

---

---

---

## Table of Contents

מערכת ביומטרית לזיהוי טביעה אבע .....	1
.....	1
תפריט ראשי .....	1
שלב משותף: טיענת תמונה ועיבוד .....	2
לוגיקה לפיה בחירה .....	2

---

---

## - מערכת ביומטרית לזיהוי טביעה אבע - main\_system.m

```
clc; clear; close all;

% --- הגדרת נטבים . 1. ---
% והתקינות של הוספה תיקית ה
addpath(genpath('src'))();
% הוספה תיקית התמונות
addpath('data');

% --- 2. ---
cfg = get_config();
dbFileName = cfg.db_filename; % שימוש בשם מהkonfiguracija
PASS_THRESHOLD = cfg.match.pass_threshold;

while true
```

## תפריט ראשי

```
choice = menu('... , תפריט ראשי' , ...
               '1. רישום משתמש' , ...
               '(Enrollment)' , ...
               '2. זיהוי משתמש' , ...
               '(Identification)' , ...
               '3. ...' , ...
               'הצגת המאגר' , ...
               '4. יציאה' , ...
               '(Exit)' );

if choice == 4 || choice == 0
    disp('!יציאה מהמערכת!');
    break;
end
```

שניהם מוחזק ב-

## שלב משותף: טעינה תמונה ועיבוד (Pipeline)

```
currentData = [];
currentImg = [];

if choice == 1 || choice == 2
    % בחרת קובץ
    [file, path] = uigetfile({'*.tif;*.png;*.jpg;*.bmp', 'Fingerprint
Files'}, ...
                            'בחר תמונה טבעת אצבע',
                            'data/');

if isequal(file, 0), continue; end

fullPath = fullfile(path, file);
currentImg = imread(fullPath);

% --- עיבוד התמונה ---
disp('>>> (Pipeline)...');
try
    % וגו את המटדים (template) עיבוד מלא: מחזיר גם את הנקודות (descriptors)
    [template, ~, ~, descriptors] = process_fingerprint(currentImg);

    % אריזה לבנה נתונים
    currentData.minutiae = template;
    currentData.descriptors = descriptors;

catch err
    errordlg(['שגיאה בעיבוד: ', err.message]);
    continue;
end

% בדיקת איכות
if size(currentData.minutiae, 1) < 8
    msgbox('איכות התמונה נמוכה מדי (מעט מדי נקודות.)', 'שגיאה', 'error');
    continue;
end
end
```

## לוגיקה לפיה בחירה

```
switch choice
    % --- רישום (Enrollment) ---
    case 1
        name = inputdlg(['הכנס שם משתמש:', 'רישום']);
        if ~isempty(name) && ~isempty(name{1})
            % קריאה לפונקציה חיצונית בתיקית
            add_user_to_db(dbFileName, name{1}, currentData, fullPath);
        end

    % --- זיהוי (Identification) ---
    case 2
```

```

=====
=====

if ~isfile(dbFileName)
    msgbox('המארג ריק. נא לבצע רישום תחילה.', 'שגיאה', 'error');
    continue;
end

load(dbFileName, 'fingerprintDB');

bestScore = 0;
bestName = 'לא ידוע';
bestAlignedPoints = [];
bestDbTemplate = [];

wb = waitbar(0, 'סורק מאגר ...');

for i = 1:length(fingerprintDB)
    waitbar(i/length(fingerprintDB), wb);

    % הכנות נתונים מהמאגר
    dbData.minutiae = fingerprintDB(i).template;
    if isfield(fingerprintDB(i), 'descriptors')
        dbData.descriptors = fingerprintDB(i).descriptors;
    else
        dbData.descriptors = [];
    end

    % השוואת
    [score, alignedData, ~] = find_best_match(dbData,
currentData, 0);

    if score > bestScore
        bestScore = score;
        bestName = fingerprintDB(i).name;
        bestAlignedPoints = alignedData;
        bestDbTemplate = fingerprintDB(i).template;
    end
end
close(wb);

% תוצאות
if bestScore >= PASS_THRESHOLD
    msgbox(['זוהה: ' bestName ' (ציון: ' num2str(bestScore) ')'],
('הצלחה'));
    % קריאה לפונקציה חיצונית בתיקית src
    visualize_match_result(currentImg, bestDbTemplate,
bestAlignedPoints, bestName, bestScore);
else
    msgbox(['לא נמצא התאמת. הציון הגבוה ביותר': ' num2str(bestScore)], ('כשלון'), 'error');
end

% --- הצעת המארג ---
case 3
    if isfile(dbFileName)
        load(dbFileName, 'fingerprintDB');

```

```
=====
=====
=====

if isempty(fingerprintDB)
    msgbox( 'המארג קיים אך ריק' );
else
    listdlg('ListString', {fingerprintDB.name}, 'Name',
    'משתמשים רשומים' );
end
else
    msgbox( 'אין נתונים.' );
end
end

end
```

Published with MATLAB® R2025b