FACE RECOGNITION REPORT

Burak Ülkü **17.01.2023**

1.Introduction

I have developed a face recognition program using Matlab, which can scan faces in videos or photos and match them with pre-registered faces to identify individuals. I believe this technology can be used in areas such as security, providing secure authentication and access control.

2.Purpose

The main purpose of this code is to create a face recognition system that uses a pre-trained deep learning model called GoogLeNet. It starts by loading a dataset of images and splitting it into training and validation sets. Then it modifies the GoogLeNet model to suit the number of classes in the dataset and performs data augmentation on the training set. The model is then trained on the augmented training set. After that, it creates a GUI window with a start button, a stop button and an input button. The start button starts the camera, detects the face and runs the trained model on the captured image to classify it. If the person in the image is the person who is expected to enter, it will show the image with a bounding box and a name of the person. If it is not the expected person, it will show an error message. The stop button stops the camera and the input button is used to input the expected person's name. The logo picture is also added to the GUI window.

3 &4. Execution and Application Results:

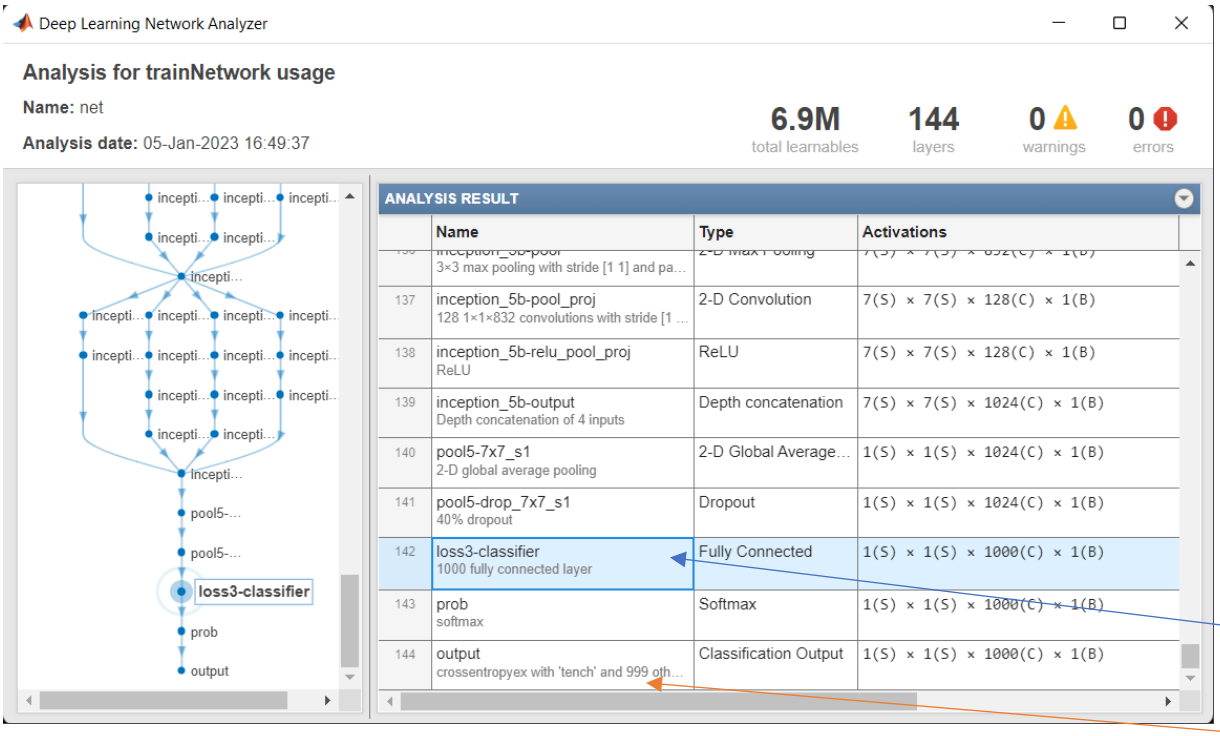
```
Dataset = imageDatastore('Dataset', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');
[Training_Dataset, Validation_Dataset] = splitEachLabel(Dataset, 0.7);
global net;
net = googlenet;
analyzeNetwork(net);

Input_Layer_Size = net.Layers(1).InputSize;

Layer_Graph = layerGraph(net);

Feature_Learner = net.Layers(142);
Output_Classifier = net.Layers(144);

Number_of_Classes = numel(categories(Training_Dataset.Labels));

New_Feature_Learner = fullyConnectedLayer(Number_of_Classes,'Name',...
    'Model the appropriate layer','WeightLearnRateFactor', 10, 'BiasLearnRateFactor', 10);
New_Classifier_Layer = classificationLayer('Name', 'Last layer');

Layer_Graph = replaceLayer(Layer_Graph, Feature_Learner.Name, New_Feature_Learner);

Layer_Graph = replaceLayer(Layer_Graph, Output_Classifier.Name, New_Classifier_Layer);
analyzeNetwork(Layer_Graph)

Pixel_Range = [-30 30];
Scale_Range = [0.9 1.1];

Image_Augmenter = imageDataAugmenter('RandXReflection', ...
    true,'RandXTranslation', Pixel_Range, 'RandYTranslation',...
    Pixel_Range,'RandXScale', Scale_Range, 'RandYScale', Scale_Range);

Augmented_Training_Image = augmentedImageDatastore(Input_Layer_Size(1:2),...
    Training_Dataset,'DataAugmentation', Image_Augmenter);

Augmented_Validation_Image = augmentedImageDatastore(Input_Layer_Size(1:2),Validation_Dataset);

Size_of_Minibatch = 5;
Validation_Frequency = floor(numel(Augmented_Training_Image.Files)/Size_of_Minibatch);
Training_Options = trainingOptions('sgdm','MiniBatchSize', Size_of_Minibatch, ...
    'MaxEpochs', 6,'InitialLearnRate', 3e-4,'Shuffle', 'every-epoch','ValidationData',...
    Augmented_Validation_Image,'ValidationFrequency', Validation_Frequency,'Verbose', false,'Plots', 'training-progress');

net = trainNetwork(Augmented_Training_Image, Layer_Graph, Training_Options);
```

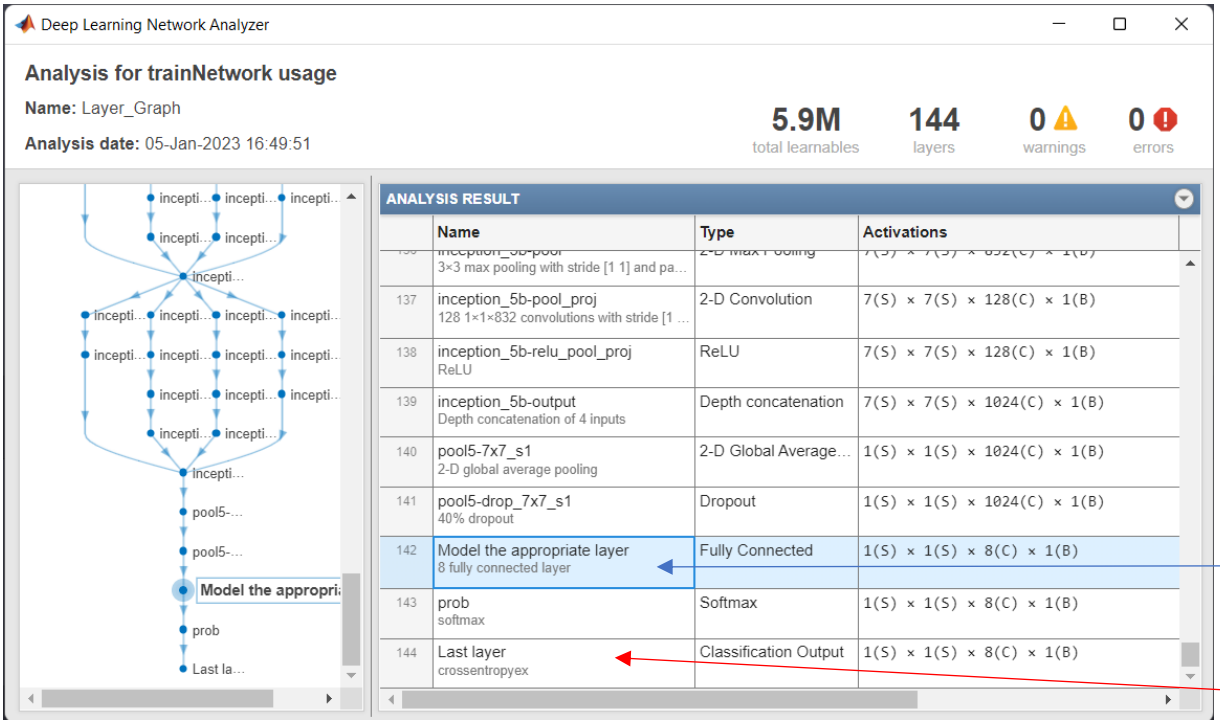
This code block is a piece of code used to create a face recognition system using MATLAB. To explain in detail:

1. *The dataset is created using the 'imageDatastore' function which includes all subfolders in the 'Dataset' folder and labels are taken from the folder names.*
2. *The dataset is then split into training and validation datasets.*
3. *An interface called "GoogLeNet" is used and the structure of the network is analyzed.*
4. *The size of the input layer is obtained.*
5. *The layer graph of the network is created.*
6. *The feature learning layer and the classification layer are determined.*
7. *A new feature learning layer is created using the number of classes in the training dataset.*
8. *A new classification layer is created.*
9. *The layer graph is replaced with the feature learning layer and the classification layer.*
10. *The structure of the network is analyzed again.*
11. *Data augmentation is performed using the "imageDataAugmenter" function which includes operations such as random rotation, scaling, reflection and translation for images.*
12. *The training and validation datasets are re-created as augmented datasets.*
13. *Training options are determined, including minibatch size, maximum number of epochs, learning rate, validation frequency, graphics etc.*
14. *The network is trained using the training dataset and the training options are used.*

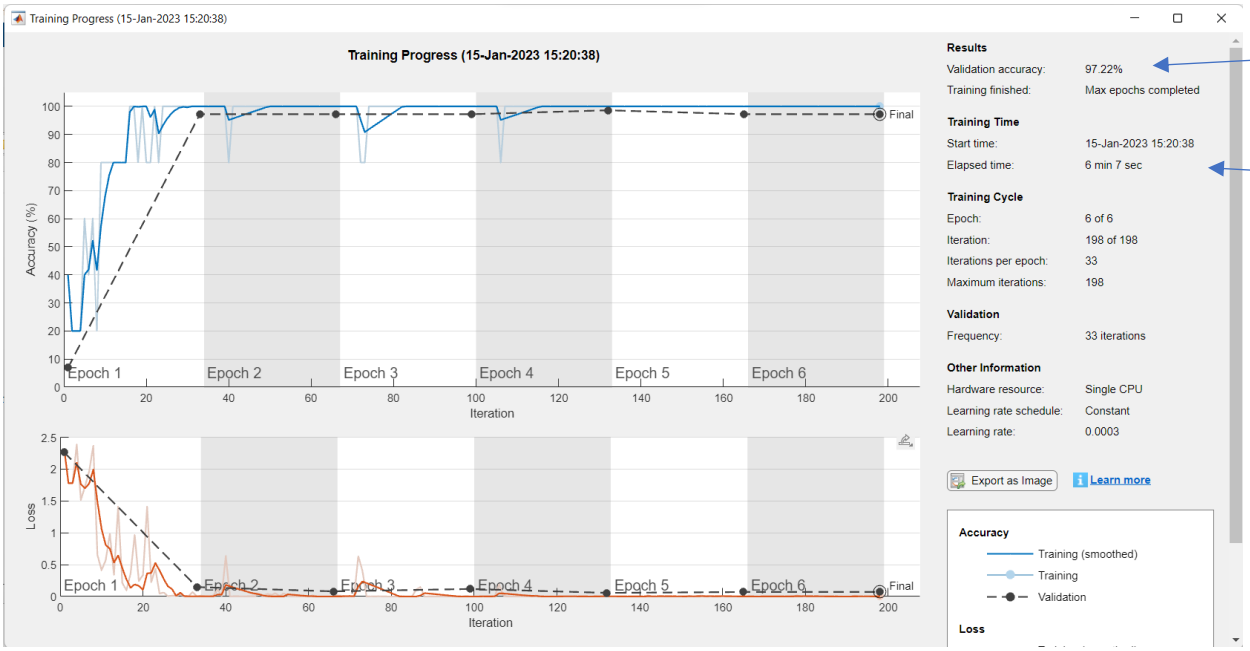
When we run the program, we need to train it first. This process takes 5-6 minutes on average.

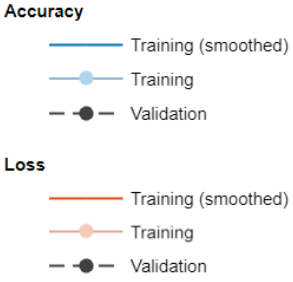
Here you can see the layers have changed. I show the start and end state so that I can understand that it is working.





Thanks to Deep Learning below, we can see how successfully the face recognition system works.





```
% Read in the video file
videoReader = VideoReader('arda.mp4');

% Loop through the frames of the video
i = 1;
while hasFrame(videoReader)
    % Read in the next frame
    frame = readFrame(videoReader);

    % Save the current frame to a file
    imwrite(frame, sprintf('frame%d.jpg', i));

    % Increment the frame counter
    i = i + 1;
end
```

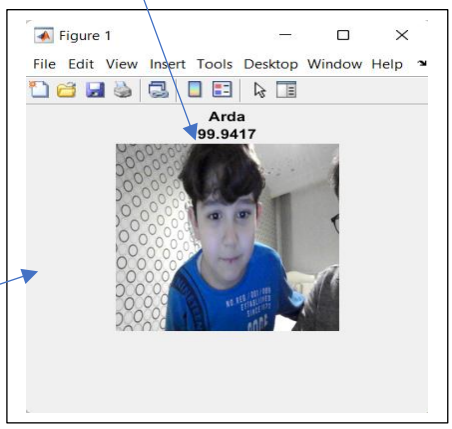
The reason I wrote this code is that it is difficult to take a photo of a person all the time thanks to the code. Thanks to this code, getting a lot of photos from a short video and this makes our face recognition system work more successfully.

Here we write the name of the video that will be separated from the computer to its frames.

In addition, I did face recognition from photos here. The reason I do this is to identify who the criminal is by comparing photos of criminals.

It is written here who the photo looks like with the highest rate of registered people.

```
function picture_(net, image)
    if ~ischar(image)
        error('The input must be a filepath or URL as a string')
    end
    I = imread(image);
    if isempty(I)
        error('The input is not a valid image file, please check the filepath or URL')
    end
    [h,w,c] = size(I);
    if h~= 224 || w~= 224
        G = imresize(I, [224, 224]);
    else
        G = I;
    end
    [Label, Prob] = classify(net, G);
    imshow(G);
    title({char(Label), num2str(max(Prob)*100)});
end
```




```
Command Window
>> picture_(net,'C:\Users\BURAK_\Desktop\MATLAB FİNAL\arda.jpg')
>>
```

Workspace

Name	Value
Augmented_T...	*1x1 augmentedI...*
Augmented_V...	*1x1 augmentedI...*
ax	*1x1 Axes*
Dataset	*1x1 ImageDatast...*
edit_box	*1x1 UIControl*
f	*1x1 Figure*
Feature_Learn...	*1x1 FullyConnect...*
Image_Augm...	*1x1 imageDataA...*
Input_Layer_Si...	[224,224,3]
Layer_Graph	*1x1 LayerGraph*
net	*1x1 DAGNetwork*
New_Classifie...	*1x1 Classification...*
New_Feature_...	*1x1 FullyConnect...*
Number_of_Cl...	7
Output_Classi...	*1x1 Classification...*
Pixel_Range	[-30,30]
Scale_Range	[0.9000,1.1000]
Size_of_Minib...	5
start_button	*1x1 UIControl*
stop_button	*1x1 UIControl*
Training_Data...	*1x1 ImageDatast...*
Training_Opti...	*1x1 TrainingOpti...*
Validation_Da...	*1x1 ImageDatast...*
Validation_Fr...	33

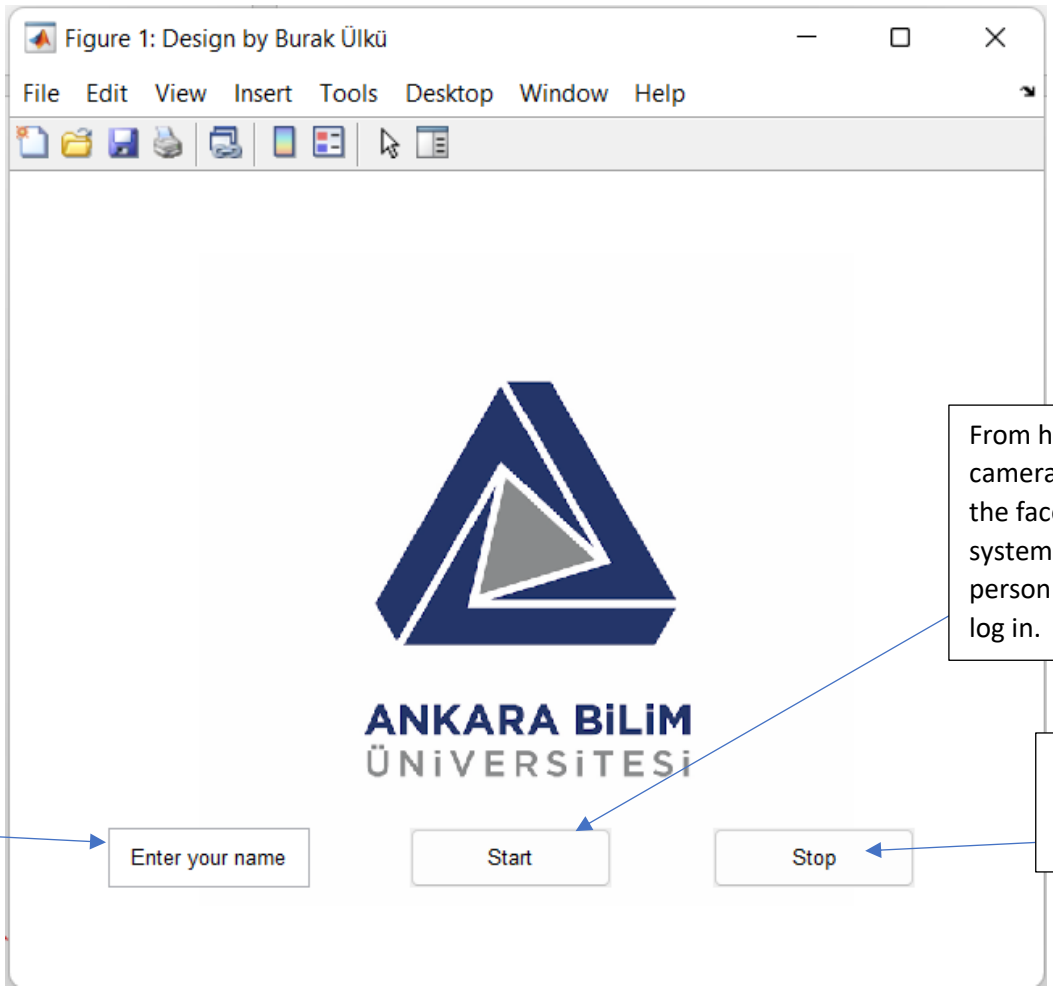
```
% Create GUI window
f = figure('Name', 'Design by Burak Ülkü', 'Color', 'white');
ax = axes('Parent', f, 'Position', [0.1, 0.1, 0.8, 0.8]);
imshow('logo.png', 'Parent', ax);

% Create start button
start_button = uicontrol(f,'Style','pushbutton','String','Start',...
    'Position',[200 50 100 30],'Callback',@start_callback);
% Create stop button
stop_button = uicontrol(f,'Style','pushbutton','String','Stop',...
    'Position',[350 50 100 30],'Callback',@stop_callback);
%Input Button
edit_box = uicontrol(f, 'Style', 'edit', 'String', 'Enter your name',...
    'Position', [50 50 100 30], 'Callback', @edit_callback);
% Return function for start button
function edit_callback(hObject, ~)
    global aranacak_kisi;
    aranacak_kisi = get(hObject, 'String');
end
function start_callback(~,~)
    global net;
    disp('Camera started');
    global web;
    web=webcam();
    sensor =vision.CascadeObjectDetector();
    global aranacak_kisi;
    image =snapshot(web);
    grey = rgb2gray(image);
    bbox = step(sensor,grey);
    picture = imresize(image, [224, 224]);
    [Label, Prob] = classify(net,picture);
    name=char(Label);
    value=num2str(max(Prob));
    detpic=insertObjectAnnotation(image,"rectangle",bbox,name+" "+value);
    if strcmp(name, aranacak_kisi)
    imshow(detpic);
    msgbox(['Welcome ', aranacak_kisi], 'Access granted');
else
    msgbox(['you are not ', aranacak_kisi],'Error');
    end
end
function stop_callback(~,~)
global web;
    disp('Camera stopped');
    delete(web);
end

```

REAL TIME CODE AND GUI

This code creates a GUI window with three buttons: "Start", "Stop", and "Enter your name".

1. *The first line creates an empty figure window for the GUI.*
2. *The next line creates a button called "Start" and adds it to the figure window. The button's position, size, and the callback function that is called when the button is pressed are also specified.*
3. *The following line creates a button called "Stop" and adds it to the figure window in the same way as the "Start" button.*
4. *Then, it creates an input button called "Enter your name" where user can type his/her name,and also specifies the position and callback function.*
5. *Then, it defines a function called "edit_callback" that is called when the "Enter your name" button is pressed. It retrieves the user's input from the button and stores it in a global variable called "aranacak_kisi"*
6. *Next, it defines a function called "start_callback" that is called when the "Start" button is pressed. It starts the camera, takes a snapshot of the current frame, uses a sensor to detect a face in the image, and classifies the face using a pre-trained neural network (net) to determine the person's name.*
7. *If the detected person's name matches the desired person (stored in the global variable "aranacak_kisi"), the image is displayed and a message box appears with the message "Welcome [aranacak_kisi]". If the names do not match, a message box appears with the message "you are not [aranacak_kisi]".*
8. *The last function called "stop_callback" is defined, it is called when the "Stop" button is pressed. It stops the camera and closes the webcam object.*

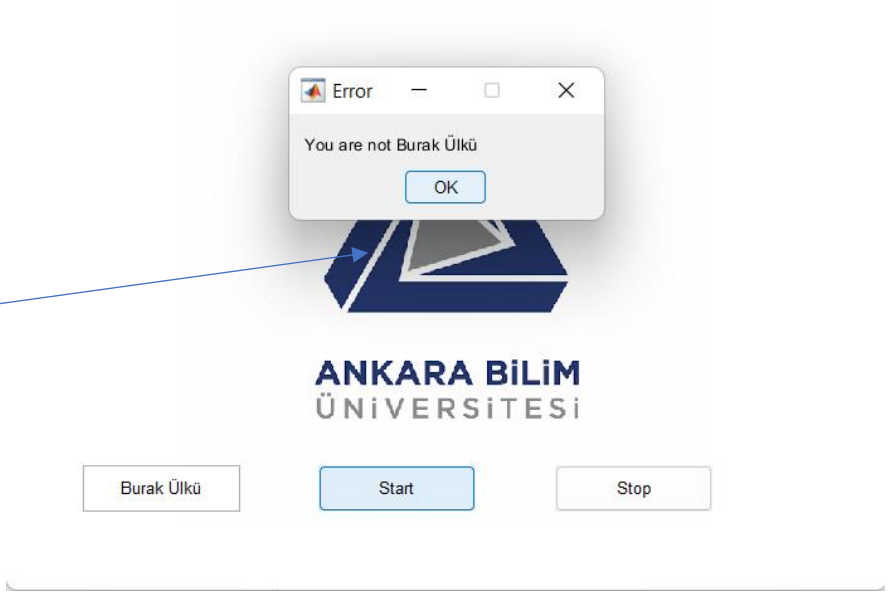
Let's run code.



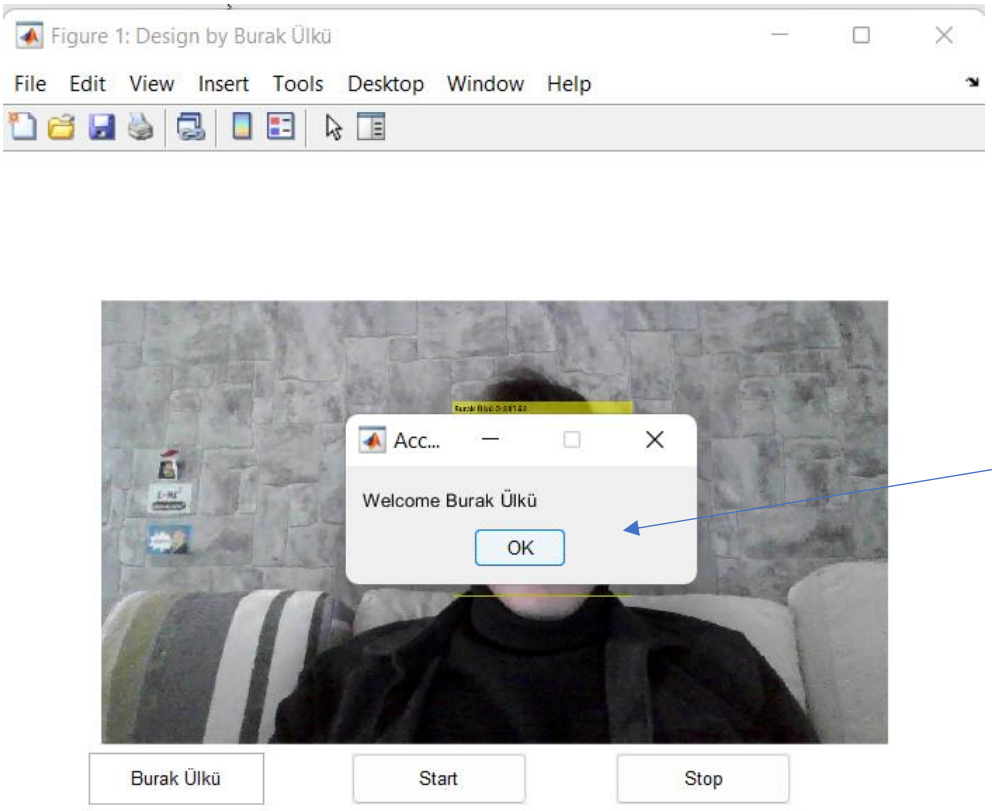
From here, it starts the camera and checks with the face recognition system whether it is the person you really want to log in.

You log in by typing your name here.

You exit here by stopping the camera.



If the person who wants to log in and the person whose face is being read are not the same person, they cannot log in.



If the login person is the right person, the pop-up screen opens and the text document appears.