**VEHICLE DETECTION,TRACKING AND SPEED ESTIMATION USING OPENCV AND PYTHON**

**1.INTRODUCTION**

Using four independently-controlled in-wheel motors for direct propulsion, four-wheel-independently-actuated electric vehicles (FWIA EVs) have incurred great attention due to its potential of flexible layout design and improved vehicle handling performance and safety . The over-actuation feature offers enormous opportunities for the applications of active control systems (ACS) like anti-lock brake system (ABS), acceleration slip regulation (ASR) and advanced driver assistance system (ADAS) . Efficient functionality realization of these ACSs hinges on accurate and reliable acquisition of longitudinal vehicle speed in real-time; but this presents a non-trivial task since the longitudinal vehicle speed is an immeasurable parameter using conventional low-cost sensors. Besides, there are no longer driven wheels in FWIA EVs for vehicle speed derivation by directly integrating the rotational wheel speed.These methods have no requirement of tire models and the underlying kinematic models are also straightforward and easy to understand.The proposed a longitudinal vehicle speed estimation algorithm by using rotational wheel speed and longitudinal acceleration to determine vehicle status and achieve different speed estimation modes. In contrast, dynamics-based approaches employ vehicle dynamics and tire models to deduce vehicle states via specialized state estimators. The key challenge for dynamics-based methods lies in real-time and accurate acquisitions of tire model parameters and road friction coefficient under varied operating conditions. Some researchers strived to develop advanced estimation schemes with no requirement of knowing road adhesion conditions.For the dual estimator structure, different estimation methods and tire models can be interchangeably used as indicated in other studies. To overcome the drawbacks of kinematics- and dynamics based methods, sensor fusion techniques are gaining increasing attention. They can make full use of kinematics and dynamics models, motor states information, and IMU and GPS signals to achieve accurate and reliable vehicle speed estimation .

**2.PROBLEM STATEMENT**

Vehicle classification models are proposed for roadside and aerial images. The first model utilises the proposed speed estimation method to extract the speed of the passing vehicles. Then, we used a fuzzy c-means algorithm to classify vehicles using their speeds and dimension features. The results show that vehicle speed is a useful feature for distinguishing different categories of vehicles. The second model employs deep neural networks to detect and classify heavy vehicles in aerial images.

**3.EXISTING SYSTEM**

This has been taken away by two means: 1)the comprehensive use of imagery in pack applications, joined with 2) updates in the size, speed and cost Manuscripts. The sufficiency of cutting edge PCs and related sign orchestrating headways. Picture managing has found a basic development in shrewd, current, space and government applications. Various structures nowadays can be displaced by picture overseeing trade systems that perform better than the past structures. SDCS system is among these structures that can declare the ordinary radars as invalid. This is ideal financially sharp system over current ones. SDCS structure can be joined with Automatic Number Plate Recognition (ANPR) system to shape a full scale radar structure. ANPR structure is a mass recognition methodology that uses optical character affirmation on pictures to research the imprints on vehicles. The makers present the key steps towards structure up the Speed Detection Radar. Here makers present another hypothesis in thing ID system, which is “flexible establishment subtraction” as it proofs that it is not sensitive to startling enlightening changes. Another part is appeared here concerning address following by making “object following blueprints”.

**4.PROPOSED SYSTEM**

Firstly, a video is given as input to the system. The given input video is at first pre processed according to the requirements. From the processed video sample, the vehicle is detected using the filters. This vehicle is then tracked and analyzed in order to find its speed. A Video Capture object is created for getting a live stream video. Its debate can be either the contraption report or the name of a video record. The video will be canny and in the event that it is incredibly high, video will be moderate (Well, that is the course by which you can demonstrate accounts in moderate movement). The number of subcomponents that apply various corrections or enhancement features to an input image. When one or more of the preprocessing options are enabled, the subcomponents operate the corrected image. After every pixel is gathered by this model framework, portions of the frontal area focuses are shown by DBSCAN (Density − based spatial social affair of organizations with tumult) gathering technique.

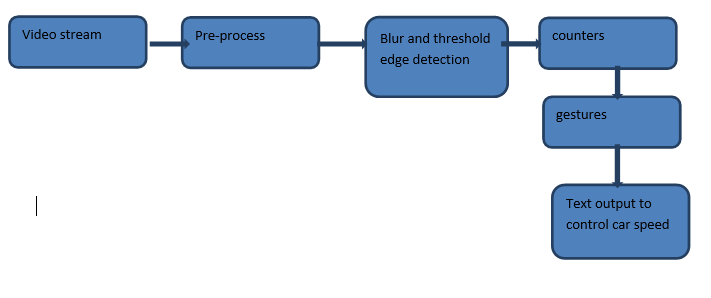
ADVANTAGES:

* Efficient functionality .
* Low accuracy.
* These methods have no requirement of tire models and the underlying kinematic models are also straight forward and easy to understand.

DISADVANTAGES:

* High accuracy.
* The variation of the wheel radius is negligible .

BLOCK DIAGRAM:



**5.SYSTEM REQUIREMENTS**

HARDWARE:

* + OS- WINDOWS 7,8,10(32 OR 64 BIT)
  + RAM-4GB–2 or 64 bit)RAM – 4GB

SOFTWARE:

* + PYTHON IDLE
  + ANACONDA
  + JUPITER NOTE BOOK

**6.DOMAIN SPECIFICATION**

DIGITAL IMAGE PROCESSING:

The identification of objects in an image and 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 areas with certain textures.

The clever bit is to interpret collections of these shapes as single objects, e.g. cars on a road, boxes on a conveyor belt or cancerous cells on a microscope slide. Another problem is deciding what features belong to what object and which are background or shadows etc. The human visual system performs these tasks mostly unconsciously but a computer requires skilful programming and lots of processing power to approach human performance. An image is usually interpreted as a two-dimensional array of brightness values, and is most familiarly represented by such patterns as those of a photographic print, slide, television screen, or movie screen. An image can be processed optically or digitally with a computer.

FUNDAMENTALS OF DIGITAL IMAGE

IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. And natural objects and phenomena, such as the human eye or water surfaces.

He word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.



Fig: Colour image to gray scale conversion process

An image is a rectangular grid of pixels. it has a definite height and a definite width counted in pixels. each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

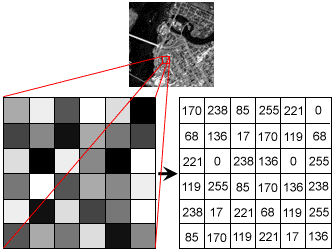


Fig: Gray scale image pixel value analysis

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

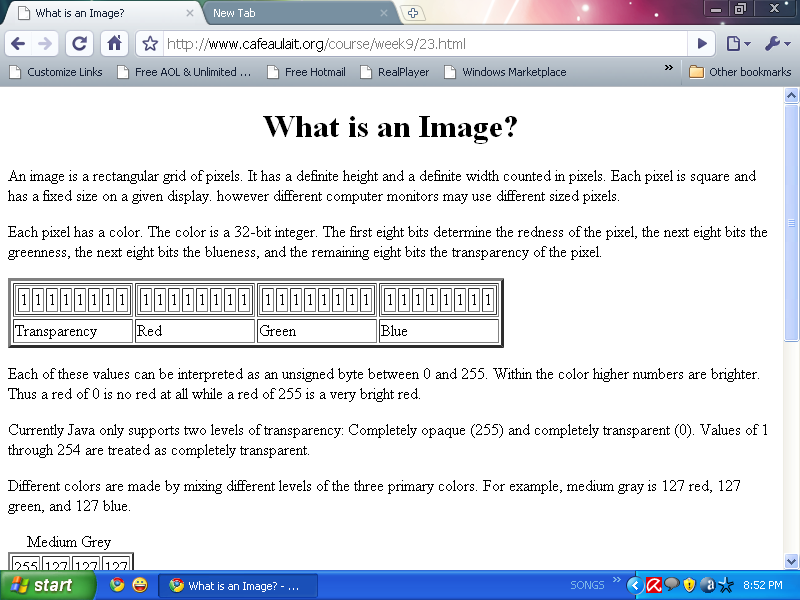


Fig: Bit transferred for red, green and blue plane (24bit=8bit red;8-bit green;8bit blue)

IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels..Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.image compression uses algorithms to decrease the size of a file. High resolution digital cameras record 12 megapixel (1mp = 1,000,000 pixels / 1 million) images, or more, in true color. for example, an image recorded by a 12 mp camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. the png, jpeg, and gif formats are most often used to display images on the internet.

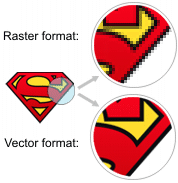


Fig: Horizontal and vertical process

in addition to straight image formats, metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most windows applications open metafiles and then save them in their own native format.

IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man’s ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (ccds) for digitizing, storage during processing and display and large low cost of image storage arrays.

FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

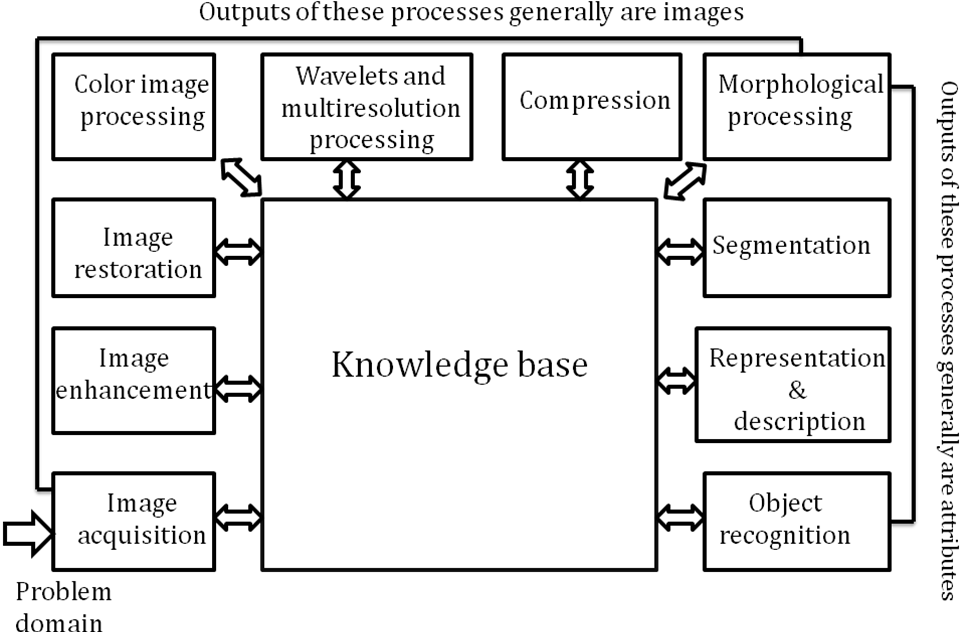
Fig: Basics steps of image processing

IMAGE ACQUISITION:

Image acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color tv camera that produces an entire image of the problem domain every 1/30 sec. The image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.



Fig: Digital camera

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.



Fig: Mobile based camera

image enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.

  
Fig: Image enhancement process for gray scale image and colour image using histogram bits

IMAGE RESTORATION:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.



Fig: Noise image🡪 image enhancement

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, where as removal of image blur by applying a deblurring function is considered a restoration technique.

COLOR IMAGE PROCESSING:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.



Fig: Gray scale image 🡪 colour image

SEGMENTATION:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

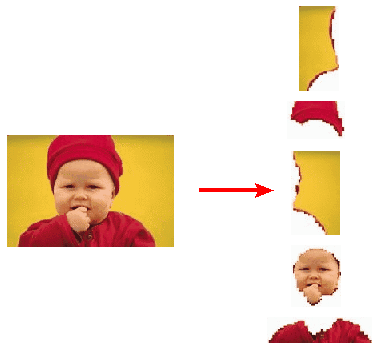


Fig: Image segment process

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

Digital image is defined as a two dimensional function f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called intensity or grey level of the image at that point. The field of digital image processing refers to processing digital images by means of a digital computer. The digital image is composed of a finite number of elements, each of which has a particular location and value. the elements are referred to as picture elements, image elements, pels, and pixels. pixel is the term most widely used.

IMAGE COMPRESSION:

Digital image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of the reduction process is removal of redundant data. From the mathematical viewpoint, this amounts to transforming a 2d pixel array into a statically uncorrelated data set. carrying units in two data sets that represent the same information, the relative data redundancy  [2] of the first data set (the one characterized by n1) can be defined as,



where  called as compression ratio [2]. it is defined as

= 

In image compression, three basic data redundancies can be identified and exploited: coding redundancy, interpixel redundancy, and phychovisal redundancy. Image compression is achieved when one or more of these redundancies are reduced or eliminated. The image compression is mainly used for image transmission and storage. Image transmission applications are in broadcast television; remote sensing via satellite, air-craft, radar, or sonar; teleconferencing; computer communications; and facsimile transmission. image storage is required most commonly for educational and business documents, medical images that arise in computer tomography (ct), magnetic resonance imaging (mri) and digital radiology, motion pictures, satellite images, weather maps, geological surveys, and so on.

IMAGE COMPRESSION MODEL:

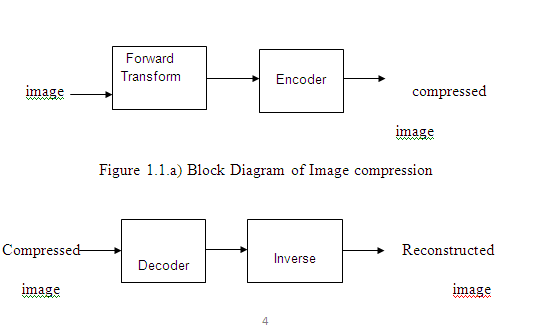


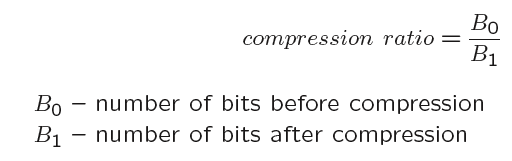
IMAGE COMPRESSION TYPES:

There are two types’ image compression techniques.

1. lossy image compression

2. lossless image compression

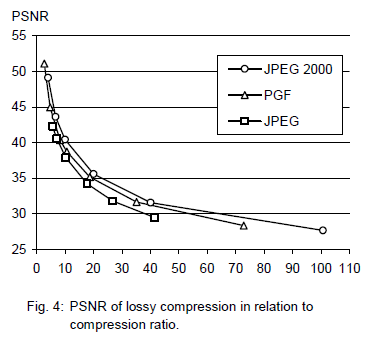
COMPRESSION RATIO:



1. Lossy image compression :

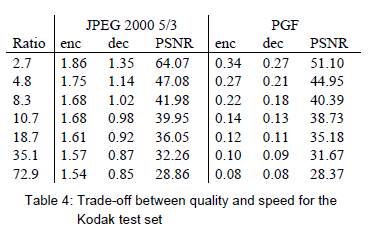
Lossy compression provides higher levels of data reduction but result in a less than perfect reproduction of the original image. It provides high compression ratio. lossy image compression is useful in applications such as broadcast television, videoconferencing, and facsimile transmission, in which a certain amount of error is an acceptable trade-off for increased compression performance. Originally, pgf has been designed to quickly and progressively decode lossy compressed aerial images..

In the next test series we evaluate the lossy compression efficiency of pgf. one of the best competitors in this area is for sure jpeg 2000. Since jpeg 2000 has two different filters, we used the one with the better trade-off between compression efficiency and runtime. On our machine the 5/3 filter set has a better trade-off than the other. however, jpeg 2000 has in both cases a remarkable good compression efficiency for very high compression ratios but also a very poor encoding and decoding speed. The other competitor is jpeg. jpeg is one of the most popular image file formats.

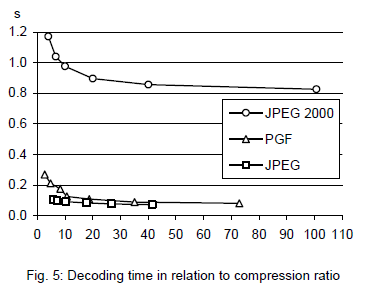


it is very fast and has a reasonably good compression efficiency for a wide range of compression ratios. the drawbacks of jpeg are the missing lossless compression and the often missing progressive decoding. fig. 4 depicts the average rate-distortion behavior for the images in the kodak test set when fixed (i.e., nonprogressive) lossy compression is used. the psnr of pgf is on average 3% smaller than the psnr of jpeg 2000, but 3% better than jpeg.

In case of pgf the encoding time is always slightly longer than the corresponding decoding time. the reason for that is that the actual encoding phase (cf. subsection 2.4.2) takes slightly longer than the corresponding decoding phase. For six of seven ratios the psnr difference between jpeg 2000 and pgf is within 3% of the psnr of jpeg 2000. only in the first row is the difference larger (21%), but because a psnr of 50 corresponds to an almost perfect image quality the large psnr difference corresponds with an almost undiscoverable visual difference. The price they pay in jpeg 2000 for the 3% more psnr is very high. the creation of a pgf is five to twenty times faster than the creation of a corresponding jpeg 2000 file, and the decoding of the created pgf is still five to ten times faster than the decoding of the jpeg 2000 file. This gain in speed is remarkable, especially in areas where time is more important than quality, maybe for instance in real-time computation.



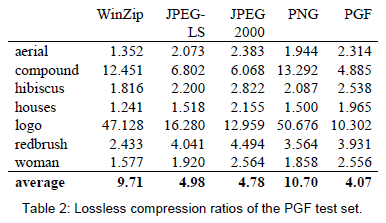
In fig. 5 we see that the price we pay in pgf for the 3% more psnr than jpeg is low: for small compression ratios (< 9) decoding in pgf takes two times longer than jpeg and for higher compression ratios (> 30) it takes only ten percent longer than jpeg. these test results are characteristic for both natural images and aerial ortho-photos. again, in the third test series we only use the ‘lena’ image. we run our lossy coder with six different quantization parameters and measure the psnr in relation to the resulting compression ratios. the results (ratio: psnr) are:



2.Lossless image compression :

Lossless image compression is the only acceptable amount of data reduction. it provides low compression ratio while compared to lossy. In lossless image compression techniques are composed of two relatively independent operations: (1) devising an alternative representation of the image in which its interpixel redundancies are reduced and (2) coding the representation to eliminate coding redundancies.

Lossless image compression is useful in applications such as medical imaginary, business documents and satellite images.table 2 summarizes the lossless compression efficiency and table 3 the coding times of the pgf test set. for winzip we only provide average runtime values, because of missing source code we have to use an interactive testing procedure with runtimes measured by hand. all other values are measured in batch mode.



In table 2 it can be seen that in almost all cases the best compression ratio is obtained by jpeg 2000, followed by pgf, jpeg-ls, and png. This result is different to the result in [sea+00], where the best performance for a similar test set has been reported for jpeg-ls. pgf performs between 0.5% (woman) and 21.3% (logo) worse than jpeg 2000. On average it is almost 15% worse. the two exceptions to the general trend are the ‘compound’ and the ‘logo’ images. both images contain for the most part black text on a white background. for this type of images, jpeg-ls and in particular winzip and png provide much larger compression ratios. however, in average png performs the best, which is also reported in [sea+00].

These results show, that as far as lossless compression is concerned, pgf performs reasonably well on natural and aerial images. In specific types of images such as ‘compound’ and ‘logo’ pgf is outperformed by far in png.

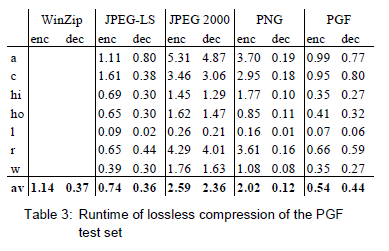


Table 3 : shows the encoding (enc) and decoding (dec) times (measured in seconds) for the same algorithms and images as in table 2. jpeg 2000 and pgf are both symmetric algorithms, while winzip, jpeg-ls and in particular png are asymmetric with a clearly shorter decoding than encoding time. jpeg 2000, the slowest in encoding and decoding, takes more than four times longer than pgf. this speed gain is due to the simpler coding phase of pgf. jpeg-ls is slightly slower than pgf during encoding, but slightly faster in decoding images.

Winzip and png decode even more faster than jpeg-ls, but their encoding times are also worse. Pgf seems to be the best compromise between encoding and decoding times.

Our pgf test set clearly shows that pgf in lossless mode is best suited for natural images and aerial ortho photos. pgf is the only algorithm that encodes the three mega byte large aerial ortho photo in less than second without a real loss of compression efficiency. For this particular image the efficiency loss is less than three percent compared to the best. These results should be underlined with our second test set, the kodak test set.

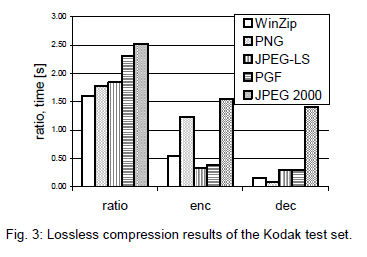


Fig. 3 shows the averages of the compression ratios (ratio), encoding (enc), and decoding (dec) times over all eight images. jpeg 2000 shows in this test set the best compression efficiency followed by pgf, jpeg-ls, png, and winzip. In average pgf is eight percent worse than jpeg 2000. The fact that jpeg 2000 has a better lossless compression ratio than pgf does not surprise, because jpeg 2000 is more quality driven than pgf.

However, it is remarkable that pgf is clearly better than jpeg-ls (+21%) and png (+23%) for natural images. jpeg-ls shows in the kodak test set also a symmetric encoding and decoding time behaviour. It is encoding and decoding times are almost equal to pgf. Only png and winzip can faster decode than pgf, but they also take longer than pgf to encode.

If both compression efficiency and runtime is important, then pgf is clearly the best of the tested algorithms for lossless compression of natural images and aerial ortho photos. In the third test we perform our lossless coder on the ‘lena’ image.

for a point operation, a pixel value in the output image depends on a single pixel value in the input image. for local operations, several neighbouring pixels in the input image determine the value of an output image pixel. in a global operation, all of the input image pixels contribute to an output image pixel value.

CLASSIFICATION OF IMAGES:

There are 3 types of images used in digital image processing. They are

1. binary image
2. gray scale image
3. colour image

BINARY IMAGE:

A binary image is a [digital image](http://en.wikipedia.org/wiki/Digital_image) that has only two possible values for each [pixel](http://en.wikipedia.org/wiki/Pixel).  typically the two colors used for a binary image are black and white though any two colors can be used.  The color used for the object(s) in the image is the foreground color while the rest of the image is the background color.

Binary images are also called bi-level or two-level. This means that each pixel is stored as a single bit (0 or 1).this name black and white, monochrome or monochromatic are often used for this concept, but may also designate any images that have only one sample per pixel, such as [grayscale images](http://en.wikipedia.org/wiki/Grayscale)

Binary images often arise in [digital image processing](http://en.wikipedia.org/wiki/Digital_image_processing) as [masks](http://en.wikipedia.org/w/index.php?title=Mask_(image_processing)&action=edit&redlink=1) or as the result of certain operations such as [segmentation](http://en.wikipedia.org/wiki/Segmentation_(image_processing)), [thresholding](http://en.wikipedia.org/wiki/Thresholding_(image_processing)), and [dithering](http://en.wikipedia.org/wiki/Dither).

GRAY SCALE IMAGE:

A grayscale image is [digital image](http://en.wikipedia.org/wiki/Digital_image) is an image in which the value of each [pixel](http://en.wikipedia.org/wiki/Pixel) is a single [sample](http://en.wikipedia.org/wiki/Sample_(signal)), that is, it carries only [intensity](http://en.wikipedia.org/wiki/Luminous_intensity) information. images of this sort, also known as [black-and-white](http://en.wikipedia.org/wiki/Black-and-white), are composed exclusively of shades of [gray](http://en.wikipedia.org/wiki/Gray)(0-255), varying from black(0) at the weakest intensity to white(255) at the strongest.

Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the [electromagnetic spectrum](http://en.wikipedia.org/wiki/Electromagnetic_spectrum) (e.g. [infrared](http://en.wikipedia.org/wiki/Infrared), [visible light](http://en.wikipedia.org/wiki/Visible_spectrum), [ultraviolet](http://en.wikipedia.org/wiki/Ultraviolet), etc.), and in such cases they are monochromatic proper when only a given [frequency](http://en.wikipedia.org/wiki/Frequency) is captured.

COLOR IMAGE:

A (digital) color image is a [digital image](http://en.wikipedia.org/wiki/Digital_image) that includes [color](http://en.wikipedia.org/wiki/Color) information for each [pixel](http://en.wikipedia.org/wiki/Pixel). each pixel has a particular value which determines its appearing color. The decomposition of a color in the three primary colors is quantified by a number between 0 and 255. For example, white will be coded as r = 255, g = 255, b = 255; black will be known as (r,g,b) = (0,0,0); and say, bright pink will be : (255,0,255).

In other words, an image is an enormous two-dimensional array of color values, pixels, each of them coded on 3 bytes, representing the three primary colors. This allows the image to contain a total of 256x256x256 = 16.8 million different colors. This technique is also known as rgb encoding, and is specifically adapted to human vision

Colors are coded on three bytes representing their decomposition on the three primary colors. it sounds obvious to a mathematician to immediately interpret colors as vectors in a three dimension space where each axis stands for one of the primary colors. therefore we will benefit of most of the geometric mathematical concepts to deal with our colors, such as norms, scalar product, projection, rotation or distance.

**7.TAKE VIDEO CAPTURING USING WEBCAMERAS**

VIDEO STREAMING:

Video streaming technology is one way to deliver video over the internet.  using streaming technologies, the delivery of audio and video over the internet can reach many millions of customer using their personal computers, pdas, mobile smartphones or other streaming devices. the reasons for video streaming technology growth are:

* Broadband networks are being deployed
* Video and audio compression techniques are more efficient
* Quality and variety of audio and video services over internet are increasing

There are two major ways for the transmission of video/audio information over the internet:

DOWNLOAD MODE:  The content file is completely downloaded and then played. this mode requires long downloading time for the whole content file and requires hard disk space.

STREAMING MODE:

The content file is not required to be downloaded completely and it is playing while parts of the content are being received and decoded.

PRE-PROCESSING:

Pre-processing is a common name for operations with images at the lowest level of abstraction -- both input and output are intensity images.ο the aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing.

RESIZING:

[Computer graphics](https://en.wikipedia.org/wiki/Computer_graphics) and [digital imaging](https://en.wikipedia.org/wiki/Digital_imaging), image scaling refers to the resizing of a digital image. in video technology, the magnification of digital material is known as upscaling or [resolution enhancement](https://en.wikipedia.org/wiki/Resolution_enhancement_technology).

When scaling a [vector graphic](https://en.wikipedia.org/wiki/Vector_graphic) image, the graphic primitives that make up the image can be scaled using geometric transformations, with no loss of [image quality](https://en.wikipedia.org/wiki/Image_quality). when scaling a [raster graphics](https://en.wikipedia.org/wiki/Raster_graphics) image, a new image with a higher or lower number of pixels must be generated. in the case of decreasing the pixel number (scaling down) this usually results in a visible quality loss. from the standpoint of [digital signal processing](https://en.wikipedia.org/wiki/Digital_signal_processing), the scaling of raster graphics is a two-dimensional example of [sample-rate conversion](https://en.wikipedia.org/wiki/Sample-rate_conversion), the conversion of a discrete signal from a sampling rate (in this case the local sampling rate) to another.

**8.CONVERSION**

RGB TO GRAY:

How do you convert a color image to grayscale? if each color pixel is described by a triple (r, g, b) of intensities for red, green, and blue, how do you map that to a single number giving a grayscale value? the [gimp](http://www.gimp.org/) image software has three algorithms.

The lightness method averages the most prominent and least prominent colors: (max(r, g, b) + min(r, g, b)) / 2.

The average method simply averages the values: (r + g + b) / 3.

The luminosity method is a more sophisticated version of the average method. It also averages the values, but it forms a weighted average to account for human perception. We’re more sensitive to green than other colors, so green is weighted most heavily

THRESHOLDING:

A binary image is one that consists of pixels that can have one of exactly two colors, usually black and white. binary images are also called bi-level or two-level. this means that each pixel is stored as a single bit—i.e., a 0 or 1. the names black-and-white, b&w, [monochrome](https://en.wikipedia.org/wiki/Monochrome) or [monochromatic](https://en.wikipedia.org/wiki/Monochrome) are often used for this concept, but may also designate any images that have only one sample per pixel, such as [grayscale images](https://en.wikipedia.org/wiki/Grayscale). in [photoshop](https://en.wikipedia.org/wiki/Adobe_Photoshop) parlance, a binary image is the same as an image in "bitmap" modebinary images often arise in [digital image processing](https://en.wikipedia.org/wiki/Digital_image_processing) as [masks](https://en.wikipedia.org/wiki/Mask_(computing)#Image_masks) or)|thresholding]], and [dithering](https://en.wikipedia.org/wiki/Dither). some input/output devices, such as [laser printers](https://en.wikipedia.org/wiki/Laser_printer), [fax machines](https://en.wikipedia.org/wiki/Fax), and bilevel [computer displays](https://en.wikipedia.org/wiki/Visual_display_unit), can only handle bilevel images.

A binary image can be stored in memory as a [bitmap](https://en.wikipedia.org/wiki/Bitmap), a packed array of bits. A 640×480 image requires 37.5 [kib](https://en.wikipedia.org/wiki/Kibibyte) of storage. Because of the small size of the image files, fax machine and document management solutions usually use this format. Most binary images also compress well with simple [run-length compression](https://en.wikipedia.org/wiki/Run-length_encoding) schemes

**9.LITERATURE SUEVEY**

Literature [survey](http://www.blurtit.com/q876299.html) is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next step is to determine which operating system and language can be used for developing the tool. Once the [programmers](http://www.blurtit.com/q876299.html) start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from [book](http://www.blurtit.com/q876299.html) or from websites. Before building the system the above considerations are taken into account for developing the proposed system. A literature review is a body of text that aims to review the critical points of current knowledge including substantive findings as well as theoretical and methodological contributions to a particular topic. Literature reviews are [secondary sources](http://en.wikipedia.org/wiki/Secondary_sources), and as such, do not report any new or original experimental work. Also, a literature review can be interpreted as a review of an abstract accomplishment. Most often associated with academic-oriented literature, such as a [thesis](http://en.wikipedia.org/wiki/Thesis), a literature review usually precedes a research proposal and results section. Its main goal is to situate the current study within the body of literature and to provide context for the particular reader.

**10.UML DIAGRAMS:**

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

* Actors
* Business processes
* (logical) Components
* Activities
* Programming language statements
* Database schemas, and
* Reusable software components.

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

SEQUENCE DIAGRAM:

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency.In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

USE CASE DIAGRAM:

* UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.
* UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.
* OMG is continuously putting effort to make a truly industry standard.
* UML stands for Unified Modeling Language.
* UML is a pictorial language used to make software blue prints

COLLABORATION:

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing

CLASS DIAGRAM:

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

COMPONENT DIAGRAM

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

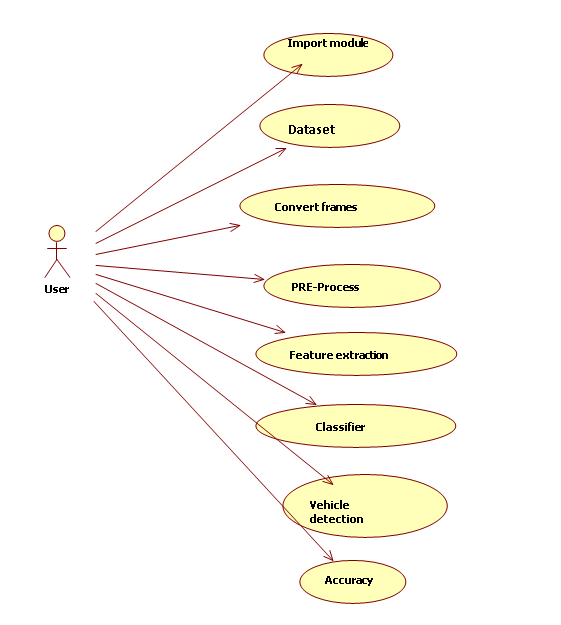
Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

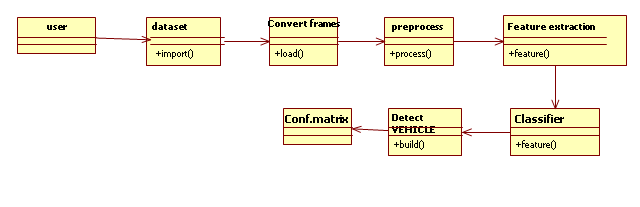
The purpose of the component diagram can be summarized as

* Visualize the components of a system.
* Construct executables by using forward and reverse engineering.
* Describe the organization and relationships of the components.

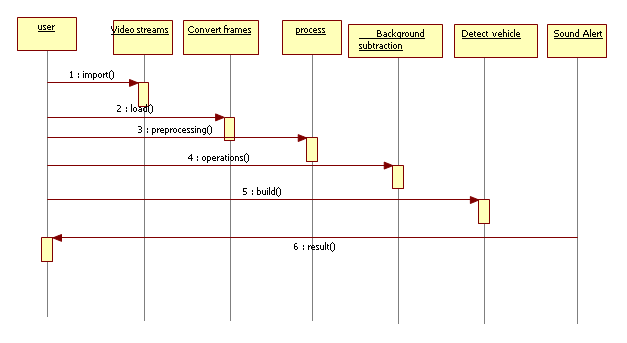
USE CASE DIAGRAM:



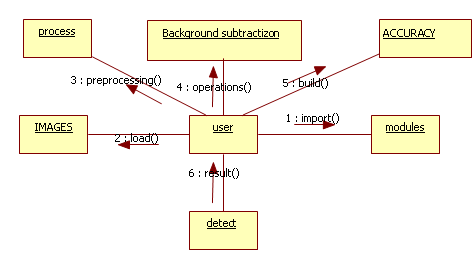
CLASS DIAGRAM:



SEQUENCE DIAGRAM:



COLLOBARATION:



**11.SOFTWARE USED**

PYTHON:

Python is an object-oriented, high level language, interpreted, dynamic and multipurpose programming language.

Python is easy to learn yet powerful and versatile scripting language which makes it attractive for application development.

Python's syntax and dynamic typing with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas.

Python supports multiple programming pattern, including object oriented programming, imperative and functional programming or procedural styles.

Python is not intended to work on special area such as web programming. that is why it is known as multipurpose because it can be used with web, enterprise, 3d cad etc.

We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to declare an integer value in a variable.

Python makes the development and debugging fast because there is no compilation step included in python development and edit-test-debug cycle is very fast.

PYTHON FEATURES:

1) Easy to use:

Python is easy to very easy to use and high level language. Thus it is programmer-friendly language.

2) Expressive language:

Python language is more expressive. The sense of expressive is the code is easily understandable.

3) Interpreted language:

Python is an interpreted language i.e. interpreter executes the code line by line at a time. this makes debugging easy and thus suitable for beginners

4) Cross-platform language:

Python can run equally on different platforms such as windows, linux, unix , macintosh etc. thus, python is a portable language.

5) Free and open source:

Python language is freely available(www.python.org).the source-code is also available. therefore it is open source.

6) Object-oriented language:

Python supports object oriented language. concept of classes and objects comes into existence.

7) Extensible:

It implies that other languages such as c/c++ can be used to compile the code and thus it can be used further in your python code.\_

PYTHON APPLICATIONS:

Python as a whole can be used in any sphere of development.

Let us see what are the major regions where python proves to be handy.

1) Console based application

Python can be used to develop console based applications. for example: ipython.

2) Audio or video based applications

python proves handy in multimedia section. some of real applications are: timplayer, cplay etc.

3) 3d cad applications

fandango is a real application which provides full features of cad.

4) Web applications

python can also be used to develop web based application. some important developments are: pythonwikiengines, pocoo, pythonblogsoftware etc.

PYTHON EXAMPLE:

Python code is simple and easy to run. Here is a simple python code that will print "welcome to python".

A simple python example is given below.

1. >>> a="welcome to python"

2. >>> print a

3. welcome to python

4. >>>

EXPLANATION:

1.Here we are using idle to write the python code. detail explanation to run code is given in execute python section.

2.A variable is defined named "a" which holds "welcome to python".

3."print" statement is used to print the content. therefore "print a" statement will print the content of the variable. therefore, the output "welcome to python" is produced.

HOW TO EXECUTE PYTHON:

There are three different ways of working in python:

1) Interactive mode:

You can enter python in the command prompt and start working with python.Press enter key and the command prompt will appear like:now you can execute your python commands.

2) Script mode:

Using script mode , you can write your python code in a separate file using any editor of your operating system.

save it by .py extension.

now open command prompt and execute it by :

3) Using IDE: (Integrated Development Environment)

You can execute your python code using a graphical user interface (gui).All you need to do is:

click on start button -> all programs -> python -> idle(python gui)

You can use both interactive as well as script mode in ide.

1) Using interactive mode:

Execute your python code on the python prompt and it will display result simultaneously.

2) Using script mode:

i) click on start button -> all programs -> python -> idle(python gui)

ii) python shell will be opened. now click on file -> new window.

a new editor will be opened . write your python code here.

run then code by clicking on run in the menu bar.

run -> run module

result will be displayed on a new python shell as:

**12.OPENCV**

INTRODUCTION TO COMPUTER VISION:

Using software to parse the world’s visual content is as big of a revolution in computing as mobile was 10 years ago, and will provide a major edge for developers and businesses to build amazing products.

Computer vision is the process of using machines to understand and analyze imagery (both photos and videos). While these types of algorithms have been around in various forms since the 1960’s, recent advances in machine learning, as well as leaps forward in data storage, computing capabilities, and cheap high-quality input devices, have driven major improvements in how well our software can explore this kind of content.

WHAT IS COMPUTER VISION ?

Computer vision is the broad parent name for any computations involving visual content – that means images, videos, icons, and anything else with pixels involved. But within this parent idea, there are a few specific tasks that are core building blocks:

• In object classification, you train a model on a dataset of specific objects, and the model classifies new objects as belonging to one or more of your training categories.

A classical application of computer vision is handwriting recognition for digitizing handwritten content (we’ll explore more use cases below). Outside of just recognition, other methods of analysis include:

• Video motion analysis uses computer vision to estimate the velocity of objects in a video, or the camera itself.

• In image segmentation, algorithms partition images into multiple sets of views.

• Scene reconstruction creates a 3d model of a scene inputted through images or video (check out selva).

Any other application that involves understanding pixels through software can safely be labeled as computer vision.

HOW COMPUTER VISION WORKS

One of the major open questions in both neuroscience and machine learning is: how exactly do our brains work, and how can we approximate that with our own algorithms? the reality is that there are very few working and comprehensive theories of brain computation; so despite the fact that neural nets are supposed to “mimic the way the brain works,” nobody is quite sure if that’s actually true. jeff hawkins has an entire book on this topic called on intelligence.

The same paradox holds true for computer vision – since we’re not decided on how the brain and eyes process images, it’s difficult to say how well the algorithms used in production approximate our own internal mental processes. for example, studies have shown that some functions that we thought happen in the brain of frogs actually take place in the eyes. we’re a far cry from amphibians, but similar uncertainty exists in human cognition.

COMPUTER VISION ON ALGORITHMIA:

Algorithmia makes it easy to deploy computer vision applications as scalable microservices. Our marketplace has a few algorithms to help get the job done:

• Salnet automatically identifies the most important parts of an image

• Emotion recognition parses emotions exhibited in images

• Deepstyle transfers next-level filters onto your image

• Face recognition…recognizes faces.

• Image memorability judges how memorable an image is.

A typical workflow for your product might involve passing images from a security camera into emotion recognition and raising a flag if any aggressive emotions are exhibited, or using nudity detection to block inappropriate profile pictures on your web application.

For a more detailed exploration of how you can use the algorithmia platform to implement complex and useful computer vision tasks.

**13.PACKAGES AND FRAMEWORKS**

Opencv – “Opencv was designed for computational efficiency and with a strong focus on real-time applications. usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.”

Simplecv – “Simplecv is an open source framework for building computer vision applications. computer vision libraries such as opencv – without having to first learn about bit depths, file formats, color spaces, buffer management, eigenvalues, or matrix versus bitmap storage.”

Mahotas – “Mahotas is a computer vision and image processing library for python. mahotas currently has over 100 functions for image processing and computer vision and it keeps growing.

NUM PY:

• Numpy, which stands for numerical python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. It also discusses the various array functions, types of indexing, etc. An introduction to matplotlib is also provided. all this is explained with the help of examples for better understanding.

In 2005, travis oliphant created numpy package by incorporating the features of numarray into numeric package. there are many contributors to this open source project.

Operations using numpy

Using numpy, a developer can perform the following operations −

• Mathematical and logical operations on arrays.

• Fourier transforms and routines for shape manipulation.

•Operations related to linear algebra. numpy has in-built functions for linear algebra and random number generation.

numpy – a replacement for matlab

Numpy is often used along with packages like scipy (scientific python) and mat−plotlib (plotting library). This combination is widely used as a replacement for matlab, a popular platform for technical computing. however, python alternative to matlab is now seen as a more modern and complete programming language.

The most important object defined in numpy is an n-dimensional array type called ndarray. It describes the collection of items of the same type. Items in the collection can be accessed using a zero-based index.

Every item in an ndarray takes the same size of block in the memory. Each element in ndarray is an object of data-type object (called dtype).

Any item extracted from ndarray object (by slicing) is represented by a python object of one of array scalar types. The following diagram shows a relationship between ndarray, data type object (dtype) and array scalar type −

An instance of ndarray class can be constructed by different array creation routines described later in the tutorial. the basic ndarray is created using an array function in numpy as follows −

IMUTILS:

A series of convenience functions to make basic image processing operations such as translation, rotation, resizing, skeletonization, and displaying matplotlib images easier with opencv and python.

TRANSLATION:

Translation is the shifting of an image in either the x or y direction. to translate an image in opencv you need to supply the (x, y)-shift, denoted as (tx, ty) to construct the translation matrix m:and from there, you would need to apply the cv2.warpaffine function.

ROTATION:

Image is to be rotated about. these calculation calls can quickly add up and make your code bulky and less readable. The rotate function inimutils helps resolve this problem.

RESIZING:

Resizing an image in opencv is accomplished by calling the cv2.resize function. however, special care needs to be taken to ensure that the aspect ratio is maintained. This resize function of imutils maintains the aspect ratio and provides the keyword arguments width and height so the image can be resized to the intended width/height while (1) maintaining aspect ratio and (2) ensuring the dimensions of the image do not have to be explicitly computed by the developer.

DISPLAYING THE MATPLOTLIB:

In the python bindings of opencv, images are represented as numpy arrays in bgr order. This works fine when using the cv2.imshow function. However, if you intend on using matplotlib, the plt.imshow function assumes the image is in rgb order. A simple call tocv2.cvtcolor will resolve this problem, or you can use the opencv2matplotlib convenience function.

**14.CONCLUSION**

The principal motivation behind this paper to tally the vehicle and to assess the vehicle speed. By applying open-CV and by applying some pre-handling strategies presently it's feasible to ascertain the speed and check the vehicle.

**15.REFERENCES**

[i] v. pavlovic, r. sharma, and t. huang, “visual interpretation of hand gestures for human-computer interaction:’ ieee trnnsnctiom 011 pntreru aiinlvsis mid mnchine iritelligerice, vol. 19.7, pp. 677495, 1997.

[2] h. hienz, k. kraiss, and b. bauer, “continuous sign language recognition using hidden markov models.” in proceedings, 21id lnt. coizjeremx on mirltin~odnl interfnces, hoizg koilg, cliinn, 1999, 1999, pp. ivioiv15.

[3] e althoff, g. mcglaun, b. schuller, p. morguet. and m. lang, “using multimodal interaction to navigate in arbitrary virtual vrml worlds,” in proceedings. pui 2001 worksliop 011 perceptive user iirterfnces. orinndo. floridn, usa, november 15-16, 2001. association for computing machinery, 2001, acm digital library: www.acm.org/uist/uist2ooi. cd-rom.

[4] y. sat0 and y. kobayashi, “fast tracking of hands and fingertips in infrared images for augmented desk interface,” in proceedings, 4th in/. conference 011 airtonmtic fnce and gesrirre recognition. greiroble, frnnce, 2000, 2000, pp. 462467.

[5] p. morguet and m. lang, “comparison of approaches to continuous hand gesture recognition for a visual dialog system,” in proceedings, icasp 1999 itit. coriceference 011 acoustics arid sigiinl processing. phoenix, arizonn, usa, marcli 15-19, 1999. ieee, 1999, pp. 3549-3552.

[6]1 c. hardenberg and e berard. “bare-hand humancomputer interaction,” in proceedings, plj1 2001 worksliop on perceptive user interfaces, orlando, florida, usa, november 15-16, 2001. association for computing machinery, 2001, acm digital library: www.acm.org/uist/uist2001. cd-rom.

[7] “jestertek inc. homepage,” [www.jestertek.com](http://www.jestertek.com).

[8] e. klarreich, “no more fumbling in the vehicle,” in iioture, glasgo~v, gmnr britniii, noveiizbei: ;!001. british association tor the advancement of science, 2001, nature news service.

[9] s. akyol, u. canzler, k. bengler, and w. hahn, “gesture control for use in automobiles,” in proceedings, mva 2000 worksliop 011 mncliine visioii applicnrions, tokjlo, japmr. no~wiiber 28-30, 2000. iapr, 2000, pp. 28-30, isbn 4-901 122-00-2.

[10] m. zobl. m. geiger, p. morguet, r. nieschulz, and m. lang, “gesture-based control of in-vehicle devices,” in vdi-berichte 1678: useware 2002 moisrlzmnsclfirre-ko~~i~firo,ikntiolrldesigi~, gma fnclltogwig useware 2002. dnrnistndt. germmy, jrrrie 11-12, 2002. dusseldorf, 2002, vdi, pp. 305-309, vdiverlag.

[i i] m. zobl, m. geiger, k. bengler, and m. lang, “a usability study on hand gesture controlled operation of in-vehicle devices,” in abri&efl proceedirigs, hci 2001 9th in/. coujei-e!ice 011 hiriiznlr mnclriur iirrernctioii, new orlen!zs. lorrisintm, usa, airgirsr 5-10, 2001. new jersey, 2001, pp. 166-168, lawrence erlbaum ass.

[12] m. geiger. m. zobl. k. bengler, and m. lang, “lntermodal differences in distraction effects while controlling automotive user interfaces,” in proceer1iiig.s vol. i: usnbi1it.v ei,nlrmrioii mid interfnce desigir , hci 2001 9th lm. corzjereizce oii hfrii~o~i mncliiiie iriterflctio!i. new orlemis, loirisinjin, usa, airgrist 5-10. 2001, new jersey, 2001, pp. 263-267, lawrence erlbaum ass.

[i3]1 m. geiger. r. nieschulz, m. zobl, and m. lang. “gesture-based control concept for in-vehicle devicess,” in vdi-berichte 1678: useware 2002 menschmnsclii~ie-koriririiriiikntioi~desigf~, gma fochtngirrig useware 2002, dnrilistndt, gen?inuy, jim 11-12, 2002, diisseldorf, 2002. vdi, pp. 299-303. vdiverlag.

[ 14]1 u. broekl-fox, uiitersuclwig rieirer, gesteiibnsierrer verfnhreri fiir die 3d-lnternktiori. phl) thesis. shaker publishing, 1995.

[ 15]1 m. hu, ‘‘visual pattern recognition by moment invariants,” ire trnrisnctioiis oii biforriiatioii tlieory, vol. it& pp. 179-187, 1962.

[ 16]1 l. r. rabiner, “a tutorial on hidden markov models and selected applications in speech recognition,” proceedings of die ieee, vol. 77, pp. 257-286, 1989.