

**UNIVERSITATEA DIN BUCUREȘTI FACULTATEA DE
MATEMATICĂ ȘI INFORMATICĂ**

LUCRARE DE LICENȚĂ

Coordonator:
Prof. Dr. Radu Ionescu

Absolvent:
Moldovan George-Alexandru

București
Iunie (fingers crossed), 2020

**UNIVERSITATEA DIN BUCUREȘTI FACULTATEA DE
MATEMATICĂ ȘI INFORMATICĂ**

Sistem pentru detectarea anomaliilor in video

Coordonator:
Prof. Dr. Radu Ionescu

Absolvent:
Moldovan George-Alexandru

București
Iunie (fingers crossed), 2020

Contents

1	Introducere	1
1.1	Motivatie	1
1.2	Context	2
1.3	Conținutul lucrării	3
2	Plictiseala	4

Abstract

Având în vedere contextul actual, detectarea anomaliilor în video este un subiect de interes în mai multe arii, în mod special în securitatea publică. Putem spune că această problemă este încă nerezolvată, deoarece sistemele actuale, deocamdată, nu depășesc omul când vine vorba de detectarea anomaliilor. De asemenea, o altă problemă a sistemelor de detectare a anomaliilor în video este nevoia acestora de resurse computaționale mari în partea de inferență, făcând aproape imposibilă rularea acestora direct pe hardware-ul existent al sistemelor de supraveghere video actuale, acolo unde acestea prezintă un maxim interes. Astfel, putem spune că dezvoltarea unui sistem capabil să transforme sistemele de supraveghere actuale în sisteme ce pot recunoaște evenimente anormale este un subiect ce poate revoluționa domeniul supravegherii video. Această lucrare își propune o implementare al sistemului state-of-the-art la momentul redactării, așa cum este prezentat de *Ionescu et al.* [5]. Obiectivul este obținerea unei arhitecturi ce folosește o soluție PaaS și expunerea etapei de inferență printr-un API astfel încât convertirea unui sistem de supraveghere clasic într-unul inteligent să devină doar o problemă de implementare, fără a fi nevoie de schimbarea hardware-ului. Utilizarea unei soluții PaaS pentru etapa de inferență rezolvă problema executării cererilor fără complexitatea creării și întreținerii unei infrastructuri de mașini virtuale sau fizice.

Chapter 1

Introducere

1.1 Motivatie

Detectarea anomaliilor in video este in strânsă legatură cu sistemele de supraveghere inteligente, un domeniu care a fost si este de interes pentru mine. La rândul lor, sistemele de supraveghere inteligente, au o mare importanță in securitatea publică. Cu toții ne dorim o lume in care apelurile de urgență in caz de incendiu se fac automat, alunecările de teren sunt descoperite înainte să fie prea târziu, iar oamenii rău intenționați sunt opriți înainte să se întâmple tragedii.

Astfel, arhitectura folosită se bazează pe detecția caracteristicilor spațio-temporale ale evenimentelor prezente în video, care mai apoi sunt împărțite in clase de normalitate. Aceste caracteristici sunt extrase trecând evenimentul printr-o serie de autoencodere pentru a folosi mai apoi reprezentarea latentă in cadrul clasificării finale. In etapa de antrenare, se folosesc filmări ce prezintă comportamentul normal in scenariul analizat. Analizând aceste video-uri putem crea un model capabil să recunoască dacă un eveniment aparține unei clase de normalitate analizate pana acum, sau dacă este un caz anormal. Deoarece un eveniment poate fii ales in multe moduri, în cadrul acestui sistem un eveniment reprezintă orice obiect aflat în cadru. Analiza asupra fiecarui obiect conține și un cadru precedente dar si unul viitor, asemănator cu modalitatea folosită de *Ionescu et al.* [5]. O mențiune in acest sens ar fi că deoarece pentru fiecare obiect sunt analizate si cadrele de la pozitie $t-3$ si $t+3$ respectiv la poziția t a cadrului analizat, atunci cand sistemul va analiza un video in sistem live-feed, analiza se va face cu un decalaj de 3 cadre. Având în vedere ca pentru un video actual viteza de redare este de minim 15 cadre pe secundă, acest decalaj este neglijabil.

Pe lângă partea algoritmica a detectării anomaliilor, o alta arie de interes a acestei lucrări este cloud computing. Această parte analizează un nou mod de rulare, ce facilitează atât dezvoltarea cât si execuția ulterioară a unor sisteme complexe. Acest nou mod constă in folosirea unei arhitecturi plasată în cloud, ce oferă dezvoltatorului posibilitatea să creeze sisteme ce necesită multe resurse in timpul rulării, fara costurile asociate creeri si menținerii unei infrastructuri proprii. Pe de altă parte, având in

vedere că toate operațiunile sunt executate în cloud, utilizatorii serviciului au nevoie doar de conexiune la internet și cerințe minime pentru sistemele proprii, fara a fi nevoiți să achiziționeze echipamente noi pentru a folosi sisteme de detecție a anomaliilor.

1.2 Context

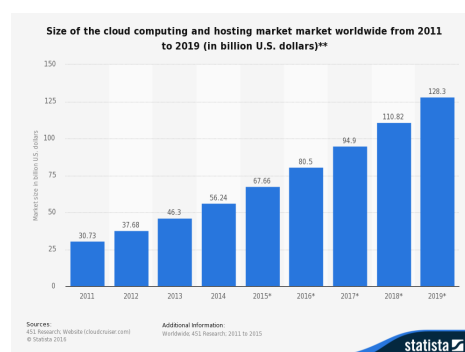
Detectarea anomaliilor in video poate fi văzută ca o problemă subiectivă, deoarece un eveniment este normal sau anormal doar dacă este luat în considerare și contextul în care acesta apare. Un exemplu foarte bun este lupta între două persoane si o persoană care se plimbă. Care dintre aceste evenimente este anormal ? Desigur, depinde de context. Dacă sistemul supraveghează o arena de lupte, atunci persoana care se plimbă in ring prezintă un comportament anormal, in timp ce luptătorii prezintă comportamentul așteptat. Din acest motiv majoritatea lucrărilor din domeniu [1, 5, 7]... abordează un mod de lucru bazat pe antrenarea folosind video-uri ce provin din aceeași locație cu cele de test. Tocmai din cauza dependenței de context, detectarea anomaliilor nu este o problemă ce poate fi generalizată, astfel fiecare scenariu necesită o antrenare si un model propriu.

Ca și moduri de expunere a soluției software catre utilizatori, aceasta se poate face in 2 moduri :

- Folosind servere proprii
- Folosind servicii cloud

Folosirea serverelor proprii presupune, pe lângă prezența fizică a serverelor și cumpărarea tuturor serviciilor conexe, cum ar fi servere de baze de date, sisteme de distribuire a fișierelor, infrastructură de rețea, ș.a.m.d., este nevoie si de o echipă dedicată pentru întreținerea infrastructurii si repararea eventualelor probleme ce pot apărea. Aceste considerente, împreuna cu faptul că scalarea soluțiilor software este foarte anevoioasă atunci când este folosită infrastructura proprie, fac această soluție să nu mai fie folosită în mod curent deoarece incetinește modul de dezvoltare a aplicației, iar rezultatul final este si el unul mai puțin calitativ decât ce se poate obține folosind soluții în cloud.

Folosirea serviciilor cloud oferă multiple posibilități de dezvoltare a aplicațiilor, de la creare unei infrastructuri complete care este administrată în totalitate de către dezvoltator, la medii de execuție serverless care sunt complet administrate de către provider-ul de servicii cloud. Deși crearea unei infrastructuri proprii înca este necesară acolo unde legislația nu permite ca datele sa



fie stocate în cloud, în toate celelalte cazuri se face migrarea spre soluții cloud, lucru vizibil și în evoluția cotei de piață a domeniului cloud-computing la nivel global, așa cum se poate observa și în 1.2.

1.3 Conținutul lucrării

Ca și structură, lucrarea este împărțită în 2 părți:

- Partea algoritmică a sistemului de detectare a anomaliilor în video
- Partea de deployment a sistemului

În prima parte sistemul va fi analizat și detaliat din punct de vedere algoritmic și teoretic studiind problema detectiei propriu zisă. Ca și tehnologii, în această parte am ales să folosesc Python3 ca și limbaj de programare pentru avantajele pe care le are. Printre acestea se numără faptul că este un limbaj orientat pe obiecte care pune la dispoziția dezvoltatorului numeroase librării specifice pentru AI/ML mutând astfel atenția dinspre detalii de implementare spre detalii de arhitectură și probleme mai abstracte ale programului care sunt cu adevărat importante pentru rezultatul final. Una dintre librăriile care s-a dovedit esențială dezvoltării este *Keras* [2], care oferă un API ce ușurează dezvoltarea unei rețele neuronale adânci/convoluționale dar și optimizează timpul de antrenare pentru modelele create.

În cea de a doua parte este analizat tipul de deployment al aplicației. Aici vor fi prezentate analize detaliate ale avantajelor și dezavantajelor soluțiilor cloud, dar și motivele pentru care modul final de deployment a fost ales.

Soluțiile de cloud-computing analizate vor fi:

- Infrastructure as a Service (IaaS) - analiză pe Amazon EC2
- Platform as a Service (PaaS) - analiză pe Amazon Elastic Beanstalk
- Function as a Service (FaaS) - analiză pe Amazon Lambda

Chapter 2

Plictiseala

In ceea ce privește FaaS, acesta este un domeniu nou, deoarece a apărut pentru prima dată în 2010 fiind oferit de câteva start-upuri la acea vreme. Acest mod de dezvoltare orientat spre microservicii a devenit trendul în industrie în ultimii ani pentru sisteme cu potențial de scalare mare, deoarece prezintă numeroase avantaje din punct de vedere al modului de dezvoltare și de execuție în industrie. În momentul de față, pentru servicii de tip FaaS sunt 3 mari jucători: Amazon cu AWS Lambda, Google cu Google Cloud Functions și Microsoft cu Azure Functions.[6]. Numeroase lucrări din domeniu [3, 8] arată că rularea algoritmilor de machine learning folosind soluții FaaS (Function as a service) precum AWS Lambda sau Google cloud functions, este în sine o problemă ce necesită soluții de optimizare a codului pentru a îndeplini restricțiile soluțiilor de rulare serverless, cum ar fi memoria limitată a mediului de execuție. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not

at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Bibliography

- [1] K. Cheng, Y. Chen, and W. Fang. Video anomaly detection and localization using hierarchical feature representation and gaussian process regression. pages 2909–2917, 2015.
- [2] F. Chollet et al. Keras description, 04 2020.
- [3] A. Christidis, R. Davies, and S. Moschogiannis. Serving machine learning workloads in resource constrained environments: a serverless deployment example. pages 55–63, 2019.
- [4] G. C. Fox, V. Ishakian, V. Muthusamy, and A. Slominski. Status of serverless computing and function-as-a-service (faas) in industry and research. *arXiv preprint arXiv:1708.08028*, 2017.
- [5] R. T. Ionescu, F. S. Khan, M. Georgescu, and L. Shao. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. pages 7834–7843, 2019.
- [6] E. Jonas, J. Schleier-Smith, V. Sreekanti, C.-C. Tsai, A. Khandelwal, Q. Pu, V. Shankar, J. Carreira, K. Krauth, N. Yadwadkar, et al. Cloud programming simplified: A berkeley view on serverless computing. *arXiv preprint arXiv:1902.03383*, 2019.
- [7] W. Sultani, C. Chen, and M. Shah. Real-world anomaly detection in surveillance videos. pages 6479–6488, 2018.
- [8] H. Wang, D. Niu, and B. Li. Distributed machine learning with a serverless architecture. pages 1288–1296, 2019.