



UNIVERSITATEA POLITEHNICA BUCURESTI

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE, DEPARTAMENTUL DE
INFORMATICĂ ȘI INGINERIE

Inferență scenelor artificiale bazată pe execuția eficientă în paralel a grafurilor tensorale aciclice direcționate

Rezumat

Andrei Ionut DAMIAN (doctorand)

Nicolae Tapus (conducator de doctorat)

2021



Contents

1	Abstract.....	3
2	Introducere.....	5
2.1	Analiza aplicativității cercetării și dezvoltării propuse în viața reală	5
2.2	Vederea de ansamblu a propunerii de arhitectură și implementare	9
2.3	Experimente vs. aplicativitate reală: provocări și așteptări.....	12
3	Rezumatul tezei	14
3.1	Descriere ipotezelor și a motivației.....	14
3.2	Rezumarea stadiului actual al cercetării.....	15
3.3	Rezumatul capitolului Arhitectură	17
3.4	Rezultate notabile ale cercetării	19
3.5	Experimentare	21
3.5.1	1.2.6 Rezumatul principalelor contribuții personale.....	22
3.5.2	1.2.7 Lucru în continuare.....	Error! Bookmark not defined.

1 Abstract

Construirea interfețelor grafice necesare interacțiunii utilizatorilor cu aplicațiile (*Graphical User Interface - GUI*) este un subdomeniu important al domeniului dezvoltării software, indiferent de mediul de implementare, scopul final al sistemelor, orizontatul sau verticalul țintă.

Încă de la începuturile istoriei *GUI-urilor* au fost dezvoltate diverse tehnici pentru proiectarea vizuala, arhitecturala și implementarea propriu-zisă a interfețelor utilizator adresându-se platforme diferite, încercându-se echilibrarea între viteza de implementare (prin *dezvoltarea rapidă a aplicației - RAD*), calitatea graficii și uzabilitate. Pe măsură ce sistemele au evoluat de-a lungul anilor, cu generație după generație, a apărut o nouă gamă de nevoi legate de *GUI*: nevoia de a regândi, re-proiecta și reconstrui interfețele grafice cu utilizatorul, precum și nevoia de automatizarea anumite comportamente. De menționat că prin „comportament automatizat” ne referim la sarcina de secvențiere și automatizare fără interacțiune umană a unei serii de comenzi care sunt de obicei date de un operator uman al aplicației.

Nevoile menționate mai sus, respectiv de reproiectare și re-structurare, au fost abordate de ceva timp cu tehnici clasice precum: reutilizarea codului (dacă este disponibil) și reproiectarea interfețelor grafice (*GUI*). Pentru cazul de automatizare a comportamentului, au fost și sunt propuse metode care abordează simularea de „acțiuni *GUP*” bazate pe euristici programabile („scriptare”) precum „clic pe coordonatele X,Y pentru activare buton” sau „mutare cursor la poziția H,W și introducere nume utilizator”.

Astfel, instrumentele software de dezvoltare rapidă a aplicațiilor (*Rapid Application Development - RAD*) au evoluat pentru a oferi cicluri de dezvoltare a *GUI* cât mai rapide și mai productive și, legat de subiectul automatizării, a apărut o nouă gamă de sisteme software: sisteme software de *automatizare robotizată a proceselor* (sau RPA) care permit utilizatorilor să construiască scenarii de automatizare în care serii de comenzi și operațiuni *GUI* sunt în fapt scriptate și executate pe baza euristiciilor.

Cu toate acestea, în toate aceste cazuri, câteva probleme inerente nu au fost rezolvate, cum ar fi: (i) reproiectarea aplicațiilor software pentru care codul sursă inițial nu mai este disponibil; (ii) iterație rapidă de la machetele produse de un designer grafic la interfețele funcționale – *de la mockup la GUI functional*; (iii) înțelegerea vizuala semantică și automata,



de catre un agent virtual intelligent, a componentelor interfeței grafice cu utilizatorul, sau mai concret înțelegerea funcționalității *GUI*-ului si a elementelor sale vizuale; (iv) înțelegerea semantică rapidă a formularelor tipărite/pre-tiparite sau grafice (pe ecran) și conversia informațiilor, fără limitarea la metodele clasice proprii *recunoașterii caracterelor* (OCR) .

Toate aceste probleme enumerate anterior au primit si primesc o atenție activă atât din partea mediului academic, cât și din partea industriei fiind încă subiecte importante de cercetare.

Pe parcursul lucrării vom aborda majoritatea acestor probleme si vom argumenta in vederea sprijinirii ipotezelor propuse atat prin prezentarea arhitecturala formala precum si prin enumerarea unora din rezultatele cercetarilor ce au fost aplicate cu success in industrie si sunt folosite commercial la scara larga.

2 Introducere

2.1 Analiza aplicativității cercetării și dezvoltării propuse în viața reală

Principala motivație din spatele muncii noastre a fost înrădăcinată în **nevoia** reală de a **converti aplicațiile vechi din medii vechi de execuție** – cum ar fi sistemele bazate pe baze de date desktop perimate sau bazate pe arhitecturi clasice client-server – **în sisteme bazate pe infrastructuri de tip Cloud**. Acest obiectiv particular are rădăcini multiple în nevoia de gestionare eficientă a resurselor și este tratat în industria software prin abordarea clasică a gestionării ciclurilor de dezvoltare software pentru migrarea aplicațiilor. În acest context este necesar să enumerăm rapid câteva din elementele cheie ce îngreunează și de multe ori blochează procesul de migrare. Din perspectiva motivației managementului resurselor, putem enumera atât costurile de contractare și de gestionare a echipei bazate pe dezvoltatori și experți în dev-ops, costuri necesare dezvoltării software pentru migrare, cât și costurile de oportunitate legate de timpul implicat în acest proces particular (*de mai jos descrie abordarea clasică a migrării și costurile directe asociate*). Aspectele tehnice adiționale ce trebuie luate în considerare la analiza abordării clasice sunt în mare parte legate de riscurile asociate cu potențialele module cu erori, probleme de configurare și implementare, problemele de acceptanță și adopție și așa mai departe.

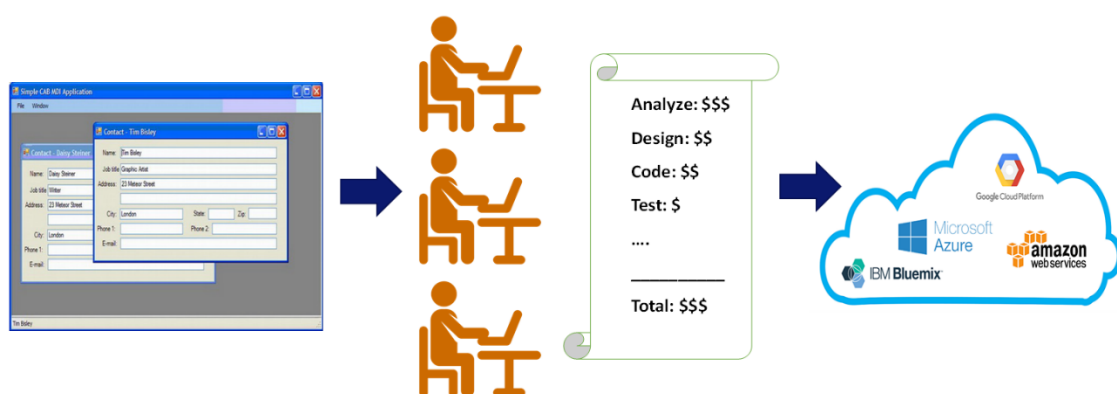


Figura 1 - Abordarea clasică a migrației sistemului - echipa de dezvoltare software rulează un întreg ciclu de dezvoltare pentru a se implementa în noul mediu țintă (poate) bazat pe Cloud Computing

Pe langa aceasta problema reala a migrarii catre medii si infrastructuri Cloud, in urma analizei nevoilor din viața reală existente actualmente în piata, au fost identificate si alte cazuri importante conectate în domeniile automatizării și prototipării rapide a aplicatiilor si interfetelor grafice cum ar fi folosirea agenților inteligenți pentru automatizarea interacțiunii operator-*GUI* prin procese robotizate (*Robotic Process Automation*). Pentru a fi și mai concreti vom analiza puțin mai detaliat un caz real aferent acestei directiei de automatizare a procesului de interactiune om-interfață prin inlocuirea agentilor de automatizare bazati pe scriptarea euristica a comenzilor cu agenți inteligenți. Vom analiza in continuare comportamentul unui agent de automatizare euristic ce trebuie sa actioneze o apasare de mouse pe un buton al interfeței cu utilizatorul. Putem rezuma sarcina în limbaj natural după cum urmează, pentru simplitatea explicației, deși de obicei este descrisă în limbaje de script: „*MUTAȚI mouse-ul în această zonă dreptunghiulară absolută a ecranului și efectuați operația CLICK*” ce se traduce în a comanda mouse-ului să se deplaseze într-o anumită locație și sa genereze un eveniment de tip *clic* pe interfața cu utilizatorul fără a necesita nici o intervenție umană. Urmatorul pas este sa va imaginați că, în timp ce locația absolută rămâne aceeași, butonul pe care trebuie să facem *clic* automat se mută cu forma sa datorită anumitor circumstanțe – spre exemplu mutarea *ferestrei* în care este respectivul buton sau rescalarea *ferestrei* aplicatiei. Pornind de la acest exemplu concret, obiectivul nostru este acela de a utiliza agenți inteligenți care să **recunoască vizual** în exemplul de mai sus dacă *formularul GUI a fost mutat* în alt loc pe ecranul interfeței sau că aspectul vizual al zonelor de introducere de date a fost modificat. Pentru claritate in figurile urmatoare este ilustrat acest caz.

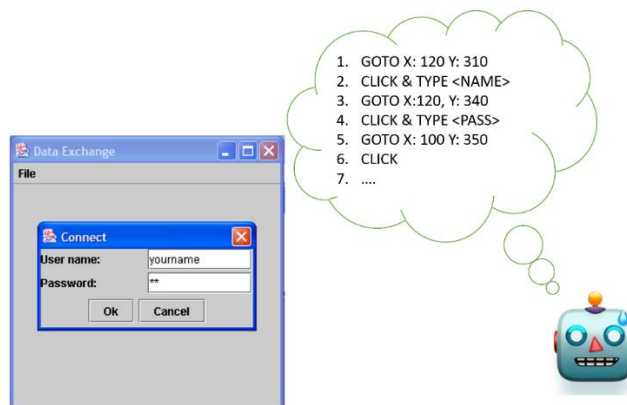


Figura 2 – agentul RPA simplu preprogramat pe baza unor reguli pentru o anumită sarcină merge la anumite coordonate și completează informațiile necesare. Sarcina este îndeplinită cu succes

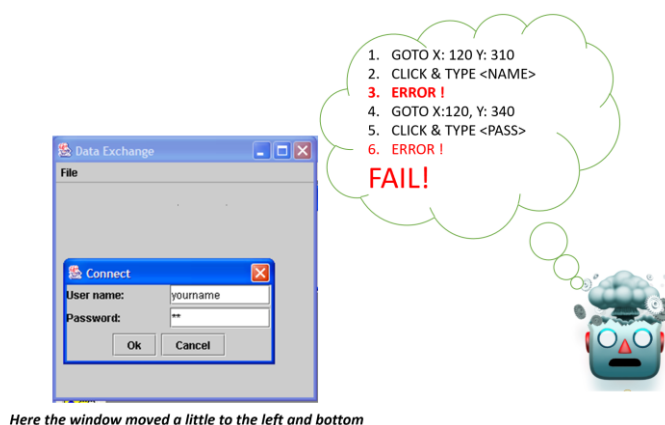


Figura 3 – În cazul urmatoru agentul RPA bazat pe euristici scriptate nu poate finaliza sarcina deoarece mediul și-a schimbat parametrii și euristica nu se mai aplică (echivalent: fereastra s-a mutat de la coordonatele preconfigurate și evenimentul „clic” implicit generează eroare)

În finalul conturării contextului motivației cercetării și dezvoltării ce sta la baza lucrării prezentăm o listă de cazuri de utilizare realiste bazate pe analiza nevoilor utilizatorilor legate de diferite scenarii, analiza realizată în cadrul etapelor inițiale ale cercetării doctorale:

- A. Îmbogățirea și îmbunătățirea proceselor aferente automatizării interfețelor GUI, procese furnizate în prezent de sistemele RPA prin metode euristice. După

cunoștințele noastre în domeniul RPA majoritatea implementarilor si sistemelor din piata se bazeaza pe metodele de scriptare euristica. Pentru acest caz particular ne propunem avansarea cercetarii si tehnologiei in directia utilizarii de agenti inteligenti cu capacitate de intelegere semantica a GUI-urilor. Doar recent (2020) companiile din acest domeniu au promovat noi cercetari si abordări bazate pe *Computer Vision* si invatare automata profunda pentru sarcina de inferenta a componentelor interfețelor grafice. Acest domeniu se află încă în faza de cercetare și experimentare la momentul redactării acestui raport.

- B. Migrarea, traducerea și întreținerea automată si la scară largă a sistemelor software vechi, cum ar fi sisteme complexe de contabilitate client-server/ER. sisteme bancare sau alte sisteme software dezvoltate si *mostente* in urma cu 15-30 de ani. Aplicațiile potențiale sunt multiple, pornind de la necesitatea de a traduce un sistem învechit dezvoltat pentru un anumit mediu de tip desktop intr-o noua infrastructura Cloud (un exemplu de bază ar fi dezvoltarea unei noi versiuni a unui sistem de contabilitate moștenit/invechit care trebuie sa migreaze dintr-un sistem bazat pe terminale text in interfete GUI online Cloud) până la necesitatea de a înțelege rapid și la fața locului interacțiunea și comportamentul utilizatorului în cadrul unui sistem complex.
- C. Prototiparea rapidă în limbaje de programare a elementelor de experiența a utilizatorului și in particular a programarea interfeței cu utilizatorul a fost în mod constant o cerință importantă în dezvoltarea *front-end*. Deși această sarcină specială este strâns legată de cea anterioară de traducere și migrare, este totuși un scenariu diferit mai ales prin prisma situatiei in care „intrarea” sistemului nu este un ecran al unei aplicatii si un mokup de aplicatie neexistenta (realizat cu unelete de grafica pe calculator sau chiar desenat manual). O altă cerință potențială și constrângere a acestui caz de utilizare este cerința pentru dezvoltarea fără cunostinte de programare – *code-less GUI development* – adică dezvoltarea unei interfețe grafice cu utilizatorul fără cunoștințe prealabile de limbaje de programare.
- D. Nu în ultimul rând, unul dintre cazurile de utilizare din viața reală se referă la înțelegerea avansată și automatizata a rapoartelor, formularelor și tabelelor electronice. În diverse domenii atât orizontale, cum ar fi contabilitatea sau logistica, cât și verticale, cum ar fi sectorul financiar, există o nevoie din ce în ce mai mare de operațiuni mai eficace și mai eficiente prin automatizarea inteligentă a

proceselor. Organizațiile folosesc diverse sisteme pentru a produce date în diferite formate și etape ale unui anumit proces, toate acestea sub forma de rapoarte sau formulare citibile de către actori umani ai procesului. Exemple reale de cazuri de utilizare sunt companiile care generează facturi fizice sau electronice sau documente de raportare a tranzacțiilor și apoi le livrează partenerilor lor, care, la rândul lor, au opțiunea de a extrage manual sau automat informații structurate fără a avea la dispoziție un sistem de analiză sau inter-schimbare a datelor structurate. În toate aceste scenarii, există o nevoie din ce în ce mai mare de post-procesare avansată a datelor dincolo de recunoașterea de bază a caracterelor (OCR). Pentru o explicație mai clară vom analiza rapid două cazuri particulare:

- i. colectarea și prelucrarea automată a datelor pe baza formularelor web online. Acest prim caz este un proces de analiză, inferență și traducere a interfeței cu utilizatorul care, în acest scenariu, va genera o ieșire de tip *cod scriptat* care poate varia de la simplu JSON la HTML5 mai complex (<https://html.spec.whatwg.org/multipage/>) și până la PHP (<https://www.php.net/docs.php>) sau alte limbaje de script pentru aplicații web.
- ii. digitizarea și preprocesarea datelor structurate aferente tabelelor și formularelor pre-tipărite sau scanate, cum ar fi rapoartele financiare lunare sau alte date similare nestructurate (imagine). În acest scenariu particular, obiectivul este de a genera o reprezentare digitală multiplatformă, cum ar fi valorile separate prin virgulă (CSV) ale unui document tipărit, sau cum ar fi o foaie de calcul tipărită care nu este disponibilă în formă digitală editabilă.

2.2 Vederea de ansamblu a propunerii de arhitectura și implementare

În cadrul domeniului Inteligenței Artificiale s-au realizat salturi de inovare considerabile în ultimii ani, în principal datorită îmbunătățirilor continue în subdomeniul învățării profunde automate și, de asemenea, datorită adoptării pe scară largă în cadrul comunității de cercetare și dezvoltare AI a calculului paralel bazat pe GPU și, nu în ultimul rând, datorită proliferării

bibliotecilor publice de seturi de date pentru o multitudine de domenii tinta ca medicina, vehicule autonome, securitate si siguranta, intelegerea limbajelor naturale, s.a.m.d. În multitudine de domenii tinta de cercetare existente și potențiale pe care le adreseaza IA trebuie să menționăm un domeniu important si anume cel al dezvoltării sistemelor software și al automatizării întreținerii sistemelor software. Acest domeniu este considerat, de majoritatea cercetatorilor si practicienilor, un domeniu de cercetare din categoria „Sfântul Graal” datorita incapacitatii chiar si a celor mai complexe modele de a rationa si intelege procesele de analiza, proiectare si imlementare pe care le parcurge un actor uman si in particular un programator software.

Viziunea noastră, prezentată în continuare în cadrul acestei lucrari, a fost să cercetăm și să dezvoltăm sisteme cu adevărat inteligente capabile să analizeze fluxurile video din experiența utilizatorului – deduse din ecrane-martor ale interacțiunii operatorului uman cu interfețele grafice GUI – fluxuri provenite din diverse surse și, în final, să inferam analize reale și utilizabile, inclusiv detalii reale la nivel de cod sursa ale respectivelor interfețe observate, cum ar fi exemplul simplu descris în figura de mai jos.

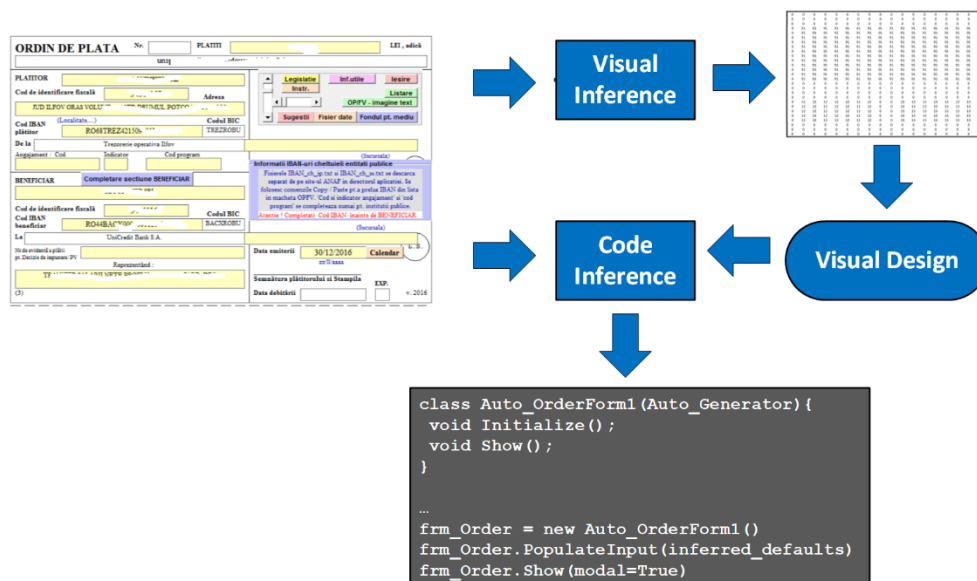


Figura 4 - De la o aplicație desktop cu un singur formular la un script de design vizual (cum ar fi HTML, JSON etc.) si până la codul sursă real într-un limbaj de programare țintă

Elementul cheie al cercetării propuse este acela de a obține un model neural bazat pe învățare automată profundă sau o suită de astfel de modele încapsulate într-o bandă de asamblare cu scopul de a realiza inferența și analiza unei scene artificiale reprezentate de o imagine a interfeței cu utilizatorul. Pe o perioadă de peste 3 ani am cercetat și dezvoltat diverse experimente care sunt menționate în cadrul tezei, cu accent deosebit pe cercetările și experimentele din ultima perioadă. Un alt obiectiv al perioadei de cercetare din cadrul studiilor doctorale a fost analiza și compararea cercetărilor și dezvoltării experimentale cu alte cercetări similare și alte inițiative existente în această zonă.

În cadrul lucrărilor științifice publicate pe durata studiilor doctorale, ca de altfel și în cadrul lucrării, argumentăm ca tehnicile și cercetările specifice stadiului actual al cercetării în zona modelelor neurale de imagistică sunt focusate, construite și antrenate pe baza seturilor de imagini naturale. Datorită acestui fapt, modelele DAG (convolution directioned acyclic graph) își vor dezvolta în timpul procesului de antrenament *detectori* de caracteristici ale imaginilor naturale, cu excepția modulelor neurale initiale din fluxul grafului orientat care vor învăța caracteristicile vizuale primitive de bază (linii, puncte, culori, etc). Ca urmare, una dintre zonele de interes ale cercetării și experimentării din cadrul studiilor doctorale a fost cercetarea și dezvoltarea modelelor DAG specializate pentru recunoașterea imaginilor sintetice și antrenarea acelor modele pe propriile noastre seturi de date experimentale.

În final modelele neurale propuse și antrenate „end-to-end” sunt în astfel capabile să deducă detalii funcționale ale *experienței utilizatorului* – primitive de GUI – din imaginile sintetice de intrare propuse. Aceste informații pot fi ulterior procesate pentru a fi traduse automat în codul sursă operațional într-o platformă țintă, cum ar fi HTML/JavaScript.

În cadrul lucrării prezente este abordat din anumite perspective stadiul actual al cercetării și tehnicii în două domenii diferite, dar conectate: cel al modelelor de învățare profundă automată pentru imagistica computerizată și zona procesării și calculului numeric paralel bazat pe GPU cu scopul de instruire eficientă a DAG-urilor (Directed Acyclical neural Graph) și al operaționalizării în producție. În continuare, vom prezenta arhitectura întregului nostru experiment end-to-end, inclusiv lucrările și abordările timpurii bazate pe arhitecturi de modele superficiale și ulterior cele din urmă clase de arhitecturi de modele bazate pe învățare automată profundă. O sub-secțiune specială este dedicată cercetării și dezvoltării setului nostru de date

de imagini artificiale care este publicat în regim Open Source pentru a beneficia de potentiala contribuție a comunității internaționale de cercetare.

În cele din urmă, lucrarea se încheie cu o revizuire a celor mai importante contribuții propuse și a concluziilor tezei, concluzii ce includ potențiale aplicații multiple din viața reală – de la implementarea de sine stătătoare a modelelor noastre până la încorporarea arhitecturilor CloudifierNet în pipeline-uri RPA externe. O observație importantă este că, în ciclurile noastre de cercetare și experimentare, am proiectat și implementat două versiuni diferite ale CloudifierNet și anume (v1 denumit *CloudifierNetV1*, v2 denumit *CloudifierNetV2*) și pentru fiecare versiune am implementat mai multe sub-versiuni *ajustând* dimensiunea graficului și cerințele sale de calcul și resurse GPU.

2.3 Experimente vs. aplicativitate reală: provocări și așteptări

Încă de la început trebuie precizat că rezultatele noastre experimentale finale au fost operaționalizate într-un prototip de sistem funcțional care a fost re-utilizat în cadrul unor aplicații cu utilitate directă industrială. Totodată, în limita resurselor disponibile, o echipă de cercetători și ingineri de date lucrează la îmbunătățirea performanței benzii de asamblare de modele și a mediului de execuție rapidă.

Una din cele mai importante probleme pe care am încercat să le tratăm cu atenție a fost aceea a reajustării continue a așteptărilor. În timpul cercetării și experimentelor din perioada studiilor doctorale ne-am confruntat cu mai multe probleme care separe în mod clar rezultatele muncii noastre experimentale de aplicarea la nivel real și la scară largă. Analiza acestor provocări individuale este importantă atât din perspectiva stabilirii așteptărilor corecte de la modulele experimentale realizate, stabilirea așteptărilor versus ambițiile de cercetare, cât și a previzualizării lucrărilor ulterioare care pot fi făcute pentru a îmbunătăți rezultatele actuale:

A. Domeniul limitat de aplicabilitate al modelelor și modulelor cercetare și dezvoltate, domeniu determinat de generarea/producția setului de date sintetice de: Una din activitățile propuse de cercetare și experimentare a fost aceea de generare a Setului de date artificiale (AD) – și anume imaginile primitivelor/comenzilor proprii interfeței cu utilizatorul GUI precum și capturile complete de ecran ale interfeței cu utilizatorul. Acestea nu pot, sub nicio formă, să surprindă toate variațiile potențiale ale interfețelor

utilizator ale oricărui standard sau abordare de UX (experiență de utilizare) disponibil anterior sau în prezent. În teza noastră, precum și în lucrările publicate, punem accent pe selecția efectivă a mai multor standarde de interfață cu utilizatorul, cum ar fi aplicațiile vechi Microsoft Windows bazate pe MFC, Delphi sau alte medii de dezvoltare similare. Cu toate acestea, un set de date universal și, prin urmare, un pipeline de modele potențial aplicabil universal depășește domeniul de aplicare al activității noastre.

- B. ***Aplicarea nefezabila a rezultatelor proiectului experimental la sisteme și aplicații ce prezintă abordări netradiționale ale UX/GUI (experienței utilizatorului) cu interfețe de utilizator care nu respectă regulile vizuale și funcționale clasice:*** Ca o problemă complementară celei prezentate anterior o avem și pe cea privind imposibilitatea suitei noastre de modele neuronale de a „înțelege” interfețele utilizator care nu urmează abordări comune în ceea ce privește fluxul experienței utilizatorului. Un astfel de exemplu sunt interfețele în care, să zicem, „butoanele” nu urmează niciun model vizual pe care îl au de obicei butoanele interfeței grafice cu utilizatorul în cadrul seturilor de date folosite la antrenare.
- C. ***Necesitatea potențială de a limita mediul de dezvoltare țintă la aplicații bazate pe web care nu necesită funcționalități complexe de partea clientului :*** În sfârșit, în urma numeroaselor experimente și analize de cazuri reale de utilizare, am ajuns la concluzia că din multitudinea de medii țintă potențiale (cum ar fi mobil-Android, mobil-iOS, web, MS-windows, Unity etc.) ne vom concentra pe aplicațiile web cu abordarea clasică model-view-controller limitând utilizarea bibliotecilor complexe de pe partea client (cum ar fi jQuery) la chestiuni simple și ușor.

3 Rezumatul tezei

În această secțiune vom prezenta pe scurt principalele aspecte descrise în fiecare dintre secțiunile principale ale tezei cu referiri la capitolele individuale respective.

3.1 Descriere ipotezelor si a motivatiei

Motivația din spatele tezei și activităților sale de cercetare și dezvoltare – prezentate cu cazurile de utilizare din industrie în *Capitolul 1.1* - a fost aceea de a construi o abordare eficientă și viabilă pentru analiza și recunoașterea cu o granularitate fină a imaginilor specifice interfețelor cu utilizatorul (GUI) fără utilizarea de euristici descrise de script-uri sau logica programabilă.

Identificarea punctul de plecare al ancorării în nevoia reală și motivația din spatele lucrării și a ipotezelor propuse poate fi urmărită în diverse cazuri de utilizare din industrie și în experimente academice – de la nevoia de automatizare a proceselor robotizate cu autoadaptare sau nevoia de a converti aplicațiile moștenite (legacy) în noi infrastructuri de tip Cloud Computing fără însă a avea disponibilitatea codului sursă inițial. Alte cazuri reale de utilizare în industrie care merită menționate sunt recunoașterea formularelor electronice completate sau a șabloanelor de formulare electronice sau extragerea sau introducerea datelor sau prototiparea rapidă a interfețelor moderne de utilizator fără a fi nevoie de instrumente de proiectare software, cum ar fi instrumentele de dezvoltare rapidă a aplicațiilor.

Pentru a avea o vedere mai concretă și o mai bună înțelegere, vom lua, analiza și împărtăși câteva cazuri de utilizare în definiția problemei, ipoteza soluției propuse și aplicația reală la nivelul industriei:

- i. Recunoașterea vizuală automată de computer a interfețelor GUI fără acces la regulile predefinite sau euristici programabile
 - **Problemă** : Având în vedere capturile de ecran ale aplicațiilor vechi, interfețele utilizatorului generează o colecție de artefacte, locații și scopuri/functiuni de interfață.
 - **Ipoteza** : Folosind modele neurale detectăm și localizăm fiecare dintre obiectele interfeței GUI

- **Aplicație** : Automatizarea proceselor robotizate beneficiază de acest caz de utilizare, renunțând la necesitatea regulilor scriptate și a euristicilor, reducând astfel timpul de implementare și configurare și crescând drastic viabilitatea generală și invariabilitatea dimensiunii/locației/aspectului elementelor GUI.
- ii. Prototiparea rapidă a interfețelor utilizator GUI funcționale, bazată pe interfețe vizuale vechi sau simple machete
- **Problemă**: Plecând de la ecrane și/sau machete de interfață cu utilizatorul, capturi de ecran ale aplicațiilor moștenite sau simple schite facute manual se generează codul de script complet al interfeței cu utilizatorul
 - **Ipoteza**: Folosind modele neurale adanci se analizează imaginile de intrare și generează direct secvențe de cod text (script)
 - **Aplicație** : prototipare rapidă de-la-desen-la-cod a interfețelor utilizator chiar și de către non-programatori – deci fără cunoștințe anterioare ale limbajului de programare tinta– sau designeri vizuali și generarea de cod sursă pentru o platformă Cloud;

3.2 Rezumarea stadiului actual al cercetării

Analiza stadiului actual al tehnicii și modul în care aceasta se leagă de cercetarea prezentată în lucrare este descrisă în *Capitolul 2* al tezei. Mai multe domenii și subiecte de cercetare active în comunicarea științifică, subiecte strâns legate de zonele de cercetare propuse, sunt prezentate și analizate în detaliu: abordări bazate pe GPU pentru optimizarea calculului numeric și arhitecturi avansate pentru construirea de grafuri aciclice direcționate pentru imagistica computerizată și prelucrarea limbajelor naturale.

Adoptarea la scară largă a unităților de calcul (PC, laptop, etc) bazate pe GPU-uri cu capacități de calcul considerabile, recenta evoluție rapidă a acestor plăci hardware și noile adăugări în special în familia Nvidia și nu numai (cum ar fi cele mai recente TPU - Tensor Processing Units) au permis atât comunității academice, cât și celei comerciale să proceseze cantități mari de date folosind paralelizare la un nivel de eficiență superior algoritmilor

numerici. Operațiile de convoluție discrete facil vectorizabile si paralelizabile, calculul paralel al mai multor fluxuri de date în grafurile tensoriale, arhitecturile de tip *Transformer* sunt doar câteva exemple în care calculul numeric paralel bazat pe GPU/TPU a contribuit major la avansul cercetării si aplicativității acesteia.

În munca noastră, câteva zone speciale și componente individuale au primit o atenție specială în ceea ce privește optimizarea calculului numeric paralel: noua abordare propusă pentru sub-grafuri cu mecanisme de tip porți și grafurile neurale pentru recunoașterea scenelor artificiale. Deși construcția *kernel*-urilor GPU este un subiect încapsulat de bibliotecile de calcul și optimizare tensoriale, cum ar fi *Tensorflow* sau *PyTorch*, am menționat tehnicile și abordările atât în conținutul tezei cât și în lucrările publicate la începutul studiilor doctorale.

Arhitecturile de grafuri neurale specifice învățării profunde automate, arhitecturi rezultate din cele mai recente lucrări si cercetări sunt prezentate atât în **Capitolul 2.1** precum și în **Capitolul 2.2**. În analiza făcută am inventariat abordările clasice cunoscute în *Computer Vision* concentrându-ne pe metode mai recente bazate pe grafuri aciclice orientate profunde. Deși principalele arhitecturi modele neurale prezentate se referă în mod direct la diferite probleme cunoscute și cazuri de utilizare clasice, principalul obiectiv al experimentelor noastre și, prin urmare, rezultatele prezentate sunt concentrate pe inferența scenei interfeței grafice cu utilizatorul (localizarea și detectarea obiectului de tip control GUI).

Un accent secundar a fost îndreptat în zona modelelor generative utilizate pentru producerea secvenței codului sursă. În aceasta zona analiza stadiului actual a fost refocusată din zona segmentării în imagistica spre analiza tendințelor moderne în procesarea neuronală a limbajului natural.

Un aspect important al lucrării a fost acela de a găsi abordări inovatoare pentru auto-ajustarea arhitecturilor de graf care să folosească metode bazate pe procesul de optimizare a grafurilor neurale în loc de abordări clasice, cum ar fi căutarea exhaustivă sau aleatorie a spațiului de opțiuni (hyper-parametrii) pentru arhitectura grafului (căutare în grilă). În acest domeniu particular, lucrările și experimentele anterioare privind mecanismele de auto-învățare au primit o atenție specială în **Capitolul 2.5** și pe scurt în **Capitolul 2.2.2**.

Pentru a rezuma principalele domenii de activitate conexe avem următoarele:

- i. **Evaluare (rularea) eficientă a grafurilor tensoriale, bazată pe calculul paralel numeric GPU**
- ii. **Arhitecturi de imagistica profundă pentru localizarea și segmentarea artefactelor vizuale**
- iii. **Metode generative de traducere imagine-la-secvență**
- iv. **Auto-învățarea hiperparametrilor specifici topologiilor de grafuri prin cautarea în spațiul de soluții folosind funcțiile de cost specifice optimizării grafului tensoral**

3.3 Rezumatul capitolului Arhitectură

Întregul proces de cercetare și experimentare s-a bazat pe o abordare ciclică și avansare pas cu pas pornind de la abordări simple către soluții din ce în ce mai profunde și mai complexe. Acest proces a condus la două tipuri de rezultate, așa cum este descris în *Capitolul 3.1*: rezultatele și livrabilele specifice domeniului, precum și rezultatele care pot fi aplicate în mai multe domenii și care au fost deja aplicate la mai multe cazuri de utilizare industrială din viața reală.

Procesul de proiectare detaliat și abordarea arhitecturală cu toate detaliile lor sunt descrise pe deplin în *Capitolul 3* al tezei. Pașii inițiali ai procesului de cercetare și experimentare s-au bazat în întregime pe abordări de bază „superficiale” (abordări de învățare automată ce nu sunt bazate pe modele neurale și grafuri complexe). Aceste abordări inițiale au fost implementate cu ajutorul modelelor de regresie logistică folosind rularea eficientă în mediul de calcul numeric paralel. Practic, abordarea inițială „de bază”, bazată pe modele simple de învățare automată, a fost aceea de a folosi simple produse matriciale între elemente ale spațiului de intrare – reprezentarea imaginii RGB a interfeței cu utilizatorul – și multiple matrice de greutate reprezentând o potențială primitivă de interfață cu utilizatorul. Toate acestea au fost implementate folosind mecanisme tip ferestre glisante. Mai concret, am folosit capacitățile de calcul numeric paralel al GPU pentru a calcula mai multe ipoteze 1-vs-all pentru toate regiunile potențiale ale imaginii interfeței cu utilizatorul.

Urmând această abordare destul de naivă de inferență a scenei reprezentată de o captură de ecran a interfeței cu utilizatorul, am început să dezvoltăm abordări mai complexe bazate pe

grafuri aciclice direcționate cu module de convoluție discrete și separabile – această abordare arhitecturală fiind prezentată pe deplin în *Capitolul 3.6*. În acest proces de rafinare a arhitecturilor, obiectivul nostru a fost acela de a construi o clasă bine optimizată și echilibrată de grafuri aciclice orientate profunde din perspectiva performanței vs costuri și consumul de energie. Astfel, am cercetat potențialul beneficiu al construirii de subgrafurilor capabile de auto-învățare a propriei topologii *ideale* așa cum este explicat și formalizat în *Capitolul 3.3*. Această abordare inovatoare permite unui model neural să-și adapteze propria topologie și implicit funcționare și structura a fluxului de date, pe baza funcției și a obiectivului general de optimizare.

Alte câteva aspecte importante privind arhitectura modelelor și modulelor cercetate sunt prezentate în continuare în *Capitolul 3*, cum ar fi analiza opțiunii de învățare-prin-transfer pe care am explorat-o în *Capitolul 3.4*, comparația dintre antrenamentul cu date artificiale versus imaginile generate manual de oameni în *Capitolul 3.5*, și respectiv detalii despre procesul de optimizare în *Capitolul 3.7*.

Pentru a sintetiza principiile principale utilizate în proiectarea arhitecturii noastre experimentale, putem enumera următoarele:

- i. Un set de **obiective și principii de bază** a fost stabilit de la bun început, cum ar fi utilizarea unor infrastructuri eficiente de calcul numeric paralel pe piața de masă bazate pe GPU urmărind minimizarea amprentei de carbon a procesului de antrenare
- ii. Experimentele au fost **începute plecând de la** abordări de **nivel scăzut, complexitate scăzută** pentru a stabili linii pornire clare
- iii. **Complexitatea a fost adăugată treptat** și mai multe provocări au fost abordate iterativ
- iv. În cele din urmă, arhitectura grafurilor și în special a sub-grafurilor MGU bazată pe mecanisme de auto-poarta permis generarea familiei noastre de DAG-uri

!

3.4 Rezultate notabile ale cercetării

Există trei domenii diferite în care cercetarea noastră realizată în perioada studiilor doctorale a produs rezultate viabile și validate, cu mențiunea importantă că unele dintre ele sunt deja utilizate în prezent în implementări de producție pe diferite cazuri de utilizare. Probabil cel mai important și relevant element de noutate atât pentru industrie cât și pentru mediul de cercetare academic constă în propunerea unei abordări inovatoare de reglare a hiperparametrilor ce definesc topologia grafurilor și subgrafurilor aciclice direcționate adânci. Noutatea propunerii mecanismului **Multi self- Gating** se încadrează în linii mari în curentul internațional de cercetări în domeniu în care obiectivul de determinare a arhitecturii ideale de graf este bazat pe auto-învățare în loc de căutarea clasică exhaustivă sau aleatorie în spațiul discret al soluțiilor. Această inovație este descrisă în detaliu **Capitolul 3.3** și este actualmente utilizată în prezent într-o serie de sisteme de producție atât în domeniul imagisticii cât și în domeniul analizei predictive – prognoza cererii și predicția evenimentelor – demonstrând astfel aplicarea inter-industrială. Agnosticismul față de domeniul de aplicare al acestei inovații propuse este foarte promițător și sunt planificate lucrări suplimentare în acest domeniu în perioada următoare. Nu în ultimul rând, în contextul acestui rezultat, un accent deosebit a fost îndreptat atât către aspectul ecologic – **eficiența energetică** – precum și către **explicabilitatea experimentului** – unul dintre cele mai fierbinți subiecte din peisajul învățării automate profunde de astăzi.

Al doilea domeniu general de cercetare în care rezultatele noastre au fost validate fie prin cercetări și experimente publicate și/sau prin dezvoltare experimentală industrială – fără însă productizare concretă - este cea a analizei imaginii artificiale-scenă (de exemplu: captură de ecran a interfeței cu utilizatorul) și generarea de cod sursă (script). În acest domeniu special, am dezvoltat – ca un al treilea rezultat principal de contribuție proprie - un nou set de date care este disponibil public pentru experimentare, precum și o metodologie pentru extinderea potențială a acestui set de date propus. Pe lângă setul de date, munca noastră s-a concentrat pe găsirea de arhitecturi eficiente de grafuri aciclice direcționate care să atingă cu succes atât obiectivele de segmentare a imaginilor GUI, cât și să genereze cod sursă (sau script) de bază pentru transformarea directă în cod sursă – fără însă a avea logica de procese de business ce nu poate fi inferată din scenele statice.

3.5 Experimentare

În ceea ce privește experimentele specifice dezvoltării experimentale realizate în perioada studiilor doctorale, detaliile se pot regăsi atât în cadrul **Capitolului 3** dar mai ales în cadrul **Capitolului 4**. Principalele două tipuri de experimente au fost: cele care au legătură directă cu obiectivul principal al tezei – experimentele din zona analizei și inferenței scenelor artificiale reprezentate de imagini GUI sau alte formulare, precum și experimente specifice inovațiilor propuse care sunt total sau parțial agnostice față de obiectivele directe studiilor doctorale. Referitor la subiectul experimentelor specifice inovațiilor este necesar să amintim de utilizarea acestora în aplicații comerciale în diverse industrii. Pentru a menționa rapid principalele aplicații din industrie, trebuie să menționăm că avem atât cazuri de utilizare legate de imagistica în domeniile medical, siguranță, securitate, cât și cazuri de utilizare în domeniul analizei predictive.

Pentru prima categorie de experimente - cele legate de inferența imaginilor GUI - munca noastră, prezentată în rezultatele descrise de **Capitolul 3.4**, s-a concentrat pe evaluarea corectă a performanței arhitecturilor DAG propuse folosind datele colectate, precum și pe creșterea iterativă - calitatea și cantitatea - a setului de date *CloudifierNet*. Pe lângă măsurarea performanței în detectarea și localizarea artefactelor interfeței cu utilizatorul, o zonă secundară de experimente a vizat scalarea și eficiența calculului numeric paralel. Mai concret, am experimentat cu diferite dimensiuni de DAG – de la o dimensiune minimă până la o extindere de 2x, 3x a nodurilor și arcelor din graf - precum și diferite capacități de calcul, cum ar fi scalarea de la o infrastructură încorporată de 256 de nuclee numerice CUDA până la peste 4000-8000 de CUDA core-uri.

Model	ArtAcc	ArtRec	NatAcc	NatRec
Cloudifier50_1	*85.1%	*91.2%	*84.3%	*88.0%
Cloudifier50_2	*88.3%	*92.1%	*86.2%	*90.7%
Cloudifier109_1	*95.2%	*97.2%	*93.1%	*95.2%
Cloudifier109_2	*98.4%	*99.7%	*96.1%	*96.1%

Rezultatele comparative ale diverselor arhitecturi CloudifierNet folosind setul de date de antrenare, validare și testare al scenelor artificiale

În această clasă specială de experimente și în special legate de setul de date *CloudifierNet* propus, am creat, de asemenea, o serie de aplicații experimentale cu diverse instrumente de dezvoltare de tip RAD (Rapid Application Development) cu scopul principal de a genera interfețe vizuale de utilizator, precum și cod sursă de proiectare a interfeței cu utilizatorul. Mai concret, am proiectat atât aplicații vizuale bazate pe *win32*, cât și aplicații vizuale portabile compatibile cu POSIX, fără însă a avea vreun proces complex sau logică de afaceri, toate cu unicul scop de a genera scene artificiale pentru setul nostru de date - imagini (capturi de ecran) ale interfețelor potențiale de utilizator pentru diverse cazuri.

3.6 Rezumatul principalelor contribuții personale și propunerile de cercetare continua

În *Capitolul 5* și respectiv *Capitolul 6* sunt sumarizate atât principalele contribuții personale ale lucrării rezultate din cercetarea și experimentarea din perioada studiilor doctorale cât și propunerile de direcții de continuare a cercetării.

Direct legat atât de rezultatele experimentării, în special în zona cercetării arhitecturilor specifice grafurilor neurale, în *Capitolul 5* avem următoarea listă de contribuții principale la cercetare și inovare, precum și aplicații practice implementate în scenarii concrete din viața reală:

- i. ***Multi Gated Unit*** : Propunerea unei noi abordări inovatoare pentru găsirea structurii optime a topologiei grafurilor neurale. Cele două rezultate principale ale acestei contribuții de cercetare pot fi rezumate după cum urmează mai jos cu ***precizarea ca actualmente abordarea MGU este încă în studiu pentru găsirea de noi metode de eliminare a redundanței și micșorare a amprentei modelelor neurale:***
 - a. Eliminarea necesității abordărilor de căutare în spațiul discret al soluțiilor care ar necesita executarea a milioane de experimente pe infrastructura de calcul paralelă și reducând astfel drastic **amprenta de carbon** a procedurii de căutare a arhitecturii DAG-urilor.

- b. Capacitatea de autoexplicare a structurii interioare a modelului pe baza activărilor porții din fiecare unitate Multi Gated pentru întregul grafic.
- ii. ***Setul de date CloudifierNet*** : Publicarea Open Source a unui nou set de date care permite cercetarea și experimentarea în domeniul recunoașterii automate a conținutului interfeței cu utilizatorul. Setul de date poate fi utilizat într-o gamă largă de experimente, cum ar fi: antrenarea agenților pentru a recunoaște elementele interfeței cu utilizatorul în scopul de a simula/genera mesaje de aplicație, de a genera cod sursă de funcționalitate de bază pentru proiectarea aspectului sau chiar de a construi abordări generative (cum ar fi *rețele adverse generative*) care ar putea automatiza procesul de proiectare pe lângă crearea codului sursă. ***Aceasta este de asemenea o zonă în care dorim să creștem o comunitate de cercetători/practicieni care să ajute la dezvoltarea setului de date.***
- iii. ***Arhitecturi CloudifierNet***: propuse se bazează pe îmbunătățiri incrementale aplicate pe diverse modele și abordări state-of-the-art..
- iv. ***Aplicații din viața reală pentru domeniile medicală, securitate și siguranță***: Folosind arhitectura CloudifierNet propusă, în ultimii 2 ani, am reușit să dezvoltăm și să propunem sisteme la nivel funcțional de producție precum:
 - a. Sistem complet reglat automat bazat pe arhitectura CloudifierNet propusă pentru dermatologi și deducerea leziunilor dermatologice severe
 - b. Aplicație capabilă să efectueze analize de colposcopie (sau *cervigramă*) pentru ginecologie oncologică pentru a detecta potențiale leziuni de col uterin și gravitatea acestora împreună cu asistent în determinarea tipologiei zonei de transfer a colului uterin



- c. Sistem de analiza in timp real cu functionalitati plug-and-play pentru domeniul sigurantei si securitatii