

# Chapter 1

## Detec'tia obiectelor cu algoritmul Viola-Jones

### 1.1 Punerea problemei

Algoritmul de detec'tie a obiectelor propus de c'atre Viola-Jones<sup>1</sup> combin'a cu succes mai multe tehnici elementare. 'In continuare vom studia pe r'znd aceste aspecte, urm'znd ca 'in final s'a vedem ansamblul func'tional. Aceste concepte sunt:

- *Imaginea Integral'a;*
- *Descriptori (caracteristici sau tr'as'aturi) primitivi (de tip Haar);*
- *Antrenare de tip AdaBoost;*
- *Calibrare prin maximizare ratei de detec'tie sa minimizare Falsurilor negative.*

Pe baza Imaginii Integrale se pot calcula rapid sume pe zone rectangulare 'intr-o imagine. Astfel putem calcula rezultatele unor descriptori de tip Haar aplicate pe diferite rezolu'tii ale imaginilor, c'aut'znd astfel obiecte (sau 'in cel mai cunoscut caz, fe'te umane) de diferite dimensiuni (scal'ari). Rezultatele clasificatorilor simpli sunt combinate cu ajutorul algoritmului AdaBoost, astfel 'inc'zt s'a rezulte un complex capabil s'a detecteze 'in timp real obiecte.

### 1.2 Descriptori Haar

Caracteristicile utilizate 'in varianta originala a algoritmului Viola-Jones amintesc de func'tiile baza de tip Haar folosite de catre Papageorgiou et al.<sup>2</sup>. Mai precis, Viola 'si Jones au folosi trei tipuri de caracteristici (vezi figura 1.1):

1. Caracteristici de dou'a dreptunghiuri 'in care se calculeaz'a diferenta dintre suma pixelilor din cele dou'a regiuni dreptunghiulare. Regiunile au aceea'si m'arime 'si forma 'si sunt orizontal sau vertical adiacente (a se vedea figura 1.1, punctele A 'si B).
2. O caracteristic'a de trei dreptunghiuri (punctul C) unde se calculeaz'a suma celor dou'a dreptunghiuri din afar'a minus dreptunghiul central.
3. O caracteristic'a de patru dreptunghiuri (punctul D din figur'a) unde se diferenta dintre suma elementelor de pe diagonala principala 'si suma elementelor din dreptunghiurile de pe diagonala secundar'a .

Fereastra de baz'a este una de  $24 \times 24$  pixeli. 'In aceast'a fereastr'a se vor considera toate caracteristicile posibile de tipurile men'tionate: cele care 'incep 'in pixelul (0,0), 'in pixelul 2,2, de la'time 2, de latime 4, 'si asa mai departe.

ACTIVITATE: Implementa'ti, utiliz'znd conceptul de Imagine Integral'a tipurile de caracteristici de tip Haar men'tionate. Aplica'ti-le pentru matricea  $b = \begin{bmatrix} 3 & 2 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{bmatrix}$  si  $a = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$ .

- *Pentru matricea a Haar 1(0,0,4,4) rez este (zn,za,H1) : 12 28 16;*

<sup>1</sup>P. Viola, M. Jones, "Robust real time object-detection", Second International Workshop on Statistical and Computational Theories of Vision Modeling, Learning, Computing and Sampling, Vancouver, Canada, July 13, 2001

<sup>2</sup>C. Papageorgiou, M. Oren, and T. Poggio. "A general framework for object detection". In Proceedings of International Conference on Computer Vision, 1998.

- Pentru matricea  $b$ , Haar  $2(0,0,2,2)$  rez este  $(zn1, zn2, za1, H_4)$ :  $4 \ 4 \ 0$ ;
- Pentru matricea  $b$ , Haar  $3(0,0,2,3)$  rez este  $(zn1, zn2, za1, H_4)$ :  $4 \ 4 \ 4 \ -4$ ;
- Pentru matricea  $a$ , Haar  $4(0,0,4,4)$  rez este  $(zn1, zn2, za1, za2, H_4)$ :  $6 \ 14 \ 6 \ 14 \ 0$

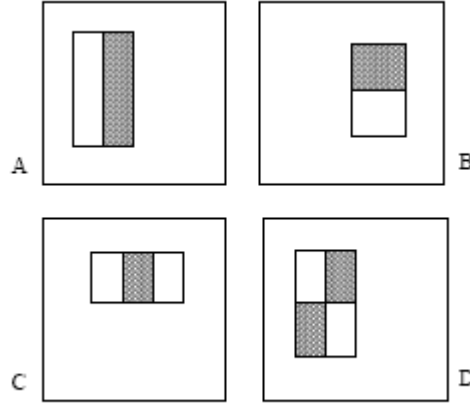


Figure 1.1: Caracteristicile de tip Haar utilizate de către Viola și Jones în algoritmul original.

Caracteristicile de tip Haar sunt oferite unui algoritm de tip Ada-Boost, în urma căruia se realizează un proces de selecție, ale celor mai utile trăsături, precum și a ponderii lor în algoritmul final.

### 1.2.1 Cascadă de clasificatori și calibrarea

Deși conform variantei originale ale algoritmului AdaBoost antrenarea s-a terminat, pentru detecția cu metoda propusă de către Viola și Jones se utilizează o abordare diferită. În acest caz clasificatorul final este obținut nu ca o sumă ponderată ci ca o cascadă de clasificatori. Din procesul de antrenare se extrage un număr limitat de clasificatori (de exemplu 100). În pasul de calibrare, se va hotărî ordinea acestor clasificatori, precum și pragurile fiecăruia.

Pentru aceasta se recomandă folosirea unui nou set de date, diferit de cel de antrenare, numit set de calibrare. Clasificatorul final se obține ca o cascadă de clasificatori, în care fiecăruia i se atribuie o pondere din valoarea țintă a detecției sau a fals pozitivelor. Pentru o cascadă de  $K$  clasificatori cu rate de detecție individuale  $d_i$ , rata de detecție totală este:

$$D = \prod_{i=1}^K d_i \quad (1.1)$$

Distribuția ratelor de detecție se poate face în mod uniform sau neuniform. Noi vom opta pentru primul caz. Trăsăturile Haar cele mai potrivite pentru detecția de fețe sunt prezentate în figura 1.2

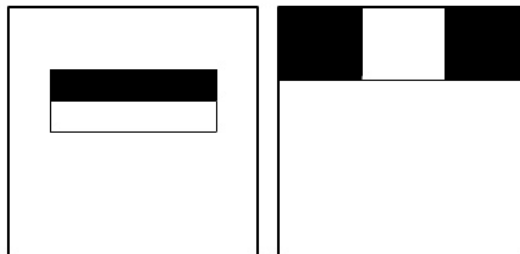


Figure 1.2: Caracteristicile de tip Haar determinate a fi cele mai eficiente. Caracteristica din stânga este în dreptunghiul  $[6 : 11; 4 : 19]$ , în timp ce cea din dreapta este în dreptunghiul  $[0 : 7; 0 : 23]$ .

**ACTIVITATE:** Aplicați caracteristicile Haar menționate pe setul de imagini oferit. Alegeți pragul astfel încât numărul de fețe identificate (TP) să fie cât mai mare.?

ACTIVITATE: Aplica'ti caracteristica 3 ca în figura 1.2, cât și în colțul din dreapta jos al imaginilor. Care este cazul mai relevant? Reprezentați grafic cele 2 cazuri

## 1.3 Algoritmul Viola-Jones

Dup'a cum am mai men'tionat 'in introducere, algoritmul Viola-Jones utilizeaz'a Imaginea Integral'a pentru calculul eficient al tr'as'aturilor de tip Haar. 'In algorimul original autorii folosesc ferestre de baz'a de  $24 \times 24$ , ceea ce conduce la 45,396 de tr'as'aturi de baz'a. Dintre acestea 'in urma antren'arii se utilizeaz'a 32 de clasificatori. 'In momentul c'aut'arii, se porne'ste de la rezolutia de baz'a, care constituie nivelul minim 'si se scaleaz'a 'in 11 niveluri consecutive, cu un factor relativ de scalare de 1.25. Pașii pentru varianta simplificată a algoritmului ar fi următorii:

1. Pentru fiecare imagine din setul de date se calculează toate cele 4 tipuri de trăsături Haar(A-D) pentru fiecare dimensiune posibilă a ferestrei
2. Se alege caracteristica Haar cea mai relevantă(TP=1.0, FP=min)
3. Se construiește primul nod al ansamblului de arbori folosind caracteristica cea mai relevantă și se elimină eşantioanele care sunt non fețe(TN)
4. Se caută următoarea cea mai relevantă caracteristică și se repetă procesul descris mai sus
5. Algoritmul se oprește dacă eroarea scade sub un anumit prag impus(TP=cât mai mare) sau dacă se atinge o adâncime maximă setată a ansamblului de arbori

*Andrei Racoviteanu*  
*Ultima modificare: 26 septembrie 2019*