норма дає менш надійні результати, ніж інші. 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) [7–9].

При проведенні досліджень з програмної реалізації алгоритму виявлення обличчя на зображенні нами обрано мову Python та бібліотеку Scikit-Learn [6, 7]. 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. База даних Wild містить зображення обличчя людей. Для кращого процесу навчання моделі Wild доповнено зображеннями з SL бібліотеки, які не містять зображення людських облич. Тобто, навчальна вибірка складається з двох частин – позитивна (13233 зображень) та негативна (30000 зображень). 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. Нижче наведено деякі результати роботи програмної реалізації алгоритму виявлення людських облич (Рис.1).

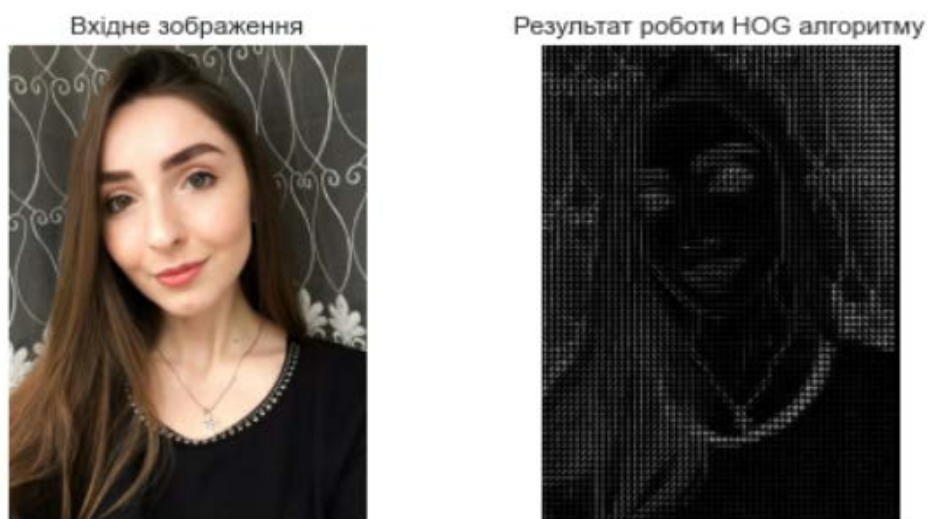


Рис.1.



Рис.2

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 [6,7], 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 [7])

```
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)
```

На Рис.2 наведено результат роботи алгоритму виявлення облич для вхідних тестових зображень.

### Список літератури

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. [http://www.machinelearning.ru/wiki/index.php?title=Машинное\\_обучение\\_\(курс\\_лекций%2C\\_К.В.Воронцов\)](http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_(курс_лекций%2C_К.В.Воронцов))
4. 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.
5. <https://www.learnopencv.com/histogram-of-oriented-gradients/>
6. <https://scikit-learn.org/stable/>
7. Плас Дж. Вандер. Python для сложных задач: наука о данных и машинное обучение. — СПб.: Питер, 2018. — 576 с.: ил.
8. Fung G.M., Mangasarian O.L. Multicategory Proximal Support Vector Machine Classifiers // *Machine Learning.*– Vol.59.– 2005.– P.77–97.
9. Chih-Wei Hsu, Chih-Jen Lin. A Comparison of Methods for Multiclass Support Vector Machines // *IEEE Transactions on Neural Networks.* – 2002, Vol.13, Iss.2. – P.415–425.