SISTEM PENTRU ASISTAREA INTRODUCERII ÎN CALCULATOR A DATELOR DIN CÂMPURILE FORMULARELOR TIPIZATE

Mihnea Horia Vrejoiu

mihnea@dossv1.ici.ro

Institutul Naţional de Cercetare-Dezvoltare în Informatică - ICI Bucureşti

Rezumat: Introducerea manuală a datelor de pe suport de hârtie în calculator reprezintă o activitate pe cât de necesară şi importantă în contextul informatizării pe scară largă în mai toate domeniile, pe atât de consumatoare de resurse umane şi de timp şi o potenţială sursă de erori de tastare, mai ales în cazul volumelor mari de date ce trebuie preluate. În acest context, sunt extrem de utile tehnici şi instrumente prin care această activitate poate fi automatizată. Adesea însă, datorită cerinţelor critice privind acurateţea de preluare a datelor este neapărat necesară măcar o verificare şi validare a datelor de către operatorul uman în lipsa altor criterii şi/sau posibilităţii de validare automată. În acest context s-a propus şi implementat experimental o soluţie software semiautomată, bazată pe tehnici OCR/ICR, destinată asistării activităţii de introducere în calculator a datelor completate în formulare tipizate, cu format fix, pentru creşterea eficienţei, productivităţii şi acurateţei. Au fost efectuate experimente de testare, au fost făcute observaţii asupra funcţionării şi rezultatelor obţinute şi au fost sintetizate câteva concluzii şi posibilităţi de îmbunătăţiri şi optimizări ulterioare.

Cuvinte cheie: introducere de date în calculator, formulare tipizate, învăţare automată, învăţare supervizată, OCR/ICR, expresii regulate.

Abstract: Manually inputting data from paper support represents, on one hand, a necessary and important activity in the context of the widely spread informatization in most areas, but, on the other hand, so human, and time, resources consuming, and also a potential source of typing errors, especially when large volumes of data are inputted. In this context, techniques and tools by which this activity can be automated are extremely useful. However, often, due to critical requirements on the accuracy of the inputted data, it is absolutely necessary at least a check and validation of these data by the human operator in the absence of other criteria and/or automated validation possibilities. In this context, it has been proposed and experimentally implemented a semi-automatic software solution based on OCR/ICR techniques, intended to assist the work of inputting the data filled in standardized forms, with fixed format, for improving the efficiency, productivity and accuracy. Testing experiments were conducted, there were made observations on the functioning and results, there were summarized some conclusions and possible further improvements and optimizations.

Keywords: data inputting, standardized forms, machine learning, supervised learning, OCR/ICR, regular expressions.

1. Introducere

În pofida evoluţiei tehnologice şi a introducerii informatizării pe scară largă în cele mai multe sectoare de activitate, precum şi a politicilor şi eforturilor ecologiste de protejare a mediului şi de sustenabilitate actuale, totuşi volumul de informaţie vehiculată pe suport de hârtie şi care trebuie introdusă în calculator este încă foarte mare.

Un exemplu este cel al formularelor tipizate. Preluarea manuală în calculator a datelor din acestea poate implica volume mari de timp şi efort şi poate introduce erori de tastare, în unele situaţii inacceptabile. Automatizarea procesului a reprezentat una din problematicile cele mai vizate de marii producători de software în domeniul analizei de imagini OCR/ICR şi mangement al documentelor. Există deja soluţii comerciale pe piaţă pentru preluarea "automată" a datelor din formulare, cu funcţiuni şi performanţe diferite, dar şi altele anunţate care nu s-au impus (încă) pentru o utilizare pe scară largă. În continuare prezentăm succint câteva idei generale, sintetice, privind acest tip de aplicaţii.

Cele mai multe soluţii sunt configurabile şi se bazează pe crearea şi folosirea de şabloane / machete asociate tipurilor de formulare care specifică poziţionarea şi caracteristicile câmpurilor acestora. Unele îşi propun chiar o identificare automată a câmpurilor de interes pe diferite criterii de analiză a imaginii formularului scanat.

Majoritatea produselor de acest tip anunţă capacitatea de recunoaştere atât a caracterelor tipărite cât şi a celor scrise de mână (separate între ele). De asemenea, unele produse permit în plus şi recunoaşterea marcajelor din check-box-uri şi a codurilor de bare;

Deşi majoritatea producătorilor anunţă rate de recunoaştere spectaculoase ale motoarelor

OCR/ICR, toate aplicațiile acordă o mare atenție etapei de corecție şi validare post-recunoaştere. Corecțiile post-recunoaştere se fac în general prin utilizarea de dicționare, informații apriori despre câmpuri, calcularea unor sume de control etc., în timp ce validarea necesită întotdeauna intervenția utilizatorului.

Astfel, se poate afirma că soluțiile existente sunt practic semiautomate, ele oferind instrumente şi interfețe de asistare a operatorului uman în această activitate.

Majoritatea produselor oferă facilități de conectare, în ceea ce priveşte exportul de date, cu diverse sisteme de baze de date standardizate şi/sau proprietare.

2. Sistemul software propus

S-a propus şi implementat experimental o modalitate semiautomată de asistare a operatorului uman pentru creşterea eficienței, productivității şi acurateței în activitatea de introducere în calculator a datelor completate din câmpurile formularelor tipizate, cu format fix. Aceasta se realizează prin identificarea şi preluarea automată utilizând tehnici de recunoaştere optică / inteligentă de caractere (OCR/ICR [1][2]) a datelor respective şi afişarea acestora cu posibilități de editare, corectare şi/sau validare de către operator înainte de a fi stocate/utilizate mai departe.

Abordarea avută în vedere a pornit de la câteva observații elementare. Completarea datelor în câmpurile formularelor se poate face computerizat, prin tipărire la imprimante diverse, cu fonturi diferite, prin dactilografiere la diferite maşini de scris sau prin scriere de mână de către diverse persoane, cu scris diferit. Pe de altă parte, din punctul de vedere al tipului de date conținute, unele câmpuri completate pot fi de tip exclusiv numeric, altele exclusiv text, iar altele mixt alfa-numeric. De asemenea, pot conține exclusiv majuscule sau minuscule în componenta de tip text, pot avea un format fix (obligatoriu) pentru caracterele completate, etc.

Astfel, o primă idee de la care s-a pornit în abordarea aleasă a constat în utilizarea unor metode şi algoritmi care permit în acelaşi timp flexibilitate şi generalitate în ceea ce priveşte partea de OCR/ICR prin folosirea unor mecanisme de învățare-clasificare şi recunoaştere [3] bazate pe acumularea evolutivă, organizată, a cunoştințelor din exemple de caractere (litere şi/sau cifre), ca alternativă mai adecvată decât „înghețarea" unor descrieri ale acestora prin programare (*hardcoded*). Această opțiune a fost motivată de faptul că formularele pot prezenta câmpuri variate, cu conținut diferit, care la rândul său poate fi completat în numeroase moduri, atât ca tip de scriere cât şi calitate a acesteia. În acest context, alegerea unui tip adecvat de reprezentare parametrică a caracterelor constituie una din problemele cheie, ea trebuind să asigure premizele unui compromis acceptabil între puterea de generalizare şi capacitatea de discriminare. A fost utilizată o reprezentare „vagă" pentru fiecare caracter, obținută pe baza porțiunii de imagine din interiorul dreptunghiului de încadrare a acestuia şi codificată ca vector de N parametri. Pornind de la astfel de reprezentări în urma procesului de învățare, algoritmul de clasificare utilizat (o variantă ne-neurală a [4] cu structuri spațiale hiperparalelipipedice pentru clasificator şi utilizând distanțe Manhattan în loc de euclidiene, mai eficientă atât ca şi capabilitate de acoperire cât şi din punct de vedere computațional) realizează acoperirea spațiului problemei cu regiuni compuse din unul sau mai multe domenii hiperparalelipipedice imbricate. Fiecare astfel de structură (regiune) specifică unei clase învățate şi etichetată cu codul ASCII al caracterului alfanumeric respectiv este definită de reuniunea domeniilor hiperparalelipipedice de acoperire a clasei respective, deformate/alungite neuniform pe axe în procesele de învățare evolutivă. Structura compusă din totalitatea regiunilor astfel construite la învățare formează o bază de cunoştințe de caractere învățate. La recunoaştere, algoritmul caută clasa a cărei structură (regiune) „acoperă" reprezentarea obținută pentru caracterul țintă (de recunoscut) curent, furnizând în caz de succes codul ASCII al acestuia sau un cod ales convențional în cazul nerecunoaşterii. S-a dovedit că algoritmul utilizat are o putere rezonabilă atât de generalizare cât şi de discriminare în acelaşi timp, permițând o stabilizare relativ rapidă a ratei de recunoaştere. Este de subliniat totuşi faptul că pentru a se atinge un nivel de performanță, stabilitate şi fiabilitate convenabil este necesară instruirea sistemului pe un număr considerabil de exemple reale, mai ales pentru recunoaşterea scrisului de mână.

O a doua idee importantă degajată din analiza problemei a fost aceea că, pentru fiecare tip de

formular (şi condiţii identice de rezoluţie la scanare) să se asocieze un fişier şablon/machetă (*template*). În acesta sunt stocate informaţii descriptive – atribute de localizare, denumire, tip, format etc., – referitoare la fiecare câmp de interes şi tipul conţinutului acestuia. Aceste cunoştinţe preliminare despre câmpuri şi conţinutul lor sunt folosite (când există) la recunoaştere, atât pentru a se diminua riscul unor confuzii, cât şi pentru a se creşte calitatea acesteia prin utilizarea numai a acelor baze de cunoştinţe potrivite pentru tipul fiecărui câmp. Mai exact s-a avut în vedere utilizarea, atunci când este posibil, a unor baze de cunoştinţe diferite specializate pentru caractere litere şi respectiv cifre, incluzând aici scrisul de mână cu caractere separate. Alegerea automată a acestor baze de cunoştinţe la recunoaşterea anumitor câmpuri se face pe baza informaţiei de tip asociat câmpului. În plus, pe de altă parte, rezultatul recunoaşterii brute mai este supus şi unor corecţii automate post-recunoaştere bazate pe eventuala informaţie de format specific asociată câmpului respectiv, dacă aceasta există în fişierul machetă/şablon amintit mai sus.

Trebuie neapărat menţionat aici că recunoaşterea automată eficientă a formularelor nu poate avea loc fără o anumită „disciplină" impusă *design*-ului acestora, modului de completare, de manipulare şi de achiziţie a imaginilor acestora. În acest sens este de preferat ca rubricile sau chenarele în care trebuie completată informaţia să fie tipărite cu o culoare sensibil mai puţin intensă (de exemplu verde deschis) decât cea cu care se va realiza completarea lor (de exemplu, albastru închis sau negru). Caracterele completate, litere sau cifre trebuie să fie cât mai lizibile şi cât mai rezonabil poziţionate în câmpurile respective, fără a se atinge sau suprapune între ele şi fără a depăşi sau atinge, pe cât posibil, eventualele chenare şi rubricaţii de separare. Formularele nu trebuie să prezinte adnotări, mâzgăleli şi/sau alte pete. Scanarea trebuie să fie făcută cât mai îngrijit cu putinţă, prin setarea cât mai potrivită a parametrilor de rezoluţie, luminozitate şi contrast, ca şi prin asigurarea unei cât mai bune reproductibilităţi a poziţionării formularelor respective în imaginea digitală rezultată. Astfel de cerinţe se înscriu în conceptul deja răspândit de „*machine readability*" pentru documente având ca scop facilitarea proceselor de analiză şi segmentare automată a imaginilor, care preced recunoaşterea propriu-zisă.

La nivel de arhitectură generală, pentru implementarea ideilor prezentate mai sus sistemul experimental realizat [6] conţine următoarele componente funcţionale:

- un subsistem de achiziţie a imaginilor formularelor (prin scanare, sau de pe disc);
- un subsistem de machetare, pentru crearea şi stocarea de şabloane (fişiere conţinând câte o macro-descriere) specifice fiecărui tip de formular, care permite definirea interactivă, *off-line*, a câmpurilor de interes din acestea, precum şi „etichetarea" lor cu atribute pe baza unor cunoştinţe apriori despre tipul şi caracteristicile datelor care vor fi completate, incluzând şi o gramatică de descriere a acestora, bazată pe expresii regulate [5];
- un subsistem de învăţare interactivă, *off-line*, dedicat instruirii pe caractere (litere sau cifre) de diverse tipuri şi fonturi, preluate din câmpurile formularelor, cu crearea şi actualizarea unor baze de cunoştinţe corespunzătoare;
- un subsistem de recunoaştere automată a caracterelor completate în câmpurile formularului curent, utilizând informaţiile despre acestea preluate dintr-un fişier machetă / şablon corespunzător precum şi baza sau bazele de cunoştinţe de caractere adecvate;
- un subsistem de afişare–editare–validare a rezultatelor recunoaşterii pe câmpurile formularului care permite operatorului uman să compare rezultatul recunoaşterii cu originalul, să facă eventuale corecţii şi/sau să valideze aceste rezultate;
- un subsistem supervizor de definire şi configurare interactivă a contextului de lucru la un moment dat (tipul formularului – şablonul cu care se va lucra, bazele de cunoştinţe ce vor fi folosite la învăţare-recunoaştere etc.), comandă a funcţiilor sistemului şi control automat al fluxului de lucru în contextul stabilit.

În Figura 1 este redată arhitectura şi schema funcţională sintetică a întregului sistem [6].

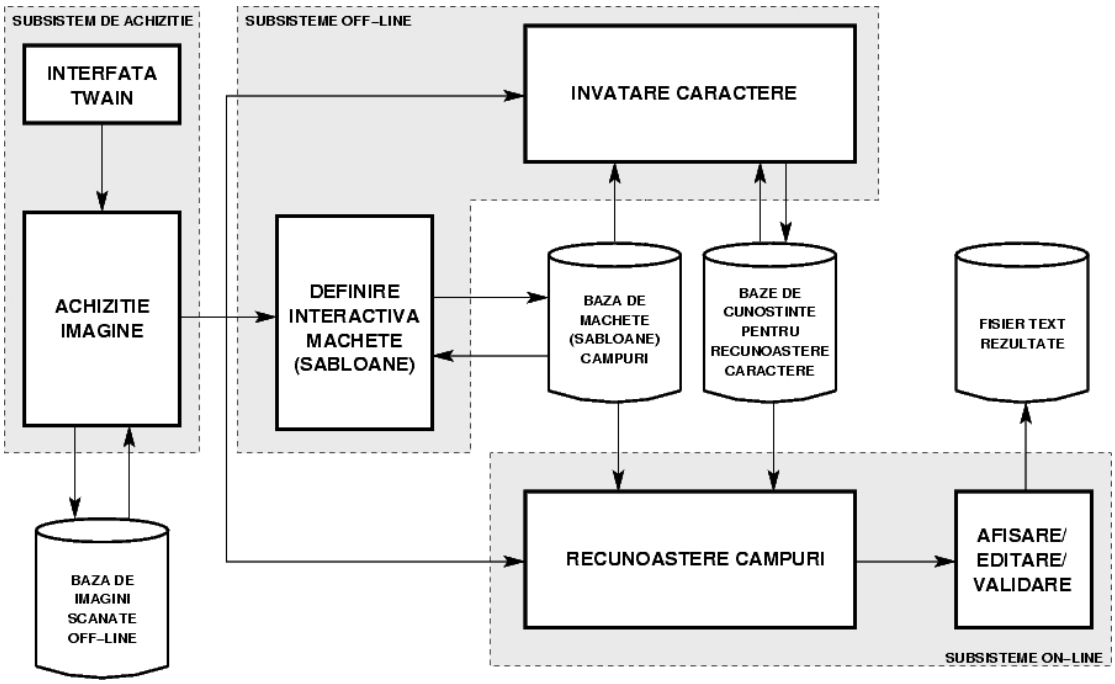

Figura 1. Arhitectura şi schema funcțională sintetică a sistemului

Subsistemul de achiziție a imaginilor formularelor permite fie preluarea/digitizarea unui nou document (comandând un scaner printr-o interfață TWAIN) şi salvarea acestuia, fie încărcarea de pe disc a unei imagini de formular scanat şi salvat anterior. Imaginea este pusă la dispoziția subsistemelor de machetare, învățare, recunoaştere şi afişare-editare-validare rezultate. Este posibilă selectarea simultană şi încărcarea apoi automată, succesivă, a mai multor imagini de pe disc. Aceasta permite lucrul pe loturi (*batch*) de imagini în activitatea de introducere de date, astfel încât să se asigure o productivitate sporită.

Subsistemul de machetare oferă posibilitatea predefinirii zonelor de interes (a câmpurilor care trebuie completate) dintr-un anumit tip de formular prin reținerea coordonatelor acestora în imagine. De asemenea, permite asocierea unor atribute acestor câmpuri:

- număr de ordine câmp;
- denumire câmp;
- tip câmp (numeric / alfa / mixt / nespecificat);
- descriere codificată format câmp (dacă informația respectivă este disponibilă) printr-o gramatică de structură a câmpului respectiv, bazată pe formalismul RE (*Regular Expressions* = expresii regulate) [5], de exemplu: cu sau fără spații, exclusiv majuscule sau minuscule, specificare format fix obligatoriu etc.

Aceste atribute permit folosirea bazelor de cunoştințe cele mai potrivite precum şi anumite corecții automate post-recunoaştere care să elimine unele eventuale confuzii (cum ar fi B cu 8, O cu 0, Z cu 2, l cu 1 etc.) şi/sau să „sincronizeze" şirul recunoscut cu un şablon predefinit obligatoriu. Odată stabilite cele de mai sus, meta-informația astfel asociată unui anumit tip de formular este stocată pe disc într-un fişier machetă/şablon urmând a fi folosită de subsistemele de învățare şi recunoaştere ori de câte ori se lucrează cu respectivul tip de document. De asemenea, ea poate fi refolosită şi de subsistemul de machetare însuşi pentru eventuale modificări ulterioare. Trebuie precizat faptul că, într-o astfel de machetă coordonatele imagine ale câmpurilor definite sau selectate sunt strâns legate de rezoluția de scanare, nefiind valabile decât pentru formulare de acelaşi tip, scanate la o aceeaşi rezoluție. Menționăm că pentru fiecare tip de câmp, pentru a se mări şi mai mult gradul de încredere în preluarea automată a informației şi a se diminua efortul aferent etapei de corecție-validare, mai este posibilă şi asocierea câte unui dicționar, pe baza căruia să se mai realizeze o corecție automată post-recunoaştere.

Subsistemul de învățare are rolul de a acumula organizat cunoştinţele necesare recunoaşterii ulterioare a caracterelor alfanumerice (litere şi cifre) completate în câmpurile formularelor, pornind de la exemple reale. Fiecare câmp din imaginea curentă este segmentat automat în caractere şi se încearcă mai întâi o recunoaştere a acestora utilizându-se starea curentă a bazei de cunoştinţe de lucru. Succesiv, pentru fiecare caracter (încă) nerecunoscut şi care se doreşte a fi învățat este generată reprezentarea sa parametrică, pe baza conținutului efectiv al imaginii acoperite de fereastra sa de încadrare. Setul (vectorul) de parametri obținuți este folosit de algoritmul de clasificare-învățare care are ca finalitate extinderea "experienței" bazei de cunoştinţe selectate în sensul recunoaşterii şi a caracterului considerat. În sens larg, aceasta se poate traduce prin eventuala adăugare a unei noi instanțe pentru clasa corespunzătoare caracterului respectiv, etichetată cu codul ASCII al acestuia în baza de cunoştinţe indicată. De asemenea, s-a avut în vedere şi posibilitatea ştergerii/excluderii unui caracter din respectiva bază de cunoştinţe la nevoie (de exemplu, în situația unei învățări greşite). Bazele de cunoştinţe astfel create şi/sau actualizate sunt stocate pe disc şi vor fi folosite la recunoaştere şi/sau pentru actualizare prin învățări ulterioare.

Subsistemul de recunoaştere analizează fiecare câmp din imaginea curentă pe baza coordonatelor definite în macheta corespunzătoare tipului de formular respectiv, realizează segmentarea acestuia în caractere şi utilizează atributele asociate câmpului şi bazele de cunoştinţe indicate pentru a furniza textul completat (litere şi/sau cifre) care a fost recunoscut, ca şir de caractere (secvență de coduri ASCII). Împreună cu acest şir recunoscut sunt furnizate mai departe subsistemului de afişare-editare-validare şi informațiile obținute din macheta utilizată curent, respectiv coordonatele câmpului şi celelalte atribute asociate lui, pentru toate câmpurile definite în macheta respectivă.

Subsistemul de afişare-editare-validare oferă o interfață prin care operatorul uman poate edita sau corecta rezultatele recunoaşterii din fiecare câmp, având totodată în față şi imaginea reală a câmpului respectiv. De asemenea poate vizualiza la dorință imaginea întregului formular. Odată acceptată corectitudinea informației preluate din formular, operatorul validează datele respective care vor fi salvate pe disc. Cu ajutorul unor interfețe de conversie potrivite, aceste date pot fi stocate/înmagazinate în formatul de bază de date specific fiecărei aplicații concrete pe care ar deservi-o sistemul de asistare a introducerii de date din formulare, sau pot fi furnizate într-un format intermediar general utilizat (cum ar fi XML), către alte aplicații specifice de prelucrare şi/sau stocare a datelor respective.

Subsistemul de configurare, comandă şi control (supervizor) are ca rol:

- integrarea celorlalte subsisteme implementate, într-un ansamblu funcțional unitar şi coerent;
- asigurarea selecțiilor şi setărilor necesare fiecăruia din acestea;
- gestionarea fluxurilor de comandă şi de date implicate.

Acest subsistem furnizează practic şi interfața utilizator din care pot fi comandate principalele funcțiuni ale sistemului:

- scanare imagini şi salvare pe disc sau selecție imagine/imagini de lucru din cele stocate şi încărcarea imaginii curente;
- definire machetă pentru un anumit tip de formular şi salvare pe disc;
- selecție şi încărcare machetă curentă;
- selecție şi încărcare set de baze de cunoştinţe curent;
- pornire sesiune de învățare din imaginea curentă;
- pornire sesiune de recunoaştere pe setul de imagini selectate, folosind macheta curentă şi setul de baze de cunoştinţe curente şi încheiată cu afişare/editare/validare succesivă pentru fiecare imagine analizată;

- ieşire din sistem (terminare program).

În plus, subsistemul asigură şi toată ordinea şi logica de apelare a funcţiilor şi procedurilor specifice pentru realizarea fiecăreia din aceste funcţiuni precum şi managementul şi controlul datelor şi fluxului acestora.

Prin integrarea subsistemelor componente amintite mai sus a fost realizat un sistem experimental [6], care a fost utilizat pentru experimentări şi testări.

3. Implementare, rezultate, observaţii şi concluzii

Menţionăm că toate implementările – pentru subsistemele componente, cât şi pentru sistemul experimental – s-au realizat pentru platforme Microsoft Windows pe 32 biţi, folosind limbajul de programare C.

Sintetic, rezultatele pot fi enumerate după cum urmează:

- a fost proiectat şi implementat un sistem experimental pentru asistarea introducerii în calculator a informaţiilor de tip text completate în câmpurile formularelor. Acesta a fost realizat prin integrarea tuturor subsistemelor descrise anterior într-un ansamblu funcţional unitar şi coerent, sub o interfaţă grafică utilizator ergonomică, bazată pe ferestre, meniuri şi *dialog box*-uri sub Windows 32 biţi. Sistemul experimental implementează modul în care se realizează selecţiile, configurarea şi comanda / controlul acestuia prin intermediul interfeţei utilizator, permiţând stabilirea contextului de lucru (ce imagine se foloseşte la un moment dat, ce machetă, ce baze de cunoştinţe, ce moduri speciale de analiză a imaginilor, etc.), precum şi lansarea funcţiilor sistemului;
- utilizându-se sistemul experimental au fost efectuate o serie de teste vizând fiecare componentă a acestuia. Au fost efectuate mai multe teste în ce priveşte modul de lucru cu interfaţa şi au fost analizate rezultatele obţinute pe diferite formulare;
- s-au efectuat teste care au pus în evidenţă faptul că algoritmul de învăţare-clasificare şi recunoaştere proiectat şi implementat are o putere rezonabilă atât de generalizare, cât şi de discriminare în acelaşi timp şi permite o stabilizare relativ rapidă în cazurile „normale”. Pe de altă parte, aşa cum era de aşteptat, în cazuri mai „dificile”, cum este recunoaşterea scrisului de mână, pentru a se atinge un nivel de performanţă, stabilitate şi fiabilitate convenabil este necesară instruirea sistemului pe un număr considerabil de exemple reale (multe instanţe pentru fiecare clasă).

Mai prezentăm în continuare alte câteva observaţii şi concluzii desprinse pe parcursul experimentelor/testelor derulate.

În urma analizei rezultatelor obţinute, s-au degajat câteva idei vizând necesitatea extinderii implementărilor anterioare pentru rezolvarea a două probleme:

- în cazul formularelor de acelaşi tip, scanate cu poziţionări variate şi/sau alegându-se setări de contrast şi/sau luminozitate (praguri de binarizare) diferite etc., s-au constatat deplasări ale coordonatelor definite (prin şablon/machetă) pentru câmpuri. Deoarece funcţionarea sistemului se bazează tocmai pe o corectă reproductibilitate a localizării câmpurilor în formulare de acelaşi tip, scanate în aceleaşi condiţii de rezoluţie, aceasta ducea la scăderea artificială a ratei de recunoaştere pe câmpurile respective, din cauza erorilor de segmentare.
    - pentru rezolvarea acestei categorii de probleme, s-a proiectat, implementat şi adăugat o metodă prin care pentru fiecare imagine se caută un punct de referinţă (x0, y0) relevant pentru conţinutul invariant acesteia, care să fie (cât mai) reproductibil de la o scanare la alta. Aceasta se realizează prin analiza matricei imaginii scanate şi identificarea celei mai din stânga coloane (x0) şi respectiv celei mai de sus linii (y0), care conţin porţiuni consistente din conţinutul invariant (tipărit) al formularului. Coordonatele câmpurilor se reţin şi se utilizează ca fiind

relative la acest punct (ale cărui coordonate, x0 şi y0, sunt folosite ca *offset*-uri pe axele x şi respectiv y);

- în cazul formularelor care prezintă marcaje / liniaturi de tip „grilă” pentru câmpurile de completat s-au observat situații în care mecanismele de segmentare a caracterelor completate în acestea nu erau suficient de eficiente.
  - ca soluție pentru acest tip de probleme s-a proiectat, implementat şi adăugat o funcție de filtrare automată a liniilor orizontale (lungi) şi a celor verticale (scurte) ce compun astfel de marcaje / liniaturi (de tip „grilă”).

De asemenea, aşa cum am mai amintit s-a degajat pregnant necesitatea antrenării sistemului pe un număr important de exemple de caractere, mai cu seamă în cazul scrisului de mână, pentru a se putea ajunge la o anumită stabilizare a gradului de generalizare şi implicit a ratei de recunoaştere automată în aceste situații.

4. Posibile dezvoltări ulterioare

Încheiem prezentând câteva posibile dezvoltări/îmbunătățiri pe care le întrezărim şi care ar mai putea fi eventual aduse sistemului. Acestea pot fi grupate în următoarele trei categorii:

(a) îmbunătățirea performanțelor OCR/ICR, de exemplu prin:

- rafinarea mecanismelor de segmentare a caracterelor din zone marcate cu chenar sau linii grilă de ghidare, respectiv de filtrare mai eficientă a acestor marcaje parazite;
- extinderea mecanismelor de ajustare automată a poziției (localizării) şi/sau dimensiunilor câmpurilor față de cele din şablon/machetă în funcție de eventuale încadrări puțin diferite în unele scanări;
- adăugarea unor mecanisme de corecție – validare automată pentru unele câmpuri pe baza unor dicționare ataşate acestora prin şablon/machetă;
- extinderea setului de atribute de format câmp cu: câmp numeric negativ, câmp numeric cu N zecimale, câmp din care sunt relevante doar primele/ultimele N caractere;
- proiectarea şi implementarea unei alte reprezentări şi mai potrivite pentru caractere (mai ales pentru a asigura o generalizare mai uşoară în cazul scrisului de mână) şi/sau chiar a unui algoritm alternativ de clasificare-recunoaştere complementar (eventual bazat pe memorii asociative) în acest caz;
- antrenarea intensivă şi extensivă (masivă) a unor baze de cunoştințe specifice, mai ales pentru scrisul de mână;

(b) extinderea/îmbunătățirea ergonomiei interfeței utilizator de exemplu prin:

- adăugarea unui obiect de tip “*selection tree*” în interfața de corectare – editare – validare care să permită o navigare mai intuitivă printre rezultatele „OCR-izării” câmpurilor unui formular;
- adăugarea unor obiecte de control de tip *entry field*, *check box*/*radio button*, în interfața de editare atribute câmp (pentru extensiile legate de şablon/machetă: format câmp, dicționar ataşat unui câmp);

(c) adaptarea funcționalității pentru aplicații specifice de exemplu prin:

- pregătirea pentru interfațarea cu – respectiv export al datelor de ieşire spre – formate specifice de baze de date, eventual printr-un format intermediar adecvat, general acceptat şi utilizat (cum ar fi XML).

Potențialii utilizatori / beneficiari ai unui astfel de sistem adaptat şi configurat specific pot fi: administrații financiare, poliție, evidența populației, bănci, oficii poştale, regii, agenții, operatori de telefonie, alte birouri de funcționari publici etc., unde este necesară preluarea pe scară largă a

informațiilor completate în câmpurile unor formulare tipizate, cu format fix (declarații tip, cereri tip, mandate tip, avize tip, foi de depunere / vărsământ tip, ordine de plată/încasare/schimb valutar tip etc.).

BIBLIOGRAFIE

1. **ONȚANU, D.-M.; VREJOIU, M. H.:** Sistem de recunoaştere optică a caracterelor bazat pe rețele neurale – produs program pentru recunoaşterea scrisului de mână, Tema A15, Institutul Național de Cercetare-Dezvoltare în Informatică - ICI Bucureşti, 1993.
2. **VREJOIU, M. H.; ONȚANU, D.-M.:** Sisteme de programe de tip OCR, PC World România, nr. 6, edit. IDG România, iunie 1995.
3. **MITCHELL, T.:** Machine Learning, McGraw-Hill, ISBN: 0070428077, March 1997.
4. **ONȚANU, D.-M.:** Learning by Evolution. A New Class of General Classifier Networks and Their Training Algorithm, Advances in Modelling & Analysis, AMSE Press, vol. 26, nr. 2, pp. 27-30, 1993.
5. **FRIEDL, J. E.:** Mastering Regular Expressions, 2nd Ed., O'Reilly, July 2002.
6. **VREJOIU, M. H.; ONȚANU, D.-M.:** Sistem de recunoaştere optică de caractere pentru citirea automată de formulare scanate. Faza a V-a. Realizare sistem experimental pentru asistarea introducerii în calculator a informațiilor de tip text completate în câmpurile formularelor, Raport de fază, proiect PN0313-0301, Institutul Național de Cercetare-Dezvoltare în Informatică - ICI Bucureşti, noiembrie 2005.