**NUMBER PLATE DETECTION USING IMAGE PROCESSING**

**ABSTRACT**

With the increasing of the vehicles around the world, the possibility of being an accident in traffic increases dramatically. The countries all over the world take precautions to prevent the accidents. Many automatic auditing systems are developed for precaution. The faulty drivers that endangers the traffic safety are identified and punished. To identify the faulty drivers, the images taken from the traffic surveillance cameras are used. This paper presents algorithms for vision-based detection and classification of vehicles in video sequences of traffic scenes recorded by a stationary camera. Vehicles are modeled as rectangular patches with certain dynamic behavior. The proposed method is based on the establishment of correspondences between regions and vehicles, as the vehicles move through the image sequence. Experimental results from highway scenes are provided which demonstrate the effectiveness of the method. We also briefly describe an interactive camera calibration tool that we have developed for recovering the camera parameters using features in the image selected by the user. The proposed classification method is based on the tensor flow of the detected moving object obtained by background subtraction. Different from other shape based classification techniques, we exploit the information available in multiple frames of the video. This approach eliminates most of the wrong decisions which are caused by a poorly extracted silhouette from a single video frame. The decision boundaries in the feature space are determined using a training set, whereas the performance of the proposed classification is measured with a test set. To ensure randomization, the procedure is repeated with the whole dataset split differently into training and testing samples.

**INTRODUCTION**

Application of computer vision paves the way for vehicle detection and identification from a digital forensic video since it allows the identification of individual vehicles and extracts the parameters related to each vehicle such as location, registration number, colour, type simultaneously. In general, the information that can be extracted through this system can be used effectively to find the suspect vehicle in a high traffic flow and also recognize the number plate of the vehicle. Due to recent progress in object detection, tracking and detection has become the leading paradigm in multiple object tracking. Within this paradigm, object trajectories are usually found in a global optimization problem that processes entire video batches at once. There is interest in vehicle tracking and identification due to its applications for identifying the suspect vehicle and for public safety. There is interest in vehicle tracking due to its applications for public safety and traffic monitoring. Vehicle tracking can be used by law enforcement to track criminals in cases of Amber Alerts and also suspect involved in other criminal activity.

The number of on-road motor vehicles has increased with the rapid growth of world’s economy and with this augmentation the need for security and monitoring of vehicles has also increased. Many successful commercial systems that employ dedicated camera systems, providing video input captured under control environments to ANPR algorithms. However, application scenarios in video surveillance and forensics such as tracking down a stolen vehicle or searching for a vehicle involved in a crime, as identified by a bystander to be of a particular registration number, requires the painstaking task of manual search, because the existing ANPR systems are not capable of efficiently working on video footage obtained via non-dedicated (for ANPR) CCTV systems.

This main objective of this project is to provide efficient and robust framework that can perform tracking, identification and recognition of multiple vehicle and their license plates in a real-time scenario (i.e., incoming video stream from cctv surveillance cameras). Also, this technology is used in suspecting criminals, obtaining information about the criminal vehicles from a captured forensic video.

**LITERATURE SURVEY**

* **“Performance assessment of model based tracking,” in Proc. IEEE Workshop Applications of Computer Vision**

Non-intrusive video vehicle detection and tracking for traffic flow surveillance and statistics is the primary alternative to conventional inductive loop detectors. Vision-based systems for traffic have an impressive spread both for their practical application and interest as research issue. This paper presents vision-based vehicle detection and tracking system which consists of environment background segmentation and subtraction, foreground moving object extraction, moving vehicles detection algorithms, object tracking algorithms, and vehicle classification. The proposed system can perform well for the video sequences acquired under different weather, illumination, and traffic conditions through the use of these technologies.

* **“A real-time computer vision system for measuring traffic parameters,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition**

This paper presents algorithms for vision-based detection and classification of vehicles in monocular image sequences of traffic scenes recorded by a stationary camera. Processing is done at three levels: raw images, region level and vehicle level. Vehicles are modeled as rectangular patches with certain dynamic behavior. The proposed method is based on the establishment of correspondences between regions and vehicles, as the vehicles move through the image sequence. Experimental results from highway scenes are provided which demonstrate the effectiveness of the method. We also briefly describe an interactive camera calibration tool that we have developed for recovering the camera parameters using features in the image selected by the user.

* **Road traffic sign detection and classification,” IEEE Trans. Ind. Electron**

A vision-based vehicle guidance system for road vehicles can have three main roles: (1) road detection; (2) obstacle detection; and (3) sign recognition. The first two have been studied for many years and with many good results, but traffic sign recognition is a less-studied field. Traffic signs provide drivers with very valuable information about the road, in order to make driving safer and easier. The authors think that traffic signs most play the same role for autonomous vehicles. They are designed to be easily recognized by human drivers mainly because their color and shapes are very different from natural environments. The algorithm described in this paper takes advantage of these features. It has two main parts. The first one, for the detection, uses color thresholding to segment the image and shape analysis to detect the signs. The second one, for the classification, uses a neural network. Some results from natural scenes are shown.

* **Image segmentation in video sequences,” in Proc. 13th Conf. Uncertainty in Artificial Intelligence**

“Background subtraction” is an old technique for finding moving objects in a video sequence—for example, cars driving on a freeway. The idea is that subtracting the current image from a time averaged background image will leave only no stationary objects. It is, however, a crude approximation to the task of classifying each pixel of the current image; it fails with slow-moving objects and does not distinguish shadows from moving objects. The basic idea of this paper is that we can classify each pixel using a model of how that pixel looks when it is part of different classes. We learn a mixture-of-Gaussians classification model for each pixel using an unsupervised technique—an efficient, incremental version of EM. Unlike the standard image-averaging approach, this automatically updates the mixture component for each class according to likelihood of membership; hence slow-moving objects are handled perfectly. Our approach also identifies and eliminates shadows much more effectively than other techniques such as thresholding. Application of this method as part of the Road watch traffic surveillance project is expected to result in significant improvements in vehicle identification and tracking.

* **Moving object recognition using an adaptive background memory, in Proc. Time-Varying Image Processing and Moving Object Recognition**

In recent years, video monitoring and surveillance systems have been widely used in traffic management. The image sequences for traffic scenes are recorded by a stationary camera. The video clip is sent to LabVIEW program to convert into image frames. NI LabVIEW vision assistant module is used to detect the moving vehicle. The method is based on the establishment of correspondences between regions and vehicles, as the vehicles move through the image sequence. Background subtraction is used which improves the adaptive background mixture model and makes the system learn faster and more accurately, as well as adapt effectively to changing environments. The resulting system robustly identifies vehicles, rejecting background and tracks vehicles over a specific period of time. Once the (object) vehicle is tracked, the attributes of the vehicle like width, length, perimeter, area etc are extracted by image process feature extraction techniques. In proposed system we use LabVIEW and Vision assistant module for image processing and feature extraction. The project will benefit to reduce cost of traffic monitoring system and complete automation of traffic monitoring system.

**EXISTING SYSTEM**

Tracking moving vehicles in video streams has been an active area of research in computer vision. In, a real time system for measuring traffic parameters is described. It uses a feature-based method along with occlusion reasoning for tracking vehicles in congested traffic scenes. In order to handle occlusions, instead of tracking entire vehicles, vehicle subfeatures are tracked. This approach however is very computationally expensive. In a moving object recognition method is described that uses an adaptive background subtraction technique to separate vehicles from the background. The background is modeled as a slow time-varying image sequence, which allows it to adapt to changes in lighting and weather conditions. In a related work described in, pedestrians are tracked and counted using a single camera. The images from the input image sequence are segmented using background subtraction. The resulting connected regions are then grouped together into pedestrians and tracked. Merging and splitting of regions is treated as a graph optimization problem. In a system for detecting lane changes of vehicles in a traffic scene is introduced. The approach is similar to the one described in [10] with the addition that trajectories of the vehicles are determined to detect lane changes. Despite the large amount of literature on vehicle detection and tracking, there has been relatively little work done in the field of vehicle classification. This is because vehicle classification is an inherently hard problem. Moreover, detection and tracking are simply preliminary steps in the task of vehicle classification. Given the wide variety of shapes and sizes of vehicles within a single category alone, it is difficult to categorize vehicles using simple parameters. This task is made even more difficult when multiple categories are desired. In real-world traffic scenes, occlusions, shadows, camera noise, changes in lighting and weather conditions, etc. are a fact of life. In addition, stereo cameras are rarely used for traffic monitoring. This makes the recovery of vehicle parameters—such as length, width, height etc., even more difficult given a single camera view. The inherent complexity of stereo algorithms and the need to solve the correspondence problem makes them unfeasible for real-time applications. a vehicle tracking and classification system is described that can categorize moving objects as vehicles or humans. However, it does not further classify the vehicles into various classes. an object classification approach that uses parameterized 3-D models is described. The system uses 3-D polyhedral model to classify vehicles in a traffic sequence. The system uses a generic vehicle model based on the shape of a typical sedan. The underlying assumption being that in typical traffic scenes, cars are more common than trucks or other types of vehicles. The University of Reading has done extensive work in three-dimensional tracking of vehicles and classification of the tracked vehicles using 3-D model matching methods. Baker and Sullivan and Sullivan] utilized knowledge of the camera calibration and that vehicles move on a plane in their 3-D model-based tracking. Three-dimensional wireframe models of various types of vehicles (e.g., sedans, hatchbacks, wagons, etc.) were developed. Projections of these models were then compared to features in the image.this approach was extended so that the image features act as forces on the model. This reduced the number of iterations and improved performance. They also parameterized models as deformable templates and used principal component analysis to reduce the number of parameters. Sullivan et al. developed a simplified version of the model-based tracking approach using orthographic approximations to attain real-time performance

**PROPOSE SYSTEM**

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the Vehicle to positive and negative. But obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that best classifies the vehicle and background images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then again same process is done. New error rates are calculated. Also new weights. The process is continued until required accuracy or error rate is achieved or required number of features are found).

* **Camera(s)** - that take the images of the car (front or rear side)
* **Illumination** - a controlled light that can bright up the plate, and allow day and night operation. In most cases the illumination is Infra-Red (IR) which is invisible to the driver.
* **Computer** - normally a PC running Windows or python. It runs application which controls the system, reads the images, analyzes and identifies the plate, and interfaces with other applications and systems.
* **Software** - the application and the recognition package. Usually the recognition package is supplied as a DLL (Dynamic Link Library).
* **Hardware** - various input/output boards used to interface the external world (such as control boards and networking boards)

There are a number of possible difficulties that the software must be able to cope with. These include:

1. Poor [image resolution](http://en.wikipedia.org/wiki/Image_resolution), usually because the plate is too far away but sometimes resulting from the use of a low-quality camera.

2. [Bad](http://en.wikipedia.org/wiki/Focus_%28optics%29) images particularly blur.

3. Poor [lighting](http://en.wikipedia.org/wiki/Light) and low contrast due to [overexposure](http://en.wikipedia.org/wiki/Exposure_%28photography%29), [reflection](http://en.wikipedia.org/wiki/Reflection_%28physics%29) or shadows.

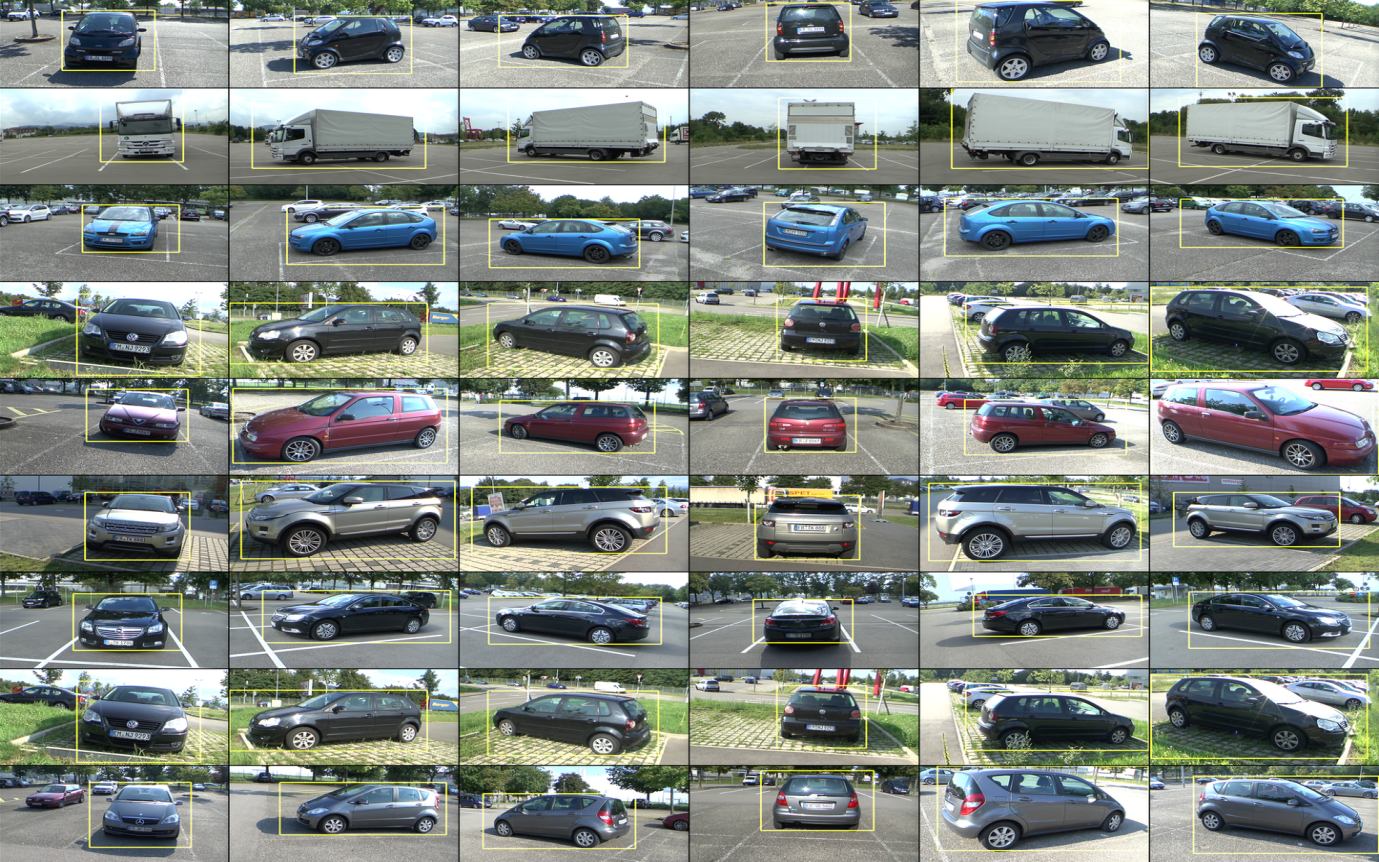
4. An object obscuring (part of) the plate, quite often a tow bar, or dirt on the plate.

5. A different font, popular for [vanity plates](http://en.wikipedia.org/wiki/Vanity_plate) (some countries do not allow such plates, eliminating the problem).

**SYSTEM IMPLIMENTATION**

**6.1 Vehicle detection:**

On-board vehicle detection systems have high computational requirements as they need to process the acquired images at real-time or close to real-time to save time for driver reaction. Searching the whole image to locate potential vehicle locations is prohibitive for real-time applications. The majority of methods reported in the literature follow two basic steps: 1) the locations of possible vehicles in an image and 2) where tests are performed to verify the presence of vehicles in an image (see Fig.3 and Fig.4). Although there is some overlap in the methods employed for each step, this taxonomy provides a good framework for discussion throughout this survey.



**Fig 6.1 Vehicle Detection**

**6.2 Pre-processing of data:**

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images(Vehicle Images) and Negative images (Background) to train the classifier. Then we need to extract features from it. For this, haar features shown in below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle.

Now all possible sizes and locations of each kernel is used to calculate plenty of features. For each feature calculation, we need to find sum of pixels under white and black rectangles. It simplifies calculation of sum of pixels, how large may be the number of pixels, to an operation involving just four pixels.

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the Vehicle to positive and negative. But obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that best classifies the vehicle and background images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then again same process is done. New error rates are calculated. Also new weights. The process is continued until required accuracy or error rate is achieved or required number of features are found).

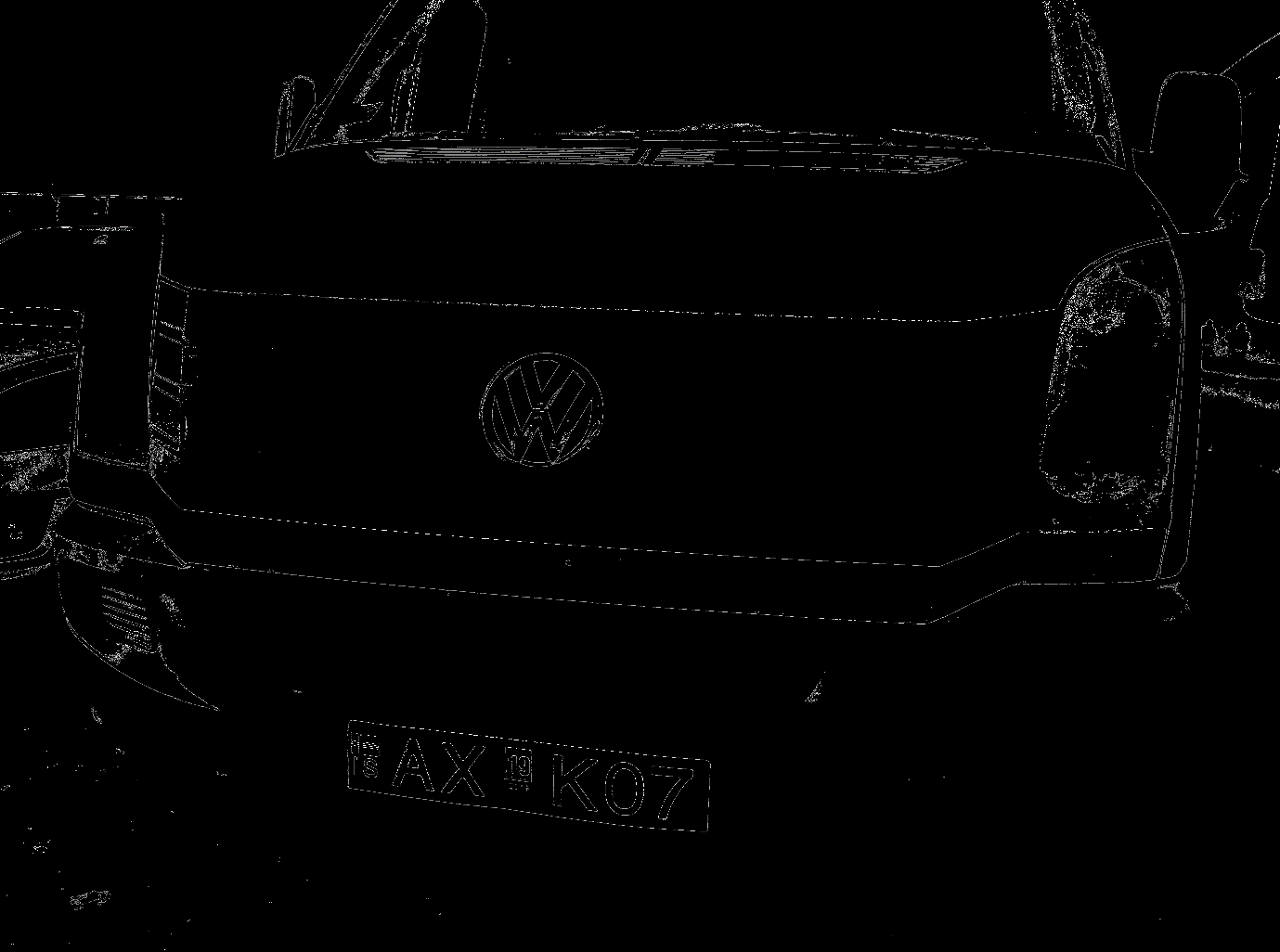
****

**Fig 6.2 Pre-processing of Vehicle**

**6.3 Number Plate Recognition :**

The current system is designed taking surveillance cameras, which usually record less frames per second and have low resolution, into consideration. For this reason, pre-processing is required to minimize noise and for sharpening edge information in the frame. This enhances plate regions and improves detection efficiency.

Fig 6.[3](https://link.springer.com/article/10.1007/s11554-011-0232-7" \l "Fig3)illustrates the effect of pre-processing. It can be seen that the original frame contains significant amount of salt and pepper noise and has poor edge information. Noise present in the foreground is removed by using well-known median filtering: a nonlinear technique that applies a sliding window on the frame pixels and removes noise while preserving the edge information. Morphological operators (erosion and dilation) are applied afterward to further refine the foreground.



**Fig 6.3 Pre-processing of Number plate**

### 6.3.1 License plate detection :

### After pre-processing, candidate regions (for license plate) are detected by finding the contours and boundaries of connected components in the frame. These connected components are further processed based on template matching criteria to void the false regions and decide about the true region of interest ‘ROI’. The ROI selection is carried out in two steps. First, the identified candidate regions are judged on the basis of their size and aspect ratio. If the size of any region is smaller or larger than a certain threshold, then that region is classified as a false region and is discarded from further processing. If the ‘width-to-height’ aspect ratio of any region is less than or greater than a certain threshold, then that region is also discarded for further processing.

### 

### Fig 6.4 Number plate Detected

**SYSTEM DESIGN**

System design is the process of defining the architecture, components, modules, interfaces and [data](http://en.wikipedia.org/wiki/Data) for a [system](http://en.wikipedia.org/wiki/System) to satisfy specified [requirements](http://en.wikipedia.org/wiki/Requirement). One could see it as the application of [systems theory](http://en.wikipedia.org/wiki/Systems_theory) to [product development](http://en.wikipedia.org/wiki/Product_development). There is some overlap with the disciplines of [systems analysis](http://en.wikipedia.org/wiki/Systems_analysis), [systems architecture](http://en.wikipedia.org/wiki/Systems_architecture) and [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering). If the broader topic of [product development](http://en.wikipedia.org/wiki/Product_development) "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing [systems](http://en.wikipedia.org/wiki/System) to satisfy specified [requirements](http://en.wikipedia.org/wiki/Requirement) of the user.

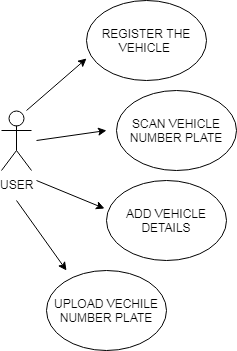
**UML Diagrams**

Unified Modeling Language (UML) is a standardized general-purpose [modeling language](http://en.wikipedia.org/wiki/Modeling_language) in the field of [object-oriented](http://en.wikipedia.org/wiki/Object-oriented) [software engineering](http://en.wikipedia.org/wiki/Software_engineering). The standard is managed, and was created, by the [Object Management Group](http://en.wikipedia.org/wiki/Object_Management_Group).

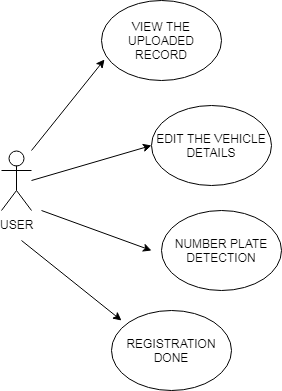
**Use Case Diagrams**

A use case diagram at its simplest is a graphical representation of a user's interaction with the system and depicting the specifications of a [use case](http://en.wikipedia.org/wiki/Use_Case). A use case diagram can portray the different types of users of a system and the various ways that they interact with the system.

**Use Case 0**

****

**Use Case 1**

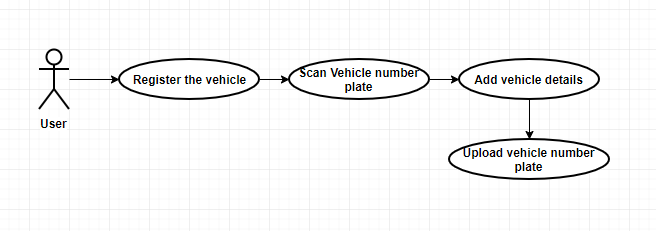
****

**6.1.2 Data Flow Diagram**

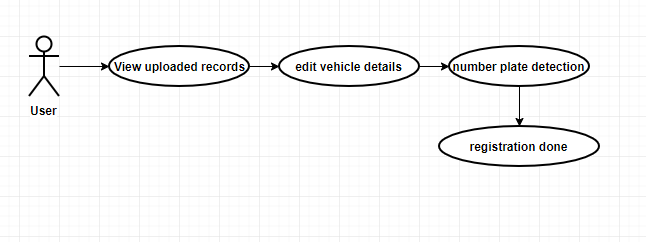
A data flow diagram is a graphical representation of the "flow" of data through an [information system](http://en.wikipedia.org/wiki/Information_system), modeling its *process* aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the [visualization](http://en.wikipedia.org/wiki/Data_visualization) of [data processing](http://en.wikipedia.org/wiki/Data_processing) (structured design).The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

**Dataflow Diagram:**

**DFD level 0**

****

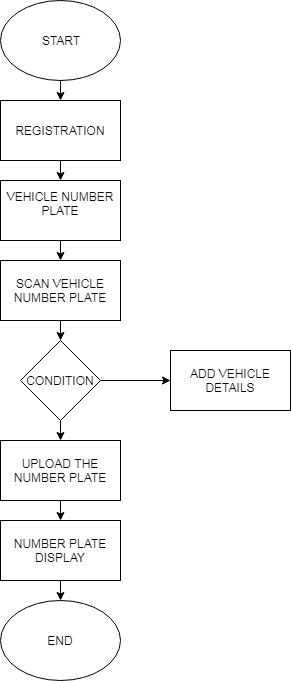
**DFD level 1**



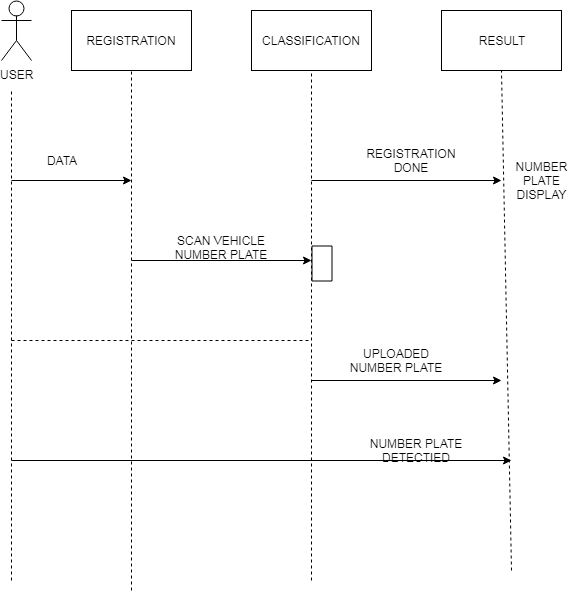
**6.2 Flowcharts**

A flow chart is a graphical or symbolic representation of a process. Each step in the process is represented by a different symbol and contains a short description of the process step. The flow chart symbols are linked together with arrows showing the process flow direction.

**Flowchart:**



**Sequence diagram:**

****

**SYSTEM REQUIREMENTS**

**3.1 Functional Requirements**

The particular necessities are UIs. The outside clients are the customers. The greater part of the customers can utilize this thing for referencing and looking.

* Hardware Interfaces: The outside gadget interface utilized for referencing and looking is PCs of the customers. The PC's quality be flexible PCs with remote LAN as the web affiliation gave will be remote.
* Software Interfaces: The working Frameworks can be any variety of windows.
* Performance Prerequisites: The PC's utilized must be atleast pentium 4 machine with the target that they can give ideal execution of the thing.

**3.2 Non-Functional Requirements**

Non utilitarian necessities are the cutoff centers offered by the structure. It merges time target and essential on the improvement framework and models. The non enormous basics are as appeared by the going with:

* Speed: The framework should set up the offered obligation to yield inside fitting time.
* Ease of usage: The thing tought to be quick. By then the clients can utilize feasibly, so it doesn't require much preparing time.
* Reliability: The rate of perplexities should be less then fundamentally the structure is rationally solid.
* Portability: It thought to be an option that is other than difficult to perceive in any structure

**3.3 Hardware requirements**

The most extensively watched game-plan of prerequisites depicted by any working framework or programming application is the physical PC assets, everything considered called fixing, A rigging necessities list is a remarkable bit of the time joined by a mechanical gathering closeness list, particularly if there should be an occasion of working structures. A HCL records endeavored, perfect, and from time to time incongruent apparatus gadgets for a specific working framework or application. The going with sub-parts review the different bits of gadget fundamentals.

All PC working frameworks are proposed for a specific PC structure. Most programming applications are constrained to express working frameworks running on unequivocal structures. Disregarding the manner by which that building free working frameworks and applications exist, most should be recompiled to keep running on another structure.

The essentialness of the focal arranging unit (CPU) is a central structure essential for anything. Most programming running on x86 building depict preparing power as the model and the clock speed of the CPU. Different highlights of a CPU that sway its speed and power, similar to transport speed, store, and MIPS are once in a while slighted. This criticalness of significance is constantly wrong, as AMD Intel Pentium CPUs at relative clock speed unpredictably have express throughput speeds.

• 10GB HDD(min)

• 128 MB RAM(min)

• Pentium P4 Processor 2.8Ghz(min)

**H/W System Configuration:-**

• RAM - 4 GB (min)

• Hard Disk - 20 GB

• Floppy Drive - 1.44 MB

• Key Board - Standard Windows Keyboard

• Mouse - Two or Three Button Mouse

• Monitor - SVGA

**Software requirements**

Programming necessities oversee depicting programming asset necessities and nuts and bolts that should be agreeable on a PC with give ideal working of an application.

These necessities or requirements are typically precluded in the thing establishment pack and should be shown self-governingly before the thing is introduced.

Python 2.7 or higher

* Pycharm
* opencv

**Outline of advances**

The movements utilized is delineated as underneath:

**Python**

* Python is a generally gainful bizarre state programming Language (human reasonable tongues are High estimation programming vernaculars)
* Python Developed by Guido Van Rossam
* 1989 National Research Institute(NRI) At Netherland
* Officially Python open to individuals when all is said in done in 1991 :: FEB twentieth 1991

Python was imagined in the late 1980s,[29] and its usage started in December 1989[30] by Guido van Rossum at Centrum Wiskunde and Informatica (CWI) in the Netherlands as a successor to the ABC vernacular (itself blended by SETL)[31]capable of expulsion managing and interfacing with the Amoeba working system.[6] Van Rossum remains Python's fundamental creator. His system with focal part in Python's advancement is reflected in the title given him by the Python social sales:

**Python Feature**

* Simple and simple to learn Python as just 33 catchphrases But JAVA as(53) watchwords
* Free thing (There is no give we can't pay anything) furthermore, Open source (we can arranged to see source code if source isn't phenomenal I can engineered to re-attempt our necessities)
* High level programming language (human sensible language) Python Is Platform Independent (It proposes I can make a program once and run any where(WORA)
* Portability Moving python program starting with one stage then onto the going with stage without changing anything
* Dynamically Typed Programming Language In python we are not required to express kind in Python
* Both Object Oriented and Procedure Oriented Language
* Interpreted Language It recommends we are not going to compose
* Extensible

We can utilize Other Programming Language in Python

**Limitations of python**

* Performance capable it isn't satisfactory Because its a deciphered language Mediator sorted out to see just a lone line (JAVA is better performance show up indisputably in relationship with python in java JIT (just in time compiler) thought is there
* Mobile applications it isn't satisfactory Dream:- python isn't fitting expansive scale try applications

**Flavors of python**

* Cpython :- It can be standard, It can be utilized to c language python
* Jpython or jpython :- it is for JAVA application
* Iron python:- to work with microsoft .net stage
* Pypy :- Internally JIT (just intime compiler) compiler is there so execution watchful very identity blowing
* Ruby python:- utilized for ruby application
* Anaconda python:- To oversee Big-information euphorically go for Anaconda python
* Stackless (python for concurrancy) :-
* parallely you execute (like mutithread) so go for stackless

**Applications of Python**

**1. GUI-Based Desktop Applications:**

Python has direct sentence structure, surveyed plan, rich substance preparing contraptions and the capacity to oversee unmistakable working frameworks which settle on it a surprising decision for making work region based applications. There are explicit GUI toolboxs like wxPython, PyQt or PyGtk accessible which help facilitators make utilitarian Graphical User Interface (GUI). The different applications made utilizing Python joins:

* **Image Processing and Graphic Design Applications:**

Python has been utilized to make 2D imaging programming, for example, Inkscape, GIMP, Paint Shop Pro and Scribus. Further, 3D vitality packs, similar to Blender, 3ds Max, Cinema 4D, Houdini, Lightwave and Maya, furthermore use Python in factor degrees.

* **Scientific and Computational Applications:**

The higher rates, limit and accessibility of contraptions, for example, Scientific Python and Numeric Python, have perceived Python changing into an essential piece of organizations pulled in with figuring and preparing of obvious information. 3D appearing, for example, FreeCAD, and bound fragment framework programming, for example, Abaqus, are coded in Python.

* **Games:**

Python has different modules, libraries and stages that help headway of beguilements. For instance, PySoy is a 3D preoccupation motor supporting Python 3, and PyGame gives quality and a library to amuse improvement. There have been various redirections amassed utilizing Python including Civilization-IV, Disney's Toontown Online, Vega Strike, and so on.

**2. Web Frameworks and Web Applications:**

Python has been utilized to make a beguilement plan of web-structures including CherryPy, Django, TurboGears, Bottle, Flask, and so forth. These structures give standard libraries and modules which improve errands identified with substance the directors, relationship with database and interfacing with various web appears, for example, HTTP, SMTP, XML-RPC, FTP and POP. Plone, a substance the store up structure; ERP5, an open source ERP which is utilized in flight, dress and banking; Odoo – a joined suite of business applications; and Google App motor are a few the standard web applications dependent on Python.

**3. Enterprise and Business Applications:**

With highlights that combine extraordinary libraries, extensibility, flexibility and effectively enormous semantic structure, Python is a reasonable coding language for tweaking progressively basic applications. Reddit, which was at first written in Common Lips, was balanced in Python in 2005. Python also contributed in a far reaching part to unfaltering quality in YouTube.

**4. Operating Systems:**

Python is reliably a basic piece of Linux spreads. For example, Ubuntu's Ubiquity Installer, and Fedora's and Red Hat Enterprise Linux's Anaconda Installer are written in Python. Gentoo Linux utilizes Python for Portage, its gathering the board structure.

**5. Language Development:**

Python's methodology and module approach has influenced improvement of various vernaculars. Boo language utilizes an article model, feature and space, like Python. Further, etymologicalstructure of tongues like Apple's Swift, CoffeeScript, Cobra, and OCaml all offer proportionality with Python.

**6. Prototyping:**

Other than being lively and simple to learn, Python in like way has the open source remarkable position of being free with the help of a wide framework. This settles on it the favored decision for model improvement. Further, the deftness, extensibility and adaptability and straightforwardness of refactoring code related with Python concede speedier headway from beginning prototype.Since its inspiration in 1989, Python has made to forefront toward convincing the opportunity to be bit of an a great deal of electronic, work a zone based, visual depiction, mindful, and computational applications. With Python accessible for Windows, Mac OS X and Linux/UNIX, it offers straightforwardness of progression for undertakings. In like way, the most recent discharge Python 3.4.3 builds up the present qualities of the language, with mind blowing improvement in Unicode support, among other new highlights.

**Versions of python**

* Python 1.0 Introduced in jan 1994
* Python 2.0 Introduced in oct 2000
* Python 3.0 introduced in dec 2008

latest version

python 3.6.3 🡺 2016

python 3.7

Any new version should provide support for old version programs

• There is no-retrogressive closeness support

• Python 3 isn't help to python 2 program

Python Machine Learning

Python is a superb stage utilized for creative work of age structures. It is a gigantic language with number of modules, parties and libraries that gives different approach for accomplishing an errand.

Python and its libraries like NumPy, SciPy, Scikit-Learn, Matplotlib are utilized in information science and information examination. They are likewise normally utilized for making versatile AI figurings. Python executes unavoidable AI structures, for example, Classification, Regression, Recommendation, and Clustering.

Python offers minute structure for performing information mining errands on enormous volumes of information adequately in lesser time. It joins a couple of utilization accomplished through estimations, for example, direct apostatize, key drop into bad behavior, Naïve Bayes, k-proposes, K closest neighbor, and Random Forest.

**Python in Machine Learning**

Python has libraries that pulls in makers to utilize improved figurings. It executes got a handle on AI structures, for example, proposition, strategy, and get-together. In like manner, it is basic to have a short prologue to AI before we move further.

**What is Machine Learning?**

Information science, AI and electronic reasoning are a portion of the top coasting subjects in the tech world today. Information mining and Bayesian examination are slanting and this is including the vitality for AI. This instructional exercise is your passage into the universe of AI.

Man-made comprehension is a control that manages programming the structures in order to make them consistently learn and improve with mindfulness. Here, learning begins seeing and understanding the information and taking encouraged choices subject to the gave information. It is hard to consider the majority of the choices subject to every potential information. To deal with this issue, checks are built up that produce information from a particular information and past experience by applying the measures of quantifiable science, likelihood, strategy for considering, numerical streamlining, post learning, and control theory.

**Occupations of Machine Learning Algorithms**

The made AI fuses are utilized in different applications, for example,

• Vision sorting out

• Language sorting out

• Forecasting things like securities exchange structures, condition

• Pattern assertion

• Games

• Data mining

• Expert structures

• Robotics

Libraries and Packages

To perceive AI, you need genuine learning of Python programming. Similarly, there are various libraries and bundles regularly utilized in performing shifting AI assignments as recorded underneath:

• numpy - is utilized for its N-dimensional get-together items

• pandas – is an information examination library that consolidates dataframes

• matplotlib – is 2D plotting library for making plans and plots

• scikit-learn - the figurings utilized for information examination and information mining assignments

• seaborn – an information confirmation library subject to matplotlib

**HARDWARE & SOFTWARE REQUIRMENT**

**H/W System Configuration:-**

* Raspberry pi
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

# S/W System Configuration:-

# Operating System : Windows xp,7,8

# Technology : Python

IDLE : Python 2.7 or higher

**TESTING**

Testing is a fundamental part which guarantees quality and reachability of the proposed structure in (fulfilling) meeting its targets. Testing is done at different stages in the System arranging and execution process with a goal of structure up an unquestionable, versatile and validated structure. Testing is a central piece of programming improvement. Testing process, in a way avows, paying little regard to whether the thing, that is made, agrees to the benchmarks, that it was required to. Testing process consolidates working of examinations, against which, the thing must be attempted.

**Test targets**

 Testing is a framework for executing a program to find a bungle.

 A amazing case is one that has a high likelihood of finding another stun.

 A surprising test is one that reveals a yet new chaos up. In the event that testing is driven reasonably (as indicated by the destinations) it will reveal crushes in the thing. Testing can't demonstrate the horrendous nonappearances of misshapenness are open. It can on a very basic level demonstrate that thing surrenders are open.

**Testing measures**

Before applying technique to structure productive examinations, a thing engineer must regard the real standard that guides programming testing. The vast majority of the tests ought to be distinguishable to client stray pieces.

**Testing plan**

Any structure thing can be attempted in one of two explicit ways:

**White box Testing**

This testing is likewise called as glass box testing. In this testing, by knowing the predefined work that a thing has been required to perform test can be supported that shows each point of confinement is absolutely progression in the interim looking for after down wrecks up in each generally extraordinary.

it is an examination plan strategy that utilizes the control structure of the procedural structure to total tests.

**Introduction Testing**

In this testing by knowing the internal endeavor of a thing, tests can be encouraged to guarantee that "all riggings work", that is inside development executes as appeared to be unequivocal and every single inward zone have been appealingly worked out. It on a fundamental estimation spotlights on the utilitarian necessities of the thing.

The methodology related with disclosure test setup is:

• Graph based testing structures

• Equivalence allotting

• Boundary respect examination

• Comparison testing

**Testing frameworks**

A thing testing framework gives a manual for the thing fashioner. Testing is a lot of exercises that can be masterminded in cutting edge and energized profitably. In this manner a course of action for programming testing a lot of undertakings into which we can put express examination structure strategies ought to be portrayed for programming building process.

Anything testing approach ought to have the going with characteristics:

a. Testing starts at the module level and works outward toward the joining of the whole PC based structure.

b. Different testing structures are sensible at various fixations in time.

c. The master of the thing and a self-coordinating test pack conducts testing.

d. Testing and looking changed exercises notwithstanding investigating must be fit in any testing system.

**Arrangement Testing**

The second piece of testing is called mix testing. Mix testing is a wary framework for structure the program structure while controlling tests to reveal bumbles related with interfacing. In this, many attempted modules are joined into subsystems, which are then endeavored. The objective here is to check whether the greater part of the modules can be enabled reasonably.

There are three sorts of trade off testing:

 Top-Down Integration: Top down mix is an immovable framework to administer improvement of program structures. Modules are merged by moving downwards heave the control dynamic structure starting with the colossal control module.

 Bottom-Up Integration: Bottom up joining as its name proposes, starts Construction and testing with changed modules.

Fall a long way from the certainty Testing: In this primer of a circuit test strategy, apostatize testing is the re execution of some subset of test that have beginning late been encouraged to guarantee.

**Unit Testing**

The focal bit of testing is called unit testing. Unit testing pays special mind to the most minute unit of programming structures the module. The unit test is consistently white box facilitated. In this, various modules are attempted against the nuances passed on amidst plan for the modules. Unit testing is on a key measurement for check of the code made amidst the coding stage, and starting now and into the not all that removed the objective is to test within side enthusiasm of the modules. It is ordinarily done by the thing originator of the module. Because of its neighboring relationship with coding, the coding stage is every once in a while called "coding and unit testing." The unit test can be driven in parallel for different modules.

The Test cases in unit testing are as follows:

The Test cases in unit testing are as follows:

Table I: Unit Test Case 1

|  |  |
| --- | --- |
| Test Case ID | Unit Test Case 1 |
| Description | Process without loading an image |
| Input | nothing |
| Expected output | It will show error message – “please load a valid image” |
| Actual Result/Remarks | Got the expected output |
| Passed(?) | Yes |

Table II: Unit Test Case 2

|  |  |
| --- | --- |
| Test Case ID | Unit Test Case 2 |
| Description | Load an image |
| Input | .jpg image |
| Expected output | Gray scale image |
| Actual Result/Remarks | Got the expected output |
| Passed(?) | Yes |

Table III: Unit Test Case 3

|  |  |
| --- | --- |
| Test Case ID | Unit Test Case 3 |
| Description | Load an image without number plate |
| Input | .jpg image |
| Expected output | It will show error message “number plate not present” |
| Actual Result/Remarks | Got the expected output |
| Passed(?) | Yes |

Table II: Unit Test Case 4

|  |  |
| --- | --- |
| Test Case ID | Unit Test Case 4 |
| Description | Load an image with noisy number plate |
| Input | .jpg image |
| Expected output | It will show error message “number plate is noisy” |
| Actual Result/Remarks | Got the expected output |
| Passed(?) | Yes |

Table III: Unit Test Case 5

|  |  |
| --- | --- |
| Test Case ID | Unit Test Case 5 |
| Description | Load an image with language other than english |
| Input | .jpg image |
| Expected output | It will show error message “templates cannot be matched” |
| Actual Result/Remarks | Got the expected output |
| Passed(?) | Yes |

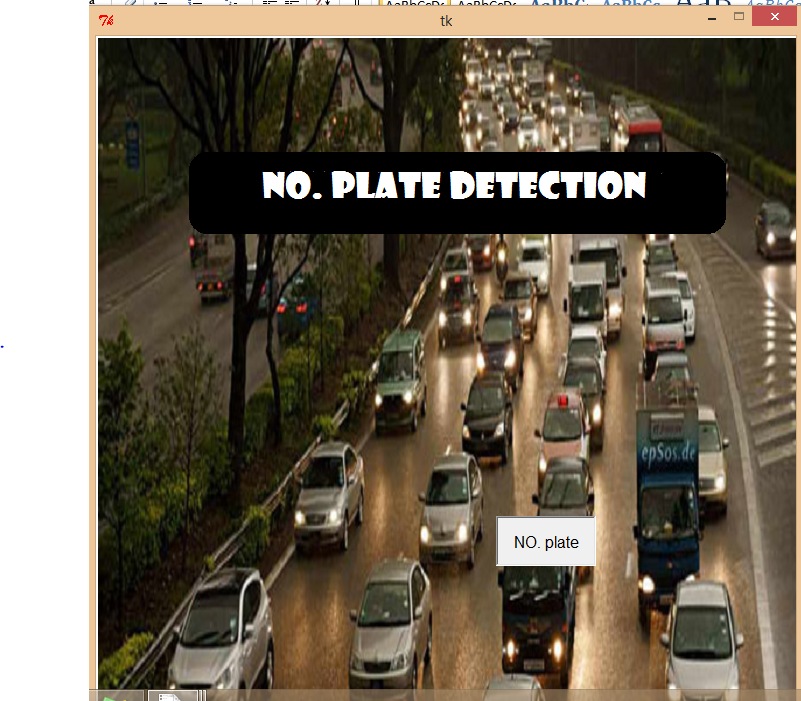
Table III: Unit Test Case 6

|  |  |
| --- | --- |
| Test Case ID | Unit Test Case 6 |
| Description | Load an image with number plate of different font |
| Input | .jpg image |
| Expected output | It will show error message “templates cannot be matched” |
| Actual Result/Remarks | Got the expected output |
| Passed(?) | Yes |

**EXPERIMENTAL RESULTS**

We have run our proposed method on desktop computer Several vehicle images are taken using 1.3 mega pixel camera as well as 12 mega pixel cameras. In the experiments, we test our proposed method on the different type car image to identify the location exactly.

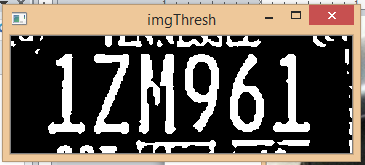
SCREENSHOTS

















**CONCLUSION AND FUTURE WORKS**

An efficient less time consuming vehicle number plate detection method is projected which performed on multifaceted image. By using, edge detection method here detects edges and fills the holes less than 8 pixels only. To removing the license plate we remove connected components less than 1000 pixels. Our anticipated algorithm is mainly based on Indian automobile number plate system. Extraction of number plate accuracy may be increased for low ambient light image.

REFERANCE

1] S. Hamidreza Kasaei, S. Mohammadreza Kasaei, S. Alireza Kasaei

International Journal of Computer Theory and Engineering, Vol. No. 2,

2 April, 2010.

[2] Serkan Ozbay, and Ergun Ercelebi 1793-8201

World Academy of Science, Engineering and Technology 9 2005

[3] Deepak Kumar Gupta-Y6154, Siddhartha Kandoi-Y6472

CS 676: Image Processing and Computer Vision 2009-10 Semester 1

Books**:**

1. International Journal of Computer Theory and Engineering, Vol. No. 2,

2 April, 2010, Pg. 57.

2. R. Gonzalez, R .Woods , Digital Image Processing, Prentice Hall, New Jersey,2002, Chapter

5. Operations on an image.

Websites**:**

1. [http://www.anpr-tutorial.com](http://www.anpr-tutorial.com/)

2. http://www.cctv-information.co.uk/i/An\_Introduction\_to\_ANPR

3. http://www.visl.technion.ac.il/projects/2003w24/

4. http://www.stackoverflow.com

5. <http://opencv.willowgarage.com/documentation>

6. http://www.aishack.com