International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Image Recognition Process through Human Eye, Computer and Artificial Intelligence

Dr. Neetu Agarwal

Assistant Professor

Abstract: ***Now days artificial intelligence is using in many fields. In this branch of science, a machine can work like a human being. It can work in any field like fraud detection, video games, smart cars, smart home devices, robotics, image recognition etc. In this paper, some basic definitions are mentioned of artificial intelligence and image processing. The process for object recognition, color identification, image formation through human eye, computer and artificial intelligence is also described.***

Keywords: Artificial Intelligence, Image Recognition, photoreceptor cells, CMYK, electron gun, Histogram of Oriented Gradients (HOG), phosphor etc.

1. Definitions of Artificial Intelligence

Artificial intelligence is the branch of computer science concerned with making computers behave like humans. The term was coined in 1956 by John McCarthy at the Massachusetts Institute of Technology.

According to the father of Artificial Intelligence, John McCarthy, it is *"The science and engineering of making intelligent machines, especially intelligent computer programs"*.

Artificial Intelligence (AI) is usually defined as the science of making computers do things that require intelligence when done by humans.

Artificial intelligence is a branch of computer science that aims to create intelligent machines. It has become an essential part of the technology industry.

Artificial intelligence is the replication of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using the rules to reach approximate or definite conclusions), and self-correction.

Artificial Intelligence is a way of making a computer, a computer-controlled robot, or a software think intelligently, in the similar manner the intelligent humans think.

AI is accomplished by studying how human brain thinks, and how humans learn, decide, and work while trying to solve a problem, and then using the outcomes of this study as a basis of developing intelligent software and systems.

2. Definitions of Image Recognition

Image recognition is the process of identifying and detecting an object or a feature in a digital image or video. This concept is used in many applications like systems for factory automation, toll booth monitoring, and security observation. Typical image recognition algorithms include:

Image recognition – also known as computer vision – is the method of acquiring, analyzing, and understanding images to produce numerical information. In other words, image recognition is a computer's way of doing what your eye does: see a picture and understand it.

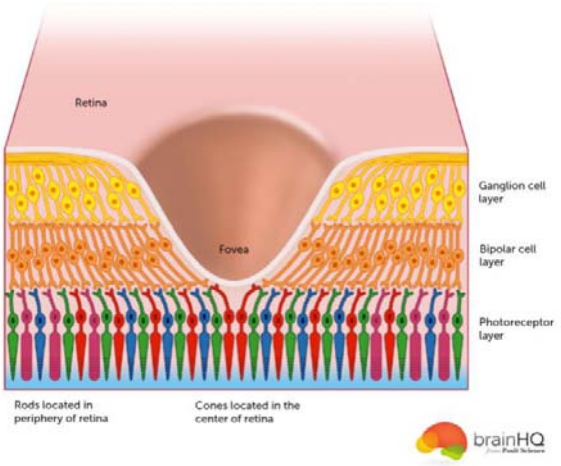
Image recognition is the identification of objects in an image. This process would probably start with image processing techniques such as noise removal, followed by (low- level) feature extraction to locate lines, regions and possibly are a swith certain textures. Image recognition technologies try to identify objects, people, buildings, places, logos, and anything else that has value to consumers and enterprises.

Smartphones and tablets equipped with cameras have pushed this technology from mainly industrial applications (for example, sorting fruit) to consumer applications. For example, logos, cars, landmarks, wine labels, and book and album covers can be identified by consumer smartphones, using a mobile app that accesses image recognition software in the cloud.

Image recognition has the potential to transform a picture into a hyperlink to something on the Internet (for example, information, service, coupon or video).It can also be used to initiate a search—which is one of the main reasons why Google and Amazon have been investing in this technology.

Image recognition has applications in security and content management. Our definition of image recognition does not include video analytics or video search, although video analysis frequently incorporates image recognition technologies.

3. Image recognition process in Human beings



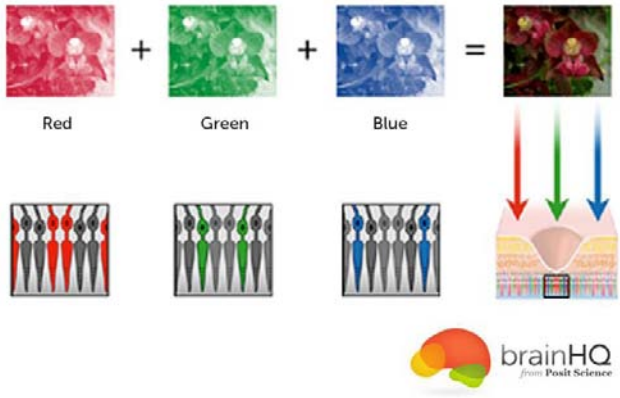
The neural mechanisms of visual perception offer rich insight into how the brain handles such computationally complex situations.

Visual perception begins as soon as the eye focuses light onto the retina, where it is absorbed by a layer of photoreceptor cells. These cells convert light into electrochemical signals, and are divided into two types, rods and cones, named for their shape. Rod cells are responsible for our night vision, and respond well to dim light. Rods are found mostly in the peripheral regions of the retina, so most people will find that they can see better at night if they focus their look just off to the side of whatever they are observing.

Cone cells are concentrated in a central region of the retina called the fovea; they are responsible for high acuity tasks like reading, and also for color vision. Cones can be subcategorized into three types, depending on how they respond to red, green, and blue light. In combination, these three cone types enable us to perceive color.

Signals from the photoreceptor cells pass through a network of interneurons in the second layer of the retina to ganglion cells in the third layer. The neurons in these two retinal layers exhibit complex receptive fields that enable them to detect contrast changes within an image; these changes might indicate edges or shadows. Ganglion cells gather this information along with other information about color, and send their output into the brain through the optic nerve.

We can only see 3 colors! Inside our eyes, the images we see pass through the lens and focus onto the retina. The retina is covered with special cells, which are sensitive to light. These cells come in three types: Red sensitive, Blue sensitive and Green sensitive. Everything we see is gathered as a collection (like a grid! again we have a grid) of red, green and blue information. This information is sent back to our wonderful brain where it is interpreted. By analyzing the strength of the red, green and blue signals from the eyes, the brain creates our perception of images and all the colors of the rainbow.



This happens continuously and in our brain and our brain is one SUPER computer!

4. Image recognition process in Computer

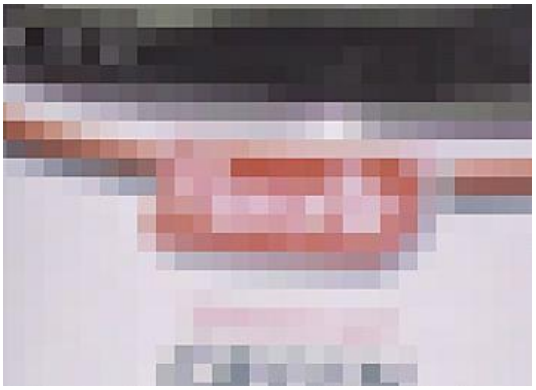
Digital Image is often used to describe a computer generated image.Digital is referring to the two digits of the binary number system...0 and 1. Each binary number is called a "bit".All the information and commands used within a computer are expressed in this binary number system. Computers may offer higher forms of notation such as our alphabet and the decimal numbers, but they are first stored as combinations of 0s and 1s.A group of eight bits is called a "byte". It might look like this: 01100111. If you add up all the possible combinations of eight 0s and 1s you will arrive at the number 256. This is 2^8. If we assign more bits to each pixel, we can get more colors out of every pixel.

Every bit doubles the number of colors: 1-bit describes 2 colors, 2-bit describes 4 colors, 3-bit describes 8 colors, 4-bit describes 16 colors...and so on. Our computers usually work with of 8-bits of data. This gives 256 colors, or levels of gray tones.

The RGB or CMYK (Cyan, Magenta, Yellow, Key (black)) channels in Photoshop are not necessarily a "color"; they are data, which displays in a channel as a gray image and will be translated to color only when interpreted by an output device - monitors, printers or printing processes.Photoshop channels in addition to the image channels are used as masking or selection tools.The pixels in a mask channel can be instructed to apply an effect on image pixels at the same raster location. Computer pictures are really information, describing tones or colors. This is stored in a file similar to a spread sheet.
Let's look at a computer picture.

This picture is really a grid with 400 squares across and 257 down.

Now take a look at an extreme close-up of the red logo on the can of beans...



Look, it's really row after row of little squares. These squares are the most basic building blocks of any computer graphic; they are called PICTURE ELEMENTS...PIXELS, for short.

Each and every one of this photo's pixels is recorded in the computer file. The file has the location, or address, of each pixel on the grid (440 x 257 squares) and the description of the color of each pixel.

Bitmapped image is a term that comes from this "map" of pixels. Below is a small portion of the map of a 24-bit image.

R:246 G:227 B:234	R:180 G:156 B:165	R:246 G:227 B:234	R:59 G:49 B:53	R:221 G:185 B:199	R:221 G:185 B:199	R:221 G:185 B:199
R:180 G:156 B:165	R:32 G:32 B:32	R:112 G:112 B:112	R:143 G:143 B:143	R:59 G:49 B:53	R:180 G:156 B:165	R:255 G:204 B:204
R:89 G:74 B:80	R:64 G:64 B:64	R:158 G:40 B:32	R:143 G:143 B:143	R:64 G:64 B:64	R:89 G:74 B:80	R:255 G:204 B:204
R:89 G:74 B:80	R:143 G:143 B:143	R:191 G:191 B:191	R:180 G:156 B:165	R:45 G:41 B:42	R:133 G:111 B:119	R:221 G:185 B:199
R:221 G:185 B:199	R:59 G:49 B:53	R:64 G:64 B:64	R:32 G:32 B:32	R:180 G:156 B:165	R:221 G:185 B:199	R:221 G:185 B:199
R:221 G:185 B:199	R:221 G:185 B:199	R:221 G:185 B:199	R:221 G:185 B:199	R:221 G:185 B:199	R:221 G:185 B:199	R:221 G:185 B:199
R:221 G:185 B:199	R:221 G:185 B:199	R:221 G:185 B:199	R:221 G:185 B:199	R:221 G:185 B:199	R:221 G:185 B:199	R:221 G:185 B:199

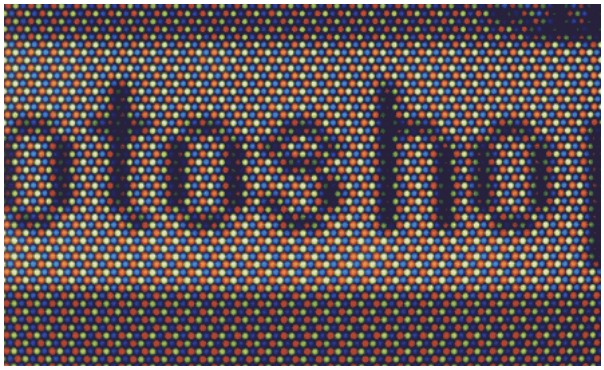
This is a huge amount of information and that is why graphics files are so large.

5. How does the Monitor do it?

The monitor uses an electron gun to shoot a stream of electrons (electricity) onto a chemical called a phosphor. When electrons strike a phosphor, the phosphor glows, sending off light. The phosphors come in three types; one emits Red light, one emits Green and one emits Blue. They are like tiny colored light bulbs, each on its own dimmer switch.

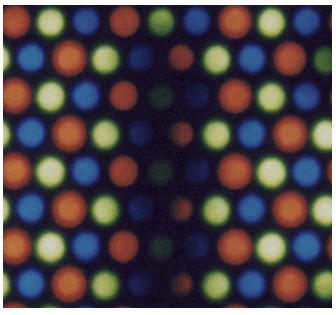
Now it gets tricky. To create a full screen image the electron gun is swept across the screen horizontally, row by row from top to bottom. It must do this cycle very quickly and repeat itself to keep those phosphors lit up. This cycling speed is what is called the refresh rate. This is usually around 75 times a second, or 75Hz.

The phosphors are arranged in a pattern and a very thin metal mask is used to define the pattern precisely. Below is a close-up of monitor (this is a very old non-multiscan model, so the pattern is very simple).

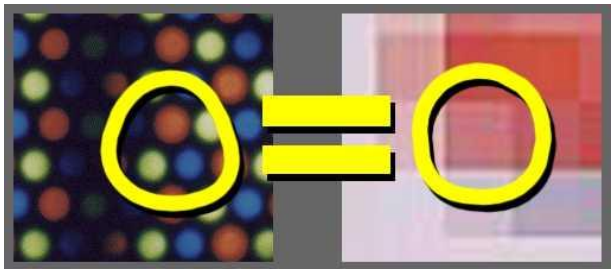


The light and dark tones are created by varying the amount of electricity the electron gun shoots at each pixel. The more electrons the brighter the color. The brightest area around the lettering is what you would see as white when you look at the screen. Because they are too small for our eyes to separate, the red, green and blue lights are seen as white when they reach our brains.

Red, green and blue are called the "Additive Primary Colors" because they form white light when added together. Here is an even closer view.



Each of those triangular clusters of a red, green and blue dot is the method your monitor uses to describe each of those pixels we saw above in the can of green beans photo. A square pixel is displayed by a 3-color cluster of phosphor dots on the monitor.



Suppose monitor is set to display 800 pixels wide and 600 pixels tall. The electron gun knows how many electrons to shoot at every red, green and blue dot on the screen. The beam of electrons zips across the rows of dots, lighting up the phosphors. It goes line-by-line, top to bottom shooting 800 red, 800 green and 800 blues dots on every line. It does each of the 600 rows to cover the screen. The phosphors only glow for an instant, so the electron gun shoots the entire screen 75 times every second to keep the image bright.

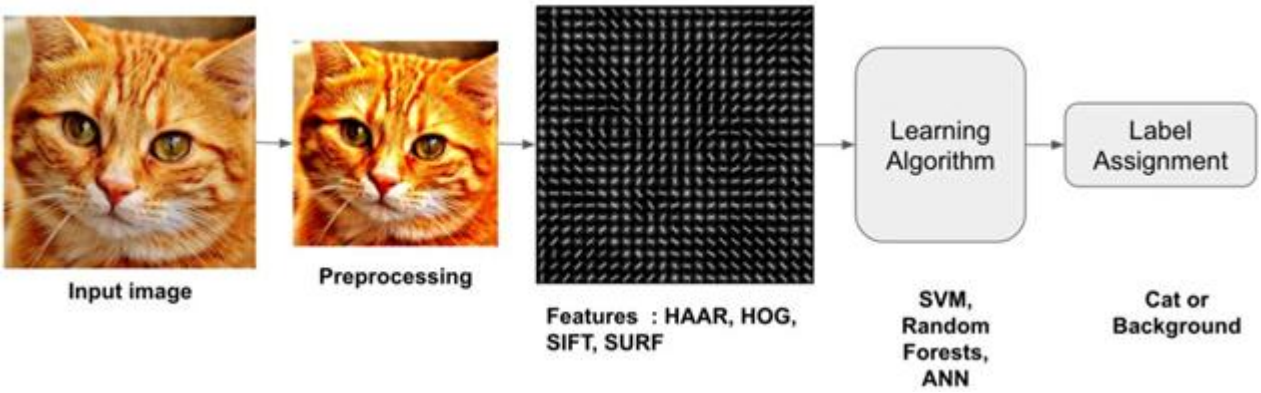
By the way, the electron gun is a stream of electrons which pass by a set of electromagnets which deflect the stream. The variance in the magnets is what moves the beam from side to side and up and down. This all happens in a vacuum inside the picture tube of your monitor.

6. Image recognition with Artificial Intelligence

Image Recognition
An image recognition algorithm takes an image (or a patch of an image) as input and outputs what the image contains. In other words, the output is a class label (e.g. ~~-c~~at", ~~-d~~og", ~~-t~~able" etc.). How does an image recognition algorithm know the contents of an image ? Well, you have to train the algorithm to learn the differences between different classes. If you want to find cats in images, you need to train an image recognition algorithm with thousands of images of cats and thousands of images of backgrounds that do not contain cats. Needless to say, this algorithm can only understand objects / classes it has learned.

To simplify things, in this post we will focus only on two-class (binary) classifiers. You may think that this is a very limiting assumption, but keep in mind that many popular object detectors (e.g. face detector and ordinary detector) have a binary classifier under the hood. E.g. inside a face detector is an image classifier that says whether a patch of an image is a face or background.

Anatomy of an Image Classifier
The following diagram illustrates the steps involved in a traditional image classifier.



Interestingly, many traditional computer vision image classification algorithms follow this pipeline, while Deep Learning based algorithms bypass the feature extraction step completely. Let us look at these steps in more details.

Step 1: Preprocessing
Often an input image is pre-processed to normalize contrast and brightness effects. A very common preprocessing step is to subtract the mean of image intensities and divide by the standard deviation. Sometimes, gamma correction produces slightly better results. While dealing with color images, a color space transformation (e.g. RGB to LAB color space) may help get better results.

As part of pre-processing, an input image or patch of an image is also cropped and resized to a fixed size. This is essential because the next step, feature extraction, is performed on a fixed sized image.

Step 2: Feature Extraction
The input image has too much extra information that is not necessary for classification. Therefore, the first step in image classification is to simplify the image by extracting the important information contained in the image and leaving out the rest. For example, if you want to find shirt and coat buttons in images, you will notice a significant variation in RGB pixel values. However, by running an edge detector on an image we can simplify the image. We can still easily distinguish the circular shape of the buttons in these edge images and so we can conclude that edge detection retains the essential information while throwing away non-essential information. The step is called**feature extraction**. In traditional computer vision approaches designing these features are crucial to the performance of the algorithm. Turns out we can do much better than simple edge detection and find features that are much more reliable. In our example of shirt and coat buttons, a good feature detector will not only capture the circular shape of the buttons but also information about how buttons are different from other circular objects like car tires.

Some well-known features used in computer vision are Haar-like features introduced by Viola and Jones,Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Feature (SURF) etc. As a concrete example, let us look at feature extraction using Histogram of Oriented Gradients (HOG).

7. Histogram of Oriented Gradients (HOG)

A feature extraction algorithm converts an image of fixed size to a feature vector of fixed size. In the case of pedestrian detection, the HOG feature descriptor is calculated for a 64×128 patch of an image and it returns a vector of size 3780. Notice that the original dimension of this image patch was 64 x 128 x 3 = 24,576 which is reduced

to 3780 by the HOG descriptor. HOG is based on the idea that local object appearance can be effectively described by the distribution (histogram) of edge directions (oriented gradients). The steps for calculating the HOG descriptor for a 64×128 image are listed below.

1) **Gradient calculation** : Calculate the x and the y gradient images, $g_x$and g_y, from the original image. This can be done by filtering the original image with the following kernels.

-1	0	1

-1
0
1

Using the gradient images $g_x$and g_y, we can calculate the magnitude and orientation of the gradient using the following equations.

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

The calculated gradients are —unsigned" and therefore θ is in the range 0 to 180 degrees.

2) **Cells**: Divide the image into 8×8 cells.
3) **Calculate histogram of gradients in these 8×8 cells** : At each pixel in an 8×8 cell we know the gradient (magnitude and direction), and therefore we have 64 magnitudes and 64 directions — i.e. 128 numbers. Histogram of these gradients will provide a more useful and compact representation. We will next convert these 128 numbers into a 9-bin histogram (i.e. 9 numbers). The bins of the histogram correspond to gradients directions 0, 20, 40 … 160 degrees. Every pixel votes for either one or two bins in the histogram. If the direction of the gradient at a pixel is exactly 0, 20, 40 … or 160 degrees, a vote equal to the magnitude of the gradient is cast by the pixel into the bin. A pixel where the direction of the gradient is not exactly 0, 20, 40 … 160 degrees splits its vote among the two nearest bins based on the distance from the bin. E.g. A pixel where the magnitude of the gradient is 2 and the angle is 20 degrees will vote for the second bin with value 2. On the other hand, a pixel with gradient 2 and angle 30 will vote 1 for both the second bin (corresponding to angle 20) and the third bin (corresponding to angle 40).
4) **Block normalization**: The histogram calculated in the previous step is not very robust to lighting changes. Multiplying image intensities by a constant factor scales the histogram bin values as well. To counter these effects we can normalize the histogram — i.e. think of the histogram as a vector of 9 elements and divide each element by the magnitude of this vector. In the original HOG paper, this normalization is not done over the 8×8 cell that produced the histogram, but over 16×16 blocks. The idea is the same, but now instead of a 9 element vector you have a 36 element vector.
5) **Feature Vector**: In the previous steps we figured out how to calculate histogram over an 8×8 cell and then normalize it over a 16×16 block. To calculate the final feature vector for the entire image, the 16×16 block is moved in steps of 8 (i.e. 50% overlap with the previous block) and the 36 numbers (corresponding to 4 histograms in a 16×16 block) calculated at each step are concatenated to produce the final feature vector.

What is the length of the final vector?

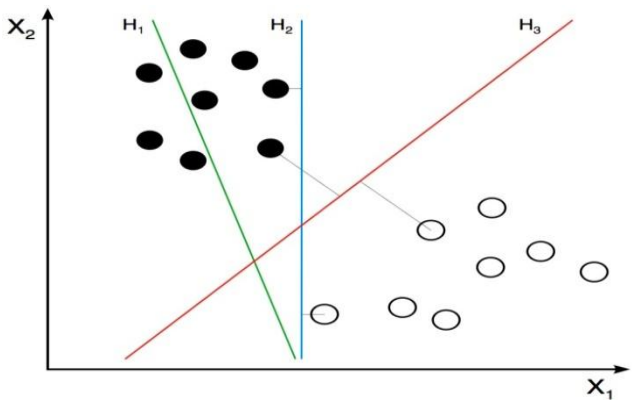
The input image is 64×128 pixels in size, and we are moving 8 pixels at a time. Therefore, we can make 7 steps in the horizontal direction and 15 steps in the vertical direction which adds up to 7 x 15 = 105 steps. At each step we calculated 36 numbers, which makes the length of the final vector 105 x 36 = 3780.

Step 3: Learning Algorithm For Classification

Before a classification algorithm can do its magic, we need to train it by showing thousands of examples of cats and backgrounds. Different learning algorithms learn differently, but the general principle is that learning algorithms treat feature vectors as points in higher dimensional space, and try to find planes / surfaces that partition the higher dimensional space in such a way that all examples belonging to the same class are on one side of the plane / surface. To simplify things, let us look at one learning algorithm called Support Vector Machines (SVM) in some detail.

How does Support Vector Machine (SVM) Work For Image Classification?

Support Vector Machine (SVM) is one of the most popular supervised binary classification algorithm. Although the ideas used in SVM have been around since 1963, the current version was proposed in 1995 by **Cortes** and **Vapnik**. Visualizing higher dimensional space is impossible, so let us simplify things a bit and imagine the feature vector was just two dimensional.



In our simplified world, we now have 2D points representing the two classes (e.g. cats and background). In the image above, the two classes are represented by two different kinds of dots. All black dots belong to one class and the white dots belong to the other class. During training, we provide the algorithm with many examples from the two classes. In other words, we tell the algorithm the coordinates of the 2D dots and also whether the dot is black or white.

Different learning algorithms figure out how to separate these two classes in different ways. Linear SVM tries to find the best line that separates the two classes. In the figure above, H1, H2, and H3 are three lines in this 2D space. H1 does not separate the two classes and is therefore not a good

classifier. H2 and H3 both separate the two classes, but intuitively it feels like H3 is a better classifier than H2 because H3 appears to separate the two classes more cleanly. Why ? Because H2 is too close to some of the black and white dots. On the other hand, H3 is chosen such that it is at a maximum distance from members of the two classes.

Given the 2D features in the above figure, SVM will find the line H3 for you. If you get a new 2D feature vector corresponding to an image the algorithm has never seen before, you can simply test which side of the line the point lies and assign it the appropriate class label. If your feature vectors are in 3D, SVM will find the appropriate **plane** that maximally separates the two classes. As you may have guessed, if your feature vector is in a 3780-dimensional space, SVM will find the appropriate **hyperplane**.

Optimizing SVM

What if the features belonging to the two classes are not separable using a hyperplane? In such cases, SVM still finds the best hyperplane by solving an optimization problem that tries to increase the distance of the hyperplane from the two classes while trying to make sure many training examples are classified properly. This tradeoff is controlled by a parameter called **C**. When the value of **C** is small, a large margin hyperplaneis chosen at the expense of a greater number of misclassifications. Conversely, when **C** is large, a smaller margin hyperplaneis chosen that tries to classify many more examples correctly. Now you may be confused as to what value you should choose for **C**. Choose the value that performs best on a **validation set** that the algorithm was not trained on.

8. Conclusion

Now a days artificial intelligence is widely using in various fields not in our daily life but also in research work, science, engineering, robotics, speech recognition, image recognition, video gams etc. In this paper some basic definitions of artificial intelligence, Image recognition, Digital image are included. We can also see the basic process of image recognition by human eye. In human eye when light focuses onto the retina, where it is absorbed by a layer of photoreceptor cells. These cells convert light into electrochemical signals, and these are divided into two types, rods and cones. Rods are responsible for proper vision at night and cones are responsible for color. The process of Computer vision is also discussed. There are only the basic colors in computer RGB and with the help of combination of bits and sequence of bits it can make many colors on the screen. The basic unit of any picture is Pixel. At the last but not least the process of Image recognition in Artificial Intelligenceis also discussed.

References

[1] http://www.alanturing.net/turing_archive/pages/reference%20articles/what%20is%20ai.html
[2] https://www.techopedia.com/definition/190/artificial-intelligence-ai
[3] http://searchcio.techtarget.com/definition/AI
[4] http://www.webopedia.com/TERM/A/artificial_intelligence.html
[5] https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_overview.htm
[6] http://www-formal.stanford.edu/jmc/whatisai/node1.html
[7] https://www.mathworks.com/discovery/image-recognition.html
[8] http://blog.dittolabs.io/post/128793605088/what-is-image-recognition
[9] http://www.dictionary.com/browse/image-recognition
[10] http://www.gartner.com/it-glossary/image-recognition/
[11] http://www.brainhq.com/brain-resources/brain-facts-myths/how-vision-works
[12] http://news.mit.edu/2014/in-the-blink-of-an-eye-0116
[13] http://www.learnopencv.com/image-recognition-and-object-detection-part1/
[14] Image Recognition and Object Detection : Part 1 November 14, 2016 By Satya Mallick