

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/281677353>

# System rozpoznawania cyfr oparty na histogramie zorientowanych gradientów

Article · September 2015

---

CITATIONS

0

---

READS

165

1 author:



[Artur Zacniewski](#)

Polish Naval Academy

11 PUBLICATIONS 0 CITATIONS

SEE PROFILE

All content following this page was uploaded by [Artur Zacniewski](#) on 10 November 2015.

The user has requested enhancement of the downloaded file. All in-text references [underlined in blue](#) are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

ZACNIEWSKI Artur<sup>1</sup>

## System rozpoznawania cyfr oparty na histogramie zorientowanych gradientów

### WSTĘP

Mimo wszechobecnych nowych technologii, odręczne pismo utrzymuje się jako środek komunikacji międzyludzkiej oraz sposób zapisu informacji w codziennym życiu. Rozpoznawanie pisma odręcznego przez maszyny ma praktyczne znaczenie m.in. przy odczycie notatek w smartfonie, adresów na kopertach pocztowych, kwot pieniędzy na czekach bankowych, na formularzach w urzędach itp. [4].

Rozpoznawanie pisma odręcznego w sposób automatyczny to problem bardzo trudny i od wielu lat nie do końca jeszcze rozwiązany. Złożoność tego zagadnienia potęguje przede wszystkim fakt, iż dany znak może być napisany różnymi charakterami pisma, więc litery za każdym razem mogą wyglądać nieco inaczej. Istnieje wiele algorytmów dotyczących rozpoznawania charakteru pisma i pojedynczych słów, a także szereg aplikacji z m.in. weryfikacją podpisu lub autentykacją piszącego [2].

W artykule przedstawiono zarys zasady działania deskryptorów obrazu HOG oraz klasyfikatora SVM, a także pokazano algorytm rozpoznawania odręcznie pisanych cyfr przy wykorzystaniu ww. technik.

Przeprowadzono badania, których celem była realizacja rozpoznawania odręcznie pisanych znaków z wybranych obrazów. Do implementacji przedstawionego algorytmu użyto bibliotek OpenCV, mahotas oraz scikit-image.

### 1. HISTOGRAMY ZORIENTOWANYCH GRADIENTÓW HOG

Histogramy zorientowanych gradientów HOG (ang. *Histograms of Oriented Gradients*) to deskryptory obrazu, które opisują kształt i pomagają w znalezieniu w obrazie konkretnego obiektu. Idea ich działania opiera się na zliczaniu występowania gradientów występujących w tej samej orientacji przestrzennej (pod określonym kątem), w pewnym ściśle określonym fragmencie obrazu. Gradienty te są liczone w równomiernie rozmieszczonych fragmentach obrazu, w przeciwieństwie do innych deskryptorów, takich jak np. SIFT (ang. *Scale-Invariant Feature Transform*). Aby poprawić skuteczność wykrywania obiektów stosuje się lokalną normalizację kontrastu w nakładających się na siebie regionach [1][3].

Deskryptory HOG są uzyskiwane poprzez podzielenie obrazu na małe fragmenty (komórki), a następnie dla każdej komórki obliczany jest histogram występowania orientacji krawędzi. Połączenie tych histogramów, obliczonych dla wszystkich komórek jest deskryptorem obrazu HOG [1][3].

Deskryptor HOG jest odporny na wszystkie transformacje geometryczne, poza rotacją. Obraz zawierający poszukiwany obiekt, po zastosowaniu na nim dozwolonej transformacji nadal będzie zawierał cechy deskryptora HOG, pozwalające na wskazanie obecności poszukiwanego obiektu [3].

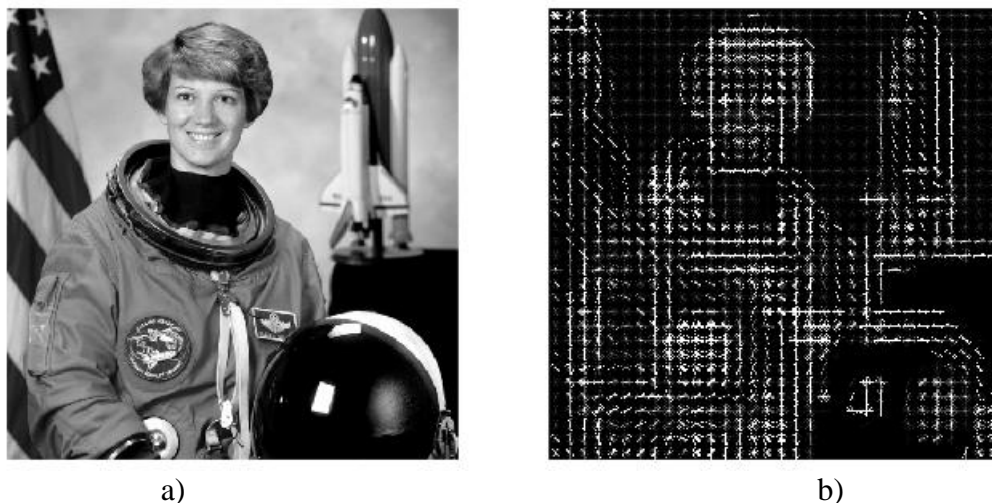
W [1] pokazano, że wstępne przetwarzanie obrazów nie pozwala na osiągnięcie lepszej jakości działania deskryptorów HOG, natomiast z powodzeniem stosowana jest normalizacja kontrastu.

Po fazie obliczenia gradientów dla każdej komórki tworzony jest histogram, który przedstawia rozkład tychże gradientów lub inaczej rozkład orientacji krawędzi we wszystkich pikselach w

<sup>1</sup>Akademia Marynarki Wojennej w Gdyni, Wydział Nawigacji i Uzbrojenia Okrętowego; 81-103 Gdynia; ul. Śmidowicza 69, Tel/fax: + 48 58 626-27-78, a.zacniewski@amw.gdynia.pl

komórce. Komórki obrazu są następnie grupowane w większe bloki, po to aby przeprowadzić operację normalizacji kontrastu [3].

Na rysunku 1.1 przedstawiono realizację HOG (jedna komórka na blok, 16x16 pikseli w komórce, 8 kanałów) dla przykładowego obrazu.



**Rys. 1.1.** Histogram zorientowanych gradientów: a) obraz wejściowy, b) histogram zorientowanych gradientów HOG dla obrazu wejściowego [3]

Deskryptor obrazu HOG to wektor, którego elementami są obliczone histogramy we wszystkich blokach. W pracy [1] zaproponowano dwie geometrie przestrzenne bloków:

- R-HOG (ang. *Rectangular HOG*), gdzie określa się liczbę komórek tworzących jeden blok, liczbę pikseli w pojedynczej komórce oraz liczbę tzw. kanałów (czyli przedziałów klasowych) histogramu;
- C-HOG (ang. *Circular HOG*), w której spotyka się wariant z centralną, pojedynczą komórką lub wariant z centralną komórką podzieloną kątowo. Określa się liczbę kątowych przedziałów, liczbę przedziałów promieniowych, promień centralnego przedziału oraz współczynnik ekspansji dla promienia z dodatkowymi przedziałami kątowymi [1][3].

## 2. KLASYFIKATOR SVM

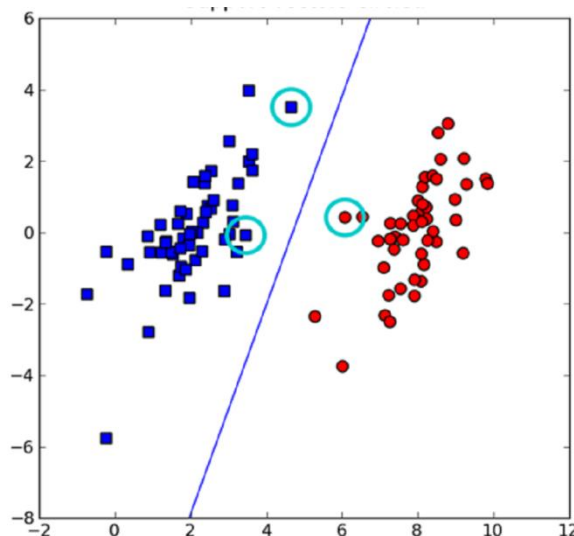
Klasyfikator SVM (ang. *Support Vector Machine*) to technika uczenia maszynowego, która pozwala na analizę danych oraz rozpoznawanie wzorców w celu klasyfikacji. SVM to klasyfikator, który jest binarny, liniowy i nieprobabilistyczny. W najprostszym przypadku SVM (po fazie uczenia) pozwala na określenie do której z dwóch klas należy zbiór danych wejściowych. Uzyskany po fazie uczenia model SVM reprezentuje dane ze zbioru uczącego, oddzielone od siebie granicą z możliwie szerokim marginesem. Punkty spoza zbioru uczącego są klasyfikowane poprzez określenie, po której stronie granicy znajduje się określony punkt i tym samym przypisywana jest mu przynależność do określonej klasy [2].

SVM w oryginalnej postaci jest klasyfikatorem binarnym, aczkolwiek istnieją odmiany pozwalające na klasyfikację wieloklasową. Przykładem może być przypisanie danemu znakowi konkretnej cyfry lub litery, co stosowane jest np. w rozpoznawaniu tekstu. Problem wielu klas w przypadku SVM rozwiązuje się poprzez podział problemu wieloklasowego na problemy dwuklasowe. Stosowane są dwa podejścia:

- *one-versus-one*, w którym tworzone są wszystkie dwuelementowe kombinacje klas i dla każdej pary uczony jest osobny klasyfikator. Rezultatem klasyfikacji jest etykieta tej klasy, która była wynikiem klasyfikacji największą liczbę razy;
- *one-versus-all*, w którym trenowanych jest tyle klasyfikatorów, ile danych jest klas. Dla każdego klasyfikatora dane podzielone są na dwie klasy: jedna z danymi z pierwotnego zbioru uczącego, a

druga z danymi z pozostałych klas. Konkretna próbka analizowana jest za pomocą wszystkich wytrenowanych klasyfikatorów, a wynikiem jest klasa która została wybrana z największym wskazaniem przez przyporządkowany jej klasyfikator [2].

Na rysunku 2.1 przedstawiono ideę klasyfikacji dla klasyfikatora SVM. Zakreślone punkty to tzw. wektory nośne (ang. *supportvectors*), a linia dzieląca zbiór danych to granica, zwana hiperpłaszczyzną (ang. *hyperplane*) [2][5].



Rys. 2.1. Idea klasyfikacji dla klasyfikatora SVM [2][4]

### 3. BADANIA

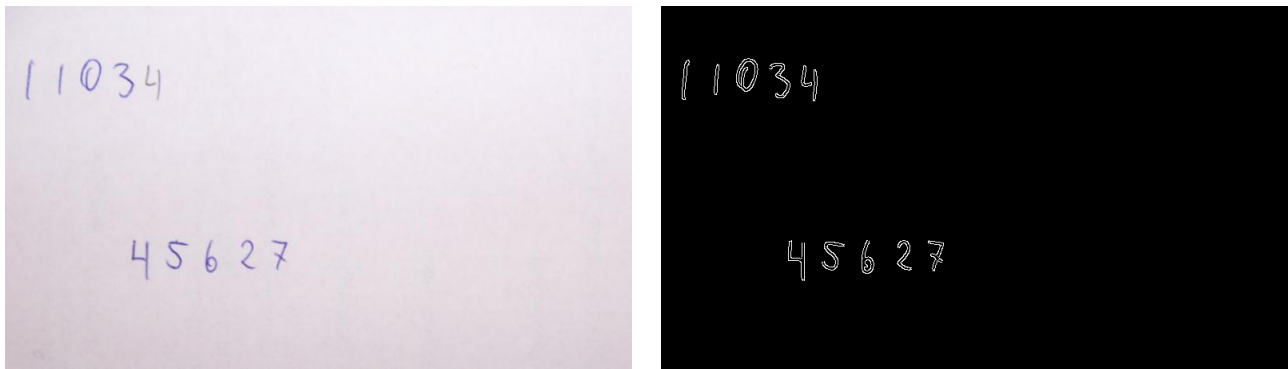
Celem badań było realizacja procesu rozpoznania odręcznego pisma, przy pomocy deskryptora obrazów HOG oraz klasyfikatora SVM. Poniżej przedstawiono poszczególne kroki zastosowanego algorytmu wraz z rezultatami otrzymanymi po każdym z etapów.

#### 3.1 Faza uczenia

- Zbiór uczący to podzbiór bazy MNIST (ang. *Mixed National Institute of Standards and Technology*), dostępny na stronie <http://yann.lecun.com/exdb/mnist/>. Zawiera on odręcznie napisane cyfry i jest powszechnie dostępny w celach badań nad rozpoznawanie znaków. Każdy obraz ma wymiary 28x28 pikseli i jest dostępny w skali szarości. Obrazy są wyśrodkowane poprzez obliczenie środka ciężkości pikseli, co może mieć wpływ na współczynnik błędów.
- Ponieważ każdy ma inny charakter pisma (np. pochylenie znaków w różne strony), należy przed procesem uczenia przystosować dane ze zbioru uczącego. Opracowana została funkcja, bazująca na momentach obrazu i transformacjach afinicznych (ang. *affinetransformations*). Jej zadanie to odpowiednia transformacja obrazu przed zastosowaniem deskryptora HOG i przed fazą uczenia.
- Do uczenia liniowego modelu SVM wybrano model LinearSVC (ang. *LinearSupportVectorClassification*), co jest zalecane w [6], jako metoda bardziej elastyczna niż np. SMO [2]. Po procesie uczenia model zostaje zapisany do pliku, w celu późniejszego wykorzystania w procesie klasyfikacji.

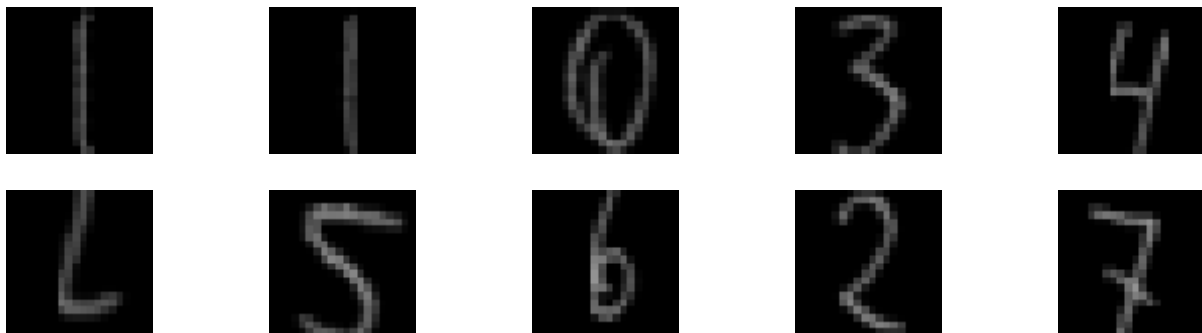
#### 3.2 Proces klasyfikacji

- W procesie klasyfikacji użyto modelu wygenerowanego w procesie uczenia, natomiast deskryptor HOG został zastosowany z identycznymi parametrami jak w pierwszej fazie.
- Rysunek 3.1.a przedstawia obraz zawierający odręcznie napisany szereg cyfr, na którym będą dokonywane dalsze operacje. Obraz ten jest konwertowany do obrazu ze skalą szarości, następnie dodawany jest szum gaussowski, po czym stosowany jest detektor krawędzi Canny'ego, co zobrazowano na rysunku 3.1.b. Każdy ze znalezionych konturów reprezentuje cyfrę, która poddana będzie klasyfikacji.



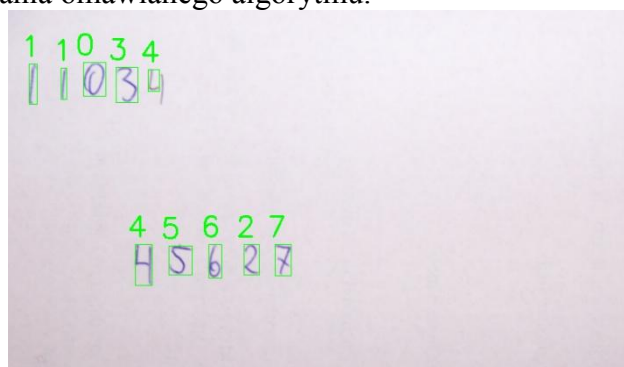
**Rys. 3.1.** Obraz zawierający odręczne pismo: a) obraz wejściowy, b) obraz po zastosowaniu detektora krawędzi Canny'ego [opracowanie własne]

- c) Dla każdego ze znalezionych konturów tworzony jest odpowiedni prostokątny obszar (tzw. *boundingbox*), sprawdzany jest jego rozmiar i jeśli nie jest zbyt mały tworzony jest tzw. obszar zainteresowania ROI (ang. *Region Of Interest*).
- d) W ww. obszarze znajduje się cyfra gotowa do klasyfikacji. Wcześniej jednak stosowane jest progowanie przy pomocy algorytmu Otsu, które pozwala na odseparowanie cyfry od tła, na którym cyfra ta jest zapisana[3][7]. Na rysunku 3.2 pokazano obszar zainteresowania ROI dla każdej cyfry z analizowanego obrazu po zastosowaniu algorytmu Otsu.



**Rys. 3.2.** Obszary zainteresowania ROI dla analizowanego obrazu [opracowanie własne]

- e) Ponownie stosowana jest metoda opisana w punkcie 3.1.b, po to by zapewnić właściwą transformację obrazu.
- f) Obliczany jest wektor cech deskryptora HOG, który przy pomocy klasyfikatora LinearSVC pozwala na predykcję cyfry znajdującej się w obszarze ROI. Na rysunku 3.3 przedstawiono pozytywny rezultat działania omawianego algorytmu.



**Rys. 3.3.** Efekt końcowy algorytmu rozpoznawania odręcznie pisanych cyfr [opracowanie własne]

- g) Kolorem zielonym zaznaczono predykcje poszczególnych cyfr w analizowanym obrazie. Nie zawsze jednak algorytm pozwalał na osiągnięcie zamierzonych rezultatów. Na rysunku 3.4 pokazano



niewłaściwą klasyfikację wybranego ciągu odręcznie pisanych cyfr. Cyfra '3' z analizowanego obrazu została sklasyfikowana jako '4', natomiast cyfra '0' jako '9'.



**Rys. 3.4.** Nieprawidłowe rozpoznanie cyfr: a) osiągnięty rezultat, b) niewłaściwie sklasyfikowana cyfra '3', c) niewłaściwie sklasyfikowana cyfra '0' [opracowanie własne]

Współczynnik prawidłowych klasyfikacji dla 200 obrazów z ręcznie pisanymi cyframi wyniósł 84%, co jest wynikiem świadczącym o dobrych właściwościach analizowanego klasyfikatora. Przyczyną niewłaściwej predykcji może być za mały lub źle dobrany zbiór uczący, nieodpowiednie parametry zastosowanych metod lub deskryptorów itd.

## WNIOSKI

Opisany i zastosowany algorytm rozpoznawania odręcznego pisma w przetwarzanym obrazie jest stosunkowo prosty, aczkolwiek bardzo elastyczny i pozwalający na różnego rodzaju modyfikacje w szerokim zakresie. Wykorzystywane są znane i popularne algorytmy (takie jak m.in. algorytm progowania Otsu, algorytm detekcji krawędzi Canny'ego itd.), które pozwalają na szeroki wachlarz kombinacji dotyczących przetwarzania obrazu i rozpoznawania pisma.

Istnieje szereg sposobów na modyfikację opisywanego algorytmu. Na pewno wart rozważenia jest inny sposób centralizacji obrazu w przypadku obrazów wchodzących w skład zbioru uczącego (informacja takowa znajduje się na ww. stronie zawierającej dane wykorzystywane w procesie uczenia). Każda z metod wykorzystywanych w realizowanym algorytmie posiada szereg parametrów do modyfikacji, co również pozwala na szerokie spektrum dalszych analiz i obliczeń.

Pokazany i zastosowany algorytm może stanowić podstawę do dalszych badań dotyczących rozpoznawania odręcznego pisma przy wykorzystaniu deskryptorów obrazu i wybranych klasyfikatorów. Dopuszczalny jest cały szereg modyfikacji, w wyniku których możliwe będzie osiągnięcie jeszcze bardziej zadowalających wyników.

## Streszczenie

W artykule pokazano możliwości użycia odręcznego pisma oraz potrzebę dalszych badań nad jego rozpoznawaniem. Przedstawiono krótką charakterystykę deskryptorów obrazu HOG oraz liniowych klasyfikatorów SVM wraz z przykładowymi zastosowaniami.

Przedstawiony został algorytm, który pozwala na rozpoznanie i klasyfikację odręcznie zapisanych cyfr za pomocą wymienionych metod wraz z technikami takimi jak algorytm progowania Otsu i algorytm detekcji krawędzi Canny'ego. Pokazano praktyczny przebieg procesu uczenia i klasyfikacji przy zastosowaniu omawianych narzędzi.

Wskazane zostały wady i zalety zastosowanej metody oraz pokazano możliwości poprawy i rozwoju algorytmu rozpoznawania odręcznego pisma poprzez zmodyfikowanie parametrów zastosowanych technik uczenia maszynowego i rozpoznawania obrazów.

**Słowa kluczowe:** deskryptor obrazu HOG, klasyfikator SVM, rozpoznawanie odręcznego pisma.

## Digits recognition system based on Histogram of Oriented Gradients

### Abstract

In article the possibility of using handwriting and the need for further research on its recognition were shown. The characteristics of the image HOG descriptors and linear SVM classifier with examples of applications were described.

Algorithm that allows the diagnosis and classification of handwritten digits by using these methods with

techniques such as Otsu thresholding algorithm and Canny edge detection algorithm was presented. The practical process of learning and classification using mentioned tools was shown.

Advantages and disadvantages of the used method were indicated and opportunities for improvement and development of handwriting recognition algorithm by modifying the parameters of the techniques of machine learning and pattern recognition were shown.

**Keywords:** Histogram of Oriented Gradients, Support Vector Machines, recognition of handwritten digits.

## BIBLIOGRAFIA

1. N. Dalal, B. Triggs, „*Histograms of oriented gradients for human detection*”, *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*, San Diego, USA, 2005.
2. P. Harrington, „*Machine Learning in Action*”, Manning Publications, New York, 2012.
3. P. Kamal, F. Rahman, „*A Robust Authentication System Handwritten Documents using Local Features for Writer Identification*”, *Journal of Computing Science and Engineering*, Vol. 8, No. 1, March 2014, pp. 11-16.
3. OpenCV, <http://docs.opencv.org/>, oficjalna dokumentacja biblioteki OpenCV, dostęp luty 2015.
4. R. Plamondon, „*Online and off-line handwriting recognition: a comprehensive survey*”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume:22, 2000, pp. 63-84.
5. J.E. Solem, „*Programming Computer Vision with Python*”, O'Reilly, Sebastopol, 2012.
6. S. van der Walt et al., „*scikit-image: Image processing in Python*”, oficjalnadokumentacjabibliotekiscikit-image, dostęp luty 2015.
7. R. Woods, R. Gonzalez, „*Digital Image Processing*”, Pearson, 2010.