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. 2 / 2018

Perioada ianuarie – iunie 2018

În etapa pe 2018 a proiectului ne-am propus testarea individuală a funcționalității fiecărui modul al aplicației de autentificare și o evaluare globală a programului prin verificarea specificațiilor de proiect. De asemenea, aplicația a fost testată pe un set de utilizatori reprezentativi și au fost detectate diferite erori și modalități de optimizare. În urma acestor teste, au fost identificate și realizate modificările software necesare pentru creșterea performanțelor aplicației și a interoperabilității cu utilizatorii.

Îmbunătățirile legate de funcționalitatea aplicației includ, printre altele: eliminarea dependentelor de biblioteca Java Bridge, optimizarea și accelerarea procesului de inserare și extragere a marcajului de autentificare prin exploatarea tehnicilor de multithreading; integrarea aplicației cu rețele de socializare; creșterea securității algoritmului prin alegerea diferită a coeficienților DCT în care se inserează marcajul de autentificare; implementarea de funcții de morfologie matematică pentru eliminarea mai eficientă a falsurilor pozitive și negative; implementarea a două variante a meniului de "Settings" (Setări), pentru utilizatori de bază (Basic Settings) și utilizatori avansați (Advanced Settings).

De asemenea, a fost testată și interfața grafică a aplicației, care în versiunea de la sfârșitul anului 2017 avea doar un rol funcțional. În urma testelor cu un set de utilizatori reprezentativi a apărut necesitatea ca interfața să fie regândită și redezvoltată de la zero pentru o mai bună funcționalitate și interoperabilitate cu utilizatorul, și pentru a corespunde tendințelor actuale de design.

În cele ce urmează vom prezenta în primul rând cele mai importante modificări și îmbunătățiri de funcționalitate a aplicației de autentificare și ulterior modificările de design și interfață grafică.

Modificări și îmbunătățiri de funcționalitate a aplicației de autentificare

Eliminarea dependentelor de biblioteca Java Bridge

În etapa din 2017 s-a construit o variantă a aplicației care folosea biblioteca Java Bridge, deoarece aceasta conținea multe funcții utile de interacționat cu sistemul de operare și alte aplicații (capturare imagini, selectare imagine din galerie etc.). În teste s-au identificat probleme legate de această bibliotecă, cum ar fi faptul că funcția de lansat o aplicație de tip galerie pentru selectarea unei imagini se comporta ca și cum ar fi fost executată de două ori, utilizatorul fiind pus în situația de a alege încă o imagine după ce o selectase deja pe cea dorită.

În noua variantă s-au folosit numai funcții din SDK-ul Android pentru a realiza aceste sarcini, iar problemele au fost rezolvate. Biblioteca Java Bridge impune descrierea interfeței

aplicației în limbaj Java folosind funcții din aceasta, având la dispoziție numai elemente de bază (butoane, chenare, containere simple etc.). Prin eliminarea ei se poate dezvolta interfața utilizând uneltele grafice oferite de Android Studio, se poate programa folosind XML și se pot oferi elemente moderne de tip „Material design".

Reducerea timpului de procesare la codor și decodor prin exploatarea tehnicilor de multithreading

S-a constatat că inserarea marcajului de securitate și testarea autenticității unei imagini cu rezoluții de ordinul zecilor de megapixeli durează prea mult (zeci de secunde sau chiar peste un minut pe telefoane mai puțin performante) dacă se execută pe un singur fir sau „thread" de execuție (imaginea este prelucrată în întregime, progresiv). Obiectivul principal al noii versiuni a aplicației a fost acela de a prelucra zone de imagine în paralel, dacă dotările hardware ale dispozitivului pe care aceasta rulează permit acest lucru. Algoritmul de autentificare a fost modificat pentru a permite acest lucru. Inițial, secvența pseudoaleatoare cu rol de securizare era generată pentru toată imaginea. Dacă prelucrarea se face în paralel pe mai multe zone disjuncte ale imaginii, se poate proceda în două feluri: se generează o secvență pseudoaleatoare pentru toată imaginea care este împărțită la rândul ei într-un număr de subsecvențe egal cu numărul de zone în care a fost împărțită imaginea sau se generează o secvență pseudoaleatoare pentru fiecare zona din imagine (cu mențiunea că aceeași secvență poate fi folosită pentru toate zonele). A doua variantă este mai ușor de implementat și, deoarece numărul de sarcini care pot fi rulate eficient în paralel de un dispozitiv mobil nu este foarte mare, imaginea nu este împărțită în foarte multe zone, iar secvența pseudoaleatoare este suficient de lungă pentru fiecare zonă pentru a asigura securitatea (aplicația dezvoltată împarte imaginea în 8 zone).

Gestionarea thread-urilor de execuție în paralel și a resurselor dispozitivului este periculos de făcut de dezvoltator deoarece sistemul de operare poate avea de executat sarcini cu priorități mai mari decât aplicația de față și poate modifica resursele disponibile ducând la un posibil comportament instabil. S-a căutat o variantă de a lăsa telefonul să gestioneze resursele alocate pentru executarea sarcinilor în cadrul aplicației. Soluția a fost dată de obiectele de tip „AsyncTask". Fiecare obiect de acest tip primește un șir de alte obiecte de prelucrat și are 3 metode: o metodă rulată la lansarea în execuție, una rulată în timpul execuției și una rulată la finalizarea execuției. În cazul de față sunt create 8 obiecte de acest tip, fiecare primind ca element de intrare o zona din imaginea de prelucrat. Zonele sunt disjuncte. Acestea sunt lansate spre a fi rulate de către un „obiect executor" numit „THREAD_POOL_EXECUTOR". Resursele sunt gestionate automat de către acesta, asigurând funcționarea stabilă indiferent de câte sarcini poate rula respectivul terminal mobil în paralel. Un alt thread de execuție monitorizează progresul făcut de cele care prelucrează imaginea și atenționează utilizatorul când operația s-a încheiat. În timpul inserării sau testării, interfața aplicației răspunde la comenzi (mai puțin la lansarea în execuție a unei noi securizări sau testări de autenticitate. În acest caz utilizatorul este informat că deja se lucrează la o astfel de operație), ceea ce reprezintă o altă funcționalitate față de varianta dezvoltată în 2017 care se bloca pe tot parcursul operațiilor de inserare sau de testare a unei imagini, utilizatorul putând considera că funcționarea nu este corespunzătoare. Mai mult, chiar sistemul de operare putea considera că aplicația este blocată, întrebând utilizatorul dacă dorește să o închidă definitiv sau să aștepte. În noua versiune, utilizatorul nu este constrâns să mențină activă aplicația pe parcursul securizării sau testării, ci poate să o minimizeze. Singura condiție este să nu o închidă definitiv din managerul de aplicații al sistemului de operare.

Soluția curentă a fost implementată progresiv. În primă fază întreaga execuție a fost mutată într-un obiect de tip „AsyncTask” pentru a garanta funcționarea corectă. După determinarea variantei corecte, imaginea a fost împărțită în 8 zone și procesul anterior reluat. Probleme au existat mai ales din cauza diferențelor de convenții dintre Java și OpenCV. De exemplu, în metodele OpenCV adresarea unui vector se face complet inclusiv la dreapta (metoda ignorând ultimul element), iar în Java se procedează precum în limbajul C (adresarea unui vector se face de la elementul 0 până la elementul $L-1$, unde L este lungimea vectorului). A fost nevoie de folosirea „debugger”-ului oferit de Android Studio pentru a observa împărțirea inițial greșită a imaginii în zone.

O comparație a duratei de execuție a proceselor de inserare și detecția a marcajului de autentificare pentru diferite dispozitive mobile și pentru alegerea diferită a numărului de coeficienți DCT utilizați pentru inserare va fi prezentată mai jos, în paragraful de rezultate experimentale.

Creșterea securității algoritmului de autentificare

Versiunea aplicației dezvoltată în etapa pe 2017 folosea un număr fix de 3 coeficienți ai Transformatei Cosinus Discrete 2D pentru a insera marcajul de autentificare în imaginea originală. Pentru a crește flexibilitatea și a securitatea algoritmului de autentificare, s-a introdus posibilitatea alegerii numărului de coeficienți DCT ce vor fi utilizați pentru inserarea marcajului de autentificare (2, 3, 4 sau 5 coeficienți). Funcțiile de inserare și testare au fost modificate pentru a permite această flexibilitate. Creșterea numărului de coeficienți are avantajul reducerii ratei de detecție de fals negativ (reducerea numărului de pixeli falsificați care sunt determinați de sistem ca fiind autentici), dar poate produce unele blocuri izolate detectate greșit ca fiind falsificate.

O altă îmbunătățire a securității algoritmului față de versiunea pe 2017 este posibilitatea utilizării unei parole alfanumerice introdusă de utilizator în meniul de setări pentru generarea marcajului de autentificare și alegerea poziției coeficienților DCT utilizați pentru inserare, după scanarea în zigzag. Parola alfanumerică este transformată într-un număr și, împreună cu numărul de coeficienți și factorul de calitate, este utilizată pe post de cheie pentru generarea secvenței pseudoaleatoare binare (watermark-ul de autentificare) și alegerea poziției coeficienților DCT utilizați pentru inserare.

Utilizarea morfologiei matematice pentru eliminarea falsurilor pozitive și negative

Morfologia matematică se ocupă cu analiza formelor reprezentate ca mulțimi de puncte, în cazul nostru pixeli. În domeniul prelucrării digitale a imaginilor, morfologia matematică poate opera asupra imaginilor binare sau a celor cu nuanțe de gri. Formele analizate și prelucrate sunt forme ce rezultă ca urmare a unei operații de binarizare sau de segmentare a unei imagini, iar în urma aplicării de procedee de morfologie matematică va fi modificată forma acestor forme/obiecte.

Pentru aplicația de autentificare dezvoltată vom lucra cu operații de morfologie matematică asupra matricei de autentificare binare A obținută după aplicarea procedeului de extragere a marcajului de autentificare dintr-o imagine de test. Imaginea binară A are dimensiunea egală cu numărul de blocuri de 8x8 pixeli din imaginea testată, un „0” logic însemnând că blocul respectiv este autentic, iar un „1” logic însemnând că blocul respectiv este posibil falsificat. Decizia finală, dacă un bloc este autentic sau a fost falsificat, se ia în urma aplicării operațiilor de morfologie matematică, deoarece este posibil ca din cauza compresiei

JPEG sau din cauza unor depășiri sau prelucrări de imagini, ce pot apărea, blocuri autentice să fie detectate de algoritm ca fiind false sau blocuri falsificate să fie detectate ca fiind autentice. Aceste blocuri sau grupuri mici de blocuri detectate greșit sunt de regulă izolate față de blocurile din categoria opusă.

Pentru a elimina falsurile negative (blocuri falsificate care sunt determinate de sistem ca fiind autentice) care apar de multe ori în interiorul unor zone mai mari detectate ca fiind falsificate, este utilă operația de morfologie matematică de „fill" (umplere). Deoarece nu există o metodă predefinită pentru această operație în Java sau OpenCV, echipa proiectului a implementat această metodă pornind de la algoritmul „floodfill". Astfel am reușit să reducem rata de detecție de fals negativ. Un exemplu de aplicare a operației de umplere morfologică pe o matrice de autentificare binară este dată în Fig. 1.

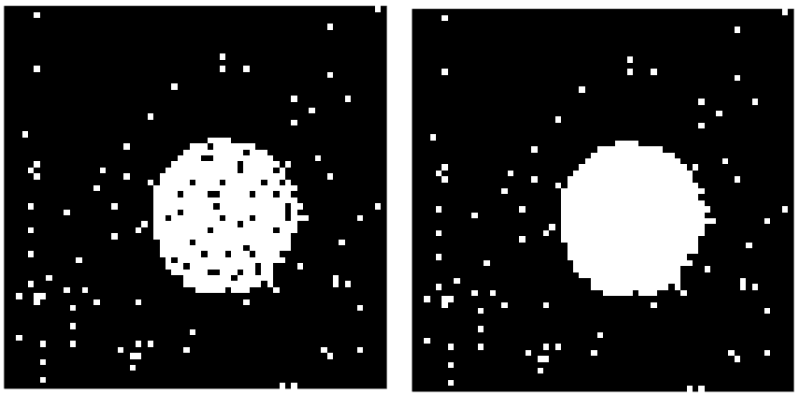

Fig. 1. Aplicarea operației de umplere morfologică pe o matrice de autentificare

Pentru a elimina falsurile pozitive (blocuri autentice clasificate de sistem ca fiind false), care apar de multe ori ca blocuri false izolate, se poate utiliza operația de morfologie matematică de „open" (deschidere) care a fost implementată în aplicația de autentificare ca o operație de eroziune morfologică, urmată de o dilatare morfologică cu același element structural. Folosind acest procedeu, am reușit să reducem semnificativ rata de detecție de fals pozitiv. Un exemplu de aplicare a operației de deschidere morfologică pe o matrice de autentificare este dată în Fig. 2.

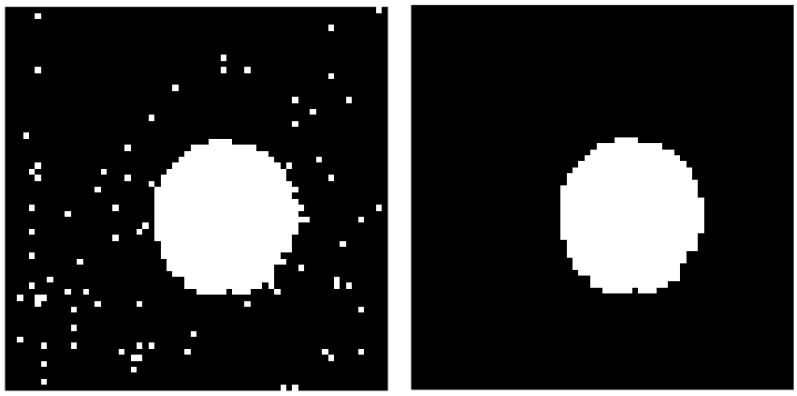

Fig. 2. Aplicarea operației de deschidere morfologică pe o matrice de autentificare

Relansarea și îmbunătățirea interfeței grafice

În ceea ce privește modificările de design și interfață grafică, s-a realizat un nou ecran de start si un nou ecran de setări pentru o funcționalitate și interoperabilitate mai bună cu utilizatorul.

Noul ecran de start dezvoltat de la zero este prezentat în Fig. 3, iar modificările și îmbunătățirile aduse sunt următoarele:

Interfață autoscalabilă

Interfața grafică a aplicației se scalează automat cu aceleași proporții ale elementelor pe orice dimensiune a ecranului (au fost ecranele fizice și cele virtuale disponibile în emulatorul de terminale mobile din SDK-ul Android, cu diagonale ale ecranului între 3 și 10 inch).

Realizarea unui logo unic al aplicației „ImFakeCheck”

Noul logo al aplicației este prezentat în Fig. 4. Acesta a fost obținut prin prelucrarea într-un program de editare foto a textului ImFakeCheck scris cu font-ul „Bullet Rain”. Imaginea folosită în completarea acesteia este disponibilă gratuit pe site-ul www.pixabay.com, cu denumirea „sign-check-tick-yes-cross-accept-145561”.

Fig. 3. Noul ecran de start al aplicației mobile de autentificare a imaginilor

Fig. 4. Noul logo al aplicației mobile de autentificare

Realizarea unui cadru de afișare a imaginilor

S-a determinat experimental că este nevoie ca imaginile să fie scalate astfel încât latura mare să aibă maxim 1000 pixeli pentru ca ele să fie afișate fără probleme pe o varietate de telefoane cu dimensiuni diferite ale ecranelor. În acest cadru se afișează replici scalate corespunzător ale imaginilor, imaginile de rezoluții mari fiind salvate în memoria telefonului (replica scalată a celei mai recente imagini de afișat în cadru este menținută în memoria RAM și nu este disponibilă ca fișier). Fotografiile pe lat sunt rotite automat pentru a utiliza întreaga zonă de afișare.

La pornirea aplicației, cadrul de afișare are și rol de informare a utilizatorului despre operațiunile pe care acesta le poate face inițial: să captureze sau să încarce o imagine. Se evită astfel necesitatea consultării instrucțiunilor de folosire a aplicației, crescând posibilitatea învățării folosirii ei în mod intuitiv. După fiecare operațiune în urma căreia rezultă o imagine (capturare, încărcare, inserare watermark, testare autenticitate imagine) acest cadru este reîmprospătat cu imaginea rezultată (eventual în miniatură). Cadrul are raportul de aspect 9:16 (sau 16:9 cu orientare „portret”), iar toate imaginile sunt prelucrate astfel încât încadrarea să se facă fără pierderi (fără a decupa imaginea). Dacă în cadru este afișată o imagine, prin apăsarea pe aceasta se va lansa o aplicație de tip galerie (dacă utilizatorul are mai multe astfel de aplicații instalate și niciuna nu este implicit selectată, el va putea selecta aplicația de galerie preferată dintr-o listă a celor instalate), care va deschide imaginea la rezoluție nativă (înaintea scalării pentru afișare în cadru). Toate operațiunile care pot fi făcute în aplicația respectivă de galerie se pot face si asupra acestei imagini (detaliere, rotire etc.). Dacă utilizatorul apasă pe cadru cât timp acesta afișează încă mesajul sugestiv de început, aplicația îi va comunica faptul că încă nu a fost încărcată nici o imagine.

S-a constatat un comportament diferit al aplicațiilor de tip galerie („Google Photos” și „Gallery”, prima oferită gratuit de Google, iar a doua fiind disponibilă implicit în sistemul de operare Android) la deschiderea imaginilor. „Google Photos” poate constata că imaginea trimisă spre ea este în format lat și o afișează corespunzător, pe când „Gallery” distorsiona imaginea, forțând-o într-un format de tip „portret” și ocupând tot ecranul telefonului. Acesta este un motiv în plus pentru care imaginea este rotită la nevoie astfel încât să corespundă orientării cadrului.

Utilizarea pe ecranul de start de icoane mai sugestive pentru realizarea / încărcarea, securizarea și testarea unei fotografii

Aplicația conține 3 butoane grafice principale, fiecare sugerând categoria de operații care se pot accesa prin apăsarea lor (vezi Fig. 3).

Primul buton (amplasat în partea superioară și având ca pictogramă două aparate foto unul în spatele celuilalt) permite selectarea sursei fotografiei (captură directă sau încărcare din memorie). În acest sens se vor deschide animat alte două butoane, inițial invizibile: „Take photo” și „Load photo”. Butonul „Take photo” lansează aplicația preferată de captură a imaginilor și

așteaptă declanșarea. În urma operațiilor de captură și validare, aplicația va afișa în cadrul special destinat o replică în miniatură și, eventual, rotită corespunzător a imaginii capturate.

Butonul „Load photo” lansează aplicația preferată de galerie și permite selectarea unei imagini pentru prelucrarea ulterioară. După selectare, o replică a sa va fi afișată în cadrul special destinat. În acest moment se pot efectua operații de securizare sau testare a autenticității.

Al doilea buton (amplasat în centru și având ca pictogramă un lacăt închis) permite selectarea operațiunii de efectuat asupra imaginii selectate (previzualizată în cadru) cu ajutorul a două butoane similare cu cele invizibile descrise mai sus: „Secure Photo” (Securizează imaginea) și „Test Photo” (Testează imaginea). Butonul „Secure photo” lansează operația de inserare a marcajului în imaginea selectată. Începutul și sfârșitul acestui proces sunt marcate prin mesaje de informare a utilizatorului, ce vor fi prezentate mai jos. Celălalt buton, „Test Photo”, pornește procesul de testare a autenticității imaginii selectate, semnalizat în mod similar.

Al treilea buton (amplasat în partea de jos și având ca pictogramă un aparat foto având în fața sa o lupă) permite deschiderea directoarelor care conțin imaginile securizate, respectiv testate cu ajutorul a două butoane inițial invizibile: „Browse Secured” (Deschide securizate) și „Browse Tested” (Deschide testate). Pentru a putea folosi aceste funcționalități, utilizatorul trebuie să aibă un manager de fișiere preinstalat pe sistemul Android, ca de exemplu „ES File Manager” sau „Total Commander”.

Utilizarea unui element grafic de tip „bottom sheet” pentru deschiderea printr-o tranziție de tip „slide” a paginii de „Settings” (Setări)

Pagina de setări este puțin vizibilă în partea de jos a interfeței aplicației și poate fi extinsă pe întregul ecran prin glisare în sus, afișând astfel mecanismele prin care se pot regla parametrii de funcționare a algoritmului folosit. S-a evitat astfel crearea unei noi activități Android pentru accesarea setărilor, simplificând ciclul de funcționare al aplicației și gestionarea datelor (nu este necesară transmiterea datelor între activități). Prima problemă descoperită în testare a fost faptul că partea vizibilă a paginii de setări putea fi închisă permanent printr-o glisare în jos pe aceasta. Acest lucru a fost rezolvat prin eliminarea proprietății „is hideable”. Un alt aspect al paginii de setări găsit mai târziu este faptul că aceasta era „transparentă” pentru click-uri și alte interacțiuni, dacă ele nu aveau ca destinație strict elemente din pagina de setări (implicit, un click pe o porțiune liberă din pagina de setări poate declanșa acțiunea elementului plasat sub aceasta). Elementul care a ajutat la găsirea acestei probleme a fost cadrul în care se afișează replicile scalate ale imaginilor deoarece el ocupă o suprafață mare a ecranului și este sensibil la click-uri (deschide în galerie imaginea originală a cărei replică este afișată in el).

Meniul de setări al aplicației a fost, de asemenea, refăcut și modernizat (vezi Fig. 5). Modificările și îmbunătățirile aduse acestuia sunt prezentate în cele ce urmează:

- *Opțiunea de a comuta de la varianta simplă („Easy”) la varianta avansată („Advanced”) a meniului*. Trecerea se face prin schimbarea poziției comutatorului dedicat, clar marcat cu cele două moduri. **Varianta simplă** a fost realizată pentru utilizatori neexperimentați, iar varianta avansată pentru utilizatori cu unele cunoștințe în domeniul prelucrării digitale a imaginilor. În această variantă, aplicația generează imagini securizate care vor putea fi testate cu succes dacă factorul de calitate folosit la recomprimarea lor JPEG este mai mare sau egal cu 40 (aplicația nu va detecta ca modificare neautorizată compresia JPEG cu condiția ca factorul de calitate folosit să fie mai mare sau egal cu această valoare). Utilizatorul poate modifica numai factorul de calitate folosit la comprimarea imaginii securizate. Cum acesta trebuie să fie mai mare sau egal cu factorul de calitate limită pentru care aplicația nu va detecta compresia JPEG

drept modificare neautorizată, valorile din care poate alege sunt în intervalul [40 , 100]. Selecția valorii se face modificând bara „Recompression Q”, valoarea exactă fiind afișată în dreapta acesteia. Aplicația va folosi aceeași cheie pentru generarea marcajului binar de autentificare pentru toate imaginile. Lungimea secvenței va varia cu rezoluția imaginii de securizat. În varianta simplă, în fiecare bloc se introduc 2 biți ai secvenței pseudoaleatoare (se folosesc 2 coeficienți DCT). În acest mod, utilizatorul poate testa numai imagini care au fost securizate strict cu acest mod de lucru.

Varianta avansată îi permite utilizatorului să modifice mai mulți parametri care intervin în securizarea sau testarea imaginii. Acesta poate selecta factorul de calitate la care imaginea este protejată după securizare („Quality factor”), factorul de calitate folosit la compresia imaginii securizate („Recompression Q”), numărul de coeficienți DCT din fiecare bloc în care se va insera marcajul de autentificare („Number of coefficients”), selectabil între 2 și 5 coeficienți, și cheie secretă a algoritmului prin setarea unei parole alfanumerice („Password”). Modificarea acestor parametri în acest mod de lucru este utilă doar în cazul securizării, nu și al testării autenticității. După inserare, aplicația va genera o nouă parola care va conține și parametrii folosiți în operația de securizare, pe care o va afișa în locul parolei introduse de utilizator. Aceasta poate fi copiată foarte ușor prin apăsare prelungită pe aceasta și prin selectarea opțiunii „copy” din meniul ce se deschide. Apoi se poate transmite folosind orice mod de comunicare cu mesaje text (SMS, Whatsapp, e-mail, etc.). În cazul testării, utilizatorul trebuie să introducă doar parola care a fost generată de aplicație la sfârșitul operației de securizare a imaginii respective. Aceasta va conține toți parametrii utilizați, aplicația extrăgându-i automat și folosind acele valori în procesul de testare a autenticității. La folosirea unei parole greșite, imaginea va fi clasificată drept neautentică, indiferent dacă a suferit modificări neautorizate sau nu.

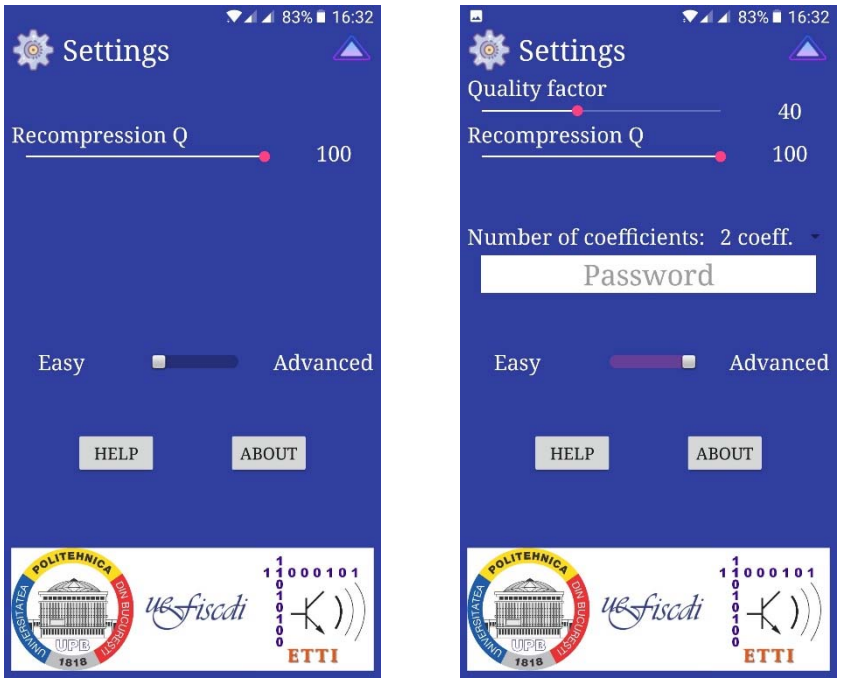

Fig. 5. Varianta „Basic” (simplă) și varianta „Advanced” (avansată) a ecranului de setări a aplicației

- *Setarea factorului de calitate q_1 la care este protejată imaginea după inserare*. Modificarea acestui parametru se realizează printr-un slider, cu valoarea maximă egală cu 100 și valoarea minimă limitată la 40. Valoarea q_1=40 a fost aleasă, astfel încât modificarea adusă imaginii originale să nu fie vizibilă pentru un ochi uman mediu. Prin setarea unei anumite valori a acestui factor de calitate, marcajul inserat în imagine este robust la recompresia JPEG a imaginii cu orice factor de calitate $q_2 \geq q_1$.

- *Setarea factorului de calitate utilizat pentru recompresia imaginii* după inserarea marcajului. Acest parametru permite compresia JPEG a imaginii securizate folosind orice factor de calitate $q_2 \geq q_1$. Astfel, valoarea lui q2 este limitată inferior la valoarea lui q1 setată de către utilizator. Această opțiune este disponibilă atât în varianta de bază, cât și în varianta avansată a ecranului de Setări.

- *Setarea numărului de coeficienți DCT utilizați la inserare și testare*. Acest parametru al algoritmului, disponibil doar in varianta avansată a meniului de Setări, este disponibil printr-un casetă de tip „dropdown". Se pot alege între 2 și 5 coeficienți.

- *Setarea unei parole alfanumerice pentru creșterea securității algoritmului*. Parola este utilizată ca o cheie secretă pentru generarea secvenței watermark-ului de autentificare și alegerea poziției coeficienților DCT utilizați pentru inserare.

- *Un buton de „Help" (Ajutor)* ce conține instrucțiuni de utilizare a aplicației, *și un buton de „About" (Despre noi)* cu informații despre echipa de cercetare și sursa de finanțare a proiectului.

- Spațiul liber rămas a fost folosit pentru afișarea *siglelor instituțiilor participante la acest proiect* (Universitatea Politehnica din București, Facultatea de Electronică, Telecomunicații și Tehnologia Informației și Unitatea Executivă pentru Finanțarea Învățământului Superior, a Cercetării, Dezvoltării și Inovării).

Tot în categoria îmbunătățirii interfeței cu utilizatorul se încadrează și implementarea unor **mesaje de informare a utilizatorului** în urma procesului de inserare a marcajului de autentificare, respectiv de testare a autenticității unei imagini. Acestea vor fi prezentate în continuare.

- Butonul „Secure photo" lansează operația de inserare a marcajului în imaginea selectată. Începutul acestui proces este marcat de un mesaj de informare a utilizatorului afișat în partea de jos a ecranului (un obiect de tip snackbar) „Working in background" (Lucrează în fundal), care rămâne afișat până când procesul de inserare s-a încheiat. În urma inserării marcajului de autentificare aplicația afișează mesajul de tip snackbar „Image secured!" (Imagine securizată!) urmat de valoarea PSNR-ului dintre imaginea marcată și cea originală, care ne dă calitatea imaginii marcate. Acest ultim mesaje poate fi înlăturat de către utilizator prin apăsarea pe textul „DISMISS" (Elimină) sau prin glisare către dreapta a mesajului. După încheierea procesului de inserare, cadrul de afișaj este reîmprospătat cu imaginea care conține marcajul de securitate.

- Butonul „Test Photo" pornește procesul de testare a autenticității imaginii selectate, semnalizat în mod similar prin mesajul „Working in background". În urma aplicării funcției de detecție aplicația afișează mesajul „Testing complete! Image is authentic!" (Testare completă! Imaginea este autentică!) în cazul în care imaginea testată este autentică și „Testing complete! Image is NOT authentic! Check red blocks!" (Testare completă! Imaginea NU este autentică! Verificați blocurile roșii!) în cazul în care imaginea a fost falsificată.

- Mesajele de informare nu conțineau inițial textul „DISMISS", iar plasarea lor se poate face numai în partea de jos a ecranului (specifică „Snackbar"), în cazul de față chiar peste bara care deschide pagina de setări. Prin testare s-a constatat că utilizatorii nu își dădeau ușor seama

cum ar putea închide mesajele. În varianta finală, utilizatorii pot apăsa pe textul „DISMISS” pentru a le închide. Varianta cu închidere automată după un timp fixat a acestora a fost luată în considerare, dar din cauză că operațiile de securizare și testare a autenticității imaginilor pot dura un timp îndelungat pe telefoane mai puțin performante și pentru că ele se desfășoară „în fundal”, utilizatorii pot folosi alte aplicații în acest timp și pot rata mesajele trimise de aplicație.

Rezultate experimentale obținute de aplicația de autentificare

Pentru a testa calitatea imaginilor securizate față de cele originale am calculat valorile PSNR medii pentru imagini de diferite rezoluții și numere diferite de coeficienți DCT utilizați pentru inserarea marcajului. Imagine securizate au fost protejate la compresie JPEG cu factori de calitate mai mari decât q_{min}=40. Rezultatele obținute sunt date în Fig. 6. Calitatea imaginilor marcate scade odată cu modificarea unui număr mai mare de coeficienți DCT, dar se observă că rămâne la valori destul de bune (42 dB) chiar și în cazul modificării unui număr de 5 coeficienți DCT, asigurând imperceptibilitatea marcajului de autentificare în imaginile securizate. De asemenea, se poate observa că influența rezoluției imaginii asupra PSNR-ului este minora, de maxim 0.32 dB.

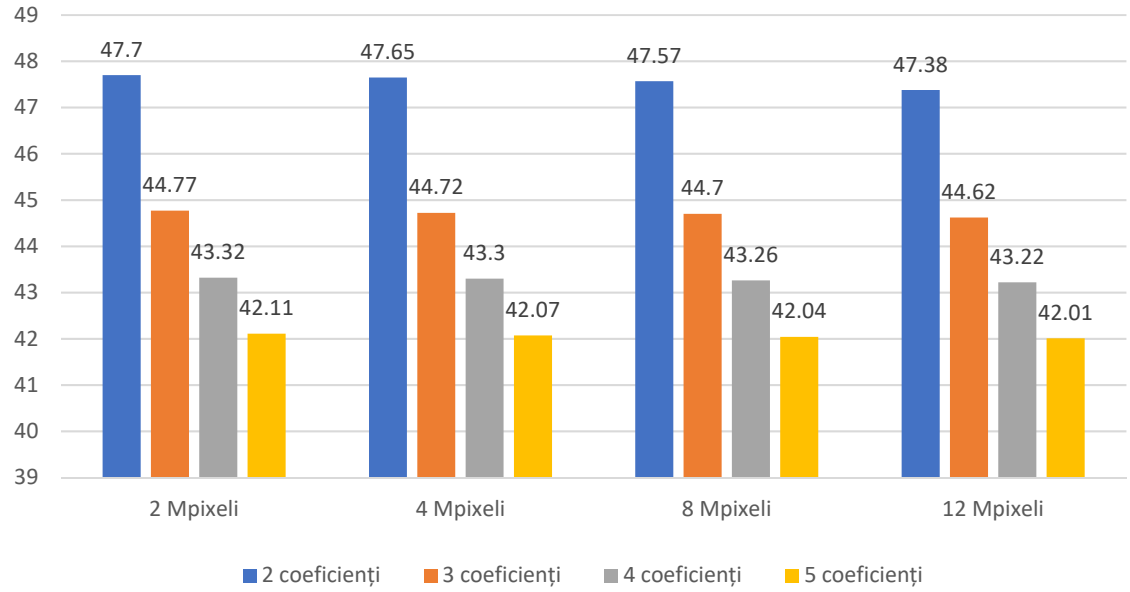

Fig. 6. Valori PSNR medii pentru imagini de diferite rezoluții și numere diferite de coeficienți DCT utilizați pentru inserarea marcajului

În urma modificărilor realizate prin exploatarea tehnicilor de tip multithreading, am măsurat timpii de execuție a procesului de inserare a marcajului și de detecție a falsificării pentru imagini de diferite rezoluții pe dispozitive mobile cu specificații și versiuni diferite ale sistemului de operare Android (vezi Tabelele 1 și 2). Pentru teste au fost utilizate dispozitive mobile prezentate în Tabelul 3. În Figurile 7 si 8 sunt date comparații grafice ale duratei de execuție a aplicației mobile pentru procesul de inserare, respectiv de detecție.

În comparație cu varianta aplicației fără procedee de multithreading, am constatat o reducere semnificativă a duratei de execuție, atât la inserare, cât și la detecție. Astfel, pentru dispozitive mobile cu procesor Quad-core (4 nuclee) timpii de execuție au scăzut în medie cu 55%, iar pentru dispozitive cu procesor Octa-core (8 nuclee) cu 64%.

Tabelul 1. Durata procesului de inserare a marcajului în secunde pentru imagini de diferite rezoluții, pe diferite dispozitive mobile (între 2 și 5 coeficienți DCT utilizați)

Rezoluție imagine	1920 x 1080 (2 Mpixeli)				2560 x 1600 (4 Mpixeli)				3840 x 2160 (8 Mpixeli)				4640 x 2610 (12 Mpixeli)			
Model Smartphone	2C	3C	4C	5C	2C	3C	4C	5C	2C	3C	4C	5C	2C	3C	4C	5C
Samsung Galaxy S6	1.68	1.49	1.63	1,74	2.04	2.37	2.15	2,32	3.41	3.51	3.77	3,97	5.34	5.33	5.65	6,21
Oneplus 3T	1.40	1.77	1.88	1.98	2.28	2.63	2.75	3.12	4.45	4.65	6.31	6.67	6.21	6.22	7.27	8.01
Oneplus 3	1.41	1.79	1.84	1.99	2.33	2.67	2.87	3.16	4.50	4.72	6.29	6.64	6.59	6.48	7.33	8.13
Motorola Moto X 2nd generation	2.97	2.96	3.22	3.46	5.54	5.62	6.31	6.62	10.96	11.84	11.65	12.53	16.68	17.30	17.44	18.09
Alcatel Pixi 4	2.97	3.19	3.16	3.48	5.11	5.38	5.96	6.27	10.10	10.08	11.29	12.72	16.07	15.52	16.41	17.47
Lenovo Tab 2 A10-30	3.34	3.38	4.08	4.45	5.83	6.12	6.96	8.62	12.51	11.95	13.23	13.28	16.91	20.56	19.18	20.39

Tabelul 2. Durata procesului de detecție a falsificării în secunde pentru imagini de diferite rezoluții, pe diferite dispozitive mobile (între 2 și 5 coeficienți DCT utilizați)

Rezoluție imagine	1920 x 1080 (2 Mpixeli)				2560 x 1600 (4 Mpixeli)				3840 x 2160 (8 Mpixeli)				4640 x 2610 (12 Mpixeli)			
Model Smartphone	2C	3C	4C	5C	2C	3C	4C	5C	2C	3C	4C	5C	2C	3C	4C	5C
Samsung Galaxy S6	1.47	1.24	1.21	1.23	1.6	1.82	1.67	1.77	2.34	2.68	2.62	2.93	4.09	3.83	4.29	5.04
Oneplus 3T	1.02	1.10	1.14	1.20	1.47	1.77	1.84	1.80	2.51	2.82	3.20	3.46	3.62	3.75	3.98	4.77
Oneplus 3	1.05	1.09	1.16	1.26	1.49	1.78	1.90	1.83	2.57	2.95	3.22	3.40	3.80	3.80	4.07	4.84
Motorola Moto X 2nd generation	1.85	2.05	2.34	2.35	3.36	3.68	3.72	3.90	6.69	6.93	7.19	7.60	11.10	10.76	11.04	11.30
Alcatel Pixi 4	2.06	2.01	2.26	2.33	3.97	4.07	3.91	4.20	6.76	6.49	7.49	7.76	10.56	10.60	11.28	11.52
Lenovo Tab 2 A10-30	2.19	2.59	3.26	3.08	4.18	4.47	4.74	4.91	7.32	7.34	8.34	9.30	12.98	12.98	14.19	14.54

Tabelul 3. Dispozitive mobile utilizate pentru testarea aplicației de autentificare

Model	**Procesor**	**Memorie RAM [GB]**	**Versiune Android**
Samsung Galaxy S6	Octa-core (4 x 2.1 GHz Cortex-A57 + 4 x 1.5 GHz Cortex-A53)	3	7.0
Oneplus 3T	Quad-core (2 x 2.35 GHz Kryo + 2 x 1.6 GHz Kryo)	6	8.0.0
Oneplus 3	Quad-core (2 x 2.15 GHz Kryo + 2 x 1.6 GHz Kryo)	6	8.0.0
Motorola Moto X 2nd generation	Quad-core (4 x 2.5 GHz Krait 400)	2	6.0
Alcatel Pixi 4	Quad-core (4 x 1.3 GHz)	1	6.01
Tabletă Lenovo Tab 2 A10-30	Quad-core (4 x 1.7 GHz Cortex-A53)	2	6.0

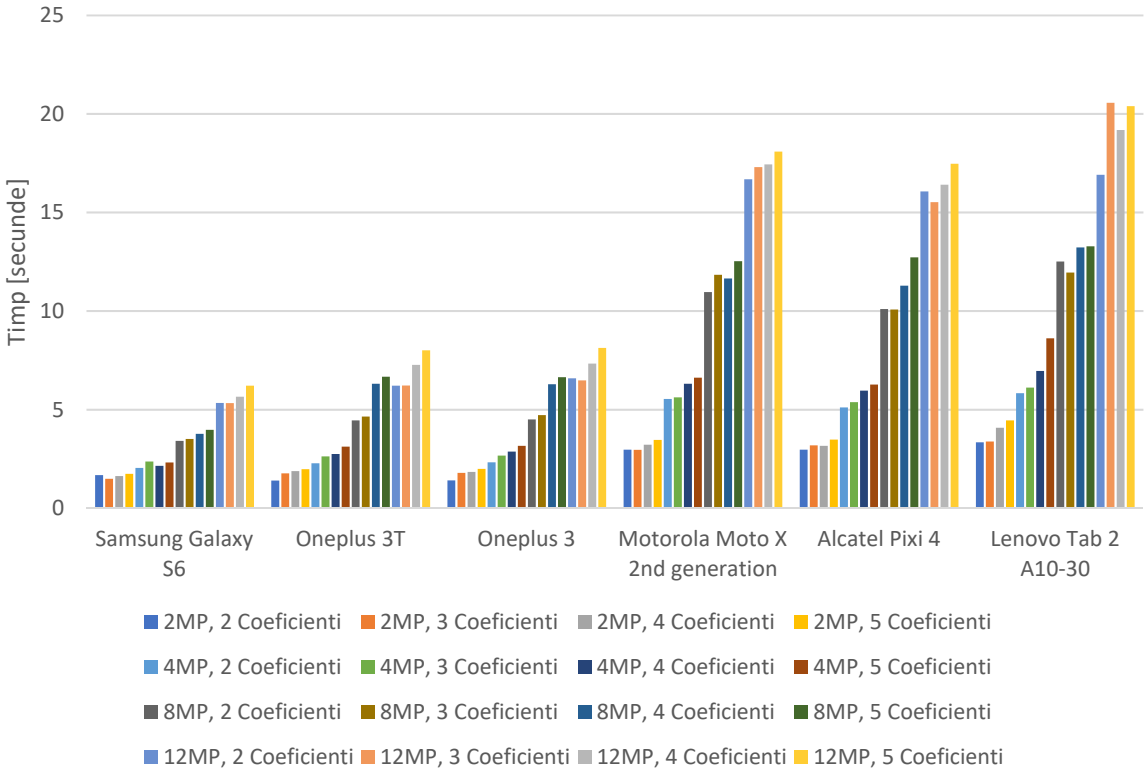

Fig. 7. Comparație a timpilor de execuție a procesului de inserare pe diferite dispozitive mobile pentru imagini de diferite rezoluții și număr diferit de coeficienți DCT utilizați

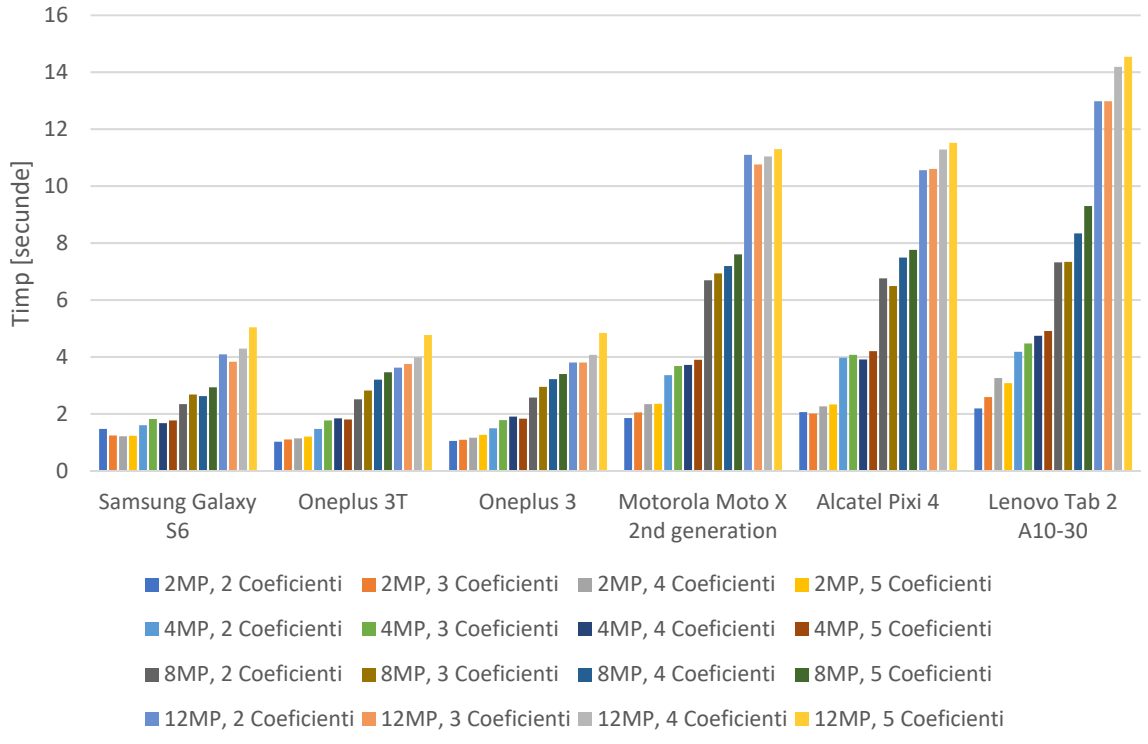

Fig. 8. Comparație a timpilor de execuție a procesului de detecție pe diferite dispozitive mobile pentru imagini de diferite rezoluții și număr diferit de coeficienți DCT utilizați

Pentru a testa robustețea la compresie JPEG pe o bază de date de 100 de imagini am utilizat mediul de dezvoltare Matlab, deoarece procesul poate fi automatizat mai ușor decât cu aplicația pentru dispozitive mobile. Imaginile utilizate au fost de rezoluție 512x512 pixeli. Fiecare imagine din baza de date a fost marcată folosind q_{min}=50, iar apoi comprimata cu factorii de calitate $q \in \{50,60,70,80,90,100\}$. Pentru toate imaginile 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 securizată cu $q_{\min} \in \{50,60,70\}$ și apoi 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, corespunzând unor proporții de falsificare de la 0,01% la 48%. Fiecare imagine falsificată a fost comprimată cu factorii de calitate $q \in \{50,60,70,80,90,100\}$. În Fig. 9 sunt date valorile DFN medii în procente pentru diferite valori ale q_{min}. Aceste valori sunt foarte apropiate de zero. Se observă că valori mai mari ale DFN se obțin doar pentru dimensiuni foarte mici ale zonelor falsificate și apar de regulă la marginile zonelor falsificate.

De asemenea, am testat robustețea schemei dezvoltate la alte operații uzuale de prelucrări de imagini. Pentru alegerea parametrilor fiecărui tip de distorsiune incidentală a fost impusă o limită de aproximativ 35 dB pentru PSNR-ul imaginii distorsionate. Dacă o operație de prelucrare de imagini modifică PSNR-ul imaginii sub 35 dB, acesta poate fi considerată malițioasă. Parametrii rezultați pentru fiecare distorsiune în parte și rata de detecție de fals pozitiv (DFP) medie pentru o bază de date de 100 de imagini testate sunt date în Tabelul 4. Se observă ca metoda dezvoltată obține valori foarte bune pentru DFP.

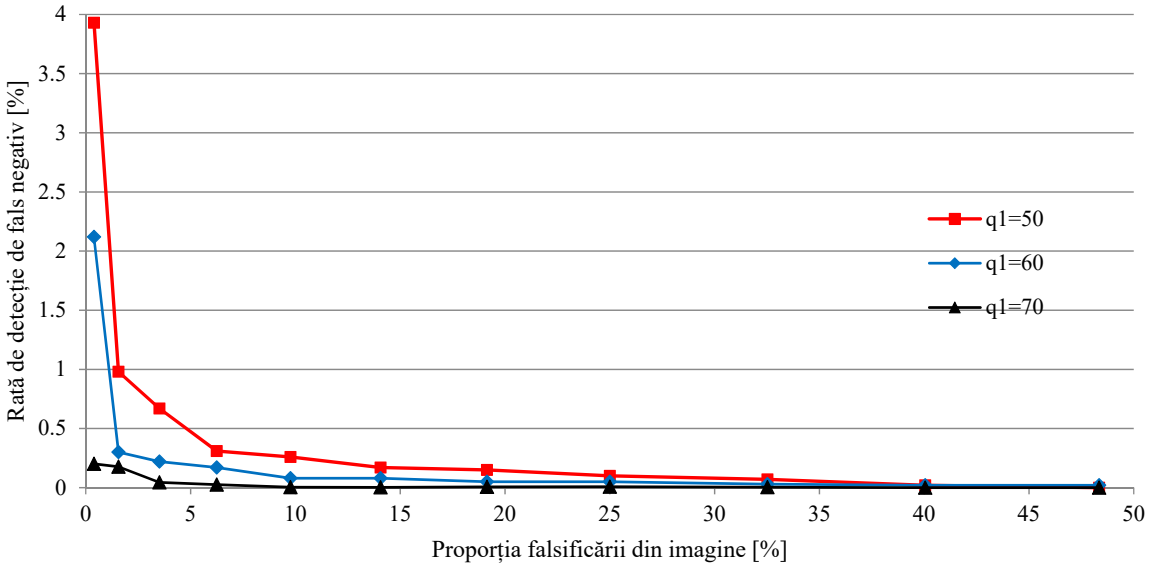

Fig. 9. Rata de detecție de fals negativ (DFN) după autentificarea de imagini falsificate și comprimate cu diferiți factori de calitate JPEG

Tabelul 4. Robustețea marcajului de autentificare la diferite operații uzuale de prelucrare de imagini

Distorsiune incidentală	**Zgomot gaussian**	**Zgomot salt&pepper**	**Ajustare de luminanță**	**Filtrare mediană**	**Blurare**
Parametri	varianță σ^2=0.0003	densitate d=0.1%	±4%	fereastră de 3×3	rază R=1
PSNR (dB)	35.23	35.20	35.04	34.84	34.61
DFP (%)	2.86	0	0	4.88	3.73

În Tabelul 5 schema dezvoltată în cadrul proiectului a fost comparată cu alte abordări relevante din domeniu. Al doilea rând al tabelului conține PSNR-ul mediu al imaginilor marcate pentru diferite metode prezentate în literatura de specialitate. Tehnica noastră obține cea mai bună calitate a imaginilor, pe când unele dintre metode obțin valori situate chiar și sub limita de vizibilitate. Al treilea rând al tabelului ne dă factorul de calitate JPEG minim la care este protejat marcajul de autentificare. Următoarele rânduri arată care dintre algoritmi sunt robuști la distorsiunile incidentale date în Tabelul 7, unde √ semnifică robustețe la distorsiunea respectivă, iar × indică fragilitate. Ultimul rând din Tabelul 8 compară rata de detecție de fals negativ (DFN) după falsificarea de conținut cu diferite proporții și compresie JPEG. Aplicația noastră obține valoarea DFN medie cea mai bună în prezența compresiei JPEG și este robustă la adăugarea de zgomot gaussian, de tip salt&pepper, ajustare de luminanță, filtrare mediană și blurare.

Tabelul 5. Comparație a tehnicii dezvoltate cu alte metode actuale de autentificare

Metodă	**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**[6]**	**Propusă**
PSNR (dB)	42.8	42	41.37	35	35.27	41	44.14
Factor de calitate q_{min}	60	85	×	65	70	50	50
Zgomot gaussian	√	√	×	√	×	√	√
Zgomot salt&pepper	×	×	×	√	√	√	√
Ajustare de luminanță	×	√	×	√	√	×	√
Filtrare mediană	×	×	×	√	√	√	√
Blurare	×	×	×	×	√	√	√
DFP (%)	2.1	5	-	0.5	-	13.8	0.26

Pentru exemplificarea funcționări aplicației, În Fig. 10 sunt prezentate două exemple de autentificare de imagini falsificate. S-au utilizat două imagini de rezoluție 8, respectiv 12 megapixeli, realizate cu telefonul mobil Oneplus 3T. Cu ajutorul aplicației de autentificare mobilă, în fiecare imagine a fost inserat marcajul de autentificare folosind 2 coeficienți DCT și un factor de calitate $q_{\min} = 40$, care asigură robustețea watermark-ului de autentificare la compresie JPEG cu orice factor de calitate $q \geq q_{\min}$. Imaginile securizate sunt prezentate în Fig. 10 (a). După obținerea imaginilor securizate, acestea au fost transferate pe un computer și au fost modificate cu o aplicație de editare de imagini (Gimp2) utilizând diferite modalități de falsificare: ștergerea/ascunderea de obiecte din imagine, duplicarea de zone din imagine (falsificare „copy-paste”), inserarea de obiecte noi în imagine, provenite din alte imagini. După falsificare imaginile au fost recomprimate cu un factor de calitate de 50.

În prima imagine capetele celor două persoane au fost înlocuite între ele și în stânga, pe iarbă, a fost adăugat un câine dintr-o altă imagine. În a doua imagine capul statuii a fost acoperit cu frunze din fundal, litera „K” din cuvântul „KAFKA” scris pe piedestalul statuii a fost înlocuită cu textura vecină și amuleta purtată de persoana din imagine a fost ștearsă. Ultimele două falsificări din a doua imagine modifică zone foarte mici din imagine.

După falsificare, imaginile au fost transferate înapoi pe telefon și s- realizat autentificarea lor cu aplicația mobilă de autentificare. În Fig. 10 (c) sunt date rezultatele autentificării, unde zonele falsificate sunt marcate cu roșu. Se observă că aplicația de autentificare detectează și localizează cu precizie toate falsificările, inclusiv cele de dimensiune foarte redusă.

(a) (b) (c)

Fig. 10. Exemple de autentificare a unei imagini falsificate folosind aplicația mobilă „ImFakeCheck”: (a) imagini securizate (b) imagini falsificate și recomprimate JPEG; (c) imagini autentificate

Tot în această a doua etapă a proiectului a fost dezvoltat un **algoritm sigur și robust de autentificare a imaginilor color codate JPEG ce utilizează o semnătură digitală a imaginii stocată pe blockchain**, ce a condus la prezentarea și publicarea articolului intitulat „Authentication of JPEG Images on the Blockchain" în volumul conferinței internaționale International Conference on Control, Artificial Intelligence, Robotics and Optimization (ICCAIRO 2018). Un rezumat al articolului este prezentate în continuare.

În domeniul autentificării imaginilor există trei direcții principale: tehnici pasive (neinvazive, ce nu modifică imaginea originală în nici un fel), active (bazate pe watermarking digital, ce inserează în imagine un marcaj de autentificare invizibil) și bazate pe semnături digitale. Tehnicile pasive oferă un grad de certitudine scăzut în comparație cu celelalte doua tipuri de autentificare. Autentificarea bazată pe semnătură creează o amprentă digitală a conținutului imaginii originale. Acest lucru implică extragerea de parametri (caracteristici) ai imaginii și stocarea lor ca o semnătură a acesteia într-un fișier separat. Dacă imaginea este falsificată, semnătura extrasă din aceasta va diferi de cea originală și se poate trage concluzia că imaginea respectivă este neautentică.

În literatura de specialitate au fost propuse diverse tehnici de autentificare a imaginilor bazate pe semnături digitale. În [7] este prezentată o metodă de autentificare bazată pe semnătură cu posibilitatea de localizare a zonelor falsificate în domeniul Wavelet, dar este dependentă de detecția de margini și autorii declară că poate fi detectată doar modificarea unor obiecte importante din imagine. În [8] autorii propun o tehnică bazată pe funcții hash ce generează o semnătură cu parametri atât globali, cât și locali. Caracteristice globale sunt bazate pe momente Zernike, iar cele locale conțin informațiile de poziție și textură a regiunilor importante din imagini. O metodă de generare a unei semnături multi-rezoluție este descrisă în [9] și utilizează tehnici adaptive de extragere a caracteristicilor locale. Semnătura hash este atașată imaginii și poate fi utilizată la decodor pentru corecția transformărilor geometrice, ce pot apărea, și poate localiza zone modificate. În [10] caracteristicile imaginii stocate în semnătură sunt extrase folosind momentele Zernike și o transformată invariantă la scalare. Semnătura imaginii este redusă ca dimensiune, dar dezavantajul este o precizie mai slabă de localizare a falsificării.

Dezavantajul principal al tehnicile de autentificare bazate pe semnături digitale este necesitatea de a stoca semnătura într-un fișier separat, ce trebuie să fie disponibil la detector. Este important de adăugat faptul că, în funcție de modul în care a fost generată semnătura, și, în special dacă se dorește și localizarea falsificării, dimensiunea semnăturii poate fi destul de mare. Pentru imparțialitate și securitate, semnătura ar trebui stocată într-un loc public și, pe cât posibil, în mod decentralizat. Proprietățile tehnologiilor de tip blockchain le fac pe acestea adecvate pentru scopul stocării semnăturii. Tehnica propusă de noi în acest articol elimină dezavantajele prezentate anterior prin stocarea pe blockchain a semnăturii de dimensiune relativ mare a imaginii, folosirea pe post de semnătură propriu-zisă doar a valorii hash a tranzacției de pe blockchain (de dimensiune mult mai redusă) și inserarea acestei valori hash criptate în metadatele header-ului imaginii codate JPEG.

Un „blockchain" sau lanț de blocuri este o listă continuu crescătoare de blocuri de informație digitală, legate între ele și securizate criptografic. Fiecare bloc de pe blockchain conține o amprentă temporală, valoarea de la ieșirea unei funcții hash criptografice pentru blocul anterior și un câmp de informație, ca, de exemplu, date electronice ale unei tranzacții monetare.

Blockchain-urile au fost inițial dezvoltate ca o rețea decentralizată pentru transferul de monede digitale, fiind singura soluție la problema „plăților duble" fără utilizarea unui server centralizat. Această primă soluție dezvoltată de Satoshi Nakamoto [11] în 2008 a fost numită

„Bitcoin” și este astăzi utilizată la scară largă pe post de monedă digitală. Poate fi privită ca un jurnal public de tranzacții, complet decentralizat și distribuit, gestionat de o rețea peer-to-peer ce poate valida orice bloc nou atașat lanțului.

Noutatea tehnologiei blockchain este posibilitatea stocării și distribuției de date fără a fi nevoie de o entitate centrală de încredere pentru a facilita transferul digital. Pentru a atinge acest obiectiv, baza de date ce conține toate blocurile generate anterior este copiată pe mii de calculatoare din întreaga lume și este public disponibilă oricui. Securitatea datelor stocate pe blockchain este garantată de algoritmul de consens utilizat, procesul prin care nodurile rețelei ajung la un consens legat de adăugarea unui bloc nou autentic la blockchain, și este bazat pe criptografia matematică cu chei asimetrice. De exemplu, pentru a modifica un bloc existent de pe blockchain, un atacator trebuie să modifice toate blocuri mai noi pe un număr foarte mare de calculatoare din rețea, care stochează o copie locală a blockchain-ului. Acest lucru face falsificarea imposibil de grea. Tehnologia blockchain nu mai este folosită doar pentru transferul de monede digitale, ci a fost adoptată în ultimul timp pentru diferite alte scenarii și aplicații, ca de exemplu urmărirea lanțurilor de distribuție de marfă, comunicarea sigură între dispozitive Internet of Things (IoT), gestionarea identităților persoanelor, proveniența documentelor, și chiar și pentru piețe de schimb financiar și magazine decentralizate, vot digital și stocare decentralizată de date.

În acest articol propunem o aplicație nouă a tehnologiei blockchain pentru autentificarea imaginilor digitale. O versiune comprimată a semnăturii imaginii originale este stocată pe blockchain și astfel nu poate fi falsificată în nici un mod de către o parte malițioasă. Versiunea criptată a valorii hash a tranzacției de scriere a semnăturii pe blockchain este inserată în metadatele EXIF ale header-ului imaginii codate JPEG. Pentru a testa autenticitatea unei imagini de test și a localiza regiunile eventual falsificate, semnătura poate fi extrasă de pe blockchain și comparată cu cea generată local pentru imaginea de test.

Imaginea originală este mai întâi convertită din spațiul de culori RGB în spațiul YC_bC_r și în continuare vom lucra doar cu componenta Y de luminanță. Caracteristicile care vor forma semnătura imaginii sunt extrase din informația conținută în coeficienții DCT ai blocurilor de 8x8 pixeli ai imaginii, pentru ca tehnica de autentificare să fie robustă la compresie JPEG. Astfel, planul Y este descompus în blocuri de 8x8 pixeli și se calculează DCT-2D pe fiecare bloc. Blocurile rezultante de coeficienți sunt împărțite la matricea de cuantizare $\mathbf{Q}(q)$ corespunzătoare factorului de calitate q și rezultatul este rotunjit la cel mai apropiat număr întreg.

După această operație, vom numi coeficienții obținuți coeficienți DCT normați. Ne dorim să extragem corect semnătura digitală chiar dacă imaginea a fost comprimată ulterior cu un factor de calitate mai mare decât o valoare minimă $q_{\min}$. O proprietate importantă a funcției de cuantizare $f(x,\delta)$ este:

$$f\left(f\left(x,\delta_2\right),\delta_1\right)=f\left(x,\delta_1\right)+e,\ \text{if}\ \delta_2\le\delta_1, \tag{1}$$

unde x este valoarea ce trebuie cuantizată, δ este pasul de cuantizare, eroarea $e\in\{-\delta_1;0;\delta_1\}$ și

$$f(x,\delta)=\delta\cdot\text{round}\left(\frac{x}{\delta}\right) \tag{2}$$

Această proprietate este demonstrată în [12]. În practică, în majoritatea cazurilor eroarea e este egală cu zero.

În concluzie, dacă alegem pe post de matrice de cuantizare JPEG matricea $\mathbf{Q}(q_{\text{low}})$ pentru o cuantizare puternică cu factor de calitate mic și extragem semnătura din coeficienții normați,

această semnătură o sa fie identică chiar dacă imaginea a fost comprimată cu un factor de calitate mai mare decât q_{low}. O observație importantă este că procedeul de extragere a semnăturii nu modifică în nici un fel imaginea originală.

Algoritmul de extragere a caracteristicilor pentru semnătură a fost ales după testarea a diferite abordări, următoarea fiind cea care a dat cele mai bune rezultate. Fie $\mathbf{C}_{\text{norm}}$ matricea de coeficienți DCT normați ai unui bloc al imaginii. Elementele sale sunt $C_{\text{norm}}(x,y)$, $x,y=\overline{0,7}$. Valoarea $C_{\text{norm}}(0,0)$ este coeficientul DC normat iar ceilalți coeficienți sunt coeficienții AC normați. Din fiecare bloc sunt extrași câte 8 biți prin compararea unor coeficienți cu cinci valori de prag t_i, $\overline{i=1,5}$ după cum urmează:

- $C_{\text{norm}}(0,0)$ este comparat cu pragurile $t_1 = C_{\text{norm}}(0,0)_{\max}/4$, $t_2 = C_{\text{norm}}(0,0)_{\max}/2$, $t_3 = 3\cdot C_{\text{norm}}(0,0)_{\max}/4$, unde gama dinamică a coeficienților DC normați este $[0, C_{\text{norm}}(0,0)_{\max}]$;
- $C_{\text{norm}}(0,1)$, $C_{\text{norm}}(1,0)$, $C_{\text{norm}}(2,0)$ sunt comparați cu $t_4 = 0$;
- $abs(C_{\text{norm}}(0,1))$ și $abs(C_{\text{norm}}(1,0))$ sunt comparați cu un prag determinat empiric $t_5 = 3$.

Rezultatul comparațiilor ne dă un bit 1 logic dacă valoarea este mai mare sau egală decât pragul și 0 altfel. Dimensiunea rezultată a semnăturii va fi 0.52% din dimensiunea imaginii necomprimate.

Pentru a exemplifica procesul de generare a semnăturii, un exemplu este dat în continuare. Să presupunem că coeficienții DCT utilizați pentru generarea semnăturii blocului curent *j* sunt $C(0,0)=722$, $C(0,1)=53.4$, $C(1,0)=-13.2$, și $C(2,0)=33.5$. Să presupunem că dorim să putem extrage corect semnătura chiar dacă imaginea a fost comprimată cu un factor de calitate $q \geq 50$. Pentru a obține valorile normate ale coeficienților DCT selectați, avem nevoie de valorile pașilor de cuantizare de pe pozițiile corespunzătoare din matrice de cuantizare $\mathbf{Q}(50)$. Acestea sunt [16 11 12 14]. Coeficienții DCT normați vor fi $C_{\text{norm}}(0,0) = \text{round}(722/16) = 45$, și în mod similar $C_{\text{norm}}(0,1)=7$, $C_{\text{norm}}(1,0)=-1$, $C_{\text{norm}}(2,0)=2$. Cei 8 biți care formează semnătura blocului curent sunt obținuți prin compararea coeficienților normați cu cele cinci praguri. Gama dinamică a coeficienților DC normați este $C_{\text{norm}}(0,0)\in[0,128]$, rezultând $t_1=32$, $t_2=64$ și $t_3=96$. Ultimele două valori de prag sunt $t_4=0$ și $t_5=3$. Astfel, semnătura binară a blocului *j* va fi $\text{sig}_j = [1\ 0\ 0\ 1\ 0\ 1\ 1\ 0]$.

Semnăturile de autentificare pentru fiecare bloc de 8x8 pixeli sunt concatenate, obținându-se un vector binar de dimensiunea $s = MN/8$ biți. Acest vector binar este criptat cu algoritmul de criptare cu cheie privată AES-128.

Majoritatea algoritmilor din literatura de specialitate se opresc la acest pas și se bazează pe transmiterea fișierul semnătură împreună cu imaginea originală. Noutatea metodei propuse constă în inserarea semnăturii pe blockchain. Pentru aplicația noastră am utilizat blockchain-ul numit BigchainDB [13] datorită posibilității utilizării de blocuri de dimensiune mai mare, dar teoretic orice alt blockchain poate fi utilizat. Pentru a putea scrie date pe blockchain-ul specificat anterior, am făcut un cont pe www.bigchaindb.com. Utilizatorii înregistrați pot trimite tranzacții pe rețea după instalarea unui driver. Driver-ul este disponibil pentru diferite platforme: Java, JavaScript or

Python. Noi am utilizat driver-ul de Python deoarece necesită cele mai puține dependențe software, care sunt instalate automat. În programul Python, semnătura extrasă din imagine trebuie transformată într-un șir de caractere ASCII. Este generat un identificator de tranzacție (transaction hash) de 64 digiți hexazecimali și tranzacția este trimisă pe blockchain.

Semnătura criptată a imaginii poate fi oricând găsită pe blockchain cunoscând ID-ul tranzacției. Datorită naturii blockchain-ului, semnătura nu poate fi alterată sau ștearsă. Pentru a extrage de pe blockchain semnătura unei imagini este necesar doar ID-ul tranzacției, care are lungime fixă în comparație cu semnătura propriu-zisă a imaginii, care variază în funcție de dimensiunea acesteia. ID-ul tranzacției este la rândul său criptat și inserat într-unul dintre câmpurile EXIF ale header-ului fișierului JPEG destul de mare pentru a stoca varianta criptată a ID-ului tranzacției, ca de exemplu câmpul „Comment" (COM) de dimensiune maximă 64 kB.

Pentru a testa autenticitatea unei imagini este nevoie la decodor de cheile private de criptare a semnăturii și a ID-ului tranzacției. Apoi tranzacția este căutată pe blockchain, este extrasă semnătura criptată, aceasta este decriptată și comparată cu semnătura generată din imaginea de test, folosind același algoritm ca la codor. Dacă cele două semnături sunt identice, imaginea este declarată ca fiind autentică. Dacă nu, imaginea de test este declarată ca fiind falsificată. Algoritmul poate realiza și localizarea zonelor neautentice. Acest lucru este posibil, deoarece semnătura a fost formată prin concatenarea caracteristicilor extrase din fiecare bloc în parte. Astfel, unele blocuri pot fi marcate ca fiind autentice, iar altele ca fiind falsificate. Se poate construi o matrice binară de autentificare, cu fiecare element corespunzând unui bloc din imagine. Un „0" logic semnalizează un bloc autentic, iar un bloc „1" logic unul falsificat.

Dacă imaginea de test a fost editată, este improbabil să se fi falsificat doar blocuri izolate. Din această cauză, înainte de a lua o decizie finală legată de autenticitatea blocurilor, blocuri izolate sunt eliminate după cum urmează: dacă un bloc este determinat ca fiind autentic și are în jur doar blocuri neautentice, acesta este marcat ca fiind și el falsificat; respectic, dacă un bloc este determinat ca fiind falsificat și are în jur doar blocuri autentice, el este marcat ca fiind autentic. Acest lucru poate fi realizat folosind operațiile de morfologie matematică prezentate mai sus.

În final, blocurile care au rămas marcate ca fiind neautentice vor fi colorate cu roșu pentru a scoate în evidență zonele falsificate.

Algoritmul de autentificare propus a fost testat în mod automat pe un număr de 100 de imagini color naturale de rezoluție 512x512 pixeli. Semnătura a fost extrasă din fiecare imagine iar apoi fiecare imagine a fost falsificată în mod automat prin înlocuirea unei zone din imagine cu alt conținut selectat aleator. Dimensiunea zonei falsificate a variat de la 32x32 pixeli la 256x256 pixeli prin creșterea arie de 4 ori la fiecare pas (4 dimensiuni diferite pentru zona falsificată). Pentru fiecare zonă falsificată au fost considerate 10 poziții diferite, rezultând un număr de 4000 de imagini falsificate. Imaginile falsificate au fost comprimate folosind factori de calitate cu valori între 20 și 100 cu un pas de 20. În total au fost analizate 20000 de imagini. Indicatorii au fost rata de detecție de fals negativ (DFN) și rata de detecție de fals pozitiv (DFP).

Valorile DFP și DFN medii sunt date în Tabelul 6. DFP în lipsa falsificării a fost întotdeauna egală cu zero, demonstrând robustețea algoritmului la compresia JPEG. În rest, se observă că metoda dezvoltată reușește să păstreze DFN la valori rezonabil de joase pentru DFP de valori foarte joase.

Tabelul 6. Valori DFP și DFN medii pentru diferiți factori de calitate

Factor de calitate JPEG de recompresie	20	40	60	80	100
DFP [%] (fără falsificare)	0	0	0	0	0
DFP [%] (cu falsificare)	0.07	0.03	0.04	0.03	0.08
DFN [%]	2.78	4.4	3.31	4.18	3.44

a) b) c)

Fig. 11. Exemple de autentificare a două imagini falsificate utilizând algoritmul propus: a) imagini originale; b) imagini falsificate și ulterior comprimate cu un factor de calitate de 50; c) imagini autentificate

Fig. 11 prezintă două exemple de autentificare, unde zonele falsificate sunt colorate cu roșu. Putem observa că schema propusă poate detecta cu succes diferite tipuri de falsificări și poate diferenția compresia JPEG de modificările intenționate ale conținutului imaginii. Pot apărea unele erori de detecție la blocurile de 8x8 pixeli de la marginea zonei falsificate, ce conțin o parte din falsificate, dar aceste erori nu au un impact important asupra rezultatului final al autentificării. Schema are o rezoluție bună de detecție de 8x8 pixeli egală cu dimensiunea blocurilor utilizate la compresia JPEG.

De asemenea, pentru a pregăti dezvoltarea unei aplicații de autentificare video, a fost dezvoltat un **algoritm de detecție a schimbării scenei pentru standardul de compresie video de actualitate High Efficiency Video Coding (HEVC)**, rezultând un articol care a fost acceptat pentru prezentare și publicare la conferința internațională 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI 2018). Un rezumat al articolului este prezentate în continuare.

High Efficiency Video Coding (HEVC) reprezintă cel mai recent standard de compresie video, fiind destinat secvențelor video de rezoluție mare precum FullHD (1920x1080) și 4K (3840x2160). La fel ca în cazul standardului anterior, H.264 / MPEG-4 AVC (Advanced Video Coding), recomandarea realizată de ITU-T Video Coding Experts Group impune sintaxa fluxului comprimat precum și etapele pentru decodare, lăsând realizarea codorului la alegerea liberă a dezvoltatorului. Una dintre problemele pe care trebuie să le soluționeze cel care dezvoltă un codor video pentru HEVC este selecția cadrelor care se vor coda intra, denumite IRAP (intra random access point). Din perspectiva raportului de compresie, cadrele IRAP ar trebui introduse cât mai rar, întrucât sunt costisitoare ca număr de biți ocupați în fluxul binar. De regulă se introduc periodic la începutul unui GOP (Group Of Pictures), servind ca referințe pentru cadrele codate inter. Compresia intercadru funcționează eficient atâta vreme cât referințele sunt actualizate corect, iar acest lucru nu este prevăzut de standard în cazul schimbărilor de scenă. La schimbarea scenei este utilă introducerea unui IRAP, astfel încât să se poată câștiga cât mai mult din compresia inter a cadrelor următoare care sunt codate pornind cu referința actualizată IRAP și nu cu o referință care face parte din altă scenă, pentru care nu se poate exploata vreo redundanță temporală.

Algoritmul propus realizează detecția schimbărilor de scenă cu un volum redus de calcule, având în vedere implementarea acestuia pe un codor de complexitate joasă destinat echipamentelor mobile de mică putere care funcționează cu acumulator. Sunt procesate numai componentele de crominanță, U și V, aferente formatului YUV 4:2:0, întrucât acestea sunt subeșantionate cu factorul 2 față de luminanță și presupun mai puține valori de prelucrat. Testele au fost realizate pe secvențe video în format Full HD, pentru care matricea de luminanță ocupă 1920x1080 pixeli. Implementarea algoritmului s-a realizat folosind codorul de referință propus de ITU-T VCEG, denumit HM (HEVC test Model).

Metoda folosită pentru detecția schimbărilor de scenă are la bază extragerea din fiecare cadru, înainte de codare, a unui vector descriptiv. Compararea cadrelor succesive, pentru identificarea cadrului care precedă / introduce o scenă nouă, se reduce astfel la compararea vectorilor descriptivi succesivi. De la compararea unor matrice de mari dimensiuni cum sunt cele de la Full HD de exemplu, se ajunge la compararea unor vectori de dimensiuni semnificativ mai mici. Un vector descriptiv caracterizează practic din punct de vedere entropic cadrul din care a fost extras. Doi vectori distincți vor proveni așadar din două cadre video cu un conținut complet diferit.

Extragerea vectorului descriptiv asociat unui cadru are la bază metoda HOG (Histograms of Oriented Gradients) introdusă de Dalal și Triggs, folosită însă pentru matricele de crominanță. Pentru o matrice de crominanță se determină gradienții pixelilor și apoi se împarte matricea gradienților în blocuri. Pentru fiecare bloc se numără gradienții cu aceeași orientare, construind o histogramă a gradienților orientați. Prin concatenarea histogramelor în ordinea parcurgerii blocurilor și apoi normări succesive, se obține un descriptor sub formă vectorială. În urma testelor și simulărilor, s-a ajuns la o metodă simplificată de comparare a acestor descriptori, folosind varianța statistică a eșantioanelor vectorului precum și medierea pe termen scurt a

varianței. Evaluarea metodei s-a realizat prin determinarea unor parametri specifici tehnicilor de detecție a schimbărilor de scenă (Precise, Recall, F1).

Rezultatele experimentale obținute utilizând codorul de referință ITU „HM (HEVC test Model)", sunt foarte bune și demonstrează eficiența și precizia înaltă de detecție a metodei prin introducerea cu frecvență redusă de cadre codate intra care îmbunătățesc raportul semnal-zgomot al cadrelor ulterioare codate inter.

Direcții viitoare de cercetare și dezvoltare

Aplicația de autentificare a imaginilor dezvoltată este complet funcțională, obținând rezultatele pe care ni le-am propus la începutul proiectului, dar ea mai poate fi îmbunătățită. Câteva dintre direcțiile de cercetare și dezvoltare pe care dorim să ne concentrăm pe viitor includ următoarele:

- *Introducerea mai multor tehnici de autentificare.* La momentul actual aplicația de autentificare poate utiliza un singur algoritm de autentificare activă. Se dorește completarea ei cu mai mulți algoritmi, unii dintre aceștia fiind deja dezvoltați și testați în mediul de programare Matlab. Un exemplu ar fi implementarea unui algoritm, care, pe lângă rezistența la compresie JPEG, să fie robust și la unele operații geometrice neintruzive de scalare, cropping sau rotație a imaginii.

- *Modificarea aplicației pentru incorporarea tehnicilor de autentificare bazate pe semnături digitale și blockchain.* O altă categorie de metode de autentificare sunt cele bazate pe o semnătură digitală a imaginii originale, având avantajul că imaginea originală nu este modificată în nici un fel. Echipa a obținut deja rezultate în acest sens, un astfel de algoritm de autentificare bazat pe semnătură digitală fiind dezvoltat în cadrul etapei pe 2018 și prezentat mai sus. Creșterea securității algoritmului și reducerea dimensiunii semnăturii pot fi obținute prin folosirea în cadrul aplicației a tehnologiei blockchain.

- *Incorporarea unei galerii proprii.* În acest moment, aplicația dezvoltată se folosește de aplicația implicită a sistemului de operare Android pentru afișarea la dimensiuni originale a imaginilor. Nu se pot afișa, de exemplu, toate imaginile securizate sau testate. Pe viitor se dorește ca aplicația să conțină un modul propriu de galerie prin care să se poată obține această funcționalitate.

- *Scalarea inițială a imaginii pentru aplicații specifice (WhatsApp, Facebook, Skype, etc.).* Multe dintre aplicație de tip mesagerie sau rețea socială, ce permit transferul de imagini, realizează o prescalare a imaginilor transmise, dacă acestea depășesc o dimensiune maximă. Se dorește introducerea unei opțiuni prin care utilizatorul să poată selecta cu ce aplicație dorește să trimită fotografia după securizare. Aplicația de autentificare va prescala imaginea și apoi va introduce marcajul de securitate, astfel încât ea să nu mai fie scalată de aplicația de comunicare și autentificarea ei să se realizeze într-un timp redus.

- *Marcarea zonei sursă la falsificarea de tip copy-paste.* În momentul de față, aplicația detectează zonele din imagine care au fost modificate. În cazul operațiilor de tip „copy-paste", la care falsificarea a fost realizată prin copierea unei zone din imaginea originală în aceeași imagine, se poate măsura gradul de corelație dintre zonele marcate ca neautentice și restul imaginii, obținându-se un maxim pentru zonele sursă. Astfel ar putea fi marcate pe imaginea de test și zonele sursă la falsificarea de tip „copy-paste".

- *Securizarea automată a unui director întreg de imagini.* La momentul actual aplicația de autentificare poate prelucra automat doar o singură imagine. Unii utilizatori sunt interesați să

securizeze un volum mai mare de imagini. Din această cauză ne dorim să adăugăm posibilitatea de a securiza și a testa directoare întregi cu imagini.

Munca de cercetare științifică depusă de către membrii echipei de cercetare în cadrul etapei pe 2018 a proiectului s-a concretizat prin publicarea unui articol științific în volumul unei conferințe internaționale indexate ISI, un al doilea fiind acceptat la o altă conferință indexată ISI. De asemenea, un articol științific este în proces de evaluare la o revistă de specialitate internațională, cotată ISI. Aceste articole sunt enumerate în cele ce urmează:

- R. A. Dobre, R. O. Preda, C. C. Oprea, I. Pirnog, *Authentication of JPEG Images on the Blockchain*, International Conference on Control, Artificial Intelligence, Robotics and Optimization (ICCAIRO 2018), 19-21 mai, 2018. (publicat, în curs de indexare ISI)
- C. Oprea, R. Preda, I. Pirnog and R. Dobre, *Video Shot Boundary Detection for Low Complexity HEVC Encoders*, 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI 2018), iunie, 2018. (acceptat pentru publicare, în curs de indexare ISI)
- R. O. Preda, *Secure signature-based authentication of JPEG compressed images*, Information Processing Letters, 2018. (în curs de evaluare la revistă ISI)

În concluzie, putem afirma că s-au îndeplinit cu succes toate obiectivele etapei pe 2018 a proiectului de cercetare intitulat „Sistem automat de autentificare activă a imaginilor digitale pentru PC și terminale mobile”.

Bibliografie

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] M.A. Kim, K.S. Yoo, W.H. Lee, Digital signature with localization for image authentication, 2011 IEEE International Conference on Consumer Electronics (ICCE), pp. 725-726, 2011.

[8] Y. Zhao, S. Wang, X. Zhang, and H. Yao, "Robust hashing for image authentication using Zernike moments and local features," IEEE Transactions on Information Forensics and Security, vol. 8, no. 1, pp.55-63, ian. 2013.
[9] C.P. Yan, C.M. Pun, X.C. Yuan, "Multi-scale image hashing using adaptive local feature extraction for robust tampering detection," Signal Processing, Vol. 121, pp. 1-16, 2016.
[10] J. L. Ouyang, Y. Z Liu. and H. Z. Shu, "Robust Hashing for Image Authentication Using Sift Feature and Quaternion Zernike Moments", Multimedia Tools and Applications, Vol. 76, Issue 2, pp. 2609-2626, 2017.
[11] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", 2008, disponibil online la adresa https://bitcoin.org/bitcoin.pdf.
[12] C.-Y. Lin, and S.-F. Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," IEEE Transactions on Circuits and Systems for Video Technology, 11, (2), pp. 153-168, 2001.
[13] https://www.bigchaindb.com; accesat pe 17.06.2018.

18.06.2018

Director proiect
conf. dr. ing. Radu Ovidiu PREDA