Doing Vision Studies with Computer Using Drone

In summary **— This document is about doing computer vision studies using drones.**

Keywords—drone; face recognition; first face detection; gender-age detection; object tracking in video or live; object counting video and live; counting object from Picture; sorting and counting objects from the Picture; Object Measurement; Licence Plate Recognition;ocr

Abstract **— This electronic document contains general information about our project and descriptions of its features.**

Keywords—drone, component, face, receognition, identification, object; counter, detection,

I. INTRODUCTION

Our drone Project basically consists of collecting data using computer vision technologies, storing this data regularly and reporting for later analysis.

From past to present, drones are used for a wide variety of functions, such as human aircraft, monitoring climate change, delivering goods, assisting in search and rescue operations, taking photographs, and recording video.

II. SCOPE

The name of the project to be presented in this document is Developing Autonomous Applications with Drone.

We tried to keep the scope of the project as wide as possible, since our application has many features. In this way, we aimed to offer our users the best experience and comfort. For this purpose, we have reviewed, tested and evaluated all drone-related projects.

Concentrating our project on computer vision studies, we determined our features in this context.

We have listed all the features of the project and grouped all these features from general to specific. These groups enabled us to identify the subsystems of our project, the main modules of these subsystems, and the submodules of these main modules. Thus, the relations of all the features in our project with each other were determined on a regular basis.

Since the project does not have a complex interface, has high mobility, and is used in a fast and comfortable way, we decided to manage all its features under the control of a simple mobile compatible website.

We recorded the data of the project as it should be, adhering to the database management system. We analyzed this data and prepared reporting studies by making the necessary associations.

We examined the scope of our project from general to specific and prepared all our module groups in line with the list given below.

Let's examine all the system modules in order according to the module group they are connected to;

III. SUBSYSTEMS

We cover the most general features of our project in the subsystems group, which is the most general group of our drone project. These features are the part where the operations related to the human face are located and the part where the features related to object identification are located. Based on these scopes, we have developed our features.

A. Face-Subsystem

It is a subsystem of our drone project, in which modules are connected to the main processes related to the human face in general. Our processes related to the human face that we have developed in this sub-system; The system recognizes the first person it sees and records it as a visitor, the system recognizes the people it knows before appearing on the system camera and performs related actions, and our last feature in this sub-system is gender and age recognition. It detects people's faces with our drone project camera and then estimates their gender and age.

B. Object Detection-Subsystem

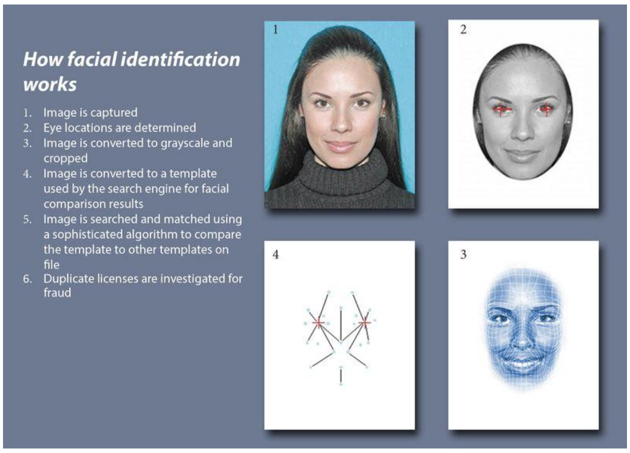
It is the subsystem of our drone project, in which modules are connected to the main processes related to object definition and related operations in general. We are defined by the drone camera and we make some features related to them; The first of these is that we track the living or inanimate objects we see in our drone camera by assigning a unique ID number according to the predetermined classes depending on the object types. Another feature; We count the animate or inanimate objects we see with our drone camera according to the predetermined classes depending on the object types. Our last feature connected to this sub-system is that we crop each animate or inanimate object in the pictures we take with the drone camera according to their predetermined classes and unique identification numbers from the relevant picture and store them in a way that is connected to the picture.

C. MAIN MODULS

It is a module group that covers all the main features of our drone project. Under the subsystems to which they are connected, they have features that directly concern the subsystem. Some of these modules are face recognition, gender and age detection, object tracking based on unique ID numbers….

Doing Vision Studies with Computer Using Drones

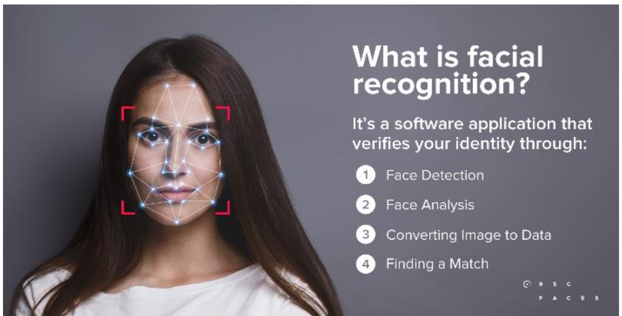
A. First Face Detection

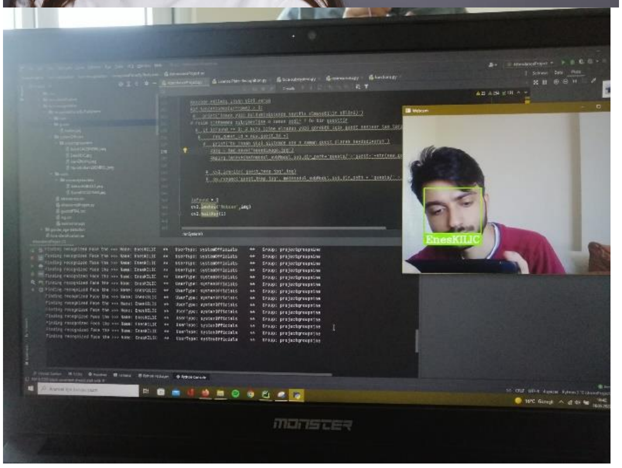


Face detection systems need computer algorithms to pick out distinctive details with a person's face. Fine details such as the distance between the eyes and the shape of the chin are translated into their mathematical form and represented as such. There are general face templates. Identification is done using these

On the web design side of our Drone Project, the first face that the webcam sees immediately after registering the login information of our visitors who want to register to our system is the face of the visitor and is a main module with t the features of recording the picture of the visitor.

B. Face Recognition



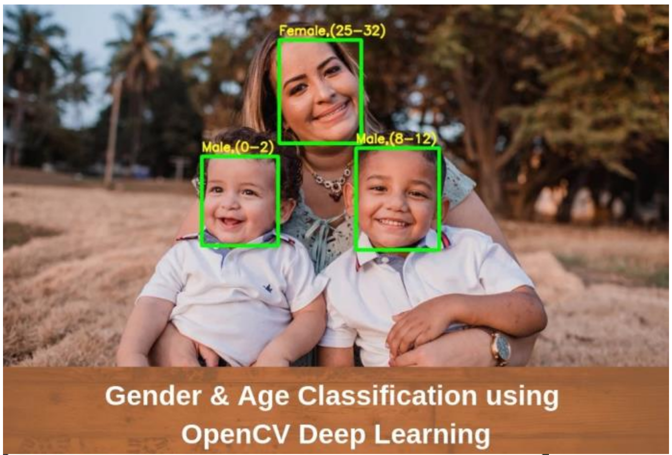


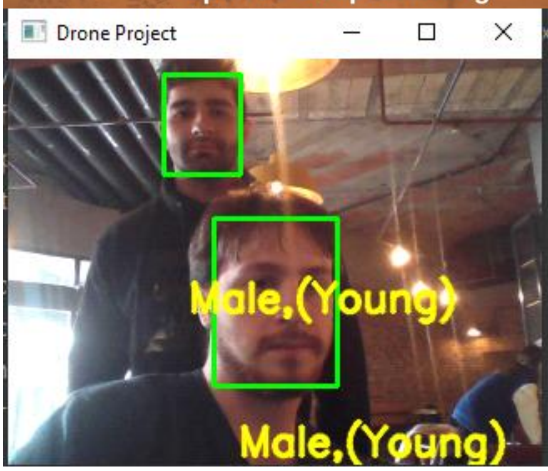
Facial recognition systems, by using artificial intelligence and machine learning, by using artificial intelligence and machine learning, knowing the mathematical state of all information, including the specific and distinctive features of a person's face, or converting and remembering the picture of the relevant person in advance into mathematical data during the comparison, and transforming the faces that can be new to the system into mathematical states. to be compared and recognized.

We start the face recognition detection phase with an algorithm that learns what a face is (opencv-python/face recognition). We did it by training with face photos. We compressed enough images to train the algorithm, thus minimizing the margin for error.

In our Drone Project, it is a main module that includes the features of ensuring that the faces registered in the system (this can be in the form of a picture, can be in the form of video, can be database data) are recognized by the system by comparing the faces of the people using the relevant feature on the drone camera. One of these features is the Atttendance feature. It is introduced to the system as a submodule of the main module.

C. Gender/Age Detection





It is a main module of our Drone Project that has the features of estimating and recording the gender and age of human faces determined by the drone camera.

I will talk about the use of Gender and Age Classification Using CNN. We have prepared this feature using a convolutional neural network architecture similar to CaffeNet and AlexNet. The details of the layers are given below.

Conv1: In the first convolution layer, it has 96 nodes with 7 kernels.
Conv2: The second convolution layer has 256 nodes with a

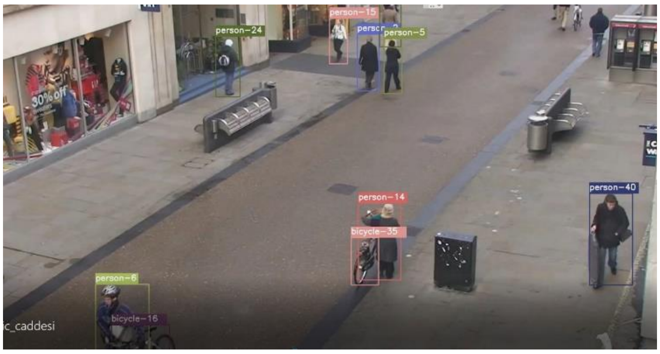
core size of 5.
Conv3: The third convolution layer has 384 nodes with a kernel size of 3.

The two fully connected layers each have 512 nodes.

We used the Adience dataset to train the model

a) *Gender Prediction : Gender Estimation was framed as a classification problem. The output layer in the network, which is used to predict gender, has 2 nodes representing two classes, 'Male' and 'Female'.*
b) *Age Estimate: Age estimation is a regression problem. It is difficult to accurately estimate age using regression. Even humans cannot make accurate age estimates in real life. We can only guess if we are in our twenties or thirties. So we can estimate the age groups of the person. There were eight age classes in the Adience data set; [(0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 – 53), (60 – 100)]. Thus, the last softmax layer of the age estimation network has eight nodes representing the mentioned age ranges.*

D. Object Tracking with ID



With the drone camera of our Drone Project, it determines the unique identification numbers of predetermined living or inanimate objects in a live image or in a pre-existing video recording, and it can record them in the desired video format. At the same time, it is a main module with the features of recording the resulting data.

Let's take a closer look at the central tracking algorithm and take a closer look at the object tracking application .

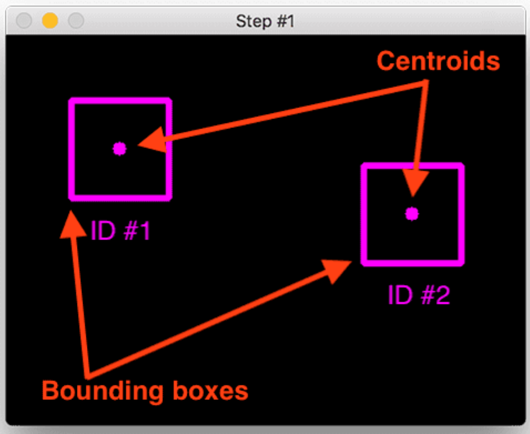


Figure 1: Bounding Box Coordinates and their center of grafity

First thing I did was I verified the bounding box coordinates and calculated their center of gravity.

Once we had the bounding box coordinates, we had to calculate the center of the bounding box, or simply the x and y coordinates.

Then we will assign them a unique identification number.

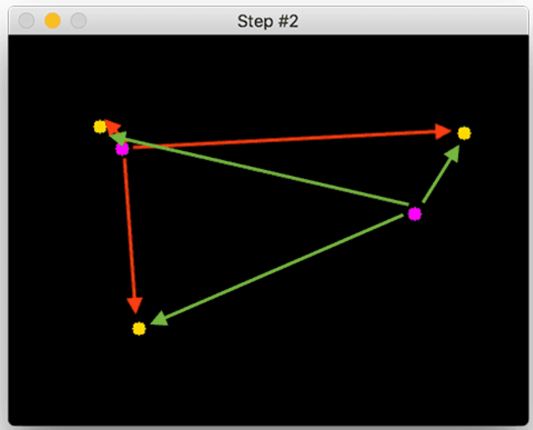


Figure 2: Bounding Box Coordinates and their center of grafity

We calculated the euclidean distance between the existing objects of the bounding boxes.

We perform this step for each frame, but instead of assigning a new unique identity to each new detected object, we first need to determine whether we can associate the new object centers (yellow) with the old object centers. To perform this operation, we calculate the Euclidean distance (indicated by the green arrows) between each pair of available object centroids and the input object centroids.

Just to give an example, you can see that we detected three objects in our figure 2' image. Two pairs close together are two objects in existence.

Doing Vision Studies with Computer Using Drones

We then calculate the Euclidean distances between each original centriole (yellow) and a new centriole (purple).

So how do we use these Euclidean distances to actually match and relate?

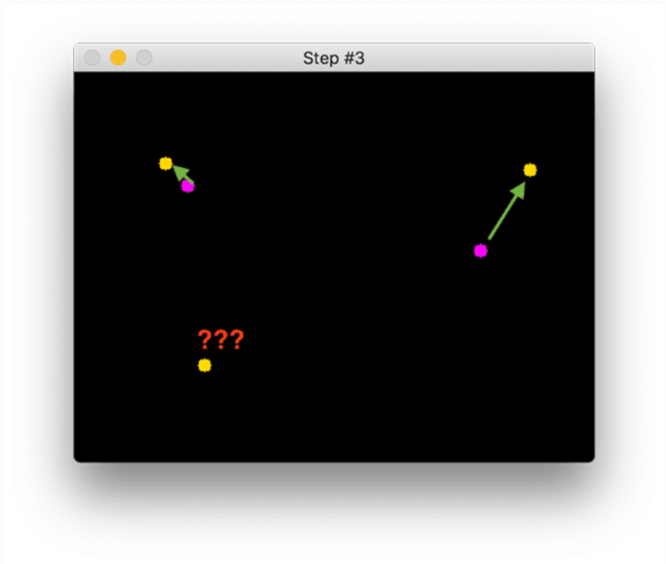


Figure 3: Updating existing object coordinates

The primary assumption of the central tracking algorithm is that a given object will potentially move between subsequent frames, but for frames, the spacing between centers will be smaller between F't and FT+1 objects than any other point. So if we choose to associate centrioles with at least minimum distances between subsequent frames, we can thus construct our object tracker.

So what do you see here, the lone dot at the bottom left?

It's not associated with anything, is it? This is the registration phase of our new object.

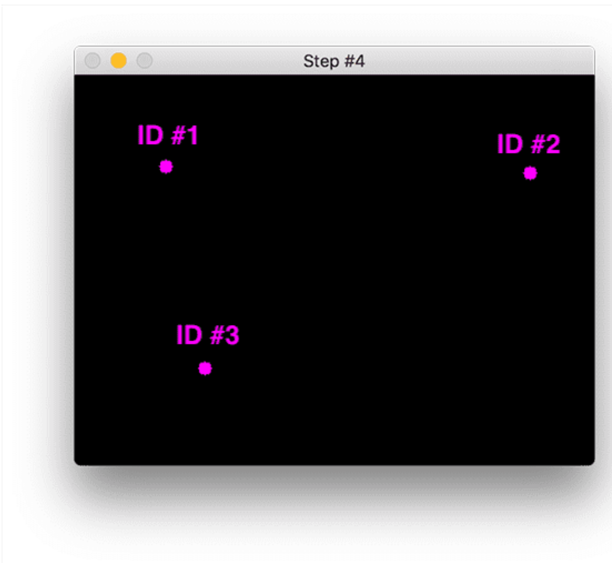


Figure 3: Updating existing object coordinates

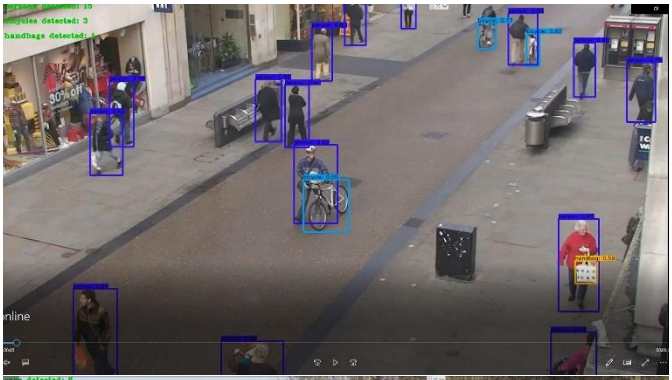
If more inputs are detected than existing monitored objects, it means it is a new object. This is what I call saving; means we add the new object in our list of watched objects as follows. We both throw new objects and hide the center of gravity of these objects.

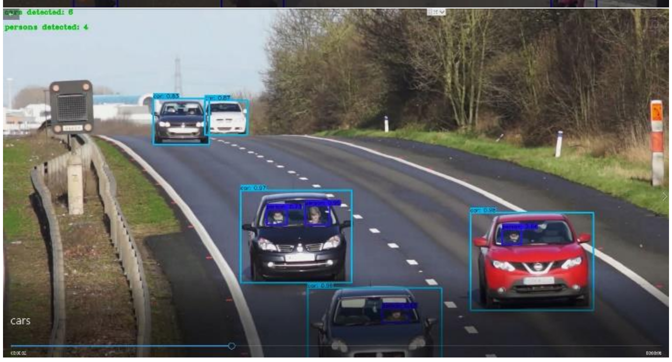
Now we clean up all that we've done 'cause it's all old.

So I mean this; Objects begin to disappear and must be able to process when they are lost or out of sight. For a total of n frames, we delete and remove the shift of old objects when they cannot be matched.

Thus, the process continues in a loop.

E. Counting object by classification(ObjectCounterVideo/Live)

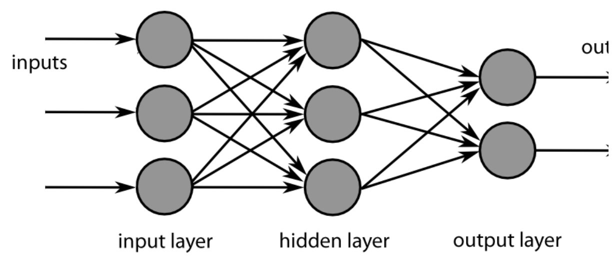




It is a main module that has the features of recording the predetermined living or inanimate objects in a pre-existing video recording with the drone camera of our Drone Project, determining their total number by determining their total number and recording them.

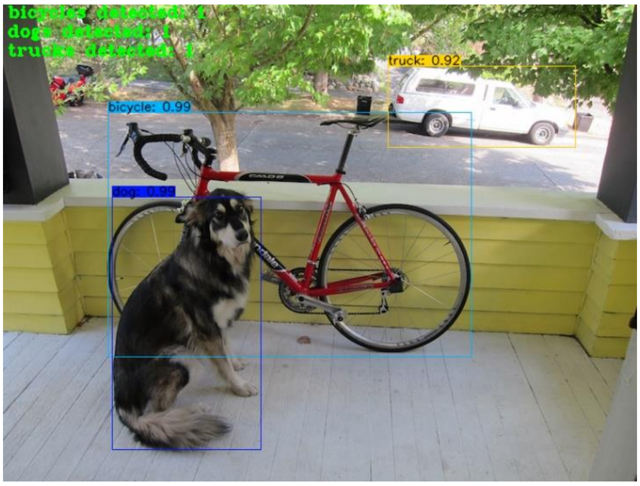
Full link layers, which are the last layers of CNN, are densely connected layers or artificial neural network (ANN). The primary function of ANN is to combine different attribute features to help analyze and classify input features. These layers represent assemblies of neurons that represent different parts of the object in question, and this collection of neurons could represent a person. If enough of these neurons are activated in response to an input image, the object will be classified.

Doing Vision Studies with Computer Using Drones



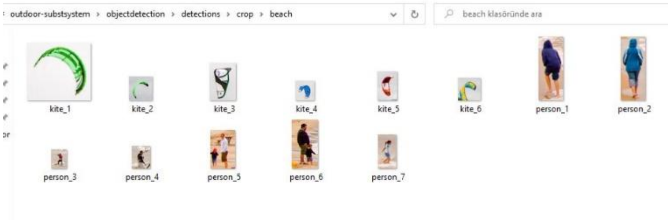
Ultimately, the fully connected layer will output the previous layer and present a one-to-one probability for each class. If the "Person" category has a value of 0.60, it is 80% certain that the image is a dog.

F. *Counting objects by classification from Image*



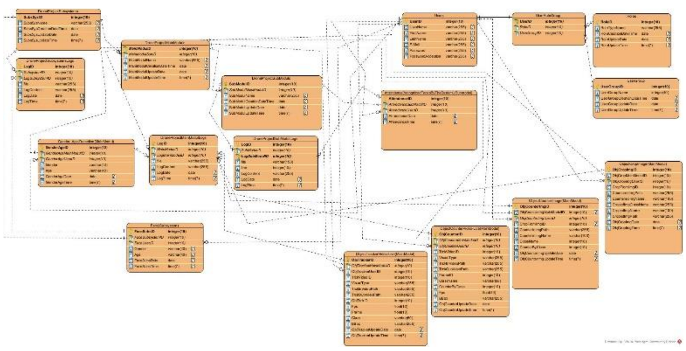
It is a main module that can determine the predetermined animate or inanimate objects according to their objects, determine the total number of the pictures taken with the drone camera of our Drone Project, and record them

G. *Crop object by classification from Image*

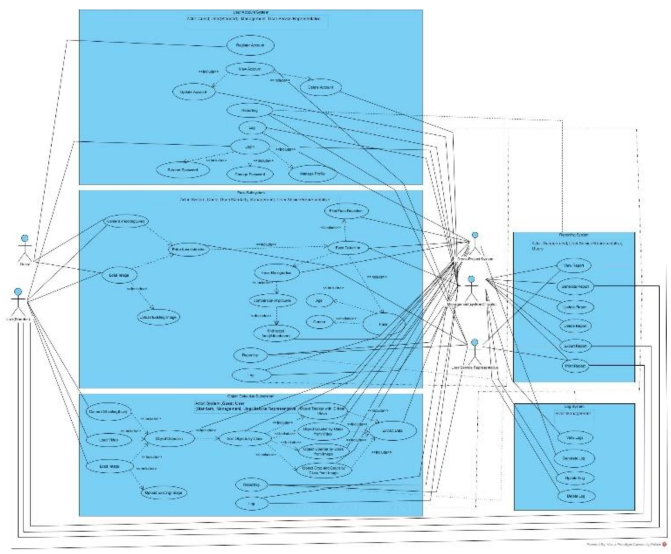


It is a main module that has the features of determining the predetermined animate or inanimate objects according to the objects of the pictures taken by the drone camera of our Drone Project, cropping them depending on the object group and the main picture of the object, naming them according to this information, and recording the pictures and their associated data.

H. *Drone Project Full Er Diagram*



İ. *Drone Project System Use Case Diagram*



J. Object Measurement

In our drone project, it is our main module that calculates the dimensions (height and width) of the objects seen by the drone camera.

By finding Countours of objects, we can more easily identify them and get their analysis. To do this, we use the CV2 structure of the OpenCv library. In this structure, we use the Canny method
First of all, we're turning the picture into a gray version..
With GaussianBlur behind it, we make the picture blurry, allowing us to see more comfortable lines of countours.
With the Canny method, we detect the object borders with the threshold value we have already determined. The higher the threshold value, the greater the density of contouring, and more prominently, it will also find contours on the surface of the object. Even the deformed dots of the wooden table are visible in the left picture. We use the dilate library to determine the iteration on the contoured picture that we have determined with Canny as 3. . (iteration > refreshes around the core)

We use the same kernel size and set the number of refreshes around the core to 2 this time, allowing spot wear to expand the image we have expanded. We also set the refreshes around the core to 2.

We've removed the image threshold and you see there are smoother edges now

We switch to the next strokes and use the find stroke function
We use our contoured image, which is the threshold, together with its external mode because we need external borders
.
Why am I doing this? Because then I can move around these

Doing Vision Studies with Computer Using Drones

conturs and find and draw their areas and see how well we detect them and then We find our Countur area, then we filter by minimum threshold area

We need to set a minimum area so there's no too little space so we can call it a space, sir.

Calculate the length of the two-point arc. First we will send our stroke as a parameter and then when it is closed (no open-ended lines in our picture so we will not say true

Now we find the corner points;
In doing so, we use the approxPolyDP method again, we just need to point out that the CV2 library is in the closed range with the resolution values of our strokes, which come in turn looping when using it.

Finally, we can find the limiting box (bbox). So we can make it more approximate and with it we can see the point limitation. I wanted to point out with the limiting rectangular box. In doing so, we will send our approxPolyD as a parameter.

All of this will reveal to us the information of our existing perceived object with a more detailed border and more detailed corner.

Well, let's just say I want to see four-pointed objects , what do I do?

This would be a filtering exercise. If our approach is equal to filter value, we will put it on our final list ; Let's create our final list; that is, let's save our last strokes to our final credit list and send the approximate number of lengths (len(approx)) and the corresponding countor area(area), then send our approximate points and then our limiting rectangular boxes (bbox), which we find, together with general function information (i) and store our final strokes in the array in its new form;

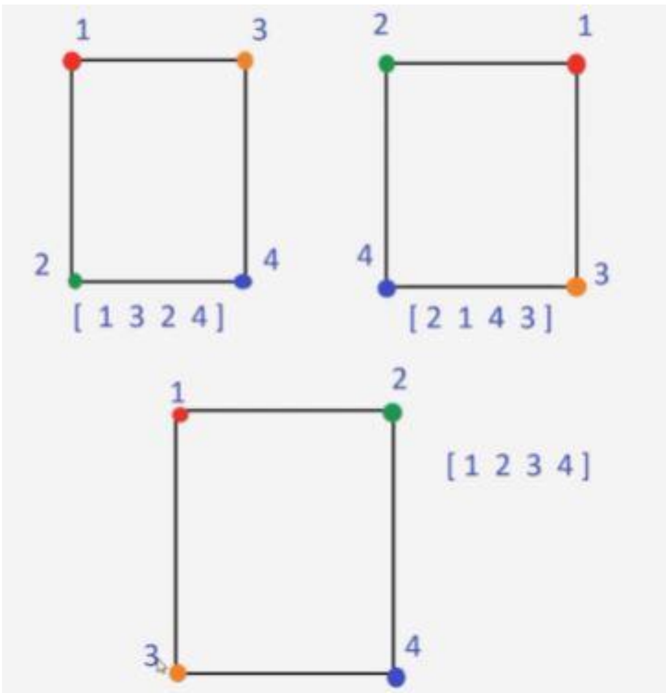
The next thing we can do is sort these strokes by size, which is useful to us , for example, we want to find the biggest contour by referencing the size of the A4 paper. If we sort it out, we'll find the biggest contour.
Let's sort our final strokes with sorted method , in doing so we will use our area countur field as the 'key' value. So that means that you're not going to be able to do anything about it. sort by size of the field.
In doing so, we use lambda inquiry ; x1 will be the area x0 length and we expect everything to be in increasing order and we reverse typing to ensure that the greatest value is the first

The last thing we're going to do is. We will use our identified contours by drawing and see if we have identified the appropriate ones.
We turn around on the list where we keep the contours I found and start drawing our contours. Since it is a multidimensional array when drawing, we need a contour fourth element, after each sequence we will determine the color values and thickness ratio, and after sending all these parameters, it will start drawing new countur.
In case our user wants to draw;
Our function ends here. As returning values, we rotate our original picture and final stroke sequence.

Let's rework our picture with our last found contours;

I had to optimize our last credit-taking phone call to find the A-4 paper. When we call our final contouring function, we start by entering a minimum area and then filtering according to 4-point objects by filter=4

We will use this for approximately 4 points by taking Len (approx) and approx values



We need to make this sorted according to the bottom frame so that we can access the picture corners comfortably and neatly, and most importantly, wrap it up by adhering to the width and height value. So we need to find out where the width and height are. For this we use the function I showed below.

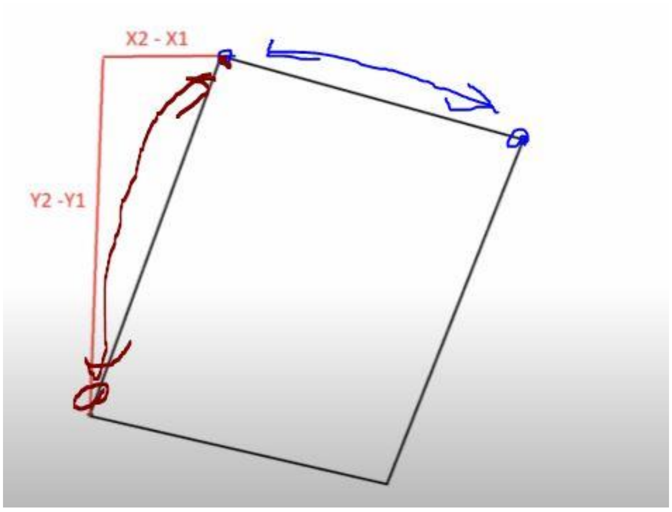
The image itself will send points, width and height values as parameters
importantly, wrap it up by adhering to the width and height value. Then we'll reorder our dots. When we run our funding. Axis will make a sum of one in 1, giving us the sum of each point. The smallest of these points will be our element, the zero element. The highest will be maximum. One will be the width, one the height will be our corner point.

Doing Vision Studies with Computer Using Drones

The new point is equal to five points zero(5.0).

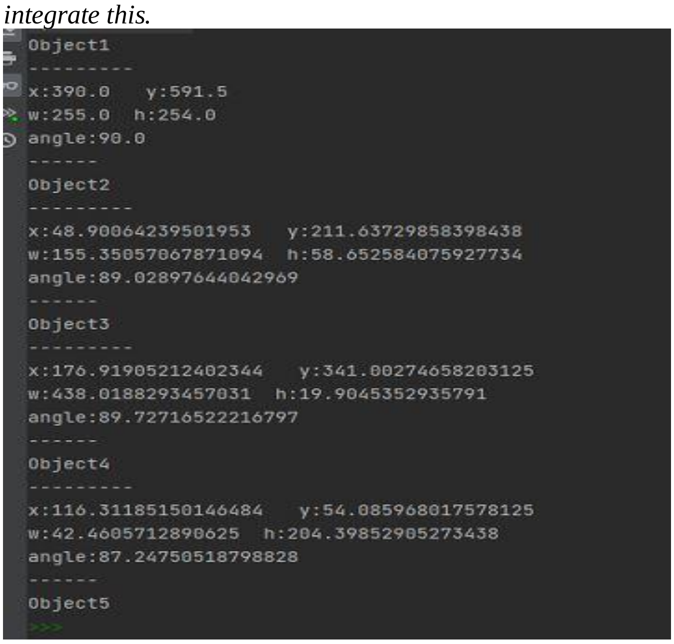
Let's assign the smallest element in the insertion list to the zero element of this new array. Now we'll get the maximum value of the new value. We adjust our points according to the axis one. We're getting minimum and maximum.

We find the lengths of the strokes with the following method and then We find the length of the object by measuring the distance between the two points

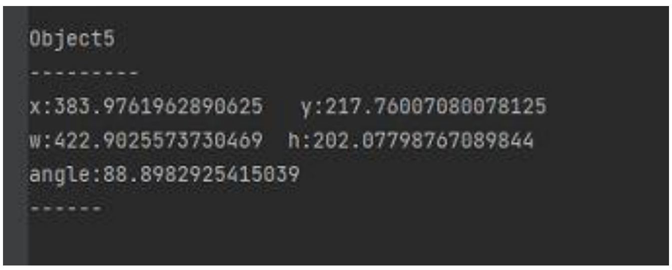


To make the edges better specific, I do the following rectangular operations

You see the x and y coordinates of the objects, the height and width of the objects, the angle values respectively, and now let's integrate this.

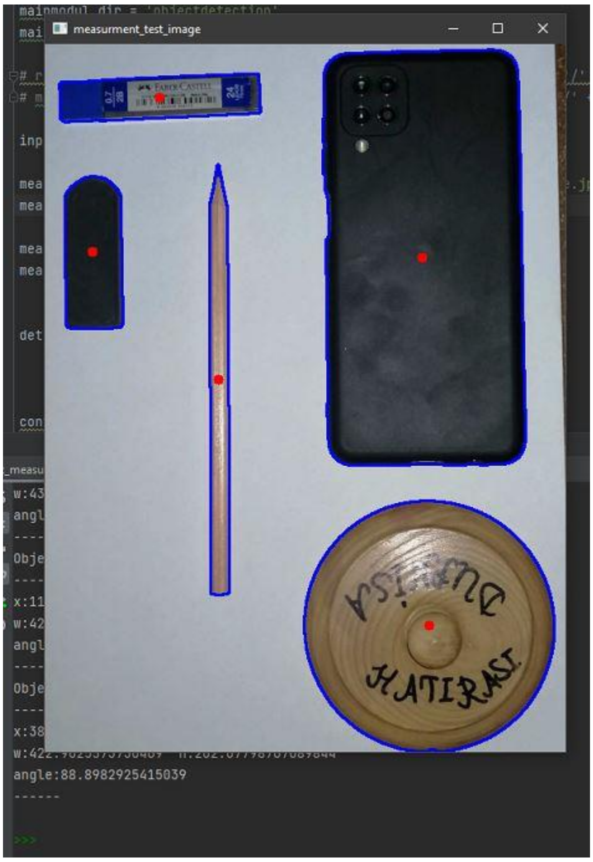


```
Object1
---------
x:390.0    y:591.5
w:255.0  h:254.0
angle:90.0
------
Object2
---------
x:48.90064239501953    y:211.63729858398438
w:155.35057067871094   h:58.652584075927734
angle:89.02897644042969
------
Object3
---------
x:176.91905212402344    y:341.80274658203125
w:438.0188293457031   h:19.9045352935791
angle:89.72716522216797
------
Object4
---------
x:116.31185150146484    y:54.085968017578125
w:42.4605712890625   h:204.39852905273438
angle:87.24750518798828
------
Object5
```



```
Object5
---------
x:383.9761962890625    y:217.76007080078125
w:422.9025573730469   h:202.07798767089844
angle:88.8982925415039
------
```

You see the centers of the objects in the picture below



Now we make our object identifications with the blue line we make using the polyLines method by means of rectangular, boxing method.

Now let's see the size of your phone live;

Doing Vision Studies with Computer Using Drones

K. Licence Plate Recognition(Anpr - Ocr)

The license plates of the vehicles are unique. If we recognize these plates and write them down, we can use them in various studies.

Step by step, let's see how we do this, and I did it with what we learned in class;

For better and better quality plate reading, we first grayscale our vehicle picture.

We move on to removing the noise from the image, softening it and edge detection, which is essentially needed for analysis.

As you can see in the picture, the writing has become more pronounced;

In the following code we copied and gave parameter edges, then tried to approximately what it looked like, so we found the key points. In a bottom line, we found the strokes of the key points we found and stored them in the countours array. This actually simplifies how our strokes are rotated, and then we reviewed and sorted them and returned the first 10 strokes. We sorted it by contour area, adjusting it from high to low.

Now we're going to go through these dots and try to find four key points, the license plate. In the following code we estimate the range from our contour, it allows us to determine how accurate or finely grained it is. There is a contour with a lot of small notches in it, it rounds it to this counter how high we put a polydp, so for example it looks at that something that is roughly rectangular and the more it is roughly the smaller the dent line and punctuation, and indicate that this is really a straight line in this regard 10. is ideal.
That's how we found the license plate box, but we can't see it. So what should we do? Of course it's masking. Let's see, let's see. Creates an empty mask in the following code, and then uses np zero to create a mask that will make our original gray image the same shape. We did this through the shape of a gray dot and then we determined how we wanted to fill it, and in this case we set the NP to eight, so we basically fill it with empty zeros, then we draw our contours into this image so that we create our mask, which is our temporary image.
And in this mask comes the image we see below.

Yes, some of our plates have been successfully identified. Now we read our license plate using ocr of this mask picture, that is, by writing the picture in writing.
Of course, as you can see above, there's a big picture of empty darkness. Let's split up the little section we need.
Here's what we're doing. We adjust the height by adjusting X to x-two plus one, that is, about 30 units, according to the pilot, and we do the same logic y-to-two in the same way. X is the end of two and y two.

Now let's write down the license plate we found and get our data;

That's how he makes a show when we read the picture.

Let's get our license plate from this show and then put it on the real car picture.

Doing Vision Studies with Computer Using Drones

L. Vanishing Point

The vanishing point line is the farthest point in a perspective drawing image plane, as if correspondingly parallel two-dimensional perspective drawings converge in 3-dimensional space.

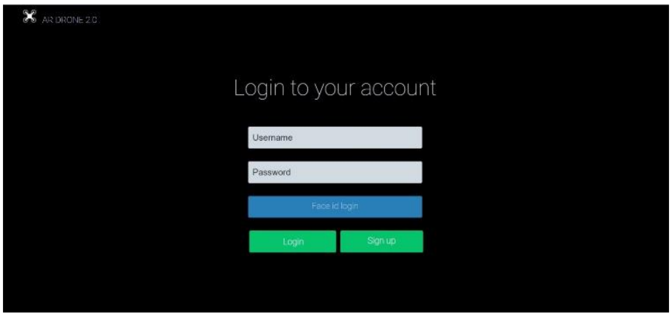
You could call it a kind of vanishing point. We can fix this problem using the steps below.

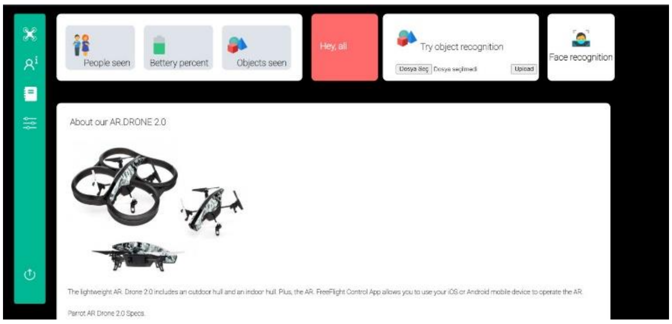
First, we find all the lines in the image, then these lines must be at least a few pixels in size.

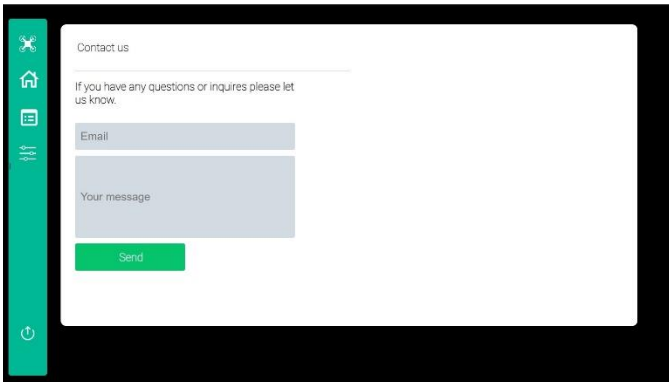
Second, we have to filter out all these rows found. Filtering will be done according to the angle and length of the line relative to the horizontal.

Finally, we will find the vanishing point with the help of the lines we have found. It should be noted that the boiling point is approximately the intersection point of these lines.

M. WEB DESGIN







Doing Vision Studies with Computer Using Drones

KAYNAKLAR

[1] https://pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/

[2] https://kediarahul.medium.com/vanishing-point-detection-with-opencv-python-c-4d1ed00a8cd5.
https://www.youtube.com/c/MurtazasWorkshopRoboticsandAI