

Vedere artificială pentru Vehicule
Documentație
Quizzzy [ZipGrade-like app]

Botescu Mihai: `mihai.botescu00@e-uvt.ro`
Beze Robert: `robert.beze00@e-uvt.ro`

2 iunie 2022

Rezumat

Aplicația presupune 2 categorii de input-uri de la utilizator, mai precis imaginea ce conține un quiz cu răspunsurile corecte și mulțimea quiz-urilor care trebuie scanate. Se extrag răspunsurile selectate din fiecare quiz, și se compară cu cele corecte și, în funcție de aceste aspecte, se calculează scorul obținut de către fiecare student în parte.

Cuprins

1	Introducere	2
1.1	Definirea problemei	2
1.2	Cazuri de utilizare ale aplicației	2
1.2.1	Cazuri de utilizare din perspectiva profesorului	2
1.3	Schimburi de mesaje între utilizatorii aplicației	2
1.4	Stocarea datelor	2
1.5	Ingineria cerințelor	4
1.5.1	Cerințele propuse pentru acest proiect	4
1.5.2	Contribuții personale	4
2	Tehnologii utilizate	5
2.1	Descrierea stivei de tehnologii	5
2.2	Descrierea metodelor utilizate	5
2.2.1	Grayscale	5
2.2.2	Gaussian blur	5
2.2.3	Canny	6
2.2.4	Bird's eye view (perspective)	6
2.2.5	Threshold	6
2.2.6	Scaling	6
3	Testarea aplicației	7
4	Concluzii și direcții viitoare	10
	Bibliografie	11

Capitolul 1

Introducere

1.1 Definirea problemei

Aplicația conține un singur actor, și anume profesorul. Profesorul are nevoie de o foaie scanată conținând răspunsurile corecte, și de foile cu răspunsuri ale studenților. Aplicația realizează recunoașterea și extragerea datelor (răspunsurilor selectate) din fiecare imagine, și le compară cu input-ul oferit de către profesor (mulțimea de întrebări însoțite de răspunsurile corecte).

1.2 Cazuri de utilizare ale aplicației

Aplicația va avea, la cel mai înalt nivel, din perspectiva utilizatorului și a modului de lucru cu aceasta, un principal actor, anume profesorul. Astfel, putem identifica un scenariu generic de utilizare din perspectiva profesorului.

1.2.1 Cazuri de utilizare din perspectiva profesorului

Diagrama de mai jos 1.1 reprezintă (sumar) etapele pe care le va parcurge studentul în utilizarea aplicației.

1.3 Schimburi de mesaje între utilizatorii aplicației

Nu există schimburi de mesaje între utilizatorii aplicației, din moment ce actorul este unul singur, adică profesorul.

1.4 Stocarea datelor

Stocarea datelor are loc local, în directorul proiectului, la același nivel, în 3 foldere diferite, mai precis: answers (pentru grila cu răspunsurile corecte), grading (pentru grilele de răspunsuri completate de studenți) și final (pentru grilele de

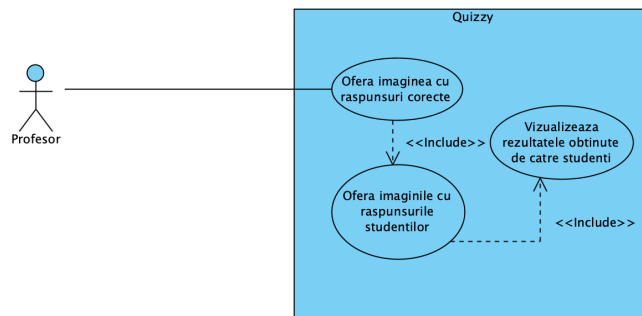


Figura 1.1: Diagrama cazurilor de utilizare din perspectiva profesorului

răspunsuri corectate automat, pe baza răspunsurilor oferite de studenți), așa cum se vede în figura 1.2.

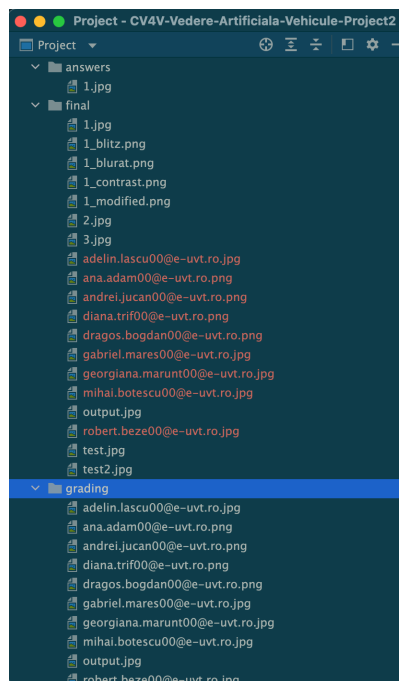


Figura 1.2: Modalitatea de stocare a datelor

1.5 Ingineria cerințelor

1.5.1 Cerințele propuse pentru acest proiect

1. Realizarea recunoașterii răspunsurilor din grilele studenților
2. Realizarea recunoașterii răspunsurilor din grila transmisă de profesor
3. Realizarea notării automate a grilelor studenților
4. Tratarea cazurilor speciale
 - Imagine foarte blurată
 - Imagine cu contrast scăzut
 - Imagine cu blitz foarte puternic
 - Imagine cu luminozitate scăzută
 - Imagine cu contrast înalt
 - Imagine rotită
5. Exportarea datelor în format CSV/JSON

1.5.2 Contribuții personale

Contribuțiile autorilor sunt listate mai sus, exceptând punctul de plecare, adică primul element enumerat.

Capitolul 2

Tehnologii utilizate

2.1 Descrierea stivei de tehnologii

1. **Limbaaj de programare:** Python3
2. **Librării externe:** opencv [1] [2] [3] (pentru detecție imagini), PIL [4] (pentru corecție imagini), numpy [5] (pentru procesarea imaginilor ca array-uri de pixeli)

2.2 Descrierea metodelor utilizate

2.2.1 Grayscaleing

Transformări în spațiul RGB [1], cum ar fi adăugarea/eliminarea canalului alfa, inversarea ordinii canalelor, conversia la/de la culoare RGB pe 16 biți (R5:G6:B5 sau R5:G5:B5) [5], precum și conversia în/din tonuri de gri folosind:

$$f([R, G, B]) = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

2.2.2 Gaussian blur

În această metodă, în loc de un filtru box, se folosește un kernel tip gaussian [2]. Ar trebui să specificăm lățimea și înălțimea nucleului care ar trebui să fie pozitive și impare. De asemenea, ar trebui să specificăm abaterea standard în direcțiile X și Y , respectiv σ_X și σ_Y [2]. Dacă este specificat doar σ_X , σ_Y este considerat la fel ca σ_X . Dacă ambele sunt 0, acestea sunt calculate din dimensiunea kernel-ului [3]. Bluring-ul gaussian este foarte eficient în eliminarea zgomotului gaussian dintr-o imagine.

2.2.3 Canny

OpenCV pune toate cele de mai sus într-o singură funcție, `cv.Canny()` [1]. Primul argument este imaginea dorită. Al doilea și al treilea argument sunt *minVal* și, respectiv, *maxVal* [3]. Al patrulea argument este `aperture_size`. Este dimensiunea nucleului Sobel folosit pentru găsirea gradientilor de imagine. În mod implicit, este 3. Ultimul argument este *L2gradient* care specifică ecuația pentru găsirea mărimii gradientului. Dacă este adevărat, folosește ecuația menționată mai sus care este mai precisă, în caz contrar folosește această funcție: $Edge_Gradient(G) = |Gx| + |Gy|$ [2]. Am utilizat Canny pentru edge detection, în aplicația noastră.

2.2.4 Bird's eye view (perspective)

Bird's eye view (BEV) [3] este o vedere ridicată a unui obiect de sus, cu o perspectivă ca și cum observatorul ar fi o pasăre. Am folosit această metodă pentru a putea să centram perspectiva asupra grilei de răspunsuri.

2.2.5 Treshold

Am utilizat treshold [2] pentru a putea recunoaște cu ușurință mulțimea tuturor răspunsurilor selectate la fiecare întrebare în parte, în quiz.

2.2.6 Scaling

Am utilizat scaling pentru a realiza mărirea luminozității imaginii în cazul în care aceasta este prea întunecată. Ne-am folosit de următoarea formulă, deoarece este nevoie de 2 parametri, α și β [2] [3]:

$$ratio = \frac{brightness}{minimum - brightness}, \alpha = \frac{1}{ratio}, \beta = 255 \cdot |1 - \alpha|$$

,iar

$$im_{nou}[x, y, z] = \alpha \cdot im[x, y, z] + \beta.$$

Capitolul 3

Testarea aplicației

S-au realizat multiple teste, pe cazurile speciale propuse mai sus. Figurile 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7 sugerează diferite cazuri speciale în care pot ajunge imaginile, în viața reală.

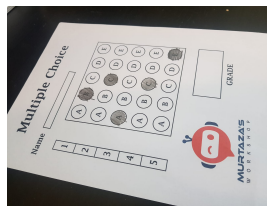


Figura 3.1: Imagine în condiții normale

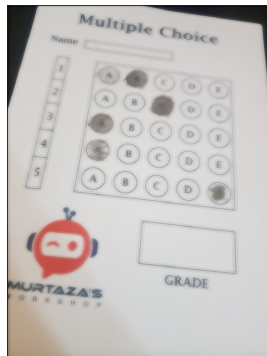


Figura 3.2: Imagine foarte blurată

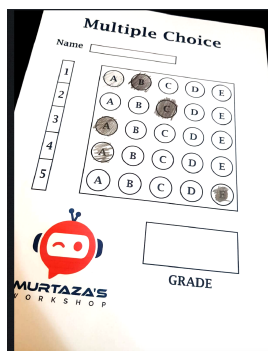


Figura 3.3: Imagine cu contrast scăzut

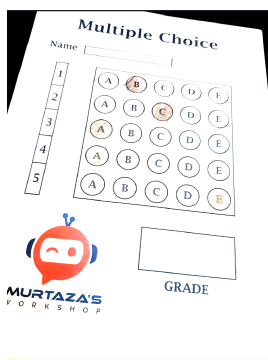


Figura 3.4: Imagine cu blitz foarte puternic

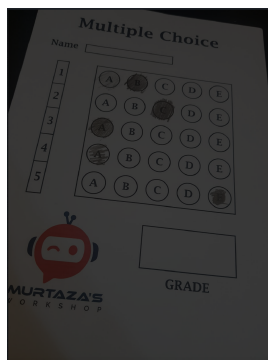


Figura 3.5: Imagine cu luminozitate foarte scăzută

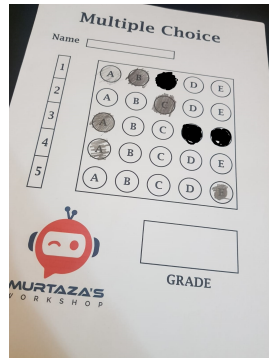


Figura 3.6: Imagine cu contrast înalt

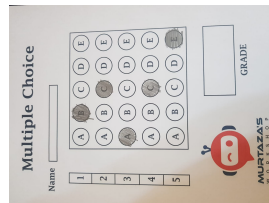


Figura 3.7: Imagine rotită

Capitolul 4

Concluzii și direcții viitoare

Ca și direcții viitoare de realizare, se dorește în primul rând extinderea aplicației spre utilizarea acesteia în mediul de web (deci crearea unui server pe care datele sunt stocate, un UI pentru interacțiunea dintre funcționalitățile aplicației și utilizator), precum și oferirea de portabilitate acesteia.

Mai apoi, se dorește identificarea și rezolvarea a câtor mai multe cazuri posibile care pot apărea în momentul fotografierii grilelor.

De asemenea, posibilitate de extindere pentru a funcționa și pe un alt șablon decât cel oferit în momentul de față.

Se dorește și posibilitatea și de recunoaștere a email-ului studentului din cadrul imaginilor scanate.

Bibliografie

- [1] “Open cv documentation for c/c++/python3.”
- [2] J. Minichino and J. Howse, *Learning OpenCV 3 Computer Vision with Python* -. Birmingham, England: Packt Publishing, 2 ed., Sept. 2015.
- [3] G. R. Bradski and A. Kaehler, *Learning OpenCV 3*. Sebastopol, CA: O’Reilly Media, Jan. 2017.
- [4] S. Dey, *Python Image Processing Cookbook*. Birmingham, England: Packt Publishing, Apr. 2020.
- [5] N. Community, “Numpy user guide.”