Raport științific și tehnic privind implementarea proiectului

SISTEM AUTOMAT DE AUTENTIFICARE ACTIVĂ A IMAGINILOR DIGITALE PENTRU PC ȘI TERMINALE MOBILE

Contract PN-III-P2-2.1-PED-2016-1465 nr. 32PED/2017

Etapa de execuție nr. 1 / 2017

Perioada ianuarie – decembrie 2017

Obiectivul proiectului este dezvoltarea unei aplicații software pentru PC si terminale mobile de autentificare activă a conținutului imaginilor digitale utilizând tehnici de watermarking digital.

Etapa pe 2017 a proiectului

Pentru etapa pe 2017 a proiectului obiectivele principale au fost stabilirea specificațiilor tehnice și implementarea diferitelor module ale aplicației de autentificare a imaginilor.

În primă fază, echipa proiectului a identificat cerințele specifice și a stabilit specificațiile pentru o aplicație eficientă de autentificare activă a imaginilor și a realizat scheme bloc și reprezentări grafice de tip wireframes ale aplicației. Apoi a fost definit setul de caracteristici și performanțe de execuție ale aplicației. Pentru a obține caracteristicile tehnice dorite ale aplicației, algoritmii de autentificare disponibili au fost îmbunătățiți și adaptați la setul de specificații ale proiectului, și au fost dezvoltat un algoritm nou special pentru aplicația de autentificare pentru dispozitive mobile cu sistem de operare Android. A fost stabilită o arhitectură modulară a aplicației pentru a permite implementarea cu ușurință a noi algoritmi de detecție a falsificării, s-au determinat responsabilitățile fiecărui modul și s-a determinat modul lor de integrare în interfața grafică a aplicației.

Cea mai lungă activitate a acestei etape a fost implementarea practică a diferitelor module funcționale ale aplicației în mediul integrat de dezvoltare Android Studio. La finalul etapei aceste module software au fost integrate într-o primă versiune alfa a aplicației de autentificare.

În cadrul etapei pe 2017 a proiectului ne-am propus și diseminarea rezultatelor obținute prin participarea la 3 manifestări științifice internaționale. Am depășit acest obiectiv, publicând 6 articole științifice în volumele unor conferințe internaționale în curs de indexare ISI, un al șaptelea articol fiind în proces de evaluare la o revistă internațională cotată ISI.

Pentru autentificarea activă a imaginilor digitale pot fi utilizate atât tehnici fragile, cât și semi-fragile de watermarking. Majoritatea tehnicilor existente în literatura de specialitate utilizează marcaje fragile pentru a asigura autentificarea conținutului imaginii [1-4], dar la astfel de metode imagine va fi considerată neautentică, chiar dacă a fost modificat un singur bit din imagine. Acest lucru face aceste metode mai puțin viabile în practică, în aplicații reale. De cealaltă parte, algoritmii semi-fragili permit anumite prelucrări ale imaginii, care nu modifică conținutul propriu-zis al acesteia, fiind în același timp capabile să detecteze falsificările de conținut.

Imaginile digitale sunt stocate și recodate în diferite formate, cu și fără opțiuni de compresie. Standardul JPEG este cel mai uzual standard de compresie cu pierderi și este utilizat

de camere digitale, dispozitive mobile și pentru transmiterea de imagini comprimate pe Internet. O tehnică semi-fragilă de autentificare a imaginilor trebuie să poată diferenția compresia JPEG de modificările ale conținutului imaginilor, ca de exemplu ștergerea sau înlocuirea unor zone din imagine.

În literatura de specialitate au fost propuse diferite metode semi-fragile de autentificare. În [5] Liu și al. utilizează amplitudinile momentelor Zernike ale imaginii originale pentru a genera marcajul de autentificare și a-l insera în subbenzile Transformatei Wavelet Discrete. Zonele falsificate sunt localizate utilizând proprietatea de separabilitate a vectorului de momente Zernike. Datorită semi-fragilității vectorului de caracteristici, algoritmul este robust la compresia JPEG cu factori de calitate mai mari decât 60, dar ratele de detecție de fals pozitiv și negativ sunt destul de mari. Al-Otum [6] propune o altă tehnica de autentificare semi-fragilă pentru imagini cu nuanțe de gri, unde trei biți de watermark sunt inserați într-un mod scalabil în subbenzile de frecvență joasă a descompunerii Wavelet de la nivelul doi de rezoluție folosind o abordare bazată pe cuantizare. Se obțin valori PSNR medii de 40 dB și o bună robustețe la modificări neintenționate ale conținutului imaginii, ca de exemplu compresia JPEG, blurarea și adăugarea de zgomot de tip Gaussian sau de tip „sare și piper”. O altă tehnică este propusă în [7], unde autorii inserează un logo binar criptat în imaginea gazdă prin înlocuirea coeficienților Transformatei Wavelet Discrete cu Valori Întregi de frecvențe înalte cu watermark-uri de autentificare. Tehnica obține rezultate bune pentru rate de falsificare între 5% și 23%, dar este robustă la compresie JPEG doar cu factori de calitate mai mari ca 85.

În [8] imaginea este împărțită în blocuri de 2x2 pixeli și din coeficienții Wavelet de aproximare a fiecărei perechi de blocuri se obține câte un watermark de 12 biți. Acest watermark este inserat în ultimii trei cei mai puțin semnificativi biți ai altei perechi de blocuri, a căror poziții sunt obținute printr-un algoritm de mapare a blocurilor. Schema obține o calitate medie a imaginilor marcate, cu valori PSNR de 41 dB, rezultate de detecție bune pentru rate de falsificare de până la 60%, dar algoritmul nu este robust la prelucrări uzuale de imagini.

În [9] Rosales-Roldan și al. propun două tehnici de autentificare în domeniul DCT, respectiv Wavelet, inserând o versiune binară a imaginii originale în coeficienții transformatei. Detecția zonelor falsificat este realizată folosind Indicele de Similitudine Structurală. Schema este robustă la compresie JPEG cu factori de calitate mai mari decât 65, dar calitatea imaginilor este relativ scăzută, cu valori PSNR de circa 35 dB.

Schema de autentificare prezentată în [10] utilizează o semnătură criptată a imaginii pe post de marcaj de autentificare și o inserează în coeficienții subbenzilor Wavelet HL și LH ale imaginii utilizând un procedeu de modulație a indicelui de cuantizare cu dither. Schema este robustă la compresie JPEG cu factori de calitate mai mari ca 70, dar, la fel ca și metoda anterioară, distorsiunea produsă de inserarea marcajului de autentificare este destul de mare la 35 dB.

Qi și al. propun o altă schemă de autentificare activă în domeniul Wavelet. Imaginea este partiționată în blocuri de 4x4 pixeli și o secvență pseudo-aleatoare binară este inserată în coeficienți Wavelet de aproximare aleși aleator. Watermark-ul este extras testând paritatea coeficienților cuantizați și rotunjiți ai imaginii de test. Calitatea imaginilor marcate este medie cu valori PSNR de 40 dB, dar schema permite o compresie JPEG redusă cu factori de calitate mai mari ca 80, iar probabilitatea de fals negativ este mare, în special pentru zone falsificate de dimensiune mică.

În cele ce urmează vom detalia activitățile desfășurate în prima etapă a proiectului.

Principalele cerințe și specificații identificate de echipa proiectului pentru o aplicație eficientă de detecție a falsificării imaginilor digitale includ:

- Transparența perceptuală (imperceptibilitate): Aplicația de watermarking trebuie să ascundă marcajul de autentificare astfel încât acest lucru să nu afecteze vizibil calitatea imaginii gazdă. O procedură de ascundere a watermark-ului este cu adevărat imperceptibilă dacă ochiul uman nu poate deosebi datele originale de cele cu marcaj inserat. Înainte de a testa calitatea imaginilor marcate folosind subiecți umani, calitatea va trebui evaluată obiectiv. Raportul semnal de vârf – zgomot (PSNR) este metrica cea mai larg utilizată pentru a măsura calitatea imaginilor. Ca o specificație tehnică a aplicației ne propunem să obținem un PSNR mediu al imaginilor marcate de peste 40 dB.
- Capacitatea (adaosul de informație): Cantitatea de informație ce poate fi stocată în imaginea gazdă depinde de aplicație. Pentru protecția la copiere, un singur bit de informație ar fi suficient. În cazul unei aplicații de autentificare trebuie găsit un compromis între capacitatea watermark-ului și calitatea imaginii marcate, deoarece inserarea unei cantități mai mari de informație produce o degradare mai mare a calității imaginii. După un studiu al literaturii de specialitate, ne-am propus să inserăm un număr maxim de 6 biți de watermark în fiecare bloc de 8x8 pixeli ai imaginii.
- Detecția falsificării conținutului: Un sistem de autentificare a imaginilor trebuie să răspundă la următoarele întrebări: „A fost modificat conținutul imaginii în vreun fel?” și „Dacă a fost modificat, care este zona modificată?”. Răspunsul la prima întrebare este de regulă mai ușor de obținut, însă a doua întrebare este mai dificilă. Autentificatorul trebuie să poată identifica locația regiunilor manipulate cu o anumită rezoluție și, pe de altă parte, să poată verifica autenticitatea altor regiuni. Pentru a putea diferenția între o modificare intenționată de conținut și distorsiuni neintenționate cauzate de prelucrări uzuale de imagini, ne propunem o rezoluție minimă de detecție a falsificării de 16x16 pixeli. De asemenea, doi parametri importanți sunt ratele de detecție de fals pozitiv și fals negativ, care trebuie să aibă valori cât mai mici.
- Robustețea la compresie JPEG: Deoarece ne dorim ca aplicația de autentificare să funcționeze pe terminale mobile, iar majoritatea acestora stochează imaginile achiziționate în format JPEG, dorim marcajul de autentificare inserat să fie unul semi-fragil și să reziste la compresie JPEG cu factori de calitate mai mari decât 50.
- Securitate: O tehnică de watermarking este cu adevărat sigură, dacă cunoașterea exactă a algoritmului de ascundere și de extragere a marcajului nu ajută o parte neautorizată să detecteze prezența marcajului. Pentru a satisface această condiție, aplicația de autentificare dezvoltată va utiliza tehnici criptografice pentru a coda marcajul de autentificare și pozițiile de inserare. Atât pentru inserare, cât și pentru autentificarea imaginii se va utiliza o cheie privată care va fi folosită la generarea watermark-ului ca o secvență pseudo-aleatoare binară și la alegerea locației de inserare a marcajului.

Pentru o implementare cât mai modulară a aplicației de autentificare, am realizat o schemă bloc de bază a aplicației, urmând ca apoi diferiți membri ai echipei proiectului să se ocupe de implementarea diferitelor blocuri/module funcționale. Schema blocurilor funcționale este dată în Fig. 1. Modulele funcționale vor fi prezentate în detaliu în descrierea modului de funcționare a aplicației mobile de autentificare.

Pentru a putea satisface cerințele și specificațiile unei aplicații care să funcționeze atât pe PC, cât și pe terminale mobile, algoritmii de autentificare deja dezvoltați de către echipa de cercetare au fost adaptați și optimizați pentru setul de specificații ale aplicați.

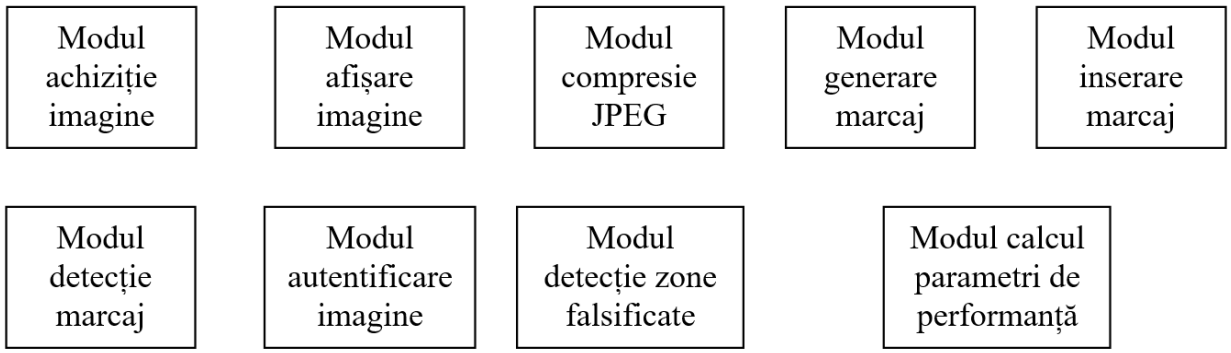


Fig. 1. Module funcționale ale aplicației de autentificare a imaginilor

În continuare vom prezenta algoritmul ce a fost implementat ulterior de echipa proiectului pentru aplicația de autentificare a imaginilor pentru terminale mobile folosind IDE-ul Android Studio. Algoritmul inițial a realizat în mediul de programare Matlab pentru PC. Tehnica a fost dezvoltată pentru imagini cu nuanțe de gri codate JPEG, dar a fost ulterior adaptat pentru imagini color prin trecerea de la spațiul de culori RGB la spațiu YCbCr și inserarea marcajului de autentificare în componenta de luminanță Y. La algoritmul propus, în fiecare bloc de luminanță de 8x8 pixeli al imaginii originale va fi inserat câte un watermark de autentificare de n biți. Marcajul de autentificare are două componente: o componentă pseudo-aleatoare și o componentă dependentă de conținutul blocului. Componenta dependentă de conținut este utilizată pentru a proteja schema împotriva atacurilor de tip cuantizare a vectorilor, la care un atacator utilizează blocuri din diferite imagini autentice pentru a obține o imagine falsificată.

O cheie privată K este utilizată pentru a obține o secvența pseudo-aleatoare binară, notată **PRBS**. Lungimea necesară a acestei secvențe poate fi calculată luând în considerare câte biți vor fi inserați în fiecare bloc de dimensiune $M \times N$ pixeli. Numărul de blocuri în cazul particular al $M=N=8$ va fi $d=W \times H / 64$, unde W și H reprezintă lățimea și înălțimea imaginii, exprimate în pixeli. Se poate observa că dacă lățimea sau înălțimea imaginii în pixeli nu se pot exprima ca un multiplu întreg de 8 pixeli, vor exista două benzi la marginea imaginii ai căror pixeli nu vor conține nici o informație securizată. Aceasta nu este o problemă deoarece zonele neasigurate ale imaginii vor fi foarte înguste (două benzi, una verticală și una orizontală, cu o lățime și respectiv o înălțime de maximum 7 pixeli) și nu pot conține detalii relevante.

Indicii de poziție a blocurilor vor fi utilizați pentru componenta dependentă de conținut a marcajului. Fiecare dintre cei doi indici vor fi convertiți în câte o secvență binară de 12 biți, iar cheie privată K este folosită pentru a selecta un număr n din cei 12 biți, obținând secvențele $\boldsymbol{b}_{1,i}$ și $\boldsymbol{b}_{2,i}$. Watermark-ul de autentificare $\mathbf{W_i}$ de n biți al blocului i este obținut aplicând XOR (sau exclusiv) între vectorii binari $\mathbf{b}_{1,i}$ și $\mathbf{b}_{2,i}$ și secvența **PRBS**:

$$\mathbf{W_i} = \mathbf{PRBS_i} \oplus \mathbf{b_{1,i}} \oplus \mathbf{b_{2,i}}, \quad i = \overline{1,d} \qquad (1)$$

Inserarea marcajului generat anterior într-un bloc al imaginii va fi prezentată în continuare. Prima operație care se aplică blocului curent de luminanță de 8x8 pixeli este transformarea DCT bidimensională. Această metodă de autentificare a imaginii este proiectată pentru a fi rezistentă la o compresia JPEG a imaginii securizate, utilizând un factor de calitate mai mare decât o limită minimă indicată q_{min}. Notând cu $\mathbf{QM}_{50}$ matricea de cuantizare JPEG

standard pentru blocuri de luminanță, de dimensiune 8 × 8 pixeli folosită pentru o compresie JPEG cu un factor de calitate egal cu 50, matricea de cuantizare notată $\mathbf{Q}_{q_{\min}}$, corespunzătoare factorului de calitate q_{min} se calculează după cum urmează:

$$\mathbf{Q}_{q_{\min}} = \begin{cases} \text{round}\left(50 \cdot \mathbf{QM}_{50} / q_{\min}\right), 1 \le q_{\min} \le 50 \\ \text{round}\left[\mathbf{QM}_{50}\left(2 - 0.02 \cdot q_{\min}\right)\right], 50 < q_{\min} \le 100 \end{cases} \quad (2)$$

Datele de autentificare nu vor fi introduse în primul coeficient al DCT (coeficientul DC) deoarece ar influența considerabil imaginea și ar cauza o pierdere notabilă de calitate. De asemenea, nici coeficienții de frecvență înaltă nu vor participa la acest proces, deoarece aceștia sunt cuantizați puternic, dacă se utilizează un factor de calitate JPEG mic. Va fi folosit un număr de *n* coeficienți de frecvențe joase spre medii pentru inserarea datelor auxiliare, aleși pe baza cheii secrete *K*. Inserarea se face folosind o variantă modificată a tehnicii modulației indicelui de cuantizare, conform ecuației (3).

$$\mathbf{C}_{i,s}(x,y) = \text{round}\left[\frac{\mathbf{C}_i(x,y)}{2 \cdot \mathbf{Q}_{q_{\min}}(x,y)} - \mathbf{W}(n \cdot i - n + j)\right] \cdot 2 \cdot \mathbf{Q}_{q_{\min}}(x,y) + \\ + \mathbf{W}(n \cdot i - n + j) \cdot \mathbf{Q}_{q_{\min}}(x,y),\ j = \overline{1,n}, \quad (3)$$

unde *x* și *y* sunt coordonatele fiecărui dintre cei *n* coeficienți DCT în care vor fi inserate datele, $\mathbf{C}_i$ este coeficientul DCT din blocul cu numărul *i* din imaginea originală și $\mathbf{C}_{i,s}$ este coeficientul DCT din blocul cu numărul *i* după înserare. După inserarea marcajului de autentificare în fiecare bloc, este aplicată Transformata DCT Inversă (IDCT) pe fiecare bloc de coeficienți DCT pentru a obține imaginea protejată.

În Fig. 2 este dat un exemplu de inserare a unui marcaj de autentificare de 6 biți W = [1 0 1 1 0 1] într-un bloc de coeficienți, protejându-l la compresie JPEG cu factori de calitate mai mari de 70. În Fig. 2(a) este figurat blocul original de coeficienți DCT, cu coeficienții ce pot fi utilizați pentru inserare reprezentați cu gri. În Fig. 2(b) doar cei 6 coeficienți DCT selectați pentru inserare prin intermediul cheii secrete K sunt reprezentați cu gri. Figura 2(c) conține matricea de cuantizare JPEG pentru factorul de calitate de 50, iar Fig. 2(d) pe cea pentru factorul de calitate de 70. Fig. 2(e) prezintă matricea de cuantizare modificată pentru a cuantiza blocul din Fig. 2(a), iar Fig. 2(f) conține blocul de coeficienți marcați. De remarcat, că doar coeficienții gri sunt modificați, toți ceilalți rămânând nemodificați.

Pentru a efectua extragerea și autentificarea datelor de securitate, împreună cu imaginea securizată, trebuie furnizată și cheia privată și factorul de calitate q_{min}. Marcajul de autentificare original va fi regenerat local pe baza cheii secrete *K*, folosind același algoritm ca cel descris la codare și va fi comparat cu marcajul extras din imaginea de test. Cel din urmă este extras prin cuantizarea coeficienților DCT selectați utilizând cheia privată cu matricea de cuantizare JPEG $\mathbf{Q}_{q_{\min}}$:

$$\mathbf{W}'(n \cdot i - n + j) = \text{mod}\left\{\text{round}\left[\frac{\mathbf{C}'_i(x,y)}{\mathbf{Q}_{q_{\min}}(x,y)}\right], 2\right\},\ j = \overline{1,n}. \quad (4)$$

Fig. 3 prezintă două exemple de aplicare a algoritmului pe două imagini cu nuanțe de gri. Imaginile au fost marcate și protejate la compresie JPEG cu factori de calitate mai mari decât q_{min}=50. Apoi imaginile au fost falsificate în diferite moduri și comprimate cu un factor de calitate *q*=50.

Algoritmul a fost testat pe o bază de date de 100 de imagini de rezoluție 512x512 pixeli. Un obiectiv principal a fost să asigurăm imperceptibilitatea marcajului în imaginile marcate. Pentru aceasta am ales empiric numărul de coeficienți utilizați pentru inserarea marcajului într-un bloc DCT pentru protecție la diferiți factori de calitate JPEG astfel încât PSNR-ul imaginilor marcate să fie aproximativ 45 dB. Astfel, pentru q_{min}=50 se poate folosi un număr de n=3 din primii 5 coeficienți AC în ordinea scanării în zigzag, pentru q_{min}=60 n=4 din primii 7 și pentru q_{min}=70 n=6 din primii 11 coeficienți AC. Aceste valori asigură imperceptibilitatea marcajului de autentificare inserat.

602	92	-57	24	9	-5	4	-3
-72	45	-56	-10	-8	10	3	-2
63	-21	-13	12	7	0	-4	-2
31	12	-7	11	-8	-4	3	2
19	-5	3	1	-1	-2	-3	0
9	-2	-1	1	0	0	2	-3
3	4	-3	3	-1	-2	3	0
5	-3	2	1	-4	-1	0	0

(a)

602	92	-57	24	9	-5	4	-3
-72	45	-56	-10	-8	10	3	-2
63	-21	-13	12	7	0	-4	-2
31	12	-7	11	-8	-4	3	2
19	-5	3	1	-1	-2	-3	0
9	-2	-1	1	0	0	2	-3
3	4	-3	3	-1	-2	3	0
5	-3	2	1	-4	-1	0	0

(b)

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(c)

10	7	6	10	14	24	31	37
7	7	8	11	16	35	36	33
8	8	10	14	24	34	41	34
8	10	13	17	31	52	48	37
11	13	22	34	41	65	62	46
14	21	33	38	49	62	68	55
29	38	47	52	62	73	72	61
43	55	57	59	67	60	62	59

(d)

1	7	6	10	1	1	1	1
1	1	1	1	1	1	1	1
8	8	1	1	1	1	1	1
8	10	1	1	1	1	1	1
11	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

(e)

602	91	-60	30	9	-5	4	-3
-72	45	-56	-10	-8	10	3	-2
64	-24	-13	12	7	0	-4	-2
31	20	-7	11	-8	-4	3	2
19	-5	3	1	-1	-2	-3	0
9	-2	-1	1	0	0	2	-3
3	4	-3	3	-1	-2	3	0
5	-3	2	1	-4	-1	0	0

(f)

Fig. 2. Exemplu de inserare a unui marcaj de 6 biți într-un bloc de coeficienți DCT

(a) (b) (c)

Fig. 3. Rezultatele autentificării a două imagini: (a) imagini originale; (b) imagini cu watermak falsificate si comprimate cu un factor de calitate de 60; (c) imagini autentificate

Ca să testăm performanțele de detecție ale algoritmului au fost utilizate ca metrici ratele de detecție de fals pozitiv (*DFP*) și fals negativ (*DFN*). *DFP* este calculată pentru a măsura performanța algoritmului în ceea ce privește probabilitatea de alarmă falsă în lipsa falsificării și este dată în ecuația (5), unde N_{FP} este numărul de pixeli nefalsificați care au fost determinați de sistem ca fiind falși și N_A este numărul total de pixeli autentici.

$$DFP = \frac{N_{FP}}{N_{\mathrm{A}}} \tag{5}$$

DFN, dată in ecuația (6), este utilizată uzual pentru a determina probabilitatea de a nu detecta falsificarea, atunci când ea a avut loc:

$$DFN = \frac{N_{FN}}{N_{\mathrm{F}}}, \tag{6}$$

unde N_{FN} este numărul de pixeli falsificați care au fost determinați de sistem ca fiind autentici și N_F este numărul total de pixeli falsificați.

Pentru a testa robustețea la compresie JPEG, fiecare imagine din baza de date a fost marcată folosind q_{min}=50, iar apoi comprimata cu factorii de calitate 50, 60, 70, 80 și 90. Pentru fiecare imagine comprimată am obținut *DFP*=0, demonstrând robustețea marcajului la compresie JPEG cu factori de calitate mai mari decât q_{min}.

Apoi am testat capacitatea sistemului de a detecta falsificări, chiar dacă imaginile falsificate sunt comprimate JPEG. Pentru aceasta fiecare imagine din baza de date a fost falsificată prin înlocuirea de regiuni de diferite dimensiuni din imagine cu zone de aceeași dimensiune din alte imagini. Zonele falsificate au fost alese de diferite dimensiuni, de la 16x16 la 356x356 pixeli. Imaginile falsificate au fost comprimate cu diferiți factori de calitate mai mari decât q_{min}. În Tabelul 1 este dată valoarea DFN medie în procente pentru diferite valori ale q_{min}. Valori mai mari ale DFN se obțin doar pentru dimensiuni foarte mici ale zonelor falsificate și apar de regulă la marginile zonei falsificate.

În concluzie, putem afirma că tehnica de autentificare propusă obține o calitate bună a imaginilor marcate de circa 45 dB și poate face diferența cu succes între o falsificare intenționată de conținut și compresia JPEG cu factori de calitate mai mari decât 50.

Tabelul 1. Valori DFN medii

Dimensiune zonă falsificată	***DFN* (%)**		
	q_{min}=50	**q_{min}=60**	**q_{min}=70**
16x16	9,27	3,35	2,02
32x32	3,93	2,12	0,20

64x64	0,98	0,3	0,18
96x96	0,67	0,22	0,04
128x128	0,31	0,17	0,03
192x192	0,17	0,08	0,00
256x256	0,1	0,05	0,01
356x356	0	0,02	0,00

Dezvoltarea aplicației de autentificare a imaginilor pentru terminale mobile (etapa 2017)

În continuare, echipa proiectului a implementat algoritmul propus anterior în Android Studio. Pentru aceasta, tehnica prezentată anterior a fost adaptată pentru a funcționa pentru imagini color, prin transformarea imaginii color din planul de culori RGB în planul de culori YC_bC_r și inserarea marcajului de autentificare doar în planul de luminanță Y.

Au fost dezvoltate următoarele module structurale ale aplicației: modulul de achiziție imagine, de afișare, de generare și inserare a watermark-ului de autentificare, detectorul de watermark și modulul de autentificare a imaginii și de localizare a falsificării.

Aplicația trebuie să fie capabilă să capteze imagini utilizând camera integrată, apoi să le protejeze astfel încât orice modificare ulterioară să poată fi detectată. Pentru a analiza imaginile și a determina dacă acestea sunt în forma originală sau au fost editate, se poate utiliza aceeași aplicație mobilă sau un software pentru PC. Aceasta implică faptul că aplicația trebuie să poată încărca fișiere din memoria telefonului.

Este foarte important să subliniem faptul că imaginea nu trebuie să fie pusă la dispoziția utilizatorului în forma capturată înainte de a fi securizată. De asemenea, aplicația nu trebuie să poată introduce date de securitate în imaginile încărcate. Aceste condiții asigură că un utilizator nu poate proteja o fotografie deja editată.

Există situații în care este necesară protejarea unei imagini fotografiate în trecut. De exemplu, utilizatorii ar dori să-și protejeze imaginile înainte de a le încărca pe Internet, pe rețelele sociale, pe galerii foto online, etc. Acesta este un caz special și imaginile securizate în acest fel pot fi marcate pentru a evidenția faptul că datele de autentificare nu au fost inserate chiar după ce au fost capturate și acele imagini au fost disponibile pentru editare înainte de a le securiza. Având în vedere aceste necesități, aplicația a fost dezvoltată pentru a proteja imagini imediat după fotografiere, dar și pentru imagini încărcate din spațiul de stocare al telefonului.

Principalele cerințe pentru aplicația mobilă în prima sa versiune sunt:

- Să poată captura fotografii utilizând camera foto a telefonului;
- Să poată încărca imagini din memoria telefonului;
- Să conțină butoane pentru pornirea proceselor de securizare și de analiză;
- Să permită vizualizarea imaginilor;
- Să afișeze mesaje de informare: imaginea a fost protejată, imaginea este autentică, imaginea este falsificată etc.;
- Să permită utilizatorului să modifice setările pentru algoritmul de autentificare.

Dezvoltarea aplicațiilor mobile pentru Android se poate realiza utilizând diferite platforme software cum ar fi Xamarin, Application Craft, Basic4Android și altele. În cadrul

proiectului a fost folosit Android Studio, mediul de dezvoltare oficial (en. IDE - Integrated Development Environment) pentru dezvoltarea aplicației mobile. Înainte de lansarea Android Studio în 2014, dezvoltarea pentru Android a fost făcută preponderent folosind IDE-ul Eclipse. Chiar dacă este instrumentul oficial de dezvoltare a aplicațiilor Android, Android Studio încă nu este recunoscut că oferă o experiență complet intuitivă. Folosirea sa necesită multe cunoștințe noi. Problema principală este că informația oficială este prea densă, iar utilizatorul care începe cu ea (și nu este neapărat un începător în domeniul programării) se regăsește frecvent solicitând informații pe forumuri și alte comunități. Din fericire, începând din 2014, multe subiecte sunt deja tratate și nu trebuie căutat prea mult înainte de a rezolva o problemă comună. Interfața aplicației Android Studio este prezentată în Fig. 4.

Android Studio oferă instrumente intuitive pentru a construi interfața de utilizare a aplicației aflate în curs de dezvoltare, dar orice altceva trebuie obținut prin scriere de cod. Necesitatea de a scrie cod pentru funcții de bază cum ar fi accesarea camerei pentru a face fotografii, accesarea galeriei pentru a încărca o fotografie salvată anterior, salvarea unei fotografii, vizualizarea unei fotografii reprezintă sarcini care consumă timpul care ar putea fi alocat pentru a dezvolta nucleul aplicației, și anume algoritmii de inserare și analiză. Din fericire, există instrumente care ușurează dezvoltarea de aplicații care implică utilizarea funcțiilor de bază ale telefonului, cum ar fi MIT App Inventor.

MIT App Inventor este o platformă de dezvoltare a aplicațiilor online. Proiectul urmărește să ușureze dezvoltarea de software astfel încât tot mai mulți oameni să fie implicați în crearea de software pentru dispozitive mobile. Codarea se face folosind blocuri funcționale care pot fi interconectate pentru a programa sarcini complexe. De asemenea, oferă instrumente pentru a genera cu ușurință o interfață cu utilizatorul. La prima vedere, pare a fi platforma perfectă pentru dezvoltarea aplicației de autentificare activă a imaginilor, dar, din păcate, nu permite utilizatorilor să-și scrie propriul cod, astfel încât algoritmii de inserare și analiză a imaginilor nu pot fi implementați cu ajutorul acestuia. Interfața de utilizare a aplicației MIT App Inventor este prezentată în Fig. 5.

Dezavantajul Android Studio este că solicită timp pentru codarea modulelor de bază ale aplicației, în timp ce problema aplicației MIT App Inventor este că nu poate fi utilizată pentru dezvoltarea de aplicații specializate, deoarece nu permite utilizatorilor să-și scrie propriul cod. Aceste două probleme ar putea fi rezolvate dacă cele două platforme ar putea fi integrate una cu cealaltă. Din fericire, acest lucru este posibil. Dezvoltatorii aplicației MIT App Inventor furnizează o bibliotecă Java (numită Java Bridge) cu funcții care pot fi apelate pentru efectuarea sarcinilor de bază. Numele funcțiilor sunt aceleași ca și blocurile pe care se bazează în App Inventor.

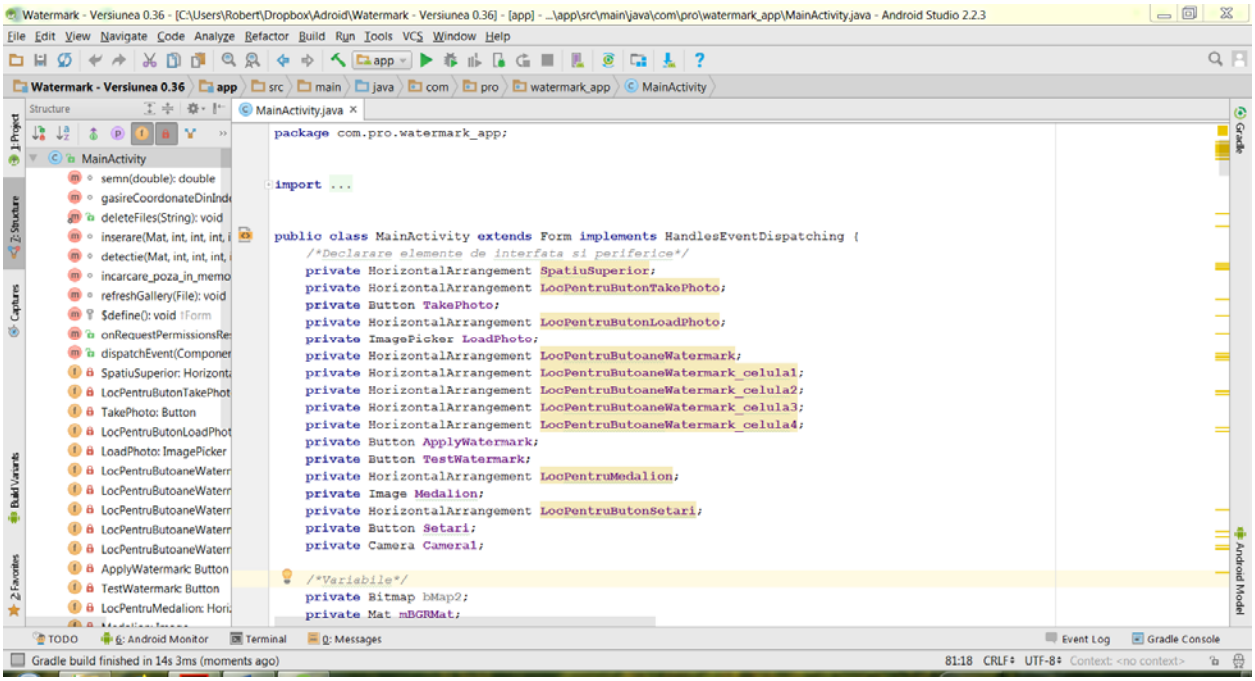


Fig. 4. Interfața Android Studio.

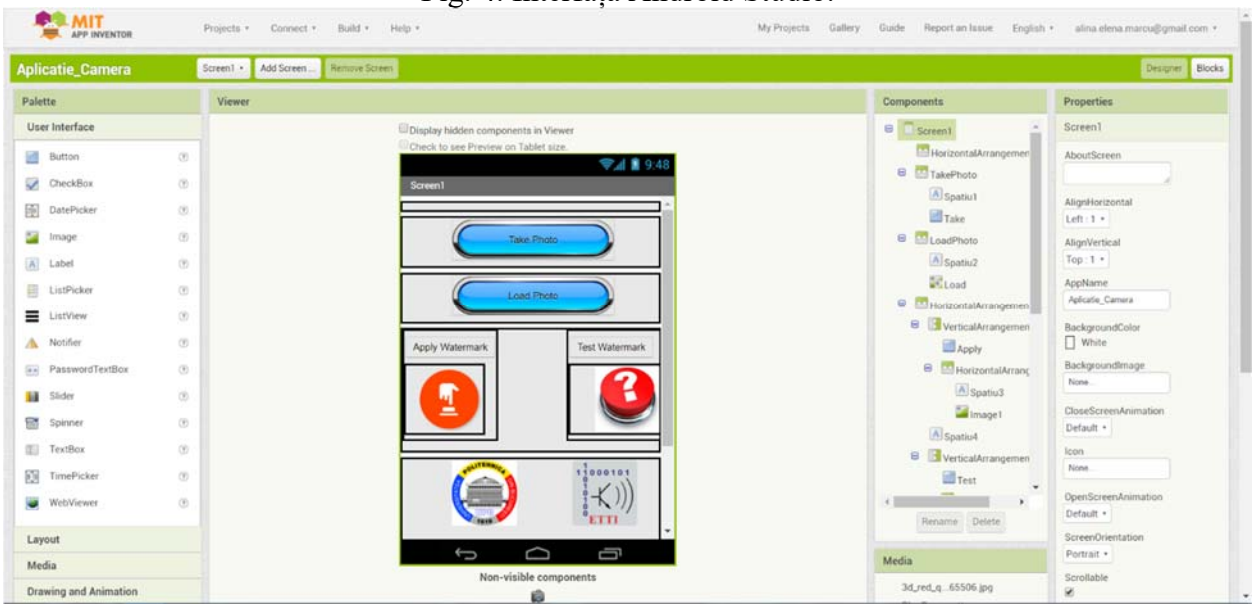


Fig. 5. Interfața MIT App Inventor.

Dezavantajul utilizării Java Bridge este că dezvoltarea interfeței cu utilizatorul nu poate fi realizată utilizând instrumentele Android Studio, deoarece elementele (butoane, câmpuri etc.) nu mai pot fi referite utilizând funcțiile Java Bridge. Din această cauză interfața trebuie să fie dezvoltată utilizând funcțiile specifice ale Java Bridge.

Algoritmul de inserare și autentificare utilizat de aplicația Android este cel dezvoltat de echipa proiectului și prezentat mai sus. Imaginile sunt protejate prin inserarea unor date auxiliare, a unui marcaj de autentificare. Inserarea se face într-un mod care să asigure că marcajul de autentificare va fi alterat la modificarea conținutului imaginii. Prin extragerea din imaginea de test a marcajului la decodor și compararea lui cu secvența originală inserată,

generată local, se poate determina dacă imaginea a fost modificată sau nu. De asemenea, introducerea datelor auxiliare nu trebuie să afecteze vizual conținutul imaginii (imaginea cu watermark inserat trebuie să pară identică cu imaginea fără marcaj inserat când este analizată cu ochiul liber). Un astfel de algoritm activ dă un grad mai mare de certitudine decât metodele pasive (care analizează pur și simplu conținutul imaginii fără a le modifica în vreun fel), dar dezavantajul lor este că nu orice imagine poate fi testată pentru autenticitate, ci doar cele care conțin datele de securitate inserate în prealabil. Tehnicile active se pretează pentru aplicații mobile, deoarece imaginile ce se doresc a fi protejate sunt de regulă captate folosind camera foto a dispozitivului mobil, iar marcajul de autentificare poate fi inserat imediat după captura foto.

Tehnicile de autentificare activă utilizate pentru realizarea aplicației Android sunt metode de inserare în domeniul transformat, deoarece acestea obțin rezultate mai bune decât cele în domeniul spațial (în valoarea luminanței sau a crominanței pixelilor imaginii), atât în ceea ce privește calitatea imaginilor marcate, cât și a robusteții la prelucrări uzuale de imagini, ce nu pot fi considerate o falsificare a imaginii. Astfel, înainte de inserare, o operație de transformare este aplicată imaginii. Există multe transformate folosite în prelucrarea digitală a imaginilor, cele mai frecvent utilizate fiind Transformata Cosinus Discretă (en. DCT – Discrete Cosine Transform) și Transformata Wavelet Discretă (en. DWT – Discrete Wavelet Transform). Deoarece aplicația propusă este destinată a fi rulată pe un dispozitiv mobil care utilizează compresia JPEG pentru a comprima imaginile captate, iar timpul de procesare trebuie să fie cât mai scurt posibil, DCT a fost aleasă ca domeniu de inserare. Astfel, datele de autentificare vor fi inserate în coeficienții DCT.

O caracteristică dorită a aplicației este de a detecta cu precizie ce zonă a imaginii a fost falsificată, nu doar dacă imaginea testată este originală sau nu. Pentru aceasta, imaginea nu va fi prelucrată în ansamblu, ci va fi împărțită în blocuri de M x N pixeli. Dimensiunea blocului ne dă rezoluția minimă de detecție a unei regiuni falsificate din imagine. Un bloc mai mare va oferi o securitate mai puternică, deoarece secvența de autentificare introdusă ar putea fi mai lungă, în timp ce un bloc mai mic ar garanta o mai bună rezoluție de localizare a zonelor falsificate, deci trebuie făcut un compromis între cele două cerințe. Dimensiunea aleasă pentru blocurile de procesare a fost M=N=8 pixeli, la fel ca cea utilizată la standardul de compresie JPEG. Această dimensiune a blocului oferă o localizare bună a zonei și, deoarece este o dimensiune standard, pot fi utilizate resurse partajate de la alți algoritmi, cum ar fi tabelele de cuantizare JPEG, simplificând dezvoltarea. Algoritmul va insera un număr de n biți de date de autentificare în fiecare bloc de 8x8 pixeli.

Prima versiune a aplicației de autentificare a imaginilor (bazată pe Android SDK)

Inițial, aplicația a fost dezvoltată utilizând Android Studio și biblioteca Java Bridge. Utilizând Kitul de Dezvoltare Software Android (en. SDK - Software Development Kit) se poate dezvolta o aplicație care utilizează maximum 256 megaocteți (MO) de memorie. O imagine de 16 Mpixeli cu 4 planuri (32 de biți per pixel) poate ocupa până la 129 MO de memorie. Chiar dacă planul de transparență nu este util în această situație, atunci când se citește imaginea, nu există nicio modalitate de a evita citirea planului respectiv folosind funcțiile standard. Aplicația a ajuns rapid la limita de memorie și toate tehnicile de optimizare încercate nu au rezolvat această problemă. O altă problemă cu Android SDK este că nu oferă funcții pentru prelucrarea ușoară a imaginilor.

A doua versiune a aplicației de autentificare a imaginilor (bazată pe Android NDK)

Pentru mai multă flexibilitate, viteză și eliminarea limitei de memorie software, trebuie folosit Android NDK (Native Development Kit). Funcțiile furnizate în Android SDK nu erau satisfăcătoare din punct de vedere performanță și s-a preferat utilizarea NDK. O soluție este folosirea unei părți din biblioteca open source de computer vision, openCV, varianta dezvoltată special pentru Android. Marele avantaj al acesteia este că este implementată utilizând NDK. Funcțiile furnizate au fost investigate pentru a traduce operațiile matematice necesare discutate mai sus în limbajul de programare corespunzător.

Depistarea problemelor utilizând doar Android Studio poate ridica probleme din cauza modului reprezentării datelor și uneori a lipsei suportului pentru explorarea detaliată a lor (nu se oferă detalii despre conținutul instantaneu al unor variabile). Trei telefoane mobile reale au fost utilizate în faza de dezvoltare pentru a asigura compatibilitatea cu diferite versiuni de Android și dimensiunile de ecrane, precum și testarea performanțelor pe diferite sisteme: OnePlus 3, Motorola Moto X și Samsung Galaxy S5 mini. Depanarea cu terminalul emulat software nu este potrivită pentru că trebuie să fie capturate imagini reale și, de asemenea, pentru că a manifestat instabilitate.

Sarcina principală a fost implementarea modulelor aplicației și testarea algoritmului, astfel încât interfața a fost păstrată pentru moment simplă, conținând doar elementele necesare: 5 butoane și un cadru de imagine pentru afișarea fotografiilor.

Două butoane sunt utilizate pentru a încărca o imagine în memorie, fie prin captură directă folosind camera dispozitivului mobil, fie prin încărcarea unei imagini salvate deja existentă în memoria internă a terminalului mobil. După încărcare, imaginea este afișată în cadrul special destinat vizualizării. S-a observat că pe unele telefoane cu ecran de rezoluție mai mică au existat unele probleme cu afișarea imaginilor de mari dimensiuni, în cadrul destinat vizualizării imaginii. Soluția a fost să se creeze o altă imagine temporară care să stocheze o replică a imaginii care urmează să fie afișată, dar cu cea mai mare dimensiune forțată la 1000 de pixeli, menținând raportul de aspect al imaginii originale.

Alte două butoane sunt utilizate pentru a porni cele două operații principale: inserarea datelor de autentificare în imaginea originală și testarea autenticității acesteia. În așteptare, aceste butoane sunt roșii și devin verzi imediat ce operațiunea respectivă este încheiată. Ultimul buton este utilizat pentru accesarea și modificarea setărilor aplicației.

După testarea autenticității unei imagini, se afișează un mesaj care arată rezultatul: „Imaginea este autentică” sau „Imaginea este falsificată”, indicând și numărul de blocuri detectate ca fiind falsificate, iar imaginea cu zonele falsificate marcate cu roșu este afișată în cadrul destinat vizualizării imaginii, astfel încât utilizatorul poată observa cu ușurință și precis regiunile falsificate.

Ecranul de pornire al aplicației dezvoltate este prezentat în Fig. 6. Deoarece cadrul de imagine trebuie să găzduiască o imagine suficient de mare, interfața a fost realizată astfel încât să se poată derula vertical.

Dacă se apasă butonul „Take photo” (captează imagine), aplicația va lansa automat aplicația principală care gestionează camera foto, așteptând ca utilizatorul să facă o fotografie. După fotografiere, aplicația încarcă automat poza în memorie și generează o replică temporară cu cea mai mare dimensiune redusă la 1000 de pixeli, menținând raportul de aspect, care va fi afișat în cadrul pentru vizualizare a imaginilor. Dacă dimensiunea cea mai mare a imaginii achiziționate este mai mică de 1000 de pixeli, imaginea originală poate fi afișată direct în cadrul imaginii. Acest lucru crește considerabil compatibilitatea cu telefoanele mai vechi sau cu costuri reduse, care au ecrane de rezoluție mică.

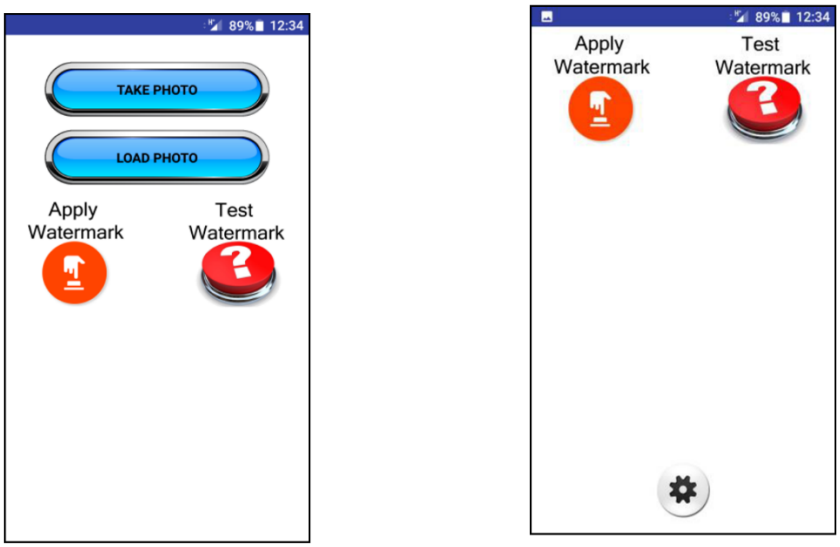


Fig. 6. Interfața aplicației dezvoltate în versiunea v1: stânga – partea de sus a ecranului de pornire; dreapta – partea de jos a ecranului de pornire

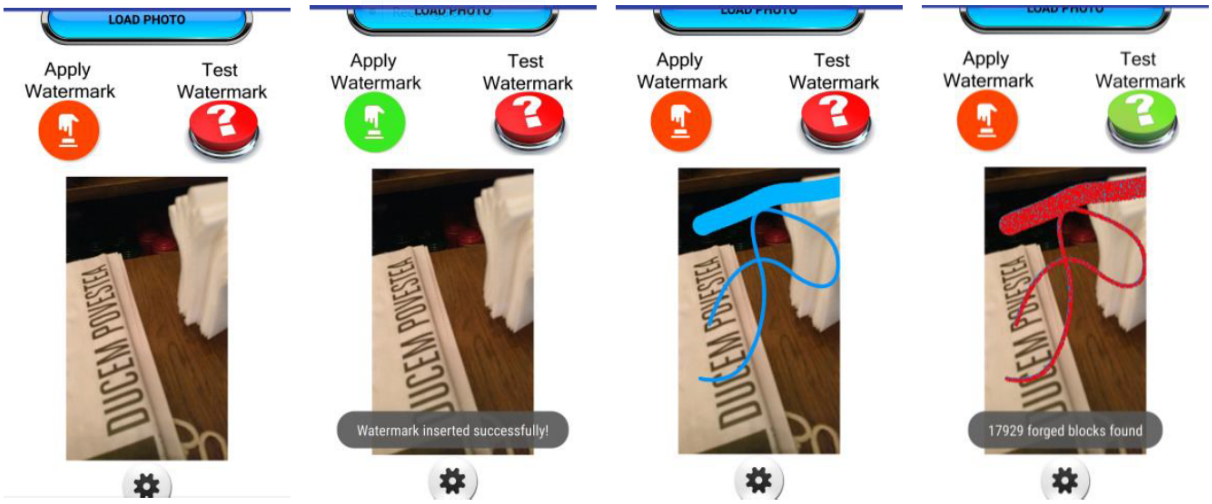
Apăsarea butonului „Încărcare fotografie” va duce la lansarea aplicației principale de galerie, permițând utilizatorului să selecteze o fotografie care să fie încărcată în memorie, pregătind-o pentru o prelucrare ulterioară.

Butonul „Apply watermark” (aplică marcaj) pornește procesul de introducere a datelor de securitate în imaginea încărcată. Introducerea cu succes este marcată prin colorarea butonului respectiv în verde și afișarea mesajului „Watermark inserted succesfully” (marcaj inserat cu succes). Pe un procesor Quad-core (2x2,15 GHz Kryo & 2x1,6 GHz Kryo), a durat 13 secunde pentru a introduce datele de securitate într-o imagine de 12 megapixeli. Procesarea se face folosind un singur thread în această versiune a aplicației.

Butonul „Test Watermark” (testează imaginea) lansează procesul de autentificare, extrăgând datele de autentificare din imaginea încărcată și comparându-le cu marcajul original generat local pe baza cheii K. La sfârșitul acestui proces, butonul va deveni verde. În funcție de rezultat, se va afișa unul dintre cele două mesaje posibile: „The image is authentic” (imaginea este autentică) sau „The image is NOT authentic. Check RED areas” (imaginea nu este autentică, verifică zonele roșii), acesta din urmă indicând și numărul blocurilor detectate ca fiind falsificate. Testarea autenticității unei imagini de 12 megapixeli pe același procesor a durat aproximativ 6 secunde, algoritmul de detecție consumând mai puține resurse.

Butonul „Settings” (setări), ilustrat ca roată dințată, permite setarea parametrului q_{min} (factorul de calitate la care este protejată imaginea), factorul de calitate cu care este comprimată imaginea după inserarea marcajului și culoarea cu care vor fi marcate zonele falsificate.

În Fig. 7 sunt prezentate toate etapele de inserare a marcajului de autentificare și de autentificare a unei imagini de test falsificate. În Fig. 7(a) imaginea capturată este afișată în cadrul pentru vizualizarea imaginilor. Aplicația este gata să introducă datele de securitate. Apăsarea butonului „Apply watermark” (aplică marcaj) va începe introducerea datelor de autentificare.



(a) (b) (c) (d)

Fig. 7. Etapele inserării marcajului de autentificare și detecției falsificării: (a) Imaginea încărcată în memorie, pregătită de securizare; (b) Inserarea marcajului de autentificare s-a efectuat cu succes; (c) Imaginea falsificată pregătită de testare; (d) Imaginea detectată ca fiind falsificată, cu zonele falsificate marcate cu roșu

În Fig. 7(b) se observă operațiunea finalizată cu succes. Imaginea protejată a fost încărcată în programul de editare foto Pixlr și a fost modificată pentru a demonstra funcționarea aplicației. Imaginea falsificată a fost încărcată în aplicația dezvoltată așa cum poate fi observat în Fig. 7(c). În Fig. 7(d) procesul de autentificare a fost încheiat și rezultatele au fost afișate. Zonele falsificate au fost marcate cu roșu și numărul de blocuri detectate ca fiind false este afișat în partea de jos a ecranului.

Există un caz special care trebuie discutat: procesarea unei imagini care conține blocuri complet negre sau complet albe. După inserarea marcajului, s-ar putea produce depășiri din punct de vedere numeric, iar ca urmare aceste blocuri ar putea fi marcate ca fiind neautentice, chiar dacă ele în realitate au fost autentice. Soluția a fost ca înainte de inserare să se crească sau să se diminueze luminanța blocurilor respective cu valori corespunzătoare pentru a le aduce în valorile 5-250 (intervalul maxim fiind 0 - 255, unde 0 înseamnă negru și 255 înseamnă alb). Aceste mici modificări ale luminanței nu sunt vizibile cu ochiul liber, iar astfel se evită eventualele depășiri.

În continuare s-a încercat reducerea timpului de procesare, în special pe partea de codor. Timpul de procesare a fost diminuat cu 30% numai prin evitarea adresării repetate a aceluiași bloc folosind funcții openCV. S-a constatat că extragerea unui bloc dintr-o imagine durează mult timp. Această operație a fost efectuată o singură dată pe bloc, iar rezultatul a fost salvat într-o matrice temporară. Apoi această variabilă temporară a fost utilizată în toate operațiile ulterioare, evitând refolosirea funcției de extragere.

Aplicația de autentificare activă a imaginilor funcționează bine atât pe PC, cât și pe dispozitivele mobile, diferențiind cu succes falsificările conținutului imaginii de compresia JPEG. Aspectele care au fost supuse unor cercetări ulterioare, în etapa pe 2018 a proiectului, au fost optimizarea și accelerarea procesului de inserare a datelor auxiliare prin exploatarea tehnicilor multithreading, integrarea cu rețelele de socializare, îmbunătățirea securității prin alegerea diferită a coeficienților în care se inserează date auxiliare pentru fiecare bloc, utilizarea

morfologiei matematice pentru eliminarea falsurilor pozitive și negative, relansarea interfeței grafice a aplicației.

Alte activități desfășurate de echipa proiectului în această prima etapă includ: dezvoltarea unei tehnici de watermarking pentru ascunderea datelor pacienților în regiuni de non-interes din imagini medicale ce conțin leziuni cutanate și segmentarea zonelor ce conțin aceste leziuni pentru evaluarea riscului ca acestea să fie maligne; evaluarea posibilităților de dezvoltare a unei aplicații de autentificare a secvențelor video codate folosind standardul de compresie video H.265/HEVC (High Efficiency Video Coding); posibilități de integrare într-o aplicație de detecție a falsificării și a unui sistem de decizie pentru autentificarea vorbitorului.

În articolul științific cu titlul „Digital watermarking and segmentation of macroscopic pigmented skin lesions images" prezentat în cadrul conferinței științifice internaționale "International Symposium on Signals, Circuits and Systems (ISSCS) 2017" a fost prezentată o **utilizare practică a watermarking-ului digital într-o aplicație de evaluare automată a leziunilor pigmentare ale pielii**. Evaluarea automată a leziunilor pielii prin procesarea digitală a imaginilor macroscopice ce prezintă leziunilor pigmentare pielii (MPSL – Macroscopic Pigmented Skin Lesions) utilizează caracteristicile leziunii pentru calculul gradului de risc de melanom al unei leziuni. Precizia gradului de risc calculat crește atunci când se iau în considerare informații suplimentare precum: vârsta pacientului, genul, tipul pielii, poziția leziunilor pe corp și prezența leziunilor multiple și similare. În această lucrare am propus o schemă de prelucrare digitală a imaginilor MPSL pentru inserarea în acestea a informațiilor relevante ale pacientului folosind metode de watermarking digital.

Din punct de vedere statistic, melanomul este al cincilea tip de cancer cel mai comun dintre bărbați și se caracterizează prin cea mai rapidă rată de incidență [12]. În stadiile avansate, melanomul este incurabil și tratamentul, cum ar fi chimioterapia sau chirurgia, este folosit doar pentru prelungirea vieții [12]. Cu toate acestea, în stadiile incipiente melanomul poate fi ușor vindecat si nu va produce metastaze. Acest fapt evidențiază importanța detectării timpurie a acestei malignități.

În zilele noastre, există o activitate semnificativă de cercetare în domeniul evaluării riscurilor pentru melanomul malign pe baza procesării imaginilor și a utilizării caracteristicilor leziunilor. Majoritatea cercetărilor sunt efectuate pentru imaginile MPLS achiziționate utilizând o cameră digitală convențională. Intenția este de a dezvolta o aplicație PC sau telefon mobil inteligent care să poată evalua riscul de malignitate al leziunilor pigmentare ale pielii folosind tehnici de procesare a imaginilor.

Procesul de evaluare a riscului de melanom prin evaluarea caracteristicilor leziunii constă în trei etape importante [14, 15]: segmentarea imaginii pentru extragerea leziunilor pigmentare cutanate, extragerea caracteristicilor pentru calculul parametrilor necesari caracterizării formei, culorii și a graniței, și clasificarea leziunilor pe baza parametrilor extrași. Rezultatul procesului de evaluare a riscului de melanom a leziunii pielii este de obicei de natură booleană, adică o leziune poate fi evaluată ca benignă sau malignă. Având în vedere natura agresivă a acestui tip de cancer de piele, procentul de rezultat fals pozitiv, leziunile clasificate ca maligne atunci când rezultatele benigne și, în special, cele fals negative trebuie minimizate.

În [16] este prezentat un sistem automat de inspecție a leziunilor pigmentare ale pielii și a diagnosticului melanomului, care folosește imaginile MPSL obținute cu ajutorul unei camere digitale convenționale. Sistemul include o componentă de susținere a deciziilor, care combină rezultatul clasificării imaginilor cu cunoașterea contextului, cum ar fi tipul pielii, vârsta

pacientului, genul și partea afectată a corpului. Rezultatele din [16] arată că cunoașterea contextului poate fi utilă în procesul de diagnosticare automată a leziunilor.

Lucrarea prezentată în cadrul conferinței ISSCS 2017 descrie o schemă de watermarking digital utilizată pentru inserarea informațiilor pacientului în imagini MPSL. În acest fel, ne putem asigura că informațiile importante despre pacienți sunt disponibile atunci când sunt necesare și că imagini de la diferiți pacienți nu pot fi încurcate. Watermarking-ul digital aplicat trebuie să fie invizibil și robust la compresie JPEG. De asemenea, leziunile din imaginea MPSL nu trebuie să fie modificate prin procesul de watermarking, iar eroarea de segmentare care va apărea în urma modificării imaginilor ar trebui să fie redusă.

Diagrama bloc a metodei propuse de generare și inserare a watermark-ului este prezentată în Fig. 8. Scopul nostru a fost de a insera datele pacientului în imaginea medicală ca watermark fără a pierde vizibil din calitatea imaginii. Deoarece majoritatea imaginilor sunt stocate în formatul popular JPEG, obiectivul secundar a fost de a proteja watermark-ul inserat împotriva compresiei JPEG.

Watermark-ul care urmează a fi inserat este alcătuit din patru câmpuri diferite ce conțin informații despre pacient: genul pacientului (1 bit), vârsta (7 biți), numele (50 de caractere) și localizarea leziunii pielii pe corpul pacientului (50 de caractere). Aceste patru componente sunt convertite în formă binară și concatenate într-un vector binar ***w*** de 808 biți. Pentru a îmbunătăți securitatea algoritmului, watermark-ul obținut ***w*** este permutat pseudo-aleator pe baza cheii secrete ***K*** a utilizatorului, obținând secvența binară ***w1***. Folosind aceeași cheie secretă, o permutare inversă va fi efectuată la partea decodorului pentru a restabili biții de watermark în poziția inițială.

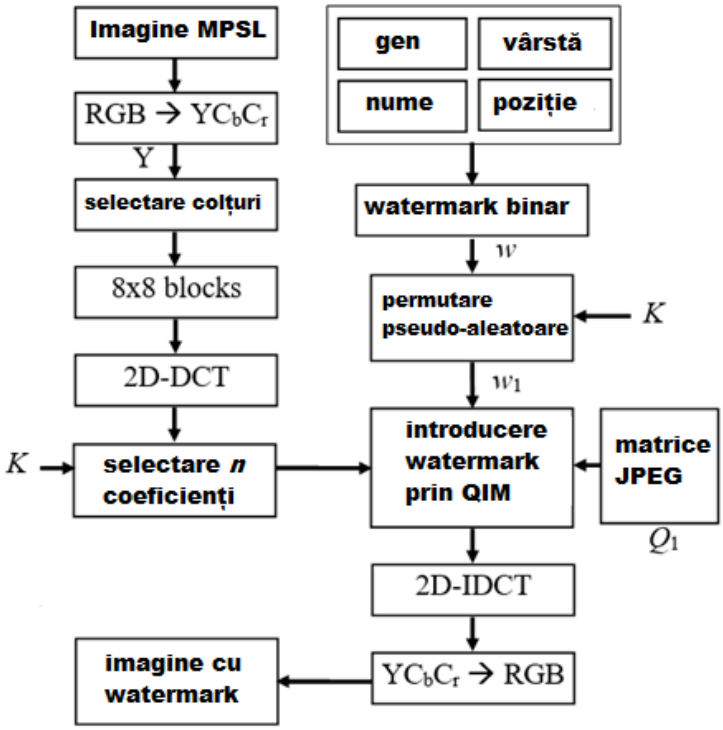


Fig. 8. Schema bloc de inserare a informațiilor pacientului în imaginile medicale de tip MPSL.

Imaginea color originală este mai întâi convertită de la spațiul de culoare RGB la spațiul de culoare YCbCr și numai componenta de luminanță Y este procesată ulterior. Deoarece

leziunea pielii este de obicei găsită în centrul imaginii și nu vrem să modificăm acea parte a imaginii în nici un fel, vom insera watermark-ul în cele patru colțuri ale imaginii. După măsurarea dimensiunii leziunilor pielii în diferite imagini, am ales dimensiunea unui colț $C = 0,2M \times 0,2N$, unde $M \times N$ este rezoluția imaginii originale. Apoi, împărțim cele patru colțuri de valoare a luminozității în blocuri care nu se suprapun de 8 × 8 pixeli și aplicăm pe fiecare bloc transformata cosinus discretă bidimensională (2D-DCT). Watermark-ul binar ***w1*** va fi inserat în coeficienți de frecvență joasă până la mijloc, utilizând o versiune modificată a algoritmului de modulație a indexului de cuantificare (QIM) bazat pe DCT prezentat în [17]. Matricea de cuantizare JPEG Q_1 pentru factorul de calitate q_1 este calculată pe baza matricea de cuantizare JPEG standard pentru blocuri de luminanță Q_{50} dată în Fig. 2(c):

$$Q_1 = \begin{cases} \text{round}(50Q_{50} / q_1), & \text{if } 1 \le q_1 < 50 \\ \text{round}(Q_{50}(2 - 0.02q_1)), & \text{if } 50 \le q_1 \le 100 \end{cases} \quad (7)$$

Din fiecare bloc de 8 × 8 coeficienți DCT numai un număr ***n*** de coeficienți de frecvență joasă spre medie va fi utilizat pentru inserarea watermark-ului. Pentru a evita o scădere importantă a calității imaginii, coeficientul DC va fi exclus. Numărul de coeficienți utilizați este $n = [s / (4C / 64)] + 1$, unde $s = 808$ este mărimea watermark-ului binar w_1, [.] reprezintă operația de calcul a părții întregi și $4C / 64$ este numărul de blocuri 8 × 8 din cele patru colțuri. După scanarea coeficienților DCT în ordine zig-zag, coeficienții ***n*** ai fiecărui bloc vor fi selectați aleator din intervalul indicilor [2; 9] în ordinea scanării în zigzag, folosind cheia secretă ***K***.

Pentru ca watermark-ul să reziste la compresia JPEG cu factori de calitate mai mari decât q_1, am exploatat următoarea proprietate a cuantizării: dacă ***x*** este un multiplu al unui pas de cuantizare predeterminat ***q1***, atunci pentru orice ***q2*** > ***q1*** este adevărat următorul lucru:

$$f\left(f\left(x, q_2\right), q_1\right) = x, \quad (8)$$

unde $f(x, q)$ denotă cuantificarea valorii ***x*** utilizând dimensiunea pasului ***q***. Aceasta înseamnă că, prin inserarea unui bit de watermark prin cuantizarea unui coeficient DCT cu o mărime a pasului corespunzător unui factor de calitate predeterminat ***q1***,watermark-ul inserat va fi elastic la compresia JPEG cu orice factor de calitate ***q2*** ≥ ***q1***, deoarece coeficientul DCT cuantizat poate fi reconstruit perfect prin cuantizarea folosind ***q1*** original.

În conformitate cu proprietatea anterioară, un bit de watermark $w_{1,j}$ va fi inserat într-un coeficient ***D (i)*** după cum urmează:

$$D_w(i) = round\left(\frac{D(i)}{2Q_1(i)} - w_{1,j}\right) 2Q_1 + w_{1,j}Q_1(i),\ j = \overline{1,s} \quad (9)$$

unde $D(i)$ este un coeficient DCT selectat la poziția ***i*** în ordinea de scanare zig-zag, $Q_1(i)$ este dimensiunea pasului de cuantizare în aceeași poziție ***i*** a matricei Q_1 de cuantizare scanată zig-zag și $D_w(i)$ este coeficientul DCT cu watermark. După ce toate blocurile DCT au fost marcate, Transformata Cosinus Discretă Inversă este aplicată fiecărui bloc pentru a obține planul de luminanță și imaginea YCbCr este transformată înapoi în spațiul de culoare RGB. Imaginea rezultată va fi robustă la compresia JPEG cu factori de calitate mai mari decât ***q1***.

Decodorul necesită cheia secretă ***K***, dimensiunea watermark-ului ***s*** și factorul de calitate JPEG q_1 utilizat în procesul de inserare pentru extragerea cu succes a watermark-ului inserat. Primii pași prezentați în secțiunea de inserare sunt de asemenea realizați și la decodor pentru a obține blocurile DCT ale valorilor luminanței pentru cele patru colțuri. Din fiecare bloc sunt selectați un număr de ***n*** coeficienți DCT folosind cheia ***K*** și un bit de watermark $w_{2,j}$ este extras

prin cuantizarea coeficienților utilizând valoarea corespunzătoare din matricea de cuantizare JPEG $Q_1(i)$ pentru factorul de calitate $\boldsymbol{q_1}$ ca pas de cuantizare și calcularea restului după împărțirea la 2:

$$w_{2,j} = round\left(D'(i)/Q_1(i)\right) \bmod 2, \quad j = \overline{1,s} \tag{10}$$

Biții de watermark extrași din blocurile DCT sunt concatenați într-un vector binar, iar permutarea inversă cu tasta secretă $\boldsymbol{K}$ este efectuată pentru a obține secvența de watermark extrasă de mărime $\boldsymbol{s}$.

Algoritmii de watermarking și de segmentare au fost testați pe o bază de date cu 31 de imagini color MPSL de dimensiuni diferite, 26 imagini din baza de date cu imagini Dermis [18] și 5 imagini capturate folosind o cameră de telefon mobil cu 13 MP, 4128 x 3096 pixeli, autofocus și LED flash. Unul dintre scopurile noastre principale a fost de a insera datele pacientului în imagini cu leziuni ale pielii prin asigurarea invizibilității watermark-ului. În acest scop, calculăm media PSNR pentru întreaga bază de date a imaginilor pentru diferite valori ale lui $q_1 \in \{50, 60, 70, 80, 90, 100\}$ și fără compresie.

Valorile PSNR obținute sunt peste 49 dB, asigurând invizibilitatea watermark-ului inserat. Desigur, după comprimarea imaginilor cu diferite valori $\boldsymbol{q_2}$, valorile PSNR vor fi mai scăzute, dar acest lucru se datorează compresiei JPEG și nu algoritmului nostru de watermarking.

În continuare, pentru a testa robustețea schemei de watermarking la compresia JPEG, fiecare imagine din baza de date a fost marcată cu q_1=50, comprimată cu $q_2 \in \{50, 60, 70, 80, 90, 100\}$ și a fost calculată rata erorii de bit (BER) între watermark-ul inițial și watermark-ul extras. Rezultatele sunt prezentate în Tabelul 2.

Putem vedea că valorile BER la decodare sunt zero sau foarte aproape de zero, dovedind robustețea watermark-ului la compresia JPEG.

Pentru evaluarea efectului inserării watermark-ului asupra segmentării leziunii am folosit 4 parametrii standard factorul DICE, rata de detecție corectă TDR, rata de fals pozitiv FPR și rata de eroare ERR.

$$DICE = \frac{2 \cdot Area\left(DS \cap GT\right)}{Area\left(DS\right) + Area\left(GT\right)} \cdot 100\left[\boldsymbol{\%}\right] \tag{11}$$

$$TDR = \frac{Area\left(DS \cap GT\right)}{Area\left(GT\right)} \cdot 100\left[\boldsymbol{\%}\right] \tag{12}$$

$$FPR = \frac{Area\left(DS \cap \overline{GT}\right)}{Area\left(GT\right)} \cdot 100\left[\boldsymbol{\%}\right] \tag{13}$$

$$ERR = \frac{Area\left(DS \oplus GT\right)}{Area\left(GT\right)} \cdot 100\left[\boldsymbol{\%}\right] \tag{14}$$

GT reprezintă masca de segmentare a imaginii fără watermark iar DS reprezintă masca de segmentare a imaginii cu watermark. Rezultatele sunt prezentate în Tabelul 3.

Tabelul 2. Valori PSNR și BER medii pentru diferiți factori de calitate

$\boldsymbol{q_1}$	$\boldsymbol{q_2}$	***PSNR* [dB]**	***BER***
50	100	49,65	0

	90	44,97	0
	80	43,27	0,0002
	70	42,74	0,002
	60	38,62	0,0018
	50	35,49	0.006

Tabelul 3. Parametrii de evaluare a segmentării.

	Valoare medie	**Deviație standard**	**Valoare minimă**	**Valoare maximă**
DICE	97.65%	2%	88.93%	99.01%
TDR	97.32%	1.46%	92.90%	98.85%
FPR	2.05%	3.62%	0.30%	19.50%
ERR	4.73%	4.28%	1.98%	23.81%

Rezultatele arată valori medii ridicate pentru DICE și TDR și valori medii scăzute pentru FPR și ERR. Aceasta arată că diferențele dintre cele două segmentări, adică segmentarea originală a imaginii și segmentarea imaginii cu marcaj, sunt mici. Valorile scăzute ale deviației standard indică faptul că rezultatele sunt fiabile.

De asemenea, a fost dezvoltat un algoritm sigur și robust de autentificare a imaginilor color codate JPEG bazat pe o semnătură digitală. Caracteristicile robuste ale imaginii utilizate pentru generarea semnăturii digitale a imaginii au fost extrase din anumiți coeficienți normați ai Transformatei Cosinus Discrete Bidimensionale. Semnătura extrasă este robustă la compresie JPEG și permite detecția și localizarea cu precizie a falsificării conținutului imaginii. În plus, algoritmul permite inserarea semnăturii criptate in metadatele EXIF ale header-ului imaginii codate JPEG și poate fi adaptat pentru a funcționa pe aplicații mobile.

Publicații

Munca de cercetare științifică depusă de către membrii echipei de cercetare în cadrul etapei pe 2017 a proiectului s-a concretizat prin publicarea a 6 articole științifice în volumele unor conferințe internaționale indexate ISI sau în curs de indexare ISI. Aceste articole sunt enumerate în cele ce urmează:

- I. Pirnog, R. O. Preda, C. Oprea and R. A. Dobre, *Digital watermarking and segmentation of macroscopic pigmented skin lesions images*, 2017 International Symposium on Signals, Circuits and Systems (ISSCS), ISBN: 978-1-5386-0674-2, pp. 1-4, Iasi, Romania, 13-14 July, 2017, DOI: 10.1109/ISSCS.2017.8034904, WOS:000425211500042. (ISI Web of Science, Scopus, IEEE Xplore)
- R. A. Dobre, R. O. Preda, I. Pirnog, C. Oprea, *Active image authentication and forgery localization for mobile devices*, 17th International Multidisciplinary Scientific GeoConference (SGEM 2017), ISBN 978-619-7408-01-0 / ISSN 1314-2704, Vol. 17, Issue 21, pp. 61-68, Albena, Bulgaria, 29 June - 5 July, 2017, DOI: 10.5593/sgem2017/21/S07.009. (SCOPUS, în curs de indexare ISI)

- R. A. Dobre, C. Negrescu, and D. Stanomir, *Improved Low Computational Method for Siren Detection*, 2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME), pp. 318-323, ISBN: 978-1-5386-1626-0, Constanta, Romania, 26-29 October, 2017, DOI: 10.1109/SIITME.2017.8259916, WOS:000428032300067. (ISI Web of Science, Scopus, IEEE Xplore)
- R. A. Dobre, R.-M. Udrea, C. Negrescu, D. Stanomir, *The Impact of the Acoustic Environment on Recovering Speech Signals Drowned in Loud Music*, The Sixteenth International Conference on Networks (ICN 2017), pp. 92-97, ISBN: 978-1-61208-546-3, Venice, Italy, 23-27 April, 2017. (publicat, în curs de indexare ISI)
- C. C. Oprea, R. O. Preda, I. Pirnog, and R. A. Dobre, *Efficient Transform Coefficient Coding in HEVC*, 3rd EAI International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures (FABULOUS 2017), Bucharest, Romania, 12-14 October, 2017. (publicat, în curs de indexare ISI)
- V. A. Cârstea, R. A. Dobre, C. C. Oprea, R. O. Preda, *A New Approach in Creating Decision Systems Used for Speaker Authentication*, 3rd EAI International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures (FABULOUS 2017), Bucharest, Romania, 12-14 October, 2017. (publicat, în curs de indexare ISI)

În concluzie, putem afirma că s-au îndeplinit cu succes toate obiectivele etapei pe 2017 a proiectului de cercetare intitulat „Sistem automat de autentificare activă a imaginilor digitale pentru PC și terminale mobile”.

Bibliografie

[1] A. T. S. Ho, Z. Xunzhan, S. Jun, și P. Marziliano, Fragile Watermarking Based on Encoding of the Zeroes of the Z-Transform, IEEE Transactions on Information Forensics and Security, 3, pp. 567-569, 2008.

[2] H. Kuo-Ming, C. Ting-Wen, S. Wen-Kai, and K. Chia-Nan, Automatic image authentication and recovery using multiple watermarks, 2012 8th International Conference on Information Science and Digital Content Technology (ICIDT), pp. 730-735, Jeju Island, South Korea, 2012.

[3] K. Wei-Chin, C. Te-Chih, W. Hsin-Lung, and C. Jen-Chun, A Fragile Watermarking Scheme for Image Authentication with Tamper Detection and Localization, Fourth International Conference on Genetic and Evolutionary Computing (ICGEC), pp. 638-641, Shenzhen, China, 2010.

[4] S. Bravo-Solorio, L. Gan, A. K. Nandi, and M. F. Aburdene, Secure private fragile watermarking scheme with improved tampering localisation accuracy, IET Information Security, 4, pp. 137-148, 2010.

[5] Liu, H., Yao, X., & Huang, J., Semi-fragile zernike moment-based image watermarking for authentication, Eurasip Journal on Advances in Signal Processing, 2010, ID 341856, 2010.

[6] H. M. Al-Otum, Semi-fragile watermarking for grayscale image authentication and tamper detection based on an adjusted expanded-bit multiscale quantization-based technique, Journal of Visual Communication and Image Representation, 25, 5, pp. 1064-1081, 2014.

[7] Y. Li; L. Du, Semi-fragile watermarking for image tamper localization and self-recovery, 2014 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), pp. 328-333, Wuhan, China, October 2014.
[8] S. Som, S. Palit, K. Dey, D. Sarkar, J. Sarkar, K. Sarkar, A DWT-based Digital Watermarking Scheme for Image Tamper Detection, Localization, and Restoration, Applied Computation and Security Systems, 2, Springer India, New Delhi, pp. 17-37, 2015.
[9] L. Rosales-Roldan, M. Cedillo-Hernandez, M. Nakano-Miyatake, H. Perez-Meana, B. Kurkoski, Watermarking-Based Image Authentication with Recovery Capability Using Halftoning Technique, Signal Processing: Image Communication, 28, 1, pp. 69-83, 2013.
[10] A. Phadikar, S.P. Maity, M. Mandal, Novel Wavelet-Based Qim Data Hiding Technique for Tamper Detection and Correction of Digital Images, Journal of Visual Communication and Image Representation, 23, 3, pp. 454-466, 2012.
[11] X. Qi, X. Xin, A quantization-based semi-fragile watermarking scheme for image content authentication, Journal of Visual Communication and Image Representation, 22, 2, pp. 187-200, 2011.
[12] K. Korotkov, J. Quintana, S. Puig, J. Malvehy, și R. Garcia, A New Total Body Scanning System for Automatic Change Detection in Multiple Pigmented Skin Lesions, IEEE Transactions on Medical Imaging, vol. 34, pp. 317-338, Dec. 2014.
[13] T. Tanaka, R. Yamada, M. Tanaka, et al, A Study on the Image Diagnosis of Melanoma, in Proc. 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vol. 1, pp. 1597-1600, Sept. 2004.
[14] M. Ramezani, A. Karimian, P. Moallem , Automatic Detection of Malignant Melanoma using Macroscopic Images, Journal of Medical Signals and Sensors, vol. 4, no. 4, pp. 281-290, 2014.
[15] S. A. Parah, J. A. Sheikh, U. I. Assad, and G. M. Bhat, Realisation and robustness evaluation of a blind spatial domain watermarking technique, International Journal of Electronics, vol. 104, no. 4, pp. 659-672, 2017.
[16] Q. T. Su, Y. G. Niu, Q. J. Wang, and G. R. Sheng, A blind color image watermarking based on DC component in the spatial domain, Optik, vol. 124, no. 23, pp. 6255-6260, 2013.
[17] A. Upadhyay and M. Dave, Robust and Imperceptible Color Image Watermarking for Telemedicine Applications, 2016 IEEE International Conference on Computing, Communication and Automation, pp. 1104-1109, 2016.
[18] P. Cavalcanti, J. Scharcanski, L. Di Persia, și D. Milone, An ICA-based method for the segmentation of pigmented skin lesions in macroscopic images, in Proc. IEEE Annu. Int. Conf. Eng. Med. Biol. Soc., pp. 5993-5996, 2011.

15.12.2017

Director proiect
conf. dr. ing. Radu Ovidiu PREDA