

Domaći zadatak 2 iz OAiS DSP 2

Radomir Zlatković, RA19/2018 i Aleksa Heler, RA22/2018

Apstrakt—Implementacija HOG algoritma i našeg posebnog algoritma prepoznavanja cifara sa slike, koristeći izdvajanje ivica i računanje gradijenta ivice kako bi se odredila obeležja cifre u slici.

Ključne reči—Canny Edge Detection, Sobel Edge Detection, Histogram of Oriented Gradient (HOG)

I. UVOD

Projekat je započet kao drugi po redu grupni domaći zadatak iz predmeta Osnovi Algoritama i Struktura DSP 2. Ceo projekat je implementiran u C++ jeziku.

II. METOD I OPIS FUNKCIONALNIH CELINA HOG ALGORITMA

U nastavku su opisani koraci HOG algoritma koji se koristi kao referenca i poređenje za naš algoritam.

a. Prelazak iz RGB u YUV domen boja

Kako nas interesuje samo *luminance* komponenta slike, izdvajamo samo Y deo YUV domena.

b. Sobel Edge Detection

Koristi se *Sobel* algoritam za detektovanje ivica na slici čime dobijamo horizontalni i vertikalni gradijent, koji koristimo da izračunamo globalni gradijent:

$$G = \sqrt{G_x^2 + G_y^2}$$

Računanje orijentacije gradijenta za svaki piksel:

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$

c. Računanje histograma (po blokovima)

Izračunati gornje granice podopsega za histogram:

$$gr_i = \frac{2 * \pi * (i+1)}{histN}$$

Gde gr_i predstavlja gornju granicu opsega, i redni broj opsega, a $histN$ ukupan broj podopsega (dužinu histograma).

Zatim za svaku tačku iz bloka proveriti kom opsegu uglova pripada njena orijentacija Θ . Potom odgovarajuću vrednost u histogramu uvećati za vrednost inzenziteta gradijenta G koji odgovara datoj tački.

d. Formiranje vektora obeležja

Konačan vektor obeležja formira se nadovezivanjem izračunatih histograma za sve blokove. Takođe je ovde potrebno normalizovati sve vrednosti histograma deljenjem sa sumom gradijenata.

III. METOD I OPIS FUNKCIONALNIH CELINA NAŠEG ALGORITMA

a. Prelazak iz RGB u YUV domen boja

Kao i u HOG algoritmu, i ovde nas zanima samo *luminance* komponenta slike.

b. Canny Edge Detection

Za razliku od referentnog algoritma, mi smo se odlučili za Canny detekciju ivica kako se u preliminarnom testiranju pokazalo kao obećavajuća zamena. U okviru funkcije detekcije ivica je takođe odrađeno i čuvanje uglova ivica u obliku 0° ili 90° kako su nam samo ova dva ugla interesantna.

c. Računanje histograma (po blokovima)

Po uzoru na *sedmosegmentni displej* koji koristi 7 pažljivo raspoređenih ivica za prikaz svih cifara, histogram treba da sadrži polje za svaku od tih ivica. Time ivice sa slike možemo vezati za određen segment gde nakon što se cela slika obradi možemo jednostavno zaključiti na osnovu postojećih segmenata koja je cifra na slici.

d. Formiranje vektora obeležja

Histogrami za svaki blok se nadovezuju i zatim se normalizuju kao i kod HOG algoritma.

IV. TESTIRANJE I REZULTATI

Parametri korišćeni u *TrainSVM.cpp* fajlu su: *kernel_type* = *LINEAR* i *C* = 1.

a. Preliminarno testiranje

Rezultati preliminarnog testiranja koje je odrađeno sa podrazumevanim parametrima HOG algoritma (*block_size* = 8x8 i *histogram_size* = 9):

- *Cross validation accuracy* = 95%
- *Prediction accuracy* = 97.3 %
- *ImageDSP accuracy* = 18/20

b. HOG algoritam sa izmenjenim parametrima

Kako bismo dobili ideju na koji način parametri utiču na tačnost algoritma, odrađeni su testovi sa raznim kombinacijama dva parametra: *block_size* i *histogram_size* i procenti kros-validacije prikazani su u sledećoj tabeli:

TABELA I
REZULTATI TESTIRANJA HOG ALGORITMA SA RAZLIČITIM PARAMETRIMA - BOJAMA SU OBOJENI PROCENTI KROS-VALIDACIJE

$\frac{histogram_size}{block_size}$	9	12	16	18	20
8x8	95%	94.56%	94.74%	94.62%	94.42%
6x6	96.20%	96.08%	96.14%	96.16%	96.10%
4x4	96.72%	96.64%	96.64%	96.90%	96.76%
2x2	96.52%	96.78%	96.40%	96.66%	96.54%

Iz tabele se vidi da je najbolji rezultat dala kombinacija veličine bloka 4x4 i veličine histograma 18.

c. Testiranje našeg algoritma

U našem algoritmu veličina histograma je konstantna (7) kako prati ideju sedmosegmentnog displeja. Zbog toga su samo izmenjene veličine bloka i procenti kros-validacije su prikazani u sledećoj tabeli:

TABELA II
REZULTATI TESTIRANJA NAŠEG ALGORITMA SA RAZLIČITIM PARAMETRIMA - BOJAMA SU OBOJENI PROCENTI KROS-VALIDACIJE

$\frac{histogram_size}{block_size}$	7
8x8	93%
6x6	93.44%
4x4	94.58%
2x2	95.28%

Iz tabele se može zaključiti da je optimalna veličina bloka 2x2.

U tom slučaju je procenat tačnosti predviđanja čak 98%, dok je u primerima dobro detektovano 18 od 20 cifara.

V. ZAKLJUČAK

U ovom zadatku pokazano je da je implementacija našeg algoritma u sličnoj meri tačan i efikasan kao i HOG algoritam. Testiranje je pokazalo da je u nekim slučajevima naša implementacija lošija, dok u nekim i bolja, u zavisnosti od parametara.

VI. DODATAK

Naš algoritam je posmatrao celu sliku, čak i u slučajevima kada cifra zauzima manji deo slike. Ovde ima prostora za testiranje i unapređenje time što će se na početku obrade odbaciti nepotreban deo slike oko cifre koji ne sadrži podatke koji nas interesuju.