Vote&Go: Aplicație mobilă pentru

recunoașterea datelor personale

din cartea de identitate

Proiect realizat în cadrul materiei *Codesign Hardware-Software*

Autori:

Enache Alexandru

Teodorescu Iulia-Maria

Capitolul 1: Introducere

1. **Contextul problemei**

În secolul al XXI-lea tehnologia continuă să ne deschidă orizonturi la care precursorii noștri puteau numai să aspire. Evoluția rapidă și fără precedent în acest domeniu a condus la digitalizarea lumii pe care o știam, totodată cauzând și majore schimbări sociale și psihologice. Viața a căpătat un ritm mai alert, omul țintește spe a fi cât mai rapid și cât mai eficient posibil în desfășurarea activităților sale cotidiene, fie ele în plan personal sau profesional.

Pe deasupra, nivelul de dezvoltare a tehnologiei se dovedește mai mult decât util și în contextul actual al problemei mondiale din domeniul sănătații publice datorat pandemiei Covid-19. Distanțarea fizică reprezintă una dintre măsurile de prevenție și stopare a transmisiei virusului și este facilitată de funcționalitățile puse la dispoziția noastră prin intermediul dispozitivelor electronice pe care le deținem.

Înclinația oamenilor spre tehnologie și eficiență este admirabilă, însă parerea noastră – autorii proiectului – este că viteza la care se desfășoară viața în zilele noastre este sursa unei considerabile părți din stresul resimțit zilnic. Chiar mai mult, angoasa noastră este alimentata în această perioadă de situația pandemiei, cu toții încercând sa ne ferim într-o măsură cât mai mare de virus.

Noi – autorii – intenționăm să scăpăm oamenii de una dintre neliniști, facilitănd distanțarea fizică în contextul participării la vot. Atât în urma alegerilor locale din luna septembrie a acestui an, cât și a celor parlamentare recente am observat cum teama și neîncrederea în respectarea măsurilor sanitare au determinat oamenii să nu meargă la vot. Televiziunile de știri au comunicat informații care au înspăimântat, care nu au crescut încrederea în stat, astfel cauzând electoratul să înghețe în casă [1]. În plus, oamenii au fost vocali cu opțiunea lor, astfel influențându-i într-o anumită măsură și pe cei care inițial nu și-au pus astfel problema.

1. **Descrierea soluției noastre**

În ultimii ani am depășit deja stadiul în care ne bazam numai pe PC-uri și desktop-uri, datorită smartphone-urilor care ne oferă o multitudine de funcționalități mereu la îndemână. Prin urmare, având in minte problema anterior prezentată, noi propunem o soluție care începe să ia avânt, și anume o aplicație mobilă de OCR Text Recognition.

**Ce înseamnă OCR?** Optical Character Recognition reprezintă conversia imaginilor ce conțin text (print sau scris de mână) provenite prin scanarea unui document, fotografierea unui afiș sau chiar subtitrarea unui film în text editabil. Această tehnică face parte din dimeniul inteligenței artificiale (computer vision, pattern recognition).

Scopul nostru este să asigurăm respectarea distanțării fizice în secțiile de votare prin diminuarea numărului de membri ai comisiei. Folosind aplicatia noastră, utilizatorul nu mai este nevoit să interacționeze cu alte persoane. Acesta își confirmă singur prezența la vot, conversia datelor personale realizându-se repede cu ajutorul propriului smartphone.

Aplicația noastră pune la dispoziție un ecran cu instrucțiuni de folosire, cu toate că interfața este minimalistă, clară și ușor de folosit. Utilizatorul fotografiază din aplicație cartea de identitate. Datele personale vor fi extrase și transformate în text editabil. Acestea pot fi vizualizate sau introduse automat într-o declarație de votare în format PDF prin care votantul se legitimează și își consemnează prezența.

Capitolul 2: State of the art

1. **Metode de Machine Learning**

Conversia imagine-text editabil (OCR) se poate realiza prin următoarele metode:

* Feature Extraction: Aceasta metodă ne spune că un caracter are anumite trăsături după care poate fi recunoscut precum înălțimea, lățimea, numărul de bucle, numărul de linii, intersecții ale liniilor etc. [2];
* Neural Networks: Această strategie presupune eșantionarea pixelilor imaginii și apoi potrivirea ei cu un model cunoscut al unui caracter [2].

1. **Aplicații mobile asemănătoare**

În urma studierii problemei am observat că aplicațiile de acest fel se încadrează în două categorii: software care realizează conversia din imagine în text editabil și software care pe lângă această conversie validează identitatea utilizatorului. Având în vedere faptul că aplicatia propusă de noi nu este prevăzută cu o funcție de confirmare a identității după scanarea cărții de identitate, nu vom insista pe acest procedeu.

Analiză comparativă

BlinkID este software-ul care ne-a inspirat și putem spune că modul de funcționare al aplicației noastre este asemănător cu al acestuia: fotografierea actului de identitate cu camera telefonului și extragerea datelor personale. Calitatea și performanța BlinkID reies din următoarele însușiri:

* viteza mare de conversie a textului (0.4 s);
* nu ține cont de poziția actului de identitate în imagine;
* oferă suport pentru mai multe tipuri de documente de identificare (carte de identitate, pașaport, permis de conducere);
* oferă suport pentru mai multe țări (formatul fizic al documentului din țara respectivă).

Spre deosebire de BlinkID, aplicația noastră nu prezintă caracteristicile enumerate mai sus; conversia textului se face în timp real, însă nu cu asemenea rapiditate. Pe de altă parte, soluția noastră Vote&Go oferă posibilitatea conversiei în format PDF.

Tabelul 1 – Comparație d.p.d.v. al funcționalitătilor între aplicații de pe piață și aplicația realizată de noi

|  |  |  |  |
| --- | --- | --- | --- |
|  | BlinkID | identity autoID | Vote&Go |
| Link store | [Play Store](https://play.google.com/store/apps/details?id=com.microblink.blinkidapp&hl=en_US&gl=US) | [Play Store](https://play.google.com/store/apps/details?id=de.identity.autoidapp&hl=en_US&gl=US) | - |
| Notă store | 4,5 | 4,6 | - |
| Număr instalări | 100.000+ | 100.000+ | - |
| Număr ratings | 976 | 6,025 | - |
| Ads/ in-app purchases |  |  |  |
| Login/user |  |  |  |
| Suport text printat | x | x | x |
| Suport text scris de mână |  |  |  |
| Editare text | x |  | x |
| Conversie în PDF |  |  | x |
| Suport pentru mai multe limbi | x |  |  |
| Suport pentru mai multe documente de identificare | x | x |  |
| Conversie în timp real | x | x | x |

Capitolul 3: Design și implementare

1. **Procesul de Machine Learning**
2. **Prelucrarea setului de date**

Pentru antrenarea rețelei neuronale am preluat setul de date The Chars74K [3]. Acesta a venit deja clasat pentru fiecare caracter dintre cele 62 de caractere din limba engleză. Fiecare director conține gruparea de poze pentru fiecare caracter.

Text

Description automatically generated Text, letter

Description automatically generated

Fig. 1 – Setul de date clasat pe caractere Fig. 2 – Conținutul unui director

Icon

Description automatically generatedA picture containing arrow

Description automatically generatedIcon

Description automatically generatedA picture containing text

Description automatically generatedIcon

Description automatically generatedIcon

Description automatically generatedIcon

Description automatically generated

Fig. 3 – Exemple de imagini

Noi am împarțit setul de date în 3 directoare pentru o clasificare mai ușoară.

Text

Description automatically generated

Fig. 4 – Împărțirea setului de date

Fiecare director conține toate clasele corespunzătoare grupării respective:

* *digits* conține 10 clase (0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
* *lowercase* conține 26 de clase (a, b, c, d, e, f, g, h, I, j, k ,l m, n, o, p, q, r, s, t, u, v, w, x, y, z);
* *uppercase* conține 26 de clase (A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z).

A picture containing text

Description automatically generated A picture containing text

Description automatically generated A picture containing logo

Description automatically generated

Fig. 5 – Gruparea finală a setului de date

1. **Antrenarea modelului**

Am format doi vectori pentru antrenare: *images* și *classNo*, unde fiecare imagine a fost redimensionata pentru a fi (32, 32), iar poziția corespunzătoare fiecărei imagini din *images* va avea clasa din care face parte în vectorul *classNo*.

Am împărțit setul de date pentru antrenare și testare astfel: 90% pentru antrenare și 10% pentru testare. Ulterior am scos 9% din setul de antrenare pentru validare avand într-un final 81% pentru antrenare, 10% pentru testare și 9% pentru validare.

În clasificare modelul nostru încearcă să găsească o ecuație pentru a determina y pentru un x dat. În cazul nostru axa X este formată din poze iar axa Y este clasa corespunzătoare fiecarei imagini.

Pentru antrenare fiecare imagine va fi redimensionata în (32, 32) și ulterior transformată într-un array, fiecare valoare din array fiind adusă la o valoare între [0, 1]. Deasemenea antrenarea se va face în 30 de epoci (epochs\_val = 30), unde o epoca va conține 512 pași (steps\_per\_epoch = 512) iar un pas va contine o grupare de 32 de poze(batch\_size\_val = 32).

1. **Folosirea modelului**

Pentru a folosi modelul ne vom lua o imagine pe care vrem să detectăm caracterele și o vom modifica pentru a obține threshold-ul acesteia.

Text

Description automatically generated

Fig. 6 – Threshold-ul folosit

Transformând imaginea în alb-negru, iar apoi obținând threshold-ul acesteia ajungem la o reprezentare mai bună a caracterelor negre pe un fundal alb, deoarece culorile care sunt mai aproape de negru vor deveni negru, iar cele mai aproape de alb vor deveni alb.

Am scos fiecare contur aplicând metoda *findContours* pe treshold-ul imaginii pe care o avem ca și input, fiecare contur a fost după aceea filtrat pentru a obține conturile care ne interesau pe noi, și anume cele din jurul caracterelor.

Apoi am sortat contururile pentru a obține contururile ordonate de la stanga la dreapta și de sus în jos folosind ecuația: ***x + y \* w***, unde x, y sunt poziția coordonatelor conturului detectat iar w este lățimea pozei. Acest lucru se traduce astfel în cod:



Fig. 6 – Ecuația transpusă în cod

Am folosit fiecare contur nefiltrat pentru a obține un ROI (Region Of Interest), ulterior făcând din fiecare contur o imagine separată pe care am trimis-o ca intrare retelei neuronale pe care o antrenăm pentru a face clasificarea. Pozele au fost salvate într-un fișier temporar iar după clasificarea acestora fișierul a fost șters pentru a avea cat mai puțin spațiu de memorie folosit.

1. **Aplicația mobilă**

Aplicația mobilă crossplatform este realizată în C# folosind Xamarin.Forms. Arhitectura soluției cuprinde trei proiecte:

* VoteAndGo: conține codul comun celor trei proiecte și se bazează pe specificațiile standardului .NET;
* VoteAndGo.Android: conține Mono SDK și toate API-urile folosite; inițializarea Xamarin.Forma se face în MainActivity.cs, după care tot de aici se intră în codul comun;
* VoteAndGo.iOS: conține toate API-urile folosite; inițializarea Xamarin.Forms se face în AppDelegate.cs, după care tot de aici se intră în codul comun.

Am folosit Shell pentru navigarea în aplicație; câteva avantaje ale acestuia sunt:

* structura declarativă a aplicației;
* navigarea simplă între ecrane;
* rutele sunt șiruri de caractere.

Meniul root al unei aplicații ce folosește Shell poartă numele de Flyout și este accesibil printr-un buton de tip burger sau prin glisarea degetului pe ecran de la stânga la dreapta. Acest tip de meniu are un Header care la noi e reprezentat de logo-ul aplicației Vote&Go și mai multe item-uri care direcționează utilizatorul spre ecranele *About, Home* și *Scan ID Card.*

Design-ul pentru ecrane(UI) este cuprins în fișierele \*.xaml (Extensive Application Markup Laguage). UI este definit declarativ. Aici se realizeaza legătura cu proprietăti, comenzi și funcționalități din fisierele ViewModel.cs care reprezintă așa zisul „creier” al aplicației.

Capitolul 4: Utilizare

1. **Elemente UI**

Aplicația Vote&Go pune la dispoziție următoarele ecrane:

* ecranul de pornire denumit *About* în meniu;
* ecranul meniului;
* ecanul cu instrucțiuni de utilizare denumit *Help*;
* ecranul in care se realizează scanarea și afișasea datelor personale.

În aplicația noastră se regăsesc următoarele elemente UI (de interfață om-dispozitiv):

* buton de tip burger pentru accesarea meniului;
* trei item-uri pentru meniu, fiecare conducând la un anumit ecran;
* butonul pentru scanarea actului de identitate;
* editorul in care apar datele extrase din cartea de identitate(datele pot fi modificate manual).

1. **Descrierea utilizării**

La deschidere, aplicația noastră pornește cu ecranul About unde e prezent logo-ul și cateva informații sugestive referitoare la scopul ei.

Meniul poate fi accesat din butonul burger prezent pe ecran sus, în stânga sau prin glisarea cu degetul pe ecran de la stânga la dreapta. Acesta conține opțiunile About, Help și Scan ID Card.

Ecranul Help conține instrucțiunile de utilizare a aplicației pas cu pas.

Ecranul Scan ID Card conține editorul de text în care vor apărea datele personale în urma scanării cărții de identitate. Sub acesta este butonul pentru scanare. În josul ecranului vor fi afișate cele doua imagini: fața și spatele cărții de identitate.

Graphical user interface, application

Description automatically generated A picture containing graphical user interface

Description automatically generated

Fig. 7 – Meniul aplicației Fig. 8 – Ecranul pentru scanare

Capitolul 5: Rezultate și concluzii

Am observat ca pentru aplicația noastră nu avem nevoie de un set de date pentru literele mici, așa că setul nostru de date va fi format din totalul pozelor pentru cifre, respectiv pentru literele mari. Am obținut un total de 38556 poze și un total de 36 de clase unde distribuția pe clase a pozelor este următoarea:

Chart, bar chart

Description automatically generated

Fig. 9 – Distribuția imaginilor pe clase

Am antrenat rețeaua neuronală pe un GTX 960M și am obținut o acuratețe de 94.55% unde graficele de acuratețe și pierdere sunt următoarele:

Chart, line chart

Description automatically generatedChart

Description automatically generated

Fig. 10 – Graficul de acuratețe Fig. 11 – Graficul de pierdere

Capitolul 6: Documentare

Referințe

[1] Cristache, I. Interviu cu Mircea Dinescu. România9. TVR1. 2020

<https://www.youtube.com/watch?v=TlBkD4ItuWM&ab_channel=Romania9>

[2]Badawy, W. ”Automatic License Plate Recognition (ALPR): A State of the Art Review”. 2012

[3] <http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/>