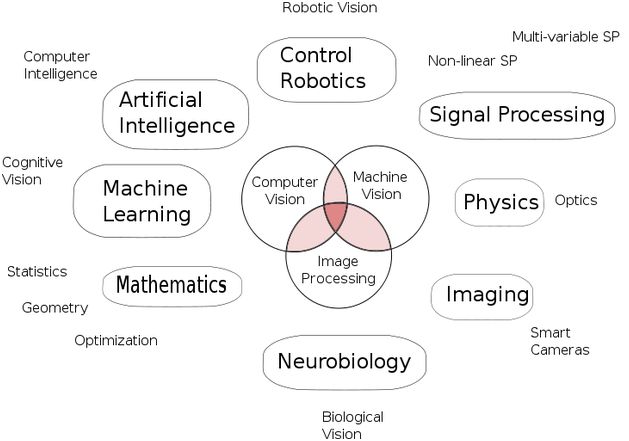
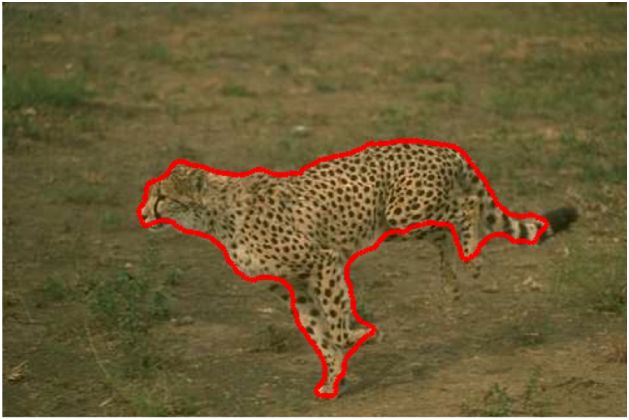
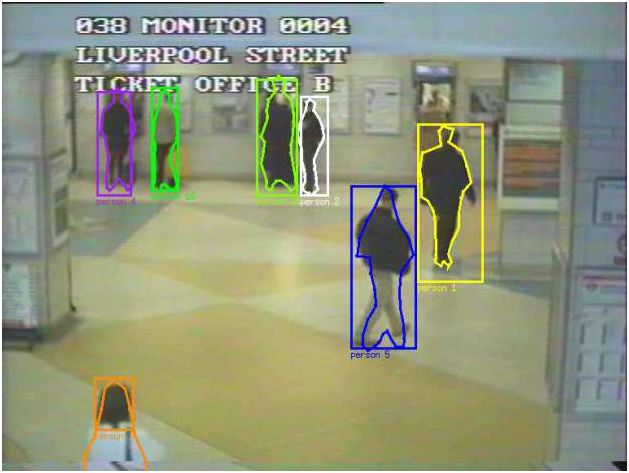
A Quick Introduction To Computer Vision Using C#

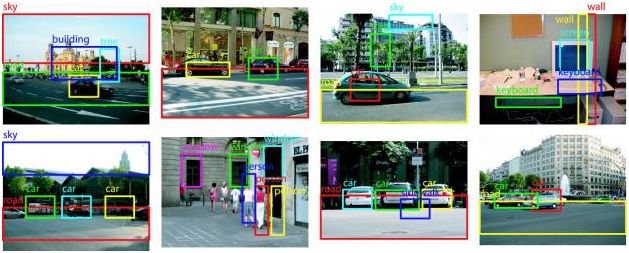
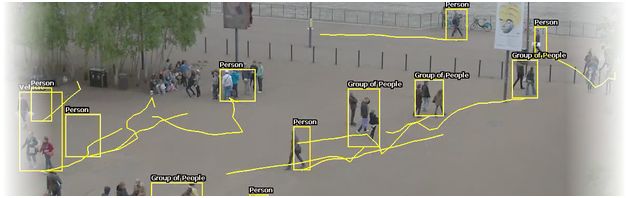
In this article you will learn about computer vision. The origins of computer vision come from the image processing field; image processing comes from signal processing.

**Introduction and Background**I have been writing many C# articles and how-to's for .NET framework which mostly cover the simplest of the tasks in C# programming such as GUI manipulation, performance enhancement, networking programming and other similar tasks. However, I have never been into computer vision or artificial intelligence, although I did talk about a few concepts a very long time ago. In this post I am going to give you a very quick but brief overview of “*computer vision*“, which is a field of artificial intelligence in computer science.  
  
  
**Figure 1-** *Computer vision and computer intelligence fields all connected up showing the relationship between fields*  
  
Now in this post what I am going to talk about is trials from computer intelligence. The origins of computer vision come from the image processing field, and  image processing comes from signal processing. I will walk you through many areas of computer science where computer vision can be deemed useful in every day usage. Computer vision is also a subject of study in practical research where you have to perform some real life image processing to detect objects, track objects, and determine where an object is. Now at this point you must understand that a face, hand, or entire human is still considered an object in image processing. It is just the algorithm that makes the program determine a face as an object, or determine an entire human person in the image as an object. That is not all, computer vision has been widely used in many other fields too, and I am going to list a few  in the following section on the application of computer vision.  
  
What you are required to know is, although computer vision is a very skilled field to dive into, you must have at least an intermediate level knowledge and understanding of computer systems, image formats, how matrices work (most of the image processing systems process the images in matrix form, MATLAB, OpenCV and others do the same, that is why it would be very helpful to you if you have the basics of matrices, not the advanced inversion, transposition etc., but the basics like what is row and column). You must also understand what the difference between an RGB image, grayscale image and black and white image is. Rest assured, I will make sure I speak English so that it is easier for you to learn and understand the concept of computer vision and how you can use it in your own applications.   
  
**Computer vision**  
*"Computer vision is the field of computer science, in which the aim is to allow computer systems to be able to manipulate the surroundings using image processing techniques to find objects, track their properties and to recognize the objects using multiple patterns and algorithms." — I made the definition myself.*  
  
I have recently stepped into the computer vision and artificial intelligence field and trust me, it is as interesting as (and sometimes even more interesting than) simple graphics programming. I have been doing a lot of graphics programming, I have been doing a lot of event driven programming and a lot of similar desktop programming in many programming languages. But trust me, this thing is very interesting, being able to make the computers “*think and see like you.*“  
  
There are many books written for this subject and many other guides that have been provided to you that cover most of the theoretical portions of this field of computer science. However, I am not going to cover the theoretical portion in this post, instead I will be walking you through using computer vision in your real world applications and other similar projects so that you may get that required image processing and have a bit of a sense of computer vision in them. The applications of computer vision range from minimal household stuff, to robotics, to security equipment such as facial recognition for allowin passage. That is again, not only it, computer vision can also support object tracking which means that your programs can recognize objects and track them in the graphics frames that can be accessed from a webcam or any other optical device that is capable of providing images to the computer. Now before I start showing all of the capabilities of computer vision systems and programs, I want to start the section where it would seem a bit more on-topic.  
  
**Applications of computer vision**  
The applications of computer vision are really vast, vast in the sense because even though there are subfields of this field that are active for research, there are more designs to be followed by computer scientists to get efficient results and to get more performance and efficiency in computer vision. Let me give you a few examples of where computer vision can be used to enhance the productivity of your work in the office, schools, homes and societies around you. Computer vision uses the technology of image processing to process the images in a fraction of a second and uses the algorithm sets to detect,

1. Objects in our images
   * As I had mentioned, matrix manipulation allows them to detect where objects are, they use the binary representation of the images.
   * Objects can be the “geometry or pattern of interest”.
   * Facial detection (not recognition) is an example of this.
2. Object recognition  
     
   * Now this field is a bit different from previous ones. Because in this field, we recognize previously “learned” patterns.
   * Facial recognition comes in this sub field of computer vision.
3. Object tracking  
     
   * It can find the objects that are in the image and then it follows them in the frame.
4. Image reconstruction  
     
   * Pass in multiple frames at the same area and get the reconstruction of the image properties using the samples provided.
5. Video processing  
     
   * It can be similar to object tracking because it processes video frames to detect the objects and then tracks them in the video frames.
   * It can also detect the speed of the objects moving.
   * It can detect when an object starts to go into “*detection zone*” such as detecting when your baby starts to come near a door or window and so on.

Let us enlist a few of the services thatthe computer vision field can provide us with.  
  
  
**Figure 2 -***Computer vision and image processing cloud***1. Object detection**  
  
The simplest fields of computer vision are object detection, to detect the objects based on a pattern of geometry, such as detecting faces, detecting human bodies, detecting animals etc. Object detection takes a bit of a pattern to follow to detect the object. There are many algorithms used to do the same thing too. For example, in the binary image you can see where an object starts and where it ends using the binary matrix. This procedure is finding connected components. Connected components are then used to determine what the boundaries of an object are. Using those boundary pixels, program can then generate a pixelated boundary of any color to show where that object is. For example, have a look at the following image where a leopard has been identified and a boundary has been drawn around it to demonstrate its boundaries.  
  
  
**Figure 3 -** *Leopard detected and bound in red boundary*Of course, a computer needs to perform a mathematical calculation and the results are not guaranteed to be 100 percent correct. If you have a look at the background, the colors match very much and that makes it hard for the computer to visualize the leopard in contrast to the background. Have a look at the following image on other hand,  
  
  
**Figure 4 -***Humans detected and labelled with their IDs and colors*In this example, you can see that the computer has started to process the human objects and has also created their boundaries. Notice that this is a live stream that goes to the computer. This is an example of “real-time” image processing and computer vision. The humans are walking around in that area and computer is able to determine the number of people, it has also labelled them in different colors. This has been all because of the image processing techniques that have been applied to get the results in computer vision. Now our program can follow these objects and see what they are doing. If we can add gesture information to it to support reading and understanding the gestures, we can allow the computer to determine if there has been a fraud in the system by a human and so on.  
  
**2. Object recognition**  
This is similar to what we have seen in the previous section, but the only difference is that in this field we already have detected the objects, we are only matching the results to what we have previously stored in the database. For example, your face has been detected by the computer, and next time you visit the system again, it won’t ask you for your name or credentials, instead it will simply recognize you as who you are.  
  
The method is very much simple. The following steps are taken to perform this,

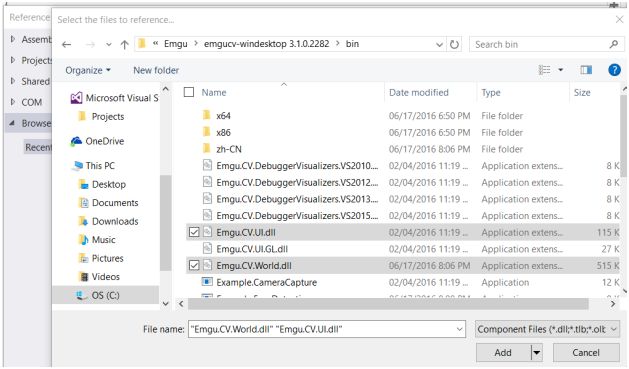
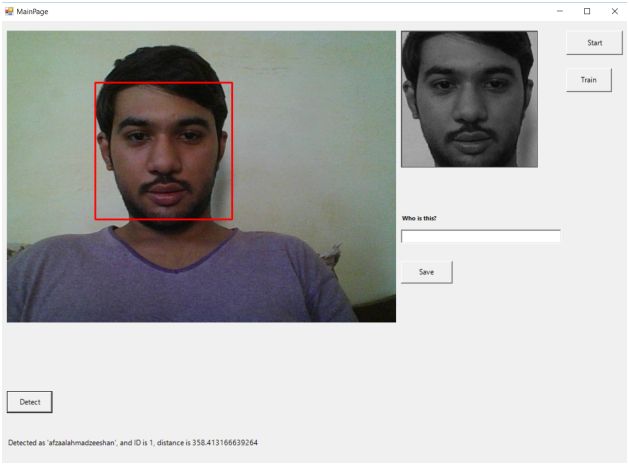
1. An object is detected.
2. The detected object and its boundaries are taken, the image is then saved at those boundaries. Of course you can get the bytes of that image.
3. You store multiple samples for each object.
4. You then again detect the object but this time after detection, you try to match it with the objects that are already in the database. You can use a bunch of algorithms, such as Fisher, Eigen etc.
5. The result is provided as either a name of the object or as unnamed.

This way, your objects (faces, bodies, materials, textures) etc are recognized in the computer and the computer tells you what it was.  
  
  
**Figure 5 -***Objects detected and recognized*As you can see, the computer program is capable of understanding what objects are present in the image and then it recognizes the “known” objects by matching them up against the already saved results in the database.  
  
**3. Object tracking**  
This type of program is based on video analysis, such as fetching the frames from a webcam and then performing image processing techniques on the image itself to learn more about the image and the objects in the image. Not just that, this field of computer vision also stores the states of the objects, such as their location in the frame. Thenthe program tracks the objects in the frames and follows them as they move around in the frames of the video that is being streamed from the webcam. A demonstration can be found in the following representation,  
  
  
**Figure 6 -***Human tracking in the frames of a webcam or surveillance camera*As seen in the image above, the program has detected the humans and has been tracking them for awhile. Notice how it has trailed their paths with a line that shows where and how they were moving in the image frames.  
This field is (and can be) only applicable in video processing, in image processing (where only one frame is available) this cannot be applied because objects are not going to move here and there.  
  
**4. Rest of the stuff**  
There are many other uses of computer vision, that, if discussed, will cause this post to become very long in length. Keeping the post considerate and short, I am going to talk about a few other uses of computer vision in regular day life.  
  
Suppose you wanted to create a security system which keeps  track of the people standing in the street. You can do so by computer vision --  you are allowed you count the number of people standing in the street and once the number rises to a “threshold”, you can trigger an alarm or do what you want to. This is one use.  
  
Otherwise, suppose you were given a few shots of a person from different angles, you can use manipulation of those images to reconstruct a 3D image of that person. 3D image reconstruction is a very open area of research in computer vision and can make good use of some popular research papers in the same field.  
  
I am sure, by now, you must be able to perceive how interesting, amazing, and helpful this field of computer science can be to your projects, to your clients, and to your household applications. I have recently build a server for my home, I am definitely going to apply these to the server and make use of my skills and the power of computer vision to provide features like, services, security and other stuff that I can make with computer vision. In the coming section, I will talk about libraries of computer vision that are actively being used in regular everyday projects by everyone.  
  
**Libraries Available**Do you remember, when you were told: Do not reinvent the wheel?  
  
I am sure you do. Likewise, you should never consider rewriting everything from scratch that would consume a lot of time and won’t give any better results at all. There are many libraries written for computer vision and image processing. I would always recommend that you use those libraries. In this section, I will walk you through different libraries and why you should use them.  
  
**The Mighty OpenCV**I have tried and used OpenCV frameworks for computer vision programming. Trust me, it is the best library available out there. There are many other libraries and tools available, in which MATLAB is one such tool that allows image processing. But who wants to spend a bunch on a tool that can be used free of cost? I have used MATLAB and I have used OpenCV. OpenCV is my personal recommendation.  
  
A few of the uses and benefits of using OpenCV are,

1. Open source and free — I just couldn’t find any better reason to use OpenCV than this, to be shared as the first benefit.  
     
   * OpenCV is led and updated by the community. So there is a lot less patent stuff, organization philosophy and “how to use it” stuff going on. You are free to use it anyway, anytime and for any project.
2. Written natively in C and C++. C++ APIs of OpenCV provide excellent performance and efficiency for real-time processing of images.
3. Cross-platform support.  
     
   * Can run on Android.
   * Can be used in iOS applications.
   * Desktop and server applications can surely use the power of OpenCV.
4. Supports the world’s well known frameworks,  
     
   * It supports OpenCL. OpenCL is the open computer language, written in C programming language that runs on any machine hardware regardless of their architectures. OpenCV can use OpenCL to enhance the performance and minimize the compile-time modification of code.
   * Intel IPP is well known for image processing. OpenCV can use Intel IPP if it detects it to be installed on the machine.
5. Many giants have contributed source code to the repository.  
     
   Google, Intel and Microsoft had contributed a good amount of source code to the repository fixing and adding a lot of features on multiple platforms, which makes OpenCV one of the best available libraries for computer vision out there.  
     
   Features are “*amazingly*” outstanding and way too much to handle,  
     
   * It supports both formats for images; 2D and 3D.
   * Object detection and recognition systems.
   * Gesture recognition.
   * Object tracking.
   * Robotics.

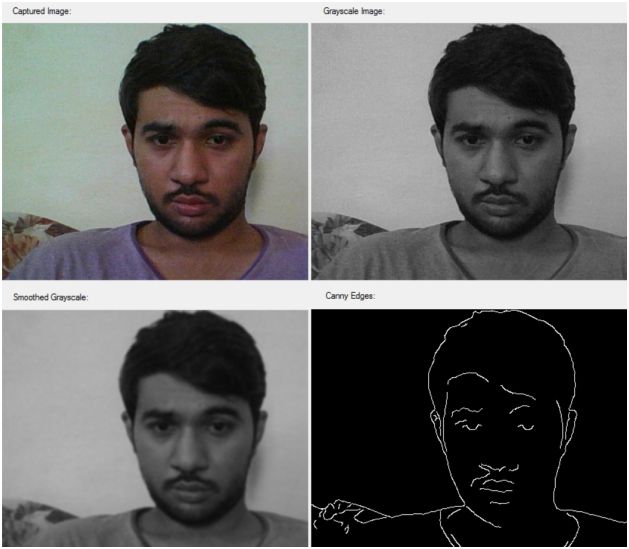
I will be writing a few of the guides for these features soon, but for this one I am just going to give you an overview of the library itself and how you can use this library in your own .NET framework applications.  
  
**Other libraries**  
There are other libraries available too, I don’t want to say they are not available or that they cannot be used. But the fact is, they all have a bit of glitches and as I have said, OpenCV is free from philosophical junk and is community-driven. What is loved in the community, stays in the library, what is not liked, gets removed. Other libraries have a few issues in them,

1. Some are not being monitored and fixed. Some were last reviewed or modified back in 2005.
2. Some have issues in programming. Lack of good programming paradigms for instance.
3. Bad performance as compared to the rest.

So on and so forth.   
 **Using OpenCV in .NET Environment**  
Since I have said OpenCV is a library written in native C and C++ programming languages, then how can you use it in C#? Luckily, C# supports some of the unsafe programming too, unsafe in the terms of unmanaged programming, where anything can go wrong. There are many wrappers written in C# that allows you to communicate with OpenCV library runtime. These wrappers allow your C# programs to be able to communicate with the APIs of the underlying library that was written with unmanaged code. Unmanaged code can be the code that manipulates the memory itself, such as in the case of C++.  
  
Emgu CV is one such wrapper, I have used it in my own .NET applications and it is very simple to use. Although, just like other wrappers, it does have some issues such as compatibilities if you are a C++ programmer, the design may bug you a bit, because the wrapper was written keeping C# in mind. From this step, I will give you a bit of an idea of using Emgu CV in your own programming in .NET framework.  
  
**Adding the library**Emgu CV must be installed before it can be used in your projects. First of all, go and download the packages from the [website](http://www.emgu.com/wiki/index.php/Download_And_Installation), one thing to know here is that the versions 2.4 and 3 are different and the version 3 has everything rewritten and rethought from the ground up! So think twice before downloading the packages, however both of the packages come shipped with the sample source code that gets you ready and set up in no time! You can try out those source samples in your own machine and see for yourself how things work. But remember, Emgu CV 3.0 has an enormous amount of performance and bug fixes but currently the most stable release is version 2.4.  
  
The library needs to be available in your executable’s directory, the output folder where your executable resides. You can add the library as an existing item to the directory, or you can use the References → Add Reference → Emgu CV libraries.  
  
  
**Figure 7 -***Addition of Emgu CV libraries*These two libraries are required at least to perform the least of the processing in your applications. They are required to be available when your application starts to execute. You can use this library in any of your .NET applications, I started with Console application and raised it to Windows Forms, which leads me to other frameworks to be also supported and so on.  
  
**Sample work**  
Sure, I would love to get you to have a look at what I built recently, sadly I am going to let yousee  only the visual of that and not the source code, I am leaving the source code for some time later.  
  
  
**Figure 8 -** *I am being detected. Application shows other controls and extra information in the window, too*  
I was able to create this application in much less time using Emgu CV in .NET framework. The frameworkis being used in Windows Forms as well, if you see thoroughly you can see that this application has quite a few features in itself.

1. Face detection.
2. Face tracking, notice that the grayscale image only holds the face that has been detected. It follows and tracks the face in the frame.
3. I added a bit of face recognition features to the application too.  
     
   1. I added the support for training the model.
   2. I added the support for saving the bitmaps of the face patterns associated with the person names. See the textbox.
   3. I added the button to start the detection of the face as a person.
   4. In the last line, notice how the application tells to whom this face belongs. I created a database system to hold the information; that is from where the ID comes from.

I will be sharing this application with you guys soon, at the moment I need to make a few more changes to the application to support even more functionality and to remove any bugs in the algorithms.  
  
**Extra features - Image processing tools**  
That was the sample of what I had created, that was not just what Emgu CV is capable of doing, there are many more features to it than just detecting the faces in the image frames. However, some of these functions are directly related to image processing way more than they are related to computer vision.

1. **Edge detection**The following image, is a sample from Emgu CV team and it shows how it can process the image coming from the webcam and detect edges using a canny algorithm, and convert it to grayscale too.  
     
     
   **Figure 9 -***My face being captured and being processed using image processing to get information such as edges, grayscale image etc*
2. **Grayscale images**You have already seen what features Emgu CV has to support grayscale image creation and processing.
3. **Histogram equalization**It is a well-known image processing technique used in many ways and for many purposes, one of the purposes that I remember in my “*Digital image processing*” class was to enhance the contrast of the images in MATLAB using histogram equalization techniques. Emgu CV uses the interfaces of OpenCV to perform the histogram equalization, the technique is pretty simple, you just call the function on the image objects. Such as the following code,
   1. Image<Bgr, **byte**>inputImage = GetImage();
   2. inputImage.\_EqualizeHist();

You will learn more on this in the coming posts that I will be writing soon. So, stay tuned.

**Performance management**Image processing and computer vision require a lot of computer power to perform the actions that you want to get results from. Even the simplest of the programs require a complex matrix manipulation and would require your computer to perform a huge amount of computation to get the results. So, I am pretty sure you need to understand how you can minimize the overheads.  
 **Parallelism**If you have a computer machine that has multi-cores, dual or quad cores, then surely you can use parallelism to enhance and increase the performance of your application using threadings and send the data to be processed as multiple threads and cores. This can help. But remember, parallelism is totally relative to the algorithm being used. Sometimes, it can help you a lot, sometimes it can just be overkill or sometimes it may be the reason for extra time being taken to process.  
  
To learn more about this, I recommend going through the following links:

1. [Parallel computing](https://en.wikipedia.org/wiki/Parallel_computing)
2. [Parallel Programming in the .NET Framework](https://msdn.microsoft.com/en-us/library/dd460693(v=vs.110).aspx)

I would still say, use with caution.  
  
**Code tuning**  
In my own applications, I consider using this mechanism to enhance the performance of the applications. I fine tune the application’s code to increase the performance. There are many reasons why this should be used, the first one being, your customers may not have the fastest computer machines.  
  
In such cases, the best way of improving the performance and efficiency of the application is to improve the code being used.  
  
**Final words**This post was created just to give you an idea of how you can use Emgu CV in your .NET applications to perform computer vision programming in your applications. In coming posts, I will show you how you can perform these actions in your application in the form of source code.  
  
I hope you enjoyed reading this, if I missed something let me know and I will talk about that in the coming posts… See you in the next posts!