UNIVERSITATEA TEHNICA “GH. ASACHI”

FACULTATEA DE AUTOMATICA SI CALCULATOARE

**DOCUMENTATIE PROIECT FINAL LA   
SISTEME DE VEDERE ARTIFICIALA**

STUDENT: BUTACU IONEL-CATALIN

GRUPA: 1305B

SPECIALIZAREA: AUTOMATICA SI INFORMATICA APLICATA

**Cuprins:**

Capitolul 1: Introducere

* Dezvoltaresi introducere in temativa
* Scopul aplicatiei si functionalitatile sale

Capitolul 2: Interfata aplicatiei

* Prezentarea interfetei aplicatiei
* Prezentarea functionalitatilor utilizatorului

Capitolul 3: Captura si procesarea imaginii

* Procesul de detectare si prelucrare
* Descrierea si corpul functiei

Capitolul 4: Captura si procesarea imaginii in timp real

* Algoritmul de procesare in timp real

Capitolul 5: Documentatie si resurse biografice

Capitolul 1: Introducere:

Dezvoltaresi introducere in temativa

**Documentatie proiect:**

Acest document contine codul si logica de functionare a aplicatiei, precum si documentatia necesara pentru intelegerea proiectului.

**Dezvoltare:**

Mediul de dezvoltare App Designer din MATLAB a fost esențial în dezvoltarea acestui proiect. Acesta a permis crearea rapidă și ușoară a interfeței grafice de utilizator, inclusiv a obiectelor grafice și a funcționalităților de bază, cum ar fi captura și procesarea imaginilor. App Designer a permis usoara integrarea/dezvoltare a metodelor propii pentru a adăuga funcționalități specifice, cum ar fi initierea a sistemului de achizitie sau procesarea imaginii achizitionate.

De asemenea, am folosit App Designer pentru a integra aceste functii in interfata grafica si pentru a adauga functionalitati specifice, cum ar fi selectarea formei de blurare si a nivelului de blur. In general, am adoptat o abordare eficienta si intuitiva in dezvoltarea aplicatiei pentru a creste experienta utilizatorului.

**Tema proiectului:**

Realizarea unui sistem de vedere artificială care să permită identificarea zonelor din imagini care sunt fețe umane si blurarea acestora.

**Etapele proiectului:**

1. Realizarea unui GUI pentru achiziție de imagine in timp real
2. Identificarea zonelor tip față umana
3. Blurarea zonelor detectate
4. Afisarea rezultatului in interfata grafică

Capitolul 1: Introducere:

Scopul aplicatiei si functionalitatile sale

**Scopul aplicației:**

Scopul acestei aplicații este de a permite utilizatorilor să captureze/achiziționeze imagini în timp real sau să încarce imagini existente pentru a detecta și estompa zonele faciale din imagini, oferind astfel utilizatorului o modalitate de a proteja confidențialitatea persoanelor din imagini.

**Funcțiile principale ale aplicației:**

Caracteristicile principale ale aplicației sunt:

Caracteristicile principale ale aplicației sunt urmatoarele:

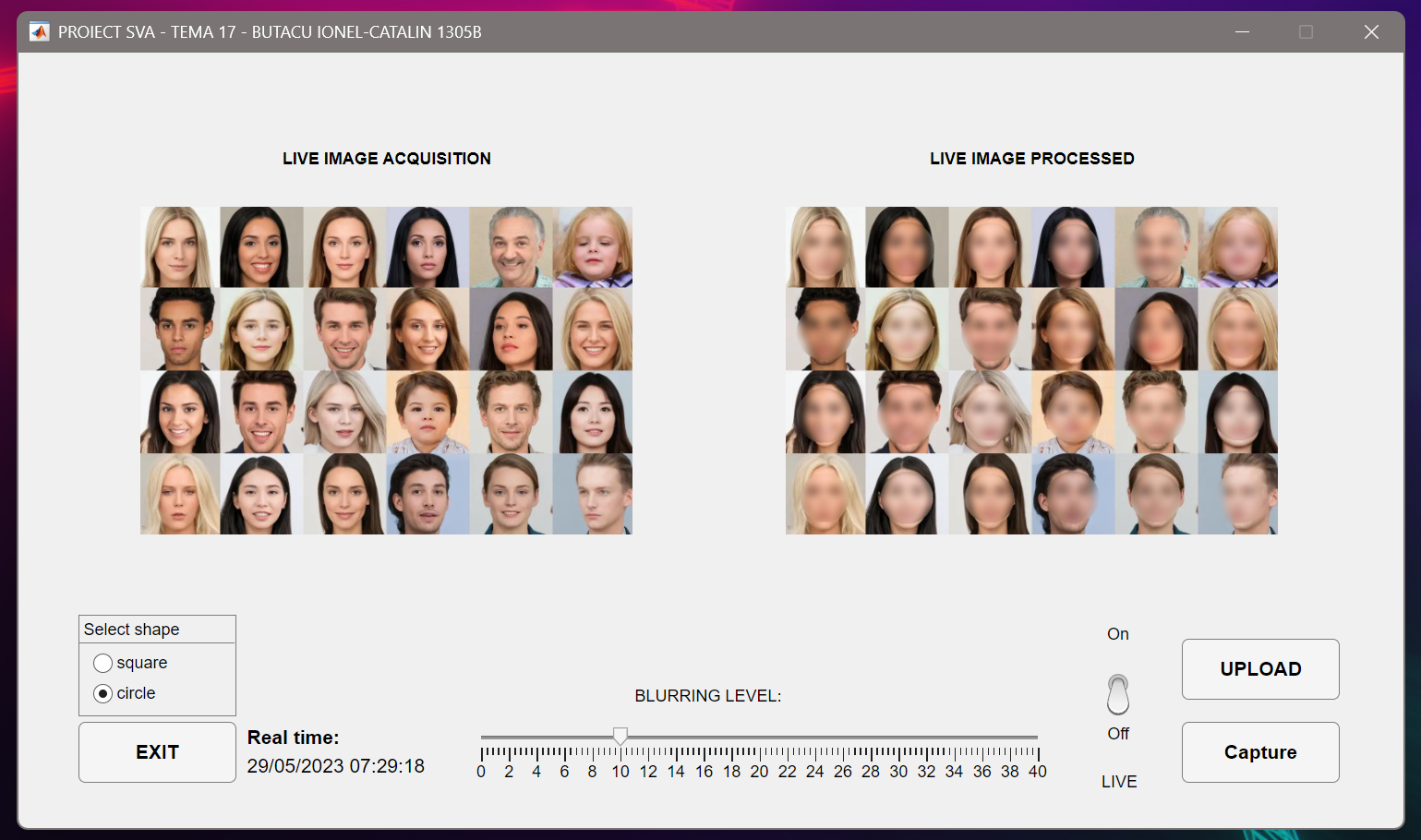
* Achizionarea de imagini în timp real de la o cameră web sau o camera externa, configurarea pentru o camera externa facandu-se din codul sursa
* Detectarea/Identificarea zonelor ce conțin fețe umane în imagini folosind un algoritm de detectare a feței prevazut de COMPUTER VISION TOOLBOX
* Selectarea formei de estompare (pătrat sau cerc) și a nivelului de estompare/blurare printr-un slider interactiv utilizatorului
* Estomparea zonelor detectate folosind un filtru de gaussian, aplicarea efectului de blur
* Afișarea imaginii procesate în interfața grafică cu utilizatorul

Aceste funcționalități combină capturarea în timp real a imaginilor, detectarea fețelor umane, estomparea zonelor respective și afișarea rezultatului într-o interfață intuitivă pentru utilizator. Prin urmare, utilizatorul poate proteja confidențialitatea persoanelor din imaginile capturate sau încărcate, oferindu-i un control asupra modului în care fețele umane sunt estompate.

Capitolul 2: Interfata aplicatiei

Prezentarea interfetei aplicatiei

\*poze aplicatie, mod utilizare\*



Capitolul 2: Interfata aplicatiei

Prezentarea functionalitatilor utilizatorului

**Funcționalități**:

*setupLiveAcquisition*

Această funcție inițializează obiectul camerei web și setează un timer pentru achiziționarea continuă a imaginilor. Funcția este apelată în metoda startupFcn pentru configurarea aplicației la pornire.

*updateLive*

Această funcție este apelată de timer pentru actualizarea imaginilor în timp real. Verifică dacă modul LIVE este activat și aplică efectul de blur corespunzător formei selectate. De asemenea, actualizează eticheta de afișare a timpului curent.

*photoProcessing*(2)

Această funcție primește o imagine și aplică efectul de blur selectat asupra fețelor umane detectate în imagine. Nivelul de blur este determinat de poziția glisorului BLURRINGLEVELSlider. Funcția returnează imaginea prelucrată.

*captureImage*

Această funcție achiziționează o imagine de la camera web și o afișează în zona corespunzătoare. Apoi, apelează funcția photoProcessing pentru a aplica efectul de blur.

*uploadImage*

Această funcție permite utilizatorului să încarce o imagine din fișier și să o afișeze în zona corespunzătoare. Apoi, apelează funcția photoProcessing pentru a aplica efectul de blur.

Capitolul 3: Captura si procesarea imaginii

Procesul de detectare si procesare

Algoritmul de detectare a feței folosit în acest proiect este implementat în funcția "photoProcessing" și se bazează pe toolbox-ul de Computer Vision din MATLAB.

În prima instanta, poza este convertită la grayscale prin utilizarea funcției "rgb2gray", pentru a simplifica procesul de identificare a trăsăturilor. Detectorrea de obiecte se realieaza folosind functia "CascadeObjectDetector", ce are by default parametrul de 'FrontalFaceCART' si dtectează fețele care sunt în poziție verticală și cu fața spre camera. Implementarea se face prin intermediul "vision.CascadeObjectDetector", pentru a extrage regiunile din imagine în care se găsesc fețe umane.

În acest proces, detectorul utilizează un set de caracteristici Haar, care sunt selectate dintr-un set de imagini pozitive și negative și încărcate într-un model de clasificator. Detectorul utilizează aceste caracteristici pentru a determina dacă o anumită regiune din imagine conține sau nu o față umană.

Dacă detectorul găsește o față umană, acesta returnează o casetă încadratoare (bounding box), care încadrează regiunea din imagine care conține fața umană. Această casetă încadratoare este stocată în variabila "bboxes".

În final, algoritmul iterază prin fiecare casetă încadratoare și aplică un filtru Gaussian de reducere a zgomotului pentru a estompa zona respectivă din imagine. Acest proces este realizat prin utilizarea funcției "imgaussfilt", care primește ca parametri imaginea, nivelul de blurare și regiunea din imagine care trebuie blurată. Rezultatul final este stocat în variabila "blurredRegion" și este suprapus peste imaginea originală prin intermediul operatorului de atribuire.

Capitolul 3: Captura si procesarea imaginii

Descrierea si corpul functiei

%% Functia de procasare a achizitiilor, blurare de tip square

function photoProcessing(app, frame)

% Instantiem/actualizam layerul pentru ACQUISITIONimage cu noul cadru, pt a observa astfel diferenta dintre imaginea achizionata si mai apoi imaginea prelucrata

app.ACQUISITIONimage.ImageSource = frame;

% Converti cadrul la grayscale pentru a simplifica procesul de identificare al trasaturilor

grayFrame = rgb2gray(frame);

% Definim detectorul de fete,

faceDetector = vision.CascadeObjectDetector;

% Folosim detectorul si poza in grayscale pentru a extrage locatiile din frame in care detectorul a identificat o fata umana

bboxes = step(faceDetector, grayFrame); % BBOXEX → {N, X, Y, WIDTH, HEIGHT}

% Obtinem nivelul de blur selectat de utilizator din app.BLURRINGLEVELSlider

blurringLevel = app.BLURRINGLEVELSlider.Value;

% Iteram prin regiuni si aplicam blur pe fiecare regiune detectata

for j = 1:size(bboxes, 1)

% Extragem treptat cate o regiune

bbox = bboxes(j, :);

% Aplicam un filtru gaussian de reducere a zgomotului/detaliilor

blurredRegion = imgaussfilt(frame(bbox(2):bbox(2)+bbox(4),… bbox(1):bbox(1)+bbox(3), :), blurringLevel);

% Suprapunem in frame-ul original, imaginea blurata/prelucrata

frame(bbox(2):bbox(2)+bbox(4), bbox(1):bbox(1)+bbox(3), :) = blurredRegion;

end

% Actualizam in UI imaginea prelucrata

app.PROCESSEDimage.ImageSource = frame;

end

Capitolul 4: Captura si procesarea imaginii in timp real

Algoritmul de procesare in timp real

Procesul prin care se apelează procesarea in timp real începe din apelarea funcției `startupFcn` la deschiderea aplicației. Apelarea funcției `setupLiveAcquisition` porneste procesul pentru a inițializa camera și perioada de achiziție a informațiilor.

% Configuram timer-ul pentru achiziționarea continuă a cadrelor, la momente fixe si apeland functia de updateLive

app.Timer = timer( ...

'ExecutionMode', 'fixedRate', ...

% Indică faptul că timer-ul va rula la o frecventa constanta/fixa

'Period', .05, ...

% Frecventa de achizionare a imaginilor ~= 1/20

'TimerFcn', @(~,~)updateLive(app), ...

% Specifica functia care va fi rulata la fiecare apelare a timerului

'BusyMode', 'drop');

% Specificam comportamentul timerului atunci cand functia precedenta nu a fost terminata inca, in cazul de fata adaugam un drop si informatiile functiile care nu s-au terminat la timp sunt pierdute

Funcția `setupLiveAcquisition` inițializează obiectul de tip cameră și configurează timer-ul astfel încât să ruleze la o frecvență constantă și să apeleze funcția `updateLive` la fiecare apelare a timer-ului. Timer-ul este pornit prin apelarea funcției `start(app.Timer)`.

Funcția `updateLive` este apoi apelată la fiecare actualizare de cadru și verifică dacă utilizatorul a activat modul LIVE din interfața grafică. Dacă modul LIVE este activat, funcția `updateLive` apelează funcția `photoProcessing` sau `photoProcessing2` în funcție de forma selectată de utilizator pentru a detecta și estompa zonele faciale din imagine. În final, funcția `updateLive` actualizează eticheta UI cu timpul curent și afișează imaginea procesată în interfața grafică cu utilizatorul.

function updateLive(app)

% Vericam daca modulul LIVE din UI este pornit de catre utilizator

if app.LIVESwitch.Value == "On"

% Selectam forma sub care bluram fata subiectului

if app.SelectshapeButtonGroup.SelectedObject == app.squareButton

% Functia de blurare pentru patrat

photoProcessing(app,snapshot(app.Camera));

else

% Functia de blurare pentru cerc

photoProcessing2(app,snapshot(app.Camera));

end

end

%% Update time in label

% Get the current system time

currentTime = datetime('now');

% Format the current time as a string

timeString = datestr(currentTime, 'dd/mm/yyyy HH:MM:SS');

% Update the UI label with the current time

app.REALTIMElabel.Text = timeString;

end

Este important să menționăm că timer-ul rulează în mod continuu, astfel încât funcțiile `updateLive` și `photoProcessing` sunt apelate în mod repetat și continuu pentru a procesa fiecare cadru capturat de cameră.

Capitolul 5: Documentatie si resurse biografice

- Documentația oficială a Computer Vision Toolbox: <https://www.mathworks.com/help/vision/index.html>

- Exemple și tutoriale pentru Computer Vision Toolbox: <https://www.mathworks.com/help/vision/examples/index.html>

<https://www.mathworks.com/help/vision/ref/vision.cascadeobjectdetector-system-object.html>

<https://www.mathworks.com/help/vision/ug/face-detection-and-tracking-using-live-video-acquisition.html>

- Resurse nemaiaplicate:

<https://www.mathworks.com/help/vision/ug/face-detection-and-tracking-using-the-klt-algorithm.html>

<https://www.mathworks.com/help/vision/ref/traincascadeobjectdetector.html>