ROMÂNIA

(19) OFICIUL DE STAT
PENTRU INVENŢII ŞI MĂRCI
Bucureşti

(11) **RO 133583 A2**

(51) Int.Cl.
G06K 9/00 (2006.01);
G06K 9/46 (2006.01);
G06T 7/20 (2006.01);
G06T 7/269 (2017.01);
G06N 3/02 (2006.01)

(12) # CERERE DE BREVET DE INVENŢIE

(21) Nr. cerere: **a 2018 00129**

(22) Data de depozit: **27/02/2018**

(41) Data publicării cererii:
30/08/2019 BOPI nr. **8/2019**

(71) Solicitant:
• **SECURIFAI S.R.L.**, *BD.MIRCEA VODĂ NR.24, ET.3, CAMERA 12, SECTOR 3, BUCUREŞTI, B, RO*

(72) Inventatori:
• **IONESCU RADU TUDOR**, *STR.OVIDIU NR.15, BL.LC5, AP.51, GALAŢI, GL, RO;*
• **SMEUREANU SORINA**, *STR.CALEA LUI TRAIAN NR.144, BL.4, SC.F, AP.8, RÂMNICU VÂLCEA, VL, RO;*
• **ALEXE DUMITRU BOGDAN**, *STR.SERGENT TACHE GHEORGHE NR.6, BL.B43, AP.4, SECTOR 4, BUCUREŞTI, B, RO;*
• **POPESCU MARIUS NICOLAE**, *BD.CONSTANTIN BRÂNCOVEANU NR.8, BL.B2, SC.1, AP.17, BUCUREŞTI, B, RO*

(54) # METODĂ ŞI ALGORITM BAZAT PE GRADIENŢI DE MIŞCARE ŞI REŢELE NEURONALE CONVOLUŢIONALE PENTRU DETECTAREA ŞI LOCALIZAREA AUTOMATĂ A EVENIMENTELOR ANORMALE DIN VIDEO

(57) **Rezumat:**

Invenţia se referă la o metodă şi la un algoritm pentru detectarea şi localizarea automată a evenimentelor anormale dintr-o înregistrare video. Algoritmul conform invenţiei se bazează atât pe trăsături ce reprezintă mişcarea din înregistrarea video, cât şi pe trăsături ce reprezintă înfăţişarea sau postura obiectelor sau a persoanelor şi cuprinde, pentru reprezentarea mişcării, extragerea gradienţilor de mişcare 3D şi acumularea lor în regiuni de dimensiune fixă, numite cuboizi spaţio-temporali din care se păstrează, pentru procesări ulterioare, doar regiunile pentru care magnitudinea gradienţilor depăşeşte un anumit prag, iar pentru reprezentarea obiectelor şi a posturii lor, se foloseşte o reţea neuronală convoluţională antrenată pe un set de date ImageNet pentru problema recunoaşterii obiectelor din imagini, trăsăturile extrase din reţeaua convoluţională fiind combinate apoi cu gradienţii de mişcare într-un vector de trăsături ce reprezintă o subregiune spaţio-temporală din înregistrarea video. Antrenarea algoritmului se realizează astfel: într-o primă etapă se foloseşte un algoritm de clusterizare pentru a grupa vectorii de trăsături în funcţie de similaritate, se utilizează un prag prestabilit pentru a elimina grupurile cu mai puţine elemente, iar pentru fiecare grup rămas se antrenează câte un clasificator SVM (Support Vector Machines) adaptat pentru o singură clasă, într-o etapă de testare se aplică clasificatorii pe cuboizi rezultaţi din fişierele video de testare şi, pentru fiecare exemplu, se consideră scorul de anomalie ca fiind maximul dintre scorurile date de clasificatorii SVM; în final se obţin scorurile la nivel de cadru, considerând scorul maxim al cuboizilor ce aparţin unui cadru video, după care scorurile astfel obţinute sunt netezite aplicând un filtru Gaussian pe dimensiunea temporală, iar pentru detectarea cadrelor ce conţin evenimente anormale se aplică un prag prestabilit peste scorurile calculate la nivel de cadru.

Revendicări: 1
Figuri: 9

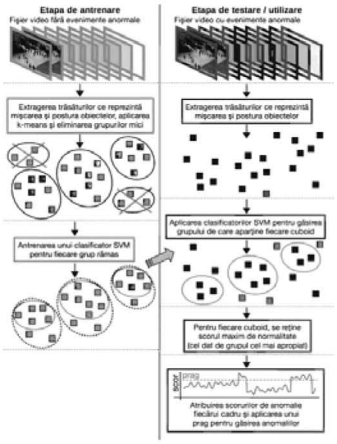

Fig. 1

RO 133583 A2

66.

Metodă şi algoritm bazat pe gradienţi de mişcare şi reţele neuronale convoluţionale pentru detectarea şi localizarea automată a evenimentelor anormale din video

Inventatori:

Radu Tudor Ionescu
Sorina Smeureanu
Bogdan Alexe
Marius Nicolae Popescu

OFICIUL DE STAT PENTRU INVENŢII ŞI MĂRCI
Cerere de brevet de invenţie
Nr. a 2018 00 129
Data depozit 27-02-2018

Aplicant:

SecurifAI
Bd. Mircea Vodă Nr. 24
Bucureşti, Romania

Rezumat:

Algoritmul propus pentru detectarea şi localizarea evenimentelor anormale din video se bazează atât pe trăsături ce reprezintă mişcarea din video cât şi pe trăsături ce reprezintă înfăţişarea sau postura obiectelor sau a persoanelor. Pentru reprezentarea mişcării, extragem gradienţi de mişcare 3D pe care îi acumulăm în regiuni de dimensiune fixă. Aceste regiuni se numesc *cuboizi spaţio-temporali*. Pentru procesarea ulterioară, păstrăm doar regiunile pentru care magnitudinea gradienţilor depăşeşte un anumit prag. Pentru reprezentarea obiectelor şi a posturii lor folosim o reţea neuronală convoluţională antrenată pe setul de date ImageNet pentru problema recunoaşterii obiectelor din imagini. Trăsăturile extrase din reţeaua convoluţională sunt combinate cu gradienţii de mişcare într-un vector de trăsături care reprezintă o sub-regiune spaţio-temporală din video. În prima etapă de antrenare, folosim algoritmul de clusterizare k-means pentru a grupa vectorii de trăsături în funcţie de similaritate. Utilizăm un prag prestabilit pentru a elimina grupurile cu mai puţine elemente. Pentru fiecare grup rămas, antrenăm câte un clasificator SVM (Support Vector Machines) adaptat pentru o singură clasă. În faza de testare, clasificatorii se aplică pe cuboizi rezultaţi din fişierele video de test. Pentru fiecare exemplu considerăm scorul de anomalie ca fiind maximul dintre scorurile întoarse de clasificatorii SVM. În final, obţinem scorurile la nivel de frame (cadru) considerând scorul maxim dintre cuboizii ce aparţin unui cadru din video. Scorurile astfel obţinute sunt netezite aplicând un filtru Gaussian pe dimensiunea temporală. Pentru detectarea şi marcarea cadrelor ce conţin evenimentelor anormale, se aplică un prag prestabilit peste scorurile calculate la nivel de cadru.

1. Introducere în domeniu

Detectarea automată a evenimentelor anormale din video este o problemă studiată în domeniul vederii artificiale, existând un număr relativ restrâns de lucrări ştiinţifice [1, 2, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 18, 21, 22, 23, 24, 25] care au abordat această problemă. Aşa cum este definit în literatura de specialitate [6, 8], un *eveniment anormal* poate fi orice eveniment care se întâmplă mai rar decât evenimentele familiare, adică cele care se întâmplă în mod frecvent. Această definiţie implică faptul că marcarea unui eveniment ca fiind anormal depinde

de context. De exemplu, un autovehicul care circulă pe stradă cu 50 de km/h în mod regulamentar este considerat un eveniment normal, în schimb un autovehicul care intră într-o zonă pietonală cu aceeaşi viteză lovind pietonii din jur este considerat eveniment anormal. Deşi în ambele situaţii maşina circulă cu aceeaşi viteză, diferenţa între marcarea evenimentului ca normal sau ca anormal este dată de context, adică de ceea ce se întâmplă în jur şi de efectele pe care le produce evenimentul. Deoarece definiţia unui eveniment anormal depinde de contextul în care se întâmplă evenimentul respectiv, problemă detectării evenimentelor anormale este una generică. Printre situaţiile care se încadrează în problema detectării evenimentelor anormale din video se numără şi:

a) Incendii şi explozii, precum şi efecte ale acestora care sunt surprinse cu camera de luat vederi. Printre exemplele de efecte produse de incendii sau explozii, care pot fi considerate evenimente anormale se numără: persoane care aleargă pentru a scăpa de incendiu, obiecte aruncate în urma suflului exploziei, etc.
b) Accidente rutiere, participanţi la trafic care execută manevre interzise, cum ar fi: pietoni care traversează prin loc nepermis, maşini care întorc pe banda dublă continuă, maşini care circulă pe o stradă cu sens interzis, etc.
c) Persoane care participă la comiterea unei infracţiuni (jaf cu sau fără armă, bătaie, altercaţie, pătrundere frauduloasă într-o incintă sau loc nepermis, etc.) având un comportament agresiv, agitat, sau pur şi simplu diferit de cel al unei persoane cu comportament normal.
d) Fenomene naturale extreme (fulgere, vânt puternic), precum şi efecte ale acestora care sunt surprinse cu camera de luat vederi. Printre exemplele de efecte produse de fenomenele naturale extreme, care reprezintă evenimente anormale pot fi: obiecte luate de vânt, crengi rupte, copaci smulşi din rădăcină, etc.

În toate situaţiile enumerate mai sus, se poate utiliza un algoritm pentru detectarea evenimentelor anormale din video, cu scopul de a detecta automat situaţiile periculoase care pot aduce pierderi de vieţi omeneşti, pagube materiale sau încălcări ale legii. Totuşi, lista de evenimente nu este una completă, fiind de fapt imposibil de realizat o listă exhaustivă de situaţii în care se poate aplica un algoritm pentru detectarea evenimentelor anormale din video. Din acest motiv, o alternativă viabilă şi des folosită în practică [1, 2, 4, 5, 7, 9, 12, 13, 14, 15, 16, 18, 21, 22, 23, 24, 25] constă în construirea unui algoritm capabil să modeleze evenimentele normale din video în baza unei etape de antrenare, folosind tehnici din vederea artificială şi învăţarea automată. Antrenarea algoritmului presupune accesul la unul sau mai multe fişiere video care conţin doar evenimente normale dintr-o anumită locaţie supravegheată cu camera de luat vederi. Deoarece evenimentele normale se întâmplă în mod frecvent, colectarea fişierelor video pentru antrenarea algoritmului implică un efort minim de timp.

2. Metode şi tehnici similare

În mod uzual, problema detectării evenimentelor anormale din video este formalizată ca o problemă de detectare a valorilor aberante [2, 4, 5, 7, 9, 12, 13, 14, 15, 16, 21, 22, 23, 24, 25], în care abordarea este în general bazată pe învăţarea unui model de normalitate din date de antrenare conţinând doar evenimente normale, şi pe considerarea valorilor aberante ca fiind evenimente anormale. Unele abordări pentru detectarea evenimentelor anormale [4, 5, 7, 13, 16, 21] se bazează pe construirea unui dicţionar de evenimente normale şi pe etichetarea evenimentelor care nu sunt reprezentate în dicţionar ca fiind anormale. Alte abordări au utilizat trăsături de tipul „deep" [23] sau filtre de tipul „locality sensitive hashing" [24] pentru a obţine rezultate mai bune.

Unele abordări mai recente se bazează pe integrarea unor paşi care nu necesită supervizare pentru detectarea evenimentelor anormale [7, 16, 22, 23]. Abordarea prezentată în [7] se bazează pe construirea unui model care să reprezinte evenimentele familiare din datele de antrenare şi pe actualizarea incrementală a modelului într-un mod nesupervizat, pe măsură ce apar noi evenimente normale în datele de testare. Într-o manieră similară, autorii lucrării [22] antrenează un model de tipul „Growing Neural Gas" pornind de la fişierele video din setul de antrenare, continuând procesul de antrenare pe măsură ce fişierele video din setul de testare sunt analizate în vederea detectării evenimentelor anormale. Autorii lucrării [16] aplică o tehnică nesupervizată numită „spectral clustering" pentru a construi un dicţionar de atomi, fiecare atom reprezentând un tip de comportament normal observat în fişierele video din setul de antrenare. Autorii lucrării [23] utilizează metoda „Stacked Denoising Auto-Encoders" pentru învăţa o reprezentare de tipul „deep" într-o manieră nesupervizată. Deşi trăsăturile sunt obţinute într-un mode nesupervizat, autorii lucrării [23] aplică mai mulţi clasificatori de tipul SVM pentru o singură clasă pentru a prezice scorurile de anomalie al evenimentelor din fişierele video de testare.

Anumite componente ale algoritmului bazat pe combinarea scorurilor ce formează obiectul cererii de brevet sunt similare cu cele descrise în lucrările [6, 8, 13, 21]. Lucrările [6, 13] sunt similare prin faptul că utilizează cuboizi spaţio-temporali, totuşi există o serie de diferenţe. O primă diferenţă constă în faptul că în cazul lucrărilor [6, 13], trăsăturile utilizate în faza de antrenare şi în cea de testare sunt obţinute în urma aplicării unei analize în componente principale. Astfel, numărul de trăsături este redus de la 500 la 100. În cazul algoritmului nostru, utilizăm toate cele 500 de trăsături. O altă diferenţă constă în faptul că metoda noastră presupune augmentarea cuboizilor cu informaţii suplimentare despre locaţia lor în fiecare cadru şi despre direcţia medie conţinută în fiecare cuboid. Aceste informaţii suplimentare conduc la o performanţă mai bună a sistemului. Totodată, trăsăturile care modelează mişcarea din cuboizi sunt combinate cu trăsături extrase cu ajutorul reţelelor neuronale convoluţionale care modelează înfăţişarea sau postura obiectelor. O altă diferenţă constă în modul de utilizare al acestor trăsături în faza de antrenare. Autorii lucrării [13] codifică structurile redundante înglobate în cuboizi printr-o mulţime de combinaţii ale unor vectori de bază. Abordarea propusă în [6] constă în detectare schimbărilor apărute într-o secvenţă de video la un moment de timp, prin comparaţie cu toate cadrele anterioare preluate de camera video. Această abordare este una nesupervizată, nefiind necesară utilizarea unor fişiere video de antrenare. Pe de altă parte, abordarea nu poate fi folosită pentru a analiza fişierele video în timp real, deoarece autorii creează permutări aleatoare ale cadrelor analizate pentru a elimina dependenţa temporală a metodei lor. Spre deosebire de [6, 13], algoritmul propus se bazează pe o metodă originală de eliminare a valorilor aberante în două etape. În prima etapă, utilizăm algoritmul k-means pentru a genera şi selecta grupuri de cuboizi ce reprezintă mişcarea şi postura normală a obiectelor. În a doua etapă, antrenăm mai mulţi clasificatori SVM adaptaţi pentru probleme cu o singură clasă. Fiecare clasificator va reprezenta un model de mişcare şi postură familiară a obiectelor din video.

De menţionat este că ideea combinării trăsăturilor ce reprezintă mişcarea din video cu trăsăturile ce reprezintă înfăţişarea sau postura obiectelor a fost studiată în [8]. Totuşi, abordarea propusă în [8] constă în combinarea scorurilor obţinute în urma analizei independente a celor două tipuri de trăsături. Obiectul prezentei cererii de brevet este un algoritm nou şi diferit prin care cele două tipuri de trăsături se pot combina într-un mod eficient, atât din punct de vedere al acurateţei cât şi al timpului de calcul. Mai precis, combinare trăsăturilor se realizează înainte de etapa de antrenare, nefiind nevoie de o procesare separată a trăsăturilor. Totodată, abordarea descrisă în [8] se bazează pe utilizarea tehnicii de „unmasking" [10] pentru detectarea evenimentelor într-un mod nesupervizat. Deşi nu necesită fişiere video pentru antrenare, abordarea descrisă în [8] produce, în general, rezultate mai slabe

decât cele ale unei tehnici supervizate, de tipul celei folosite în cadrul algoritmului propus şi descris în continuare. O altă abordare similară cu algoritmul propus este descrisă în [21]. Spre deosebire de algoritmul bazat pe combinarea trăsăturilor ce reprezintă mişcarea din video cât şi a trăsăturilor ce reprezintă postura obiectelor, abordarea propusă în lucrarea [21] utilizează doar trăsăturilor ce reprezintă postura obiectelor. Totodată, metoda supervizată din [21] se bazează pe antrenarea unui singur clasificator SVM pentru o clasă, în timp ce algoritmul nostru utilizează o metodă de antrenare diferită şi originală, fiind formată din două etape.

3. Algoritmul bazat pe combinarea scorurilor

Algoritmul propus pentru detectarea şi localizarea evenimentelor anormale din video se bazează pe extragerea a două tipuri de trăsături, unele care reprezintă mişcarea obiectelor şi altele care reprezintă înfăţişarea sau postura obiectelor. Trăsăturile extrase sunt combinate la nivel local, rezultând o mulţime de cuboizi spaţio-temporali. Antrenarea algoritmului este formată din două etape succesive. În prima etapă, utilizăm algoritmul k-means pentru a genera şi selecta grupuri de cuboizi ce reprezintă mişcarea şi postura normală a obiectelor. În a doua etapă, antrenăm mai mulţi clasificatori SVM adaptaţi pentru probleme cu o singură clasă, câte un clasificator pe fiecare grup obţinut în prima etapă de antrenare. În faza de testare, clasificatorii se aplică pe cuboizi rezultaţi din fişierele video de test. Pentru fiecare exemplu considerăm scorul de anomalie ca fiind maximul dintre scorurile întoarse de clasificatorii SVM. Etapele algoritmului sunt ilustrate în Figura 1. În continuare, sunt descrise modul de extragere al celor două tipuri de trăsături, procesul de antrenare format din două etape, precum şi metoda de calculare a scorurilor.

3.1. Extragerea trăsăturilor ce reprezintă mişcarea obiectelor

Fiind dat un fişier video cu rezoluţie mai mare de 320 (lăţime) x 240 (înăţime) pixeli şi cu un număr între 15 şi 30 de cadre pe secundă, având un număr total de N cadre, se execută următorii paşi pentru obţinerea trăsăturilor de mişcare:

a) În cazul în care cadrele sunt reprezentate în format color, se transformă fiecare imagine cadru păstrând doar intensitatea pixelilor, rezultând o imagine cadru reprezentată prin nivelurile de gri, între 0 şi 255.
b) Se aplică un filtru Gaussian de dimensiune 7x7 cu magnitudine 1.2 pentru eliminarea zgomotului din fiecare cadru (frame).
c) Se redimensionează fiecare cadru din video la dimensiunea 160 x 120 de pixeli utilizând metoda de interpolare biliniară pentru aproximarea valorilor pixelilor.
d) Se estimează gradienţii de mişcare din fiecare două cadre consecutive, prin calcularea diferenţelor în modul între valorile pixelilor corespunzători celor două cadre consecutive, aşa cum rezultă din Figurile 2 şi 3. Din cele N cadre al fişierului video, în urma acestui pas, rezultă N – 1 „cadre-gradient" ale căror pixeli indică magnitudinea gradienţilor de mişcare a obiectelor în video.
e) După calcularea gradienţilor pe întreaga dimensiune spaţială a fişierului video (160 x 120), se trece la extragerea cuboizilor spaţio-temporali. Pe dimensiunea spaţială, fiecare cadru-gradient se împarte în regiuni adiacente de 10 x 10 pixeli, ca în Figura 4.
f) Regiunile aflate în aceeaşi poziţie în fiecare 5 cadre-gradient consecutive, sunt stivuite pentru a obţine un cuboid 3D de dimensiune 10 x 10 x 5. Din fiecare 5 cadre-gradient consecutive, rezultă un număr de 16 x 12 = 192 de cuboizi.

g) Fiecare cuboid rezultat conține gradienții de mișcare 3D dintr-o regiune relativ mică din video. Totuși, dacă regiunea respectivă este aproape statică pe durata celor 5 cadre din care a fost extras cuboidul, atunci se recurge la eliminarea cuboidului prin aplicarea unui prag minim asupra sumei dată de componentele cuboidului, componente ce indică magnitudinea gradienților de mișcare.
h) Pentru etapa de antrenare a clasificatorului SVM, fiecare cuboid păstrat este liniarizat sub forma unui vector de 500 de componente ce conține trăsături de mișcare.
i) Fiecare vector este normalizat prin împărțirea fiecărei componente din vector la norma L_2 a vectorului. Fiind dat un vector $x = (x_1, x_2, ..., x_m)$, obținem vectorul normalizat conform următoarei formule:

$$\hat{x} = \frac{x}{\sqrt{x_1^2 + x_2^2 + \cdots + x_m^2}}. \quad (1)$$

Cuboizii astfel obținuți sunt augmentați cu informații suplimentare despre locația lor spațială, aplicând următorii pași:

a) Pe dimensiunea spațială, fiecare cadru-gradient (de 160 x 120 pixeli) se împarte în 4 x 4 regiuni adiacente de 40 x 30 pixeli. Astfel, fiecare regiune spațială va conține cel mult 4 x 3 cuboizi.
b) Fiecare regiune spațială este etichetată cu un număr de la 1 la 16.
c) Se adaugă la reprezentarea fiecărui cuboid 16 componente egale cu 0, rezultând astfel un vector de 516 componente.
d) Pentru fiecare cuboid se asociază valoarea 1 acelei componente (din cele 16 adăugate la pasul anterior) care corespunde etichetei regiunii spațiale care conține cuboidul respectiv, așa cum se explică în Figura 5.

Totodată, fiecare cuboid este augmentat și cu o histogramă ce codifică direcția medie a mișcării conținute în cuboidul respectiv. Pentru estimarea direcției medii și codificarea sub formă de histogramă, se aplică următorii pași:

a) Se consideră separat fiecare regiune de 10 x 10 pixeli din cele 5 regiuni consecutive care formează un cuboid spațio-temporal.
b) Se calculează centrul de greutate al gradienților de mișcare conținuți în fiecare regiune în parte.
c) Luând regiunile consecutive, două câte două, se calculează diferența între centre de greutate pe cele două axe (orizontală și verticală), rezultatul fiind interpretat ca un vector ce codifică direcția de mișcare între două regiuni consecutive.
d) Spațiul 2D în care sunt reprezentați vectorii de mișcare este împărțit în 8 cadrane disjuncte de 45° fiecare.
e) Pentru cele 8 cadrane construim o histogramă cu 8 componente, având inițial toate componentele cu valoarea 0.
f) Fiecare vector de mișcare este încadrat într-unul din cele 8 cadrane, iar magnitudinea vectorului de mișcare este adunată la componenta corespunzătoare din histogramă, așa cum este ilustrat în Figura 6. Magnitudinea vectorului x este dată de:

$$mag_x = \sqrt{x_1^2 + x_2^2}. \quad (2)$$

g) Întregul proces se repetă prin împărțirea spațială a fiecărui cuboid în 2 x 2 sub-cuboizi de 5 x 5 x 5 componente, calculând centrul de greutate al fiecărui sub-regiuni de 5 x 5 în parte.

h) Împreună cu histograma de 8 componente, mai adăugăm o componentă care conţine suma magnitudinii tuturor vectorilor de mişcare.
i) În totala, sunt 9 componente care se adaugă la reprezentarea fiecărui cuboid, rezultând un număr total de 525 de componente.

Vectorii care conţin trăsături de mişcare, rezultaţi în urma procesării fişierelor video de antrenare, sunt concatenaţi cu vectorii care conţin trăsături de postură a obiectelor. Pentru etapa în care se aplică algoritmul de detectare a evenimentelor anormale pe fişiere video noi (de testare), vectorii de trăsături trebuie calculaţi, în prealabil, din fişierele video ce trebuie analizate în vedere detectării evenimentelor anormale.

3.2. Extragerea trăsăturilor ce reprezintă postura obiectelor

Pentru extragerea obţinerea trăsăturilor ce modelează înfăţişarea şi postura obiectelor din video, se utilizează o reţea neuronală convoluţională pre-antrenată pe setul de date ImageNet [17] pentru recunoaşterea obiectelor din imagini. Arhitectura reţelei pre-antrenate este de tipul VGG-f [3]. Configuraţia acestei arhitecturi de reţea convoluţională este detaliată în Figura 7. În vederea utilizării reţelei pentru extragerea trăsăturilor din video, se elimină cele 7 straturi de la sfârşit (cu indecşii 15 până la 21), păstrând doar starturile de la intrarea reţelei până la stratul denumit „relu5" (cu indecşii 0 până la 14).

Fiind dat un fişier video cu rezoluţie mai mare de 320 (lăţime) x 240 (înăţime) pixeli şi cu un număr între 15 şi 30 de cadre pe secundă, având un număr total de N cadre, se execută următorii paşi pentru obţinerea trăsăturilor ce modelează înfăţişarea şi postura obiectelor din video:

a) Se aplică un filtru Gaussian de dimensiune 3x3 cu magnitudine 1 pentru eliminarea zgomotului din fiecare cadru.
b) Se redimensionează fiecare cadru din video la dimensiunea 224 x 244 de pixeli utilizând metoda de interpolare biliniară pentru aproximarea valorilor pixelilor.
c) În cazul în care cadrele sunt reprezentate prin nivelurile de gri, se produce o imagine color cu 3 canale (RGB) corespunzătoare fiecărui cadru, prin copierea valorilor pixelilor iniţiali pe fiecare canal în parte. În cazul în care fişierul video este color, acest pas nu este necesar.
d) Din fiecare cadru color de 224 x 224 se scade imaginea medie a reţelei convoluţionale. Imagine medie se obţine din setul de imagini de antrenare, prin calcularea valorii medii pentru fiecare pixel, prin însumarea valorilor din toate imaginile şi apoi, prin împărţirea sumei corespunzătoare fiecărui pixel la numărul de imagini din setul de antrenare. De menţionat că toate imaginile din setul de antrenare au 224 x 224 pixeli.
e) După scăderea imaginii medie, cadrul este copiat pe stratul de input al reţelei convoluţionale prezentate în Figura 4. După aplicarea operaţiilor corespunzătoare straturilor 1-14, se obţine un tensor (matrice 3D) de dimensiune 13 x 13 x 256.
f) Se redimensionează fiecare tensor la dimensiunea 16 x 12 x 256 utilizând metoda de interpolare biliniară pentru aproximarea valorilor.
g) Fiecare vector de dimensiune 1 x 1 x 256 este normalizat prin împărţirea fiecărei componente din vector la norma L_2 a vectorului, ca în Ecuaţia (1).
h) În final, fiecare vector de 256 de componente de la poziţia (x,y) din cardul t este concatenat cu cuboidul corespounzător, anume cuboidul de la poziţia (x,y) din cardul t, unde x este un număr natural între 1 şi 16 iar y este un număr natural între 1 şi 12. Rezultatul este un vector cu 256 + 525 = 781 componente.

Pentru etapa în care se aplică algoritmul de detectare a evenimentelor anormale pe fişiere video noi (de testare), vectorii de 781 componente trebuie calculaţi, în prealabil, din fişierele video ce trebuie analizate în vedere detectării evenimentelor anormale.

3.3. Prima etapă de antrenare: Eliminarea anomaliilor folosind k-means

În prima etapă de antrenare, aplică un algoritm nesupervizat pentru gruparea cuboizilor spaţio-temporali, ce conţin atât trăsături de mişcare cât şi trăsături de postură a obiectelor, obţinuţi din fişierele video de antrenare pentru a găsi grupuri ce reprezintă diferite tipuri de mişcări şi posturi normale ale obiectelor din video. Următorul pas constă în eliminarea grupurilor cu mai puţini cuboizi, pe baza presupunerii că aceste grupuri conţin în cea mai mare parte valori aberante. Pentru a susţine această presupunere, considerăm exemplul ilustrat în Figura 8 cu 400 de exemple (cercuri roşii) provenite dintr-o mixtură de două distribuţii normale de medii diferite. Exemplele generate sunt grupate în 30 de grupuri cu algoritmul k-means. Numărul de exemple din fiecare cluster este prezentat în histograma din Figura 9. În acest exemplu, am considerat că grupurile cu mai puţin de 10 exemple trebuie eliminate. Centroizii acestor grupuri sunt marcaţi cu un pătrat albastru în Figura 8. Spre deosebire, centroizii grupurilor care rămân sunt marcaţi cu un disc albastru. În acest exemplu, putem observa cu uşurinţă faptul că grupurile marcate pentru eliminare sunt într-adevăr cele care conţine valori aberante. Aceste grupuri reprezintă fie elemente de mişcare sau postură a obiectelor mai puţin reprezentative sau care conţin mai mult zgomot.

În exemplu prezentat în Figura 8, algoritmul k-means şi pasul de eliminare al grupurilor cu mai puţine exemple sunt aplicate pe un set de vectori cu două componente (puncte în plan). În cadrul algoritmului de detectare a evenimentelor anormale, algoritmul k-means şi pasul de eliminare sunt de fapt aplicate pe vectori de 781 componente (cuboizi spaţio-temporali). În practică, numărul de clustere se poate determina în mod automat în funcţie de numărul total de cuboizi astfel încât să rezulte în medie un număr de p de cuboizi în fiecare grup. Experimental, am determinat că valorile potrivite pentru p sunt între 1.000 şi 10.000. Tot experimental, am stabilit că pragul pentru eliminarea grupurilor de cuboizi este p / 2.

3.4. A doua etapă de antrenare: Eliminarea anomaliilor folosind SVM pentru o singură clasă

După eliminarea grupurilor cu mai puţini cuboizi, rămân o mulţime de grupuri $C = \{c_1, c_2, ..., c_r \mid r \leq k\}$ care modelează mişcările şi postura obiectelor reprezentative din punct de vedere al normalităţii. Totuşi, algoritmul k-means nu produce o graniţă suficient de strânsă în jurul fiecărui grup din C, astfel că un grup poate cuprinde şi valori aberante. De exemplu, graniţa unor grupuri care rămân după eliminare din Figura 8 se continuă spre infinit. Pentru a soluţiona această problemă, a doua etapă de antrenare constă în antrenarea unor clasificatori SVM pentru o singură clasă conform [19], pentru a determina graniţe de separare mai strânse în jurul grupurilor rămase. Pentru a antrena clasificatorii, considerăm că fiecare cuboid spaţio-temporal este un exemplu de antrenare independent de celelalte, neluând astfel în considerare relaţiile spaţiale şi temporale dintre cuboizi. Considerăm mulţimea de antrenare $\mathcal{X} = \{x_1, x_2, ..., x_n \mid x_i \in \mathbb{R}^m\}$ formată din vectorii de 781 de trăsături dintr-un grup c_j. În această formulare, clasificatorul SVM pentru o singură clasă va învăţa să separe o regiune restrânsă ce cuprinde un tip cuboizi normali (familiari) de restul spaţiului de trăsături, prin maximizarea distanţei de la origine la hiperplanul de separare al clasificatorului SVM. În urma antrenării clasificatorului SVM, va rezulta o funcţie de clasificare binară g, ce va produce un eticheta +1 pentru regiunile

din spațiul de trăsături ce aparțin densității de probabilitate a evenimentelor normale și eticheta -1 în rest. Funcția învățată are forma:

$$g(z) = sign\left(\sum_{i=1}^{n} \alpha_i \cdot k(z, x_i) - \rho\right), \qquad (3)$$

unde $x_i \in \mathcal{X}$ este un exemplu (cuboid) de antrenare, k este o funcție nucleu, α_i sunt ponderile asociate vectorilor suport x_i, ρ este distanța de la origine la hiperplan, iar z este un exemplu (cuboid) de test ce trebuie clasificat într-una din cele două clase de evenimente: normal sau anormal. Coeficienții α_i sunt stabiliți prin găsirea soluției următoarei probleme duale:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \cdot \alpha_j \cdot k(x_i, x_j) \; cu \; 0 \leq \alpha_i \leq \frac{1}{v \cdot n}, \sum_{i=1}^{n} \alpha_i = 1, \qquad (4)$$

unde $v \in [0,1]$ este parametrul de regularizare al clasificatorului SVM, prin intermediul căruia se controlează procentajul de valori aberante (anomalii) ce trebuie excluse din modelul de familiare învățat. Distanța de la origine la hiperplan ρ poate fi găsită prin exploatarea proprietății satisfăcute de un exemplu x_i, pentru orice α_i care nu reprezintă o limită inferioară sau o limită superioară, conform [19]:

$$\rho = \sum_{j=1}^{n} \alpha_j \cdot k(x_j, x_i). \qquad (5)$$

Funcția nucleu k pe care o utilizăm în clasificatorul SVM este funcția nucleu liniară, dată de produsul scalar între două exemple x și z:

$$k(x, z) = \langle x, z \rangle. \qquad (6)$$

În vederea obținerii un scoruri care să reflecte gradul de anomalie al fiecărui cuboid, renunțăm la funcția semn (sign) din Ecuația (3). De menționat că pentru fiecare grup de cuboizi c_j din C, vom avea câte un clasificator cu funcția de scor asociată g_{cj}. Aplicând apoi clasificatorii SVM pe un cuboid extras din fișierul video de testare vom obține o mulțime de r scoruri de normalitate. Deorece grupurile de cuboizi sunt independente (aparțin unor zone disjuncte din spațiul de trăsături), putem presupune în mod natural că fiecare cuboid de test poate aparține unui singur grup de cuboizi. Astfel, dintre cele r scoruri, alegem scorul maxim de normalitate, adică scorul asociat grupului care este cel mai apropiat de cuboidul de test. Dacă scorul maxim este pozitiv, înseamnă că grupul cel mai apropiat cuprinde cuboidul de test. Dacă scorul maxim este negativ, înseamnă că niciun grup nu cuprinde cuboidul de test, în această situație, cuboidul respectiv fiind considerat a fi anormal. Această procedură se aplică fiecărui cuboid de test.

3.5. Calcularea scorurilor de anomalie la nivel de cadru și la nivel de pixel

Pentru a interpreta scorul maxim de normalitate al unui cuboid de test ca scor de anomalie schimbăm semnul scorului. Punând împreună scorul cuboizilor din fiecare cadru, obținem o mapare cu 16 x 12 componente de predicție (scoruri de anomalie) pentru fiecare cadru. Pentru o obține o mapare cu predicții la nivel de pixel, redimensionăm maparea de 16 x 12 componente la dimensiunea fișierului video de testare utilizând metoda de interpolare biliniară. Pentru a obține scoruri de predicție la nivel de cadru, alegem scorul maxim de anomalie din maparea de 16 x 12 componente. Pe axa temporală, aplicăm un filtru Gaussian pentru a netezi scorurile la nivel de cadru. În practică, am obținut rezultate mai bune cu filtre Gaussiane cu suport de la 20 până la 100 de componente. În final, evenimentele anormale sunt detectate în urma aplicării unui prag pe scorurile de anomalie la nivel de cadru. Cadrele consecutive care depășesc pragul de anomalie se consideră că formează un singur eveniment

anormal. Pragul de anomalie poate fi ajustat în funcţie de aplicaţie pentru a obţine raportul dorit dintre detecţiile adevărate şi cele false.

3.6. Elemente de noutate ale algoritmului propus

Algoritmul pentru detectarea evenimentelor anormale conţine o serie de elemente de noutate:

a) Metoda de augmentare a cuboizilor spaţio-temporali cu informaţii despre locaţie şi despre mişcare medie (descrisă în secţiunea 3.1.).
b) Metoda de calcul a trăsăturilor pentru postura obiectelor (descrisă în secţiunea 3.2.).
c) Metoda de eliminare a valorilor aberante folosind algoritmul k-means în prima etapă de antrenare (descrise în secţiunea 3.3.).
d) Metoda de calculare a scorului de anomalie pentru fiecare cuboid de test (descrisă în secţiunea 3.4.).

4. Referinţe

[1] Adam, A., Rivlin, E., Shimshoni, I., Reinitz, D.: Robust Real-Time Unusual Event Detection Using Multiple Fixed- Location Monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3), 555–560 (2008)

[2] Antic, B., Ommer, B.: Video parsing for abnormality detection. In: Proceedings of ICCV. pp. 2415–2422 (2011)

[3] Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the Devil in the Details: Delving Deep into Convolutional Nets. In: Proceedings of BMVC (2014)

[4] Cheng, K.W., Chen, Y.T., Fang, W.H.: Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression. In: Proceedings of CVPR. pp. 2909–2917 (2015)

[5] Cong, Y., Yuan, J., Liu, J.: Sparse reconstruction cost for abnormal event detection. In: Proceedings of CVPR. pp. 3449–3456 (2011)

[6] Del Giorno, A., Bagnell, J., Hebert, M.: A Discriminative Framework for Anomaly Detection in Large Videos. In: Proceedings of ECCV. pp. 334–349 (2016)

[7] Dutta, J.K., Banerjee, B.: Online Detection of Abnormal Events Using Incremental Coding Length. In: Proceedings of AAAI. pp. 3755–3761 (2015)

[8] Ionescu, R.T., Smeureanu, S., Alexe, B., Popescu, M.: Unmasking the abnormal events in video. In: Proceedings of ICCV (2017)

[9] Kim, J., Grauman, K.: Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates. In: Proceedings of CVPR. pp. 2921–2928 (2009)

[10] Koppel, M., Schler, J., Bonchek-Dokow, E.: Measuring Differentiability: Unmasking

Pseudonymous Authors. Journal of Machine Learning Research, vol. 8, 1261–1276 (2007)

[11] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Proceedings of NIPS. pp. 1106–1114 (2012)

[12] Li, W., Mahadevan, V., Vasconcelos, N.: Anomaly detection and localization in crowded scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 36(1), 18–32 (2014)

[13] Lu, C., Shi, J., Jia, J.: Abnormal Event Detection at 150 FPS in MATLAB. In: Proceedings of ICCV. pp. 2720–2727 (2013)

[14] Mahadevan, V., LI, W.X., Bhalodia, V., Vasconcelos, N.: Anomaly Detection in Crowded Scenes. In: Proceedings of CVPR. pp. 1975–1981 (2010)

[15] Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: Proceedings of CVPR. pp. 935–942 (2009)

[16] Ren, H., Liu, W., Olsen, S.I., Escalera, S., Moeslund, T.B.: Unsupervised Behavior-Specific Dictionary Learning for Abnormal Event Detection. In: Proceedings of BMVC. pp. 28.1–28.13 (2015)

[17] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., A., K., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. International Journal of Computer Vision (2015)

[18] Saligrama, V., Chen, Z.: Video anomaly detection based on local statistical aggregates. In: Proceedings of CVPR. pp. 2112–2119 (2012)

[19] Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Computation 13(7), 1443–1471 (Jul 2001)

[20] Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: Proceedings of ICLR (2014)

[21] Smeureanu, S., Ionescu, R.T., Popescu, M., Alexe, B.: Deep Appearance Features for Abnormal Behavior Detection in Video. In: Proceedings of ICIAP (2017)

[22] Sun, Q., Liu, H., Harada, T.: Online growing neural gas for anomaly detection in changing surveillance scenes. Pattern Recognition 64(C), 187–201 (2017)

[23] Xu, D., Ricci, E., Yan, Y., Song, J., Sebe, N.: Learning Deep Representations of Appearance and Motion for Anomalous Event Detection. In: Proceedings of BMVC. pp. 8.1–8.12 (2015)

[24] Zhang, Y., Lu, H., Zhang, L., Ruan, X., Sakai, S.: Video anomaly detection based on locality sensitive hashing filters. Pattern Recognition 59, 302–311 (2016)

[25] Zhao, B., Fei-Fei, L., Xing, E.P.: Online Detection of Unusual Events in Videos via Dynamic Sparse Coding. In: Proceedings of CVPR. pp. 3313–3320 (2011)

Anexa 1 – Figura 1

Metodă şi algoritm bazat pe gradienţi de mişcare şi reţele neuronale convoluţionale pentru detectarea şi localizarea automată a evenimentelor anormale din video

Aplicant:

SecurifAI
Bd. Mircea Vodă Nr. 24
Bucureşti, Romania

Revendicare:

Metoda şi algoritmul bazat pe gradienţi de mişcare şi reţele neuronale convoluţionale pentru detectarea şi localizarea automată a evenimentelor anormale din video **se caracterizată prin aceea că** se bazează pe două tipuri de trăsături, unele care reprezintă mişcarea obiectelor cât şi pe trăsături care reprezintă înfăţişarea sau postura obiectelor sau a persoanelor.

Securifai SRL
Prin administrator
Vlad Mihu

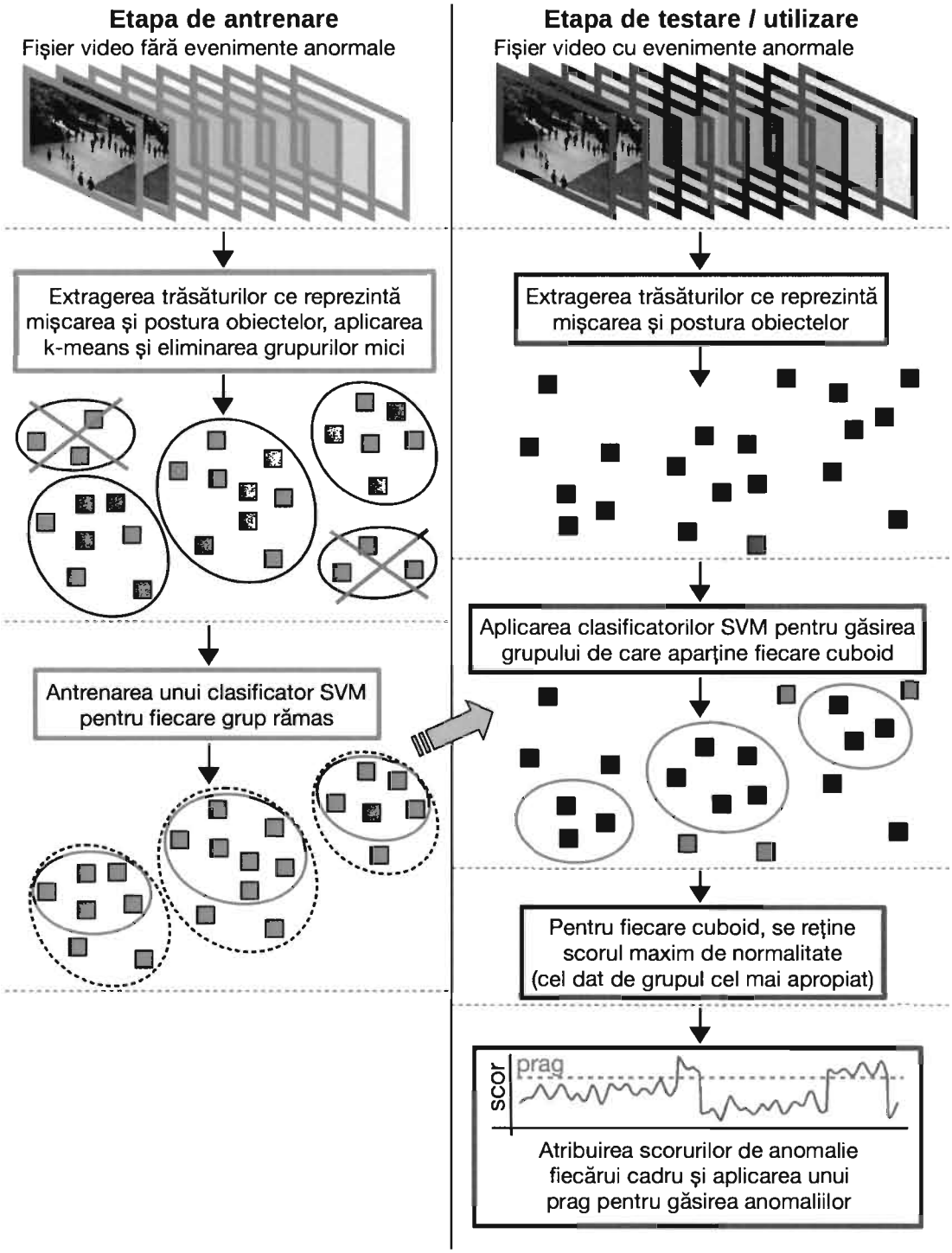

Figura 1. Etapele prevăzute în algoritmul pentru detectarea şi localizarea evenimentelor anormale din video pe baza unei metode de eliminare a valorilor aberante formată din două etape de antrenare. Metoda combină trăsăturile ce reprezintă mişcarea obiectelor şi trăsăturile ce reprezintă postura obiectelor prin intermediul unor cuboizi.

Anexa 2 – Figura 2

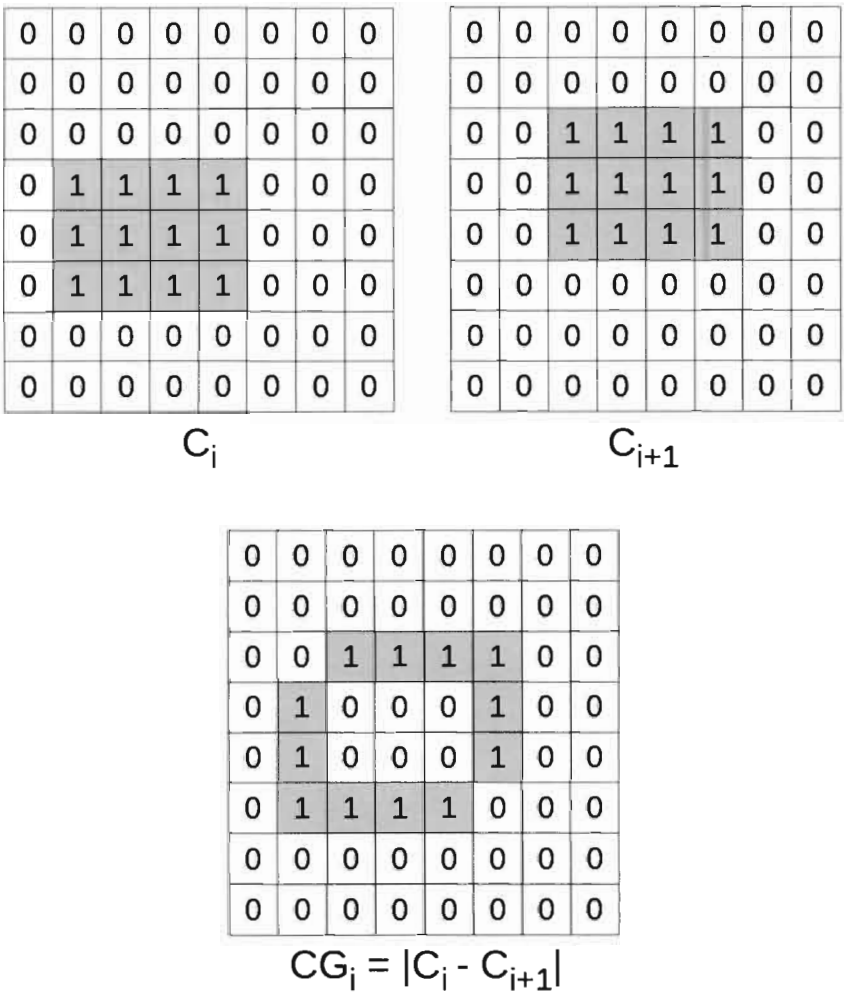

Figura 2. Pentru orice două cadre consecutive C_i şi C_{i+1}, se calculează cadrul-gradient CG_i ca diferență în modul între pixelii din cadrul C_i şi cadrul C_{i+1}. În cadrul C_i avem un obiect pătrat format din pixeli cu valoarea 1, care se deplasează spre dreapta sus în cadrul următor C_{i+1}. Cadrul-gradient CG_i conține magnitudinea gradienților de mișcare ai obiectului pătrat. În acest exemplu ilustrativ, cadrele au 8 x 8 pixeli, dar în practică, procedura se aplică pe cadre de 160 x 120 pixeli.

Anexa 3 – Figura 3

C_i C_{i+1}

$CG_i = |C_i - C_{i+1}|$

Figura 3. Pentru orice două cadre consecutive C_i şi C_{i+1}, se calculează cadrul-gradient CG_i ca diferență în modul între pixelii din cadrul C_i şi cadrul C_{i+1}. În cadrul C_i avem o intersecție prin care trec maşini, care se deplasează în cadrul următor C_{i+1}. Cadrul-gradient CG_i conține magnitudinea gradienților de mişcare ai maşinilor. În acest exemplu, cadrele au 160 x 120 pixeli.

Anexa 4 – Figura 4

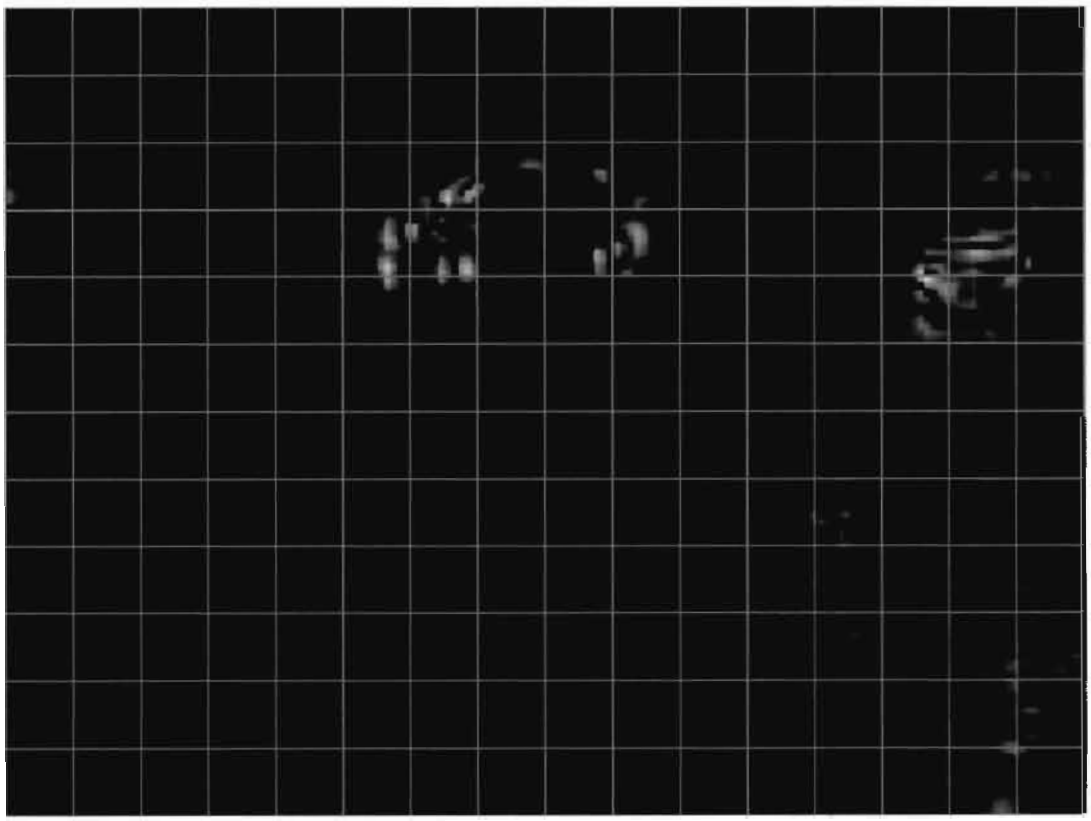

Figura 4. Cadrul-gradient CG_i de 160 x 120 de pixeli din Figura 2 este împărţit în regiuni adiacente de 10 x 10 pixeli, rezultând un număr de 16 x 12 = 192 de regiuni. Împărţirea pe regiuni este ilustrată printr-o grilă de culoare roşie aplicată peste cadrul-gradient CG_i.

Anexa 5 – Figura 5

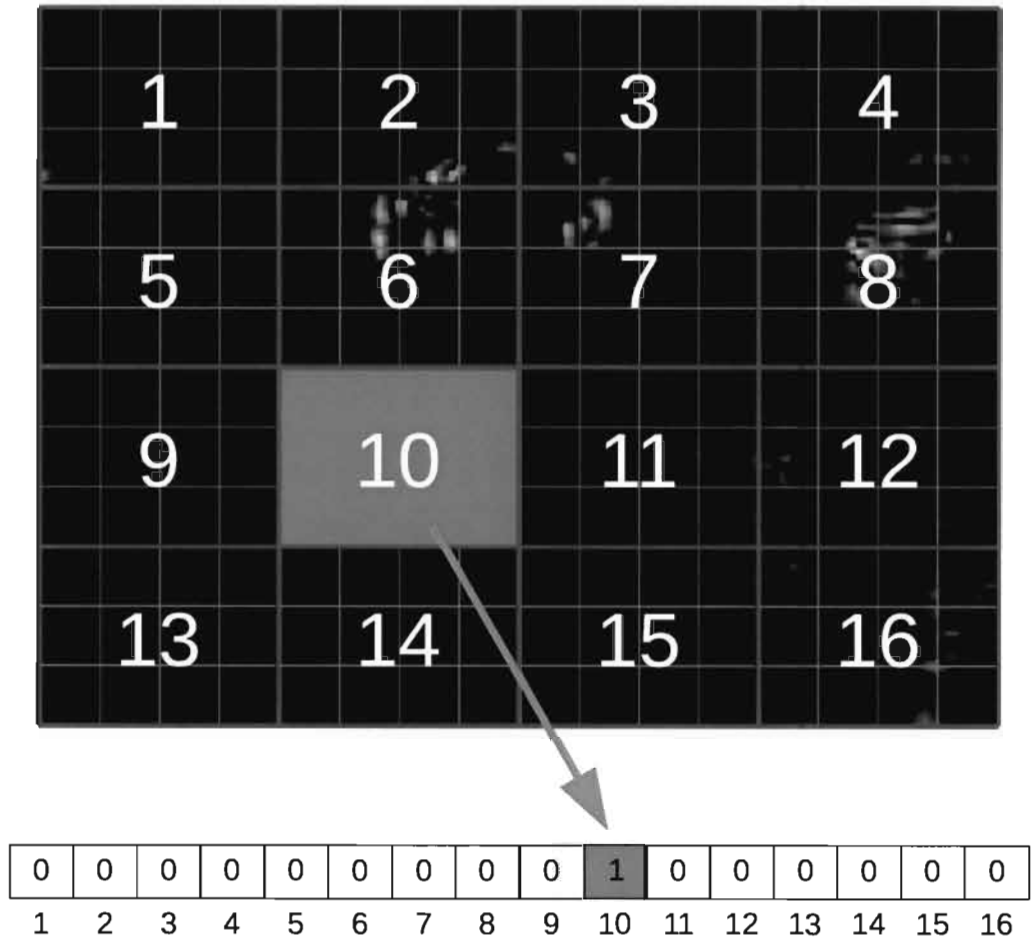

Figura 5. Pentru codificarea locației unui cuboid, se construiește un vector cu 16 componente cu 15 valori de 0 și o singură valoare de 1 ce corespunde regiunii spațiale din care provine cuboidul. În acest exemplu, toții cuboizii din regiunea 10 sunt augmentați cu vectorul de 16 componente în care valoarea 1 se află pe poziția 10.

Anexa 6 – Figura 6

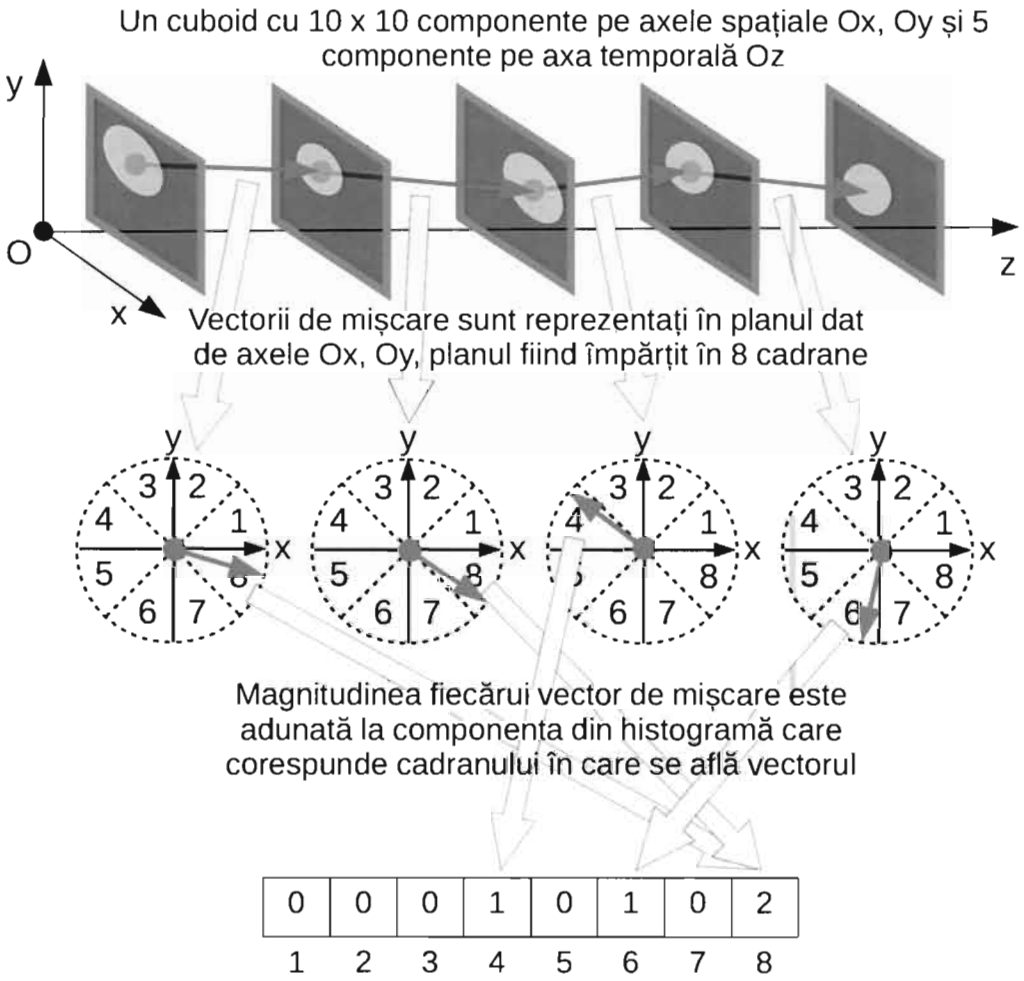

Figura 6. Pentru codificarea direcției medii a unui cuboid de 10 x 10 x 5 componente, luăm regiunile consecutive, două câte două, şi calculăm diferența între centre de greutate pe cele două axe (orizontală - Ox şi verticală - Oy), rezultatul fiind interpretat ca un vector ce codifică direcția de mişcare între două regiuni consecutive. Planul în care sunt reprezentați vectorii se împarte în 8 cadrane egale. Adunăm magnitudinea fiecărui vector de mişcare la componenta din histogramă ce corespunde cadranului în care se află vectorul respectiv.

Anexa 7 – Figura 7

Index	Tip	Nume	Suport	Adâncime filtre	Număr filtre	Spaţiere	Distanţă margini	Dimensiune date	Adâncime date
0	input	-	-	-	-	-	-	224 x 224	3
1	conv	conv1	11 x 11	3	64	4	0	54 x 54	64
2	relu	relu1	1 x 1	-	-	1	0	54 x 54	64
3	norm	norm1	1 x 1	-	-	1	0	54 x 54	64
4	maxpool	pool1	3 x 3	-	-	2	0, 1, 0, 1	27 x 27	64
5	conv	conv2	5 x 5	64	256	1	2	27 x 27	256
6	relu	relu2	1 x 1	-	-	1	0	27 x 27	256
7	norm	norm2	1 x 1	-	-	1	0	27 x 27	256
8	maxpool	pool2	3 x 3	-	-	2	0	13 x 13	256
9	conv	conv3	3 x 3	256	256	1	1	13 x 13	256
10	relu	relu3	1 x 1	-	-	1	0	13 x 13	256
11	conv	conv4	3 x 3	256	256	1	1	13 x 13	256
12	relu	relu4	1 x 1	-	-	1	0	13 x 13	256
13	conv	conv5	3 x 3	256	256	1	1	13 x 13	256
14	relu	relu5	1 x 1	-	-	1	0	13 x 13	256
15	maxpool	pool5	3 x 3	-	-	2	0	6 x 6	256
16	conv	fc6	6 x 6	256	4096	1	0	1 x 1	4096
17	relu	relu6	1 x 1	-	-	1	0	1 x 1	4096
18	conv	fc7	1 x 1	4096	4096	1	0	1 x 1	4096
19	relu	relu7	1 x 1	-	-	1	0	1 x 1	4096
20	conv	fc8	1 x 1	4096	1000	1	0	1 x 1	1000
21	softmax	softmax	1 x 1	-	-	1	0	1 x 1	1000

Figura 7. Configuraţia reţelei neuronale convoluţionale utilizată pentru extragerea trăsăturilor ce modelează înfăţişarea şi postura obiectelor dintr-un cadru al fişierului video. După antrenarea reţelei, straturile 15-21 se elimină. În urma procesării unui cadru, rezultă un tensor de dimensiune 13 x 13 x 256, conform stratului relu5.

Legendă:
input = strat de intrare (o imagine RGB de 224 x 224 pixeli)
conv = strat convoluţional (presupune aplicarea operaţiei de convoluţie)
relu = strat non-liniar (presupune aplicarea funcţiei f(x) = max(0, x))
norm = strat de normalizare (presupune normalizarea răspunsului local)
maxpool = strat de pooling (presupune păstrarea valorii maxime din fiecare regiune de 3 x 3)
softmax = stratul de decizie (presupune aplicarea funcţiei de decizie softmax)

Anexa 8 – Figura 8

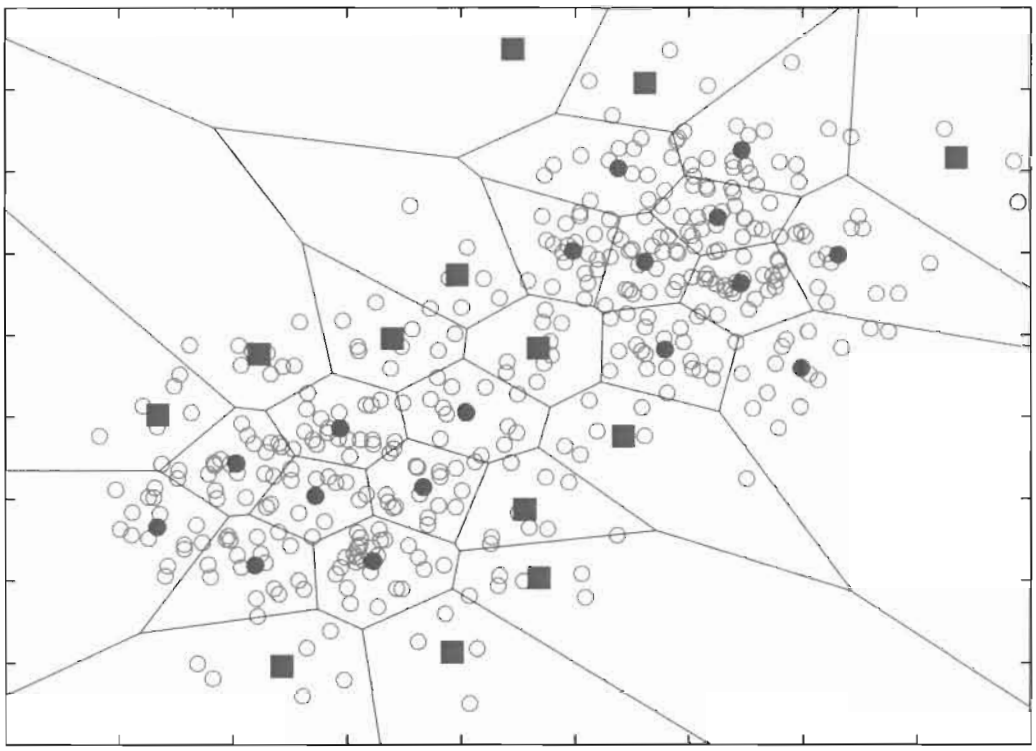

Figura 8. O mulțime de 400 de puncta în plan provenite din două distribuții normale de medii diferite. Punctele sunt clusterizate în 30 de grupuri folosind algoritmul k-means. Centroizii grupurilor ce conțin mai puțin de 10 exemple sunt marcați cu pătrate albastre. Aceste gruprui conțin valori aberante (depărtate de mediile distribuțiilor normale).

Anexa 9 – Figura 9

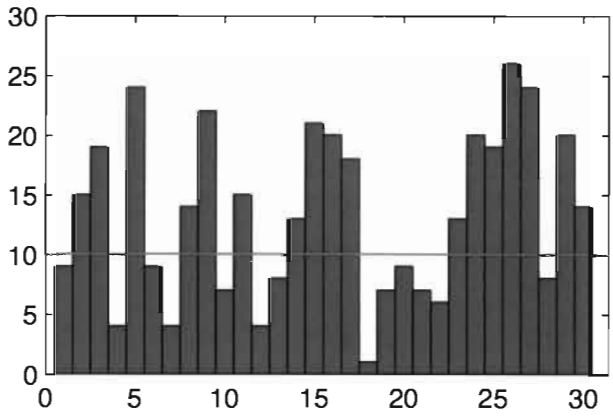

Figura 9. O histogramă cu numărul de puncte din fiecare grup de puncta reprezentat în Figura 8. Prin aplicarea unui prag de 10 puncte peste histogramă se obțin grupurile cu valori aberante.