Project Report
On

AUTOMATIC LICENSE PLATE DETECTION & RECOGNITION USING STANDALONE IMAGE PROCESSING DEVICE

Project Report Submitted for Partial Fulfilment of the Requirement for B.Tech Degree in Instrumentation Engineering
From the
Department of Applied Physics
University of Calcutta

Under the Guidance of
Dr. Ujjwal Mondal

Submitted by

SOUMIL BANDYOPADHYAY
4 YEAR B.TECH 8TH SEMESTER
ROLL: T91/IE/174093
REG NO: D01-1111-0204-17

SHUBHA ROY
4 YEAR B.TECH 8TH SEMESTER
ROLL: T91/IE/174090
REG NO: D01-1112-0147-17

RAVINANDAN KUMAR
4 YEAR B.TECH 8TH SEMESTER
ROLL: T91/IE/174098
REG NO: D01-1111-0162-17

DEPARTMENT OF APPLIED PHYSICS
UNIVERSITY COLLEGE OF TECHNOLOGY
UNIVERSITY OF CALCUTTA
92, Acharya Prafulla Chandra Road, Kolkata - 700 009, West Bengal, India
ফলিত পদার্থবিদ্যা বিভাগ, কলিকাতা বিশ্ববিদ্যালয়

CERTIFICATE OF APPROVAL

The Project Report titled **"AUTOMATIC LICENSE PLATE DETECTION & RECOGNITION USING STANDALONE IMAGE PROCESSING DEVICE"** prepared and submitted by **SHUBHA ROY, SOUMIL BANDYOPADHYAY and RAVINANDAN KUMAR**, 8th Semester, B.Tech, Instrumentation Engineering, is hereby approved and certified as a creditable study in technological subjects performed in a way sufficient for its acceptance for particular fulfilment of the degree for which submitted.

It is to be understood by this approval, the undersigned does not, necessarily endorse or approve any statement made, opinion expressed or, conclusion drawn therein, but approved for the purpose for which it is submitted.

……………………………
Examinars'

Ujjwal Mondal

Dr. Ujjwal Mondal
Associate Professor, Instrumentation Engineering
Dept. of Applied Physics
University of Calcutta
92, APC Road, Kolkata - 700 009, India

……………………………..
Dr.Ujjwal Mondal
(Supervisor)
Associate Professor,
Instrumentation Engineering,
Department of Applied Physics
University of Calcutta

Professor Rajarshi Gupta
Head, Dept. of Applied Physics
University of Calcutta
92 APC Road, Kolkata-700009, India
rgaphy@caluniv.ac.in, 033-23508386 (E-471)

Digitally signed by Dr. Rajarshi Gupta
DN: C=IN, OU=Applied Physics, O=University of Calcutta, CN=Dr. Rajarshi Gupta, E=rgaphy@caluni.ac.in
Reason: I am the author of this document
Location: your signing location here
Date: 2021.08.26 06:16:16+05'30'
Foxit PhantomPDF Version: 10.1.4

…………………………………..
Prof.(Dr).Rajarshi Gupta
Head of the Department
Dept. of Applied Physics,
University of Calcutta

DEPARTMENT OF APPLIED PHYSICS
UNIVERSITY COLLEGE OF TECHNOLOGY
UNIVERSITY OF CALCUTTA
92, Acharya Prafulla Chandra Road, Kolkata - 700 009, West Bengal, India

ফলিত পদার্থবিদ্যা বিভাগ, কলিকাতা বিশ্ববিদ্যালয়

RECOMMENDATION

It is hereby recommended that the Project Report entitled **"AUTOMATIC LICENSE PLATE DETECTION & RECOGNITION USING STANDALONE IMAGE PROCESSING DEVICE"** prepared and submitted by **SHUBHA ROY, SOUMIL BANDYOPADHYAY and RAVINANDAN KUMAR** be accepted in partial fulfilment of the B.Tech degree in Instrumentation Engineering from the Department of Applied Physics, University of Calcutta for which it is submitted.

(Supervisor)
Dr. Ujjwal Mondal
Associate Professor,
Department of Applied Physics,
University of Calcutta

ACKNOWLEDGEMENTS

We are thankful to the Department of Applied Physics, University of Calcutta for giving us the opportunity to carry on our project work with all the necessary guidance, inspiration and support.

It is our privilege to express our profound gratitude to Dr. Ujjwal Mondal, Associate Professor, Department of Applied Physics, and University of Calcutta for his guidance, valuable assistance and inspiration throughout the course without which it would have been difficult to complete our project work.

We are also thankful to the Head of the Department of Applied Physics, University of Calcutta for providing us the opportunity to pursue the work.

We are thankful to Prof.(Dr.) Madhuchchanda Mitra, Professor, Department of Applied Physics, University of Calcutta, Dr. Saurabh Pal, Associate Professor, Department of Applied Physics, University of Calcutta, Dr.Chanchal Dey, Associate Professor, Department of Applied Physics, University of Calcutta, for their valuable assistance without which it would have been difficult to complete our work.

We also express our heartiest gratitude towards the faculties of the Department of Applied Physics and the Department Library, Applied Physics for their consistent support in supplying the necessary resources.

Shubha Roy.

SHUBHA ROY

Soumil Bandyopadhyay

SOUMIL BANDYOPADHYAY

Ravinandan kumar

RAVINANDAN KUMAR

ABSTRACT

The number plate recognition (NPR) system is one of the categories of smart transportation and detection mechanism (STDM). This is a combination of the technology in which the application enables the system to detect and automatically read the license id of number plate of vehicle from digitally captured images. Automatically capturing the license plate is the process of detecting and transforming the pixels data of a digital image into the plain text data or ASCII text of the number plate. Our project contains a method for the vehicle number plate recognition from the image using mathematical morphological operations (erosion, dilation). The main objective is to use and combine different morphological operations in such a way that the license plate of the certain vehicle can be detected and translated effectively. This is based on various operation such as image improvement, Gray scale transformation, Bilateral Filtering edge detection and getting the number plate from the picture of vehicle. After the completion of the above-mentioned steps, now the process of segmentation is being applied to detect the text present on number plate by making use of matching of template and OCR. This system is able to detect the license number accurately as well as quickly from the vehicle’s picture.

Automatically capturing the license plate is the process of detecting and transforming the pixels data of a digital image into the plain text data or ASCII text of the number plate. Our project contains a method for the vehicle number plate recognition from the image using mathematical morphological operations (erosion, dilation). The main objective is to use and combine different morphological operations in such a way that the license plate of the certain vehicle can be detected and translated effectively. This is based on various operation such as image improvement, Gray scale transformation, Bilateral Filtering edge detection and getting the number plate from the picture of vehicle. After the completion of the above-mentioned steps, now the process of segmentation is being applied to detect the text present on number plate by making use of OCR. This system is able to detect the license number accurately as well as quickly from the vehicle’s picture.
After successful detection of the Vehicle Number, we are matching the number with Firestore Cloud Database and detecting the owner of the Vehicle. Also the real-time data is shared with Google Sheets storage Cloud.

CONTENTS

CHAPTER: 1

INTRODUCTION

Number plates are used for identification of vehicles all over the nations. Vehicles are identifying either manually or automatically. Automatic vehicle identification is an image processing technique to identify vehicles by their number plates. Automatic vehicle identification systems are used for the purpose of effective traffic control and security applications such as access control to restricted areas and tracking of wanted vehicles. Number plate recognition (NPR) is easier method for Vehicle identification. NPR system for Indian license plate is difficult compared to the foreign license plate as there is no standard followed for the aspect ratio of licence plate.

Experimentation of number plate detection has been conducted from many years, it is still a challenging task. Number plate detection system investigates an input image to identify some local patches containing license plates. Since a plate can exist anywhere in an image with various sizes, it is infeasible to check every pixel of the image to locate it. In parking, number plates are used to calculate duration of the parking. When a vehicle enters an input gate, number plate is automatically recognized and stored in database.

In NPR system spectral analysis approach is used were acquiring the image, extract the region of interest, character segmentation using SVM feature extraction techniques. The advantage of this approach is success full recognition of a moving vehicle .It is difficult to detect the boundary of the Number plate from the input car images in outdoors scene due to colour of characters of the number plate and Background of the Number plate the gradients of the original image is adopted to detect candidate number plate regions. There are also algorithms which are based on a combination of morphological operation, segmentation and canny edge detector. License plate location algorithm consist of steps like as Edge Detection, Morphological operation like dilation and erosion, Smoothing, segmentation of characters and recognition of plate characters are described later.

1.1 LITERATURE SURVEY:

Muhammad Tahir Qadrihas started the recognition the OCR techniques is used which is susceptible to misalignment and to various sizes. The transformation achieved by him can be used to advance the OCR identification from various size and angles in real-time. This programmed system for vehicle detection and recognition system by making use of vehicle license number plate is exhibited. A combination of image processing techniques of the model for detecting and recognizing the vehicle from the databases stored in the system.

ANPR is used in the presentation of stolen and crime suspected vehicles.
ANPR can be easily used in various manners by using to identify it stolen vehicle on the highway or in parking lot or in traffic signals. This is all about the literature survey of the project “Automatic Vehicle Number Plate Recognition System.”

1.2 MOTIVATION:

- A simple, cheap, configurable, easy to handle imaging based system has been proposed to detect and recognise license plate of a vehicle automatically.

- Advancement of technology made it possible to run such algorithms on low powered cheap electronic hardware.

1.3 AIM OF THE PROJECT:

- The aim of the project is to develop vehicle number plate detection system using OPEN-CV PYTHON.

- Phase-2 covers how the entire system works

- Understanding and being able to work with the software components of OPEN-CV PYTHON.

- Being able to use the program to generate results from static images , matching the results with a created database and storing it to the cloud

1.4 APPLICATION OF THIS PROJECT:

1. Parking: **-** The NPR is used to automatically enter prepaid members and calculate parking fee for non-members.

2. Access control: **-** A gate automatically opens for authorized members in a secured area, thus replacing or assisting the security guard.

3. Tolling: **-** The car number is used to calculate the travel fee in a toll-road or used to double check the ticket.

4. Border Security: **-** The car number is registered in the entry or exits to the country and used to monitor the border crossings.

5. Traffic Control: **-** The vehicles can be directed to different lanes according to their entry permits. The system reduces the traffic congestions and number of attendants.

6. Airport Parking: **-** In order to reduce ticket frauds or mistakes, the NPR unit is used to capture the number plate and image of the car.

1.5 Some Limitations of This Project:

1. Broken number plate.
2. Blurry images.
3. Number plate not within the legal specification.
4. Low resolution of the characters.
5. Poor maintenance of the vehicle plate.
6. Similarity between certain characters, namely, O and D; 5 and S; 8 and B, E; O, etc.
7. Internet connection is required for cloud based services.

1.6 Software and Hardware Requirements:-

1.6.1 Hardware Required

1. PC with minimum 4 GB RAM
2. Stable Internet Connection

1.6.2 Software Required

1. Python 3
2. PyCharm IDE
3. OpenCv
4. Tesseract OCR

CHAPTER: 2

FUNDAMENTALS OF IMAGE PROCESSING

Images define the world, each image has its own story, and it contains a lot of crucial information that can be useful in many ways. This information can be obtained with the help of the technique known as **Image Processing**.

It is the core part of computer vision which plays a crucial role in many real-world examples like robotics, self-driving cars, and object detection. Image processing allows us to transform and manipulate thousands of images at a time and extract useful insights from them. It has a wide range of applications in almost every field.

Python is one of the widely used programming languages for this purpose. Its amazing libraries and tools help in achieving the task of image processing very efficiently.

Through this article, you will learn about classical algorithms, techniques, and tools to process the image and get the desired output.

What is image processing?

As the name says, image processing means processing the image and this may include many different techniques until we reach our goal.

The final output can be either in the form of an image or a corresponding feature of that image. This can be used for further analysis and decision making.

But what is an image?

An image can be represented as a 2D function F(x,y) where x and y are spatial coordinates. The amplitude of F at a particular value of x,y is known as the intensity of an image at that point. If x,y, and the amplitude value is finite then we call it a digital image. It is an array of pixels arranged in columns and rows. Pixels are the elements of an image that contain information about intensity and color. An image can also be represented in 3D where x,y, and z become spatial coordinates. Pixels are arranged in the form of a matrix. This is known as an **RGB image**.

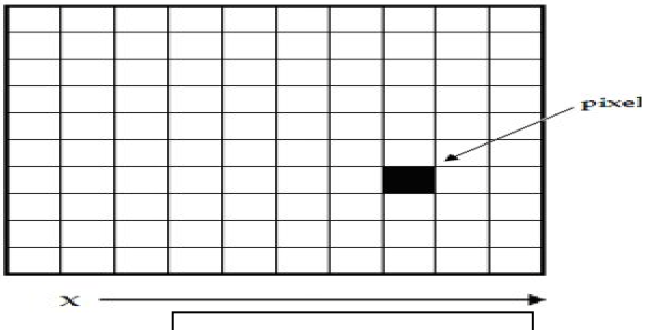


Fig 2.1 Matrix of an image

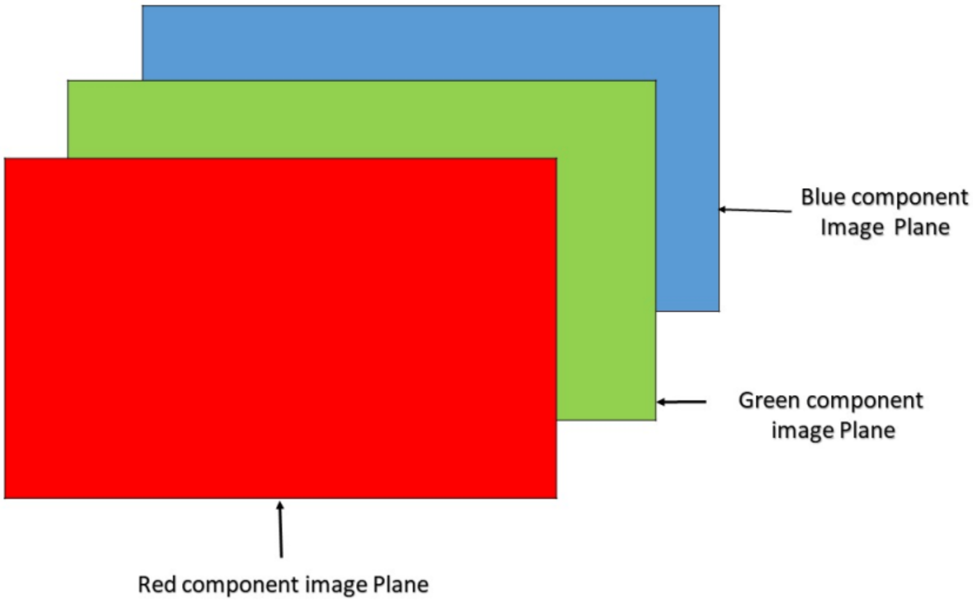


Fig. 2.2 Three layered RGB image

There are various types of images:

- RGB image: It contains three layers of 2D image, these layers are Red, Green, and Blue channels.
- Grayscale image: These images contain shades of black and white and contain only a single channel.

CHAPTER: 3

THE NPR SYSTEM

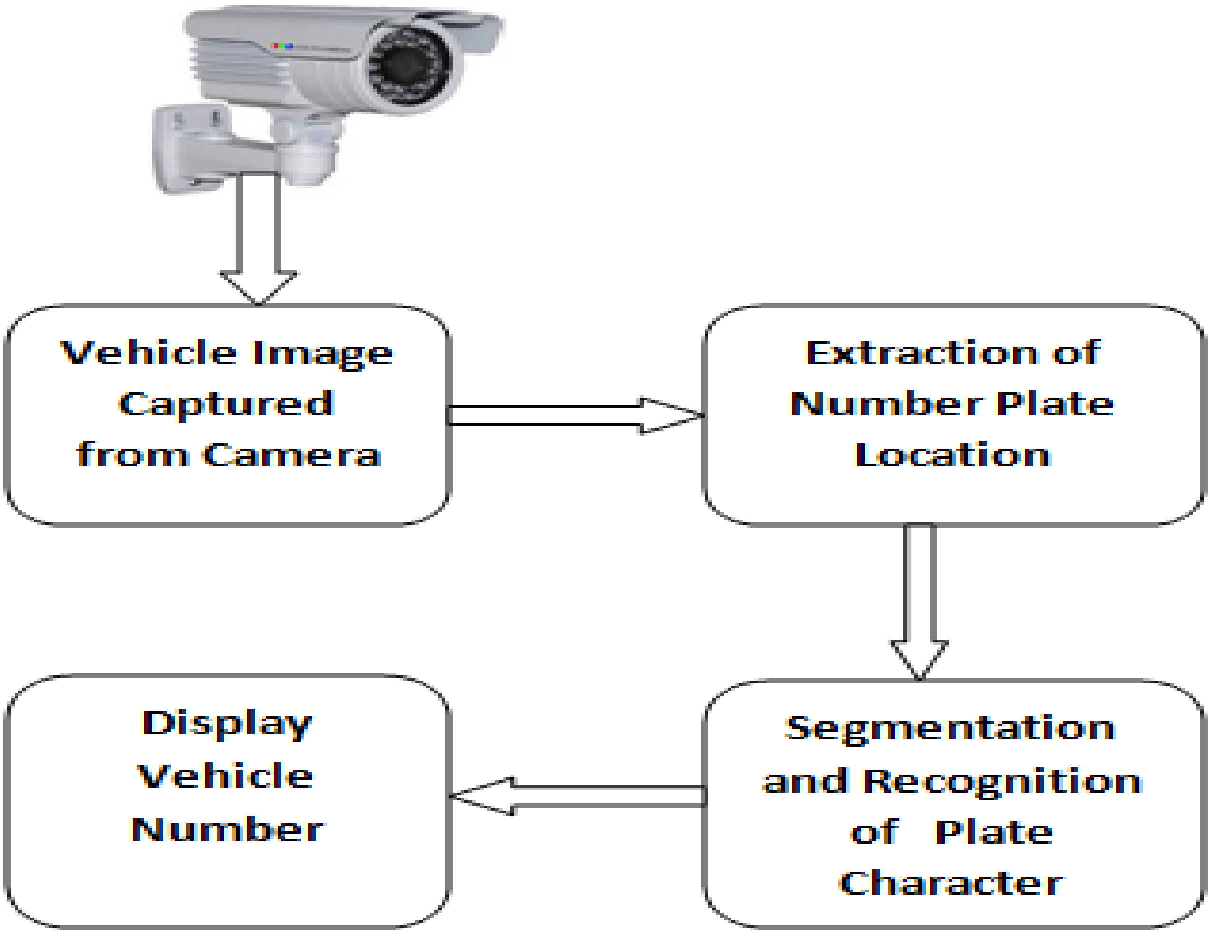

Vehicle Image Captured from Camera
Extraction of Number Plate Location
Segmentation and Recognition of Plate Character
Display Vehicle Number

Fig 3.1 Number Plate Recognition System

Blocks of general NPR system are discussed below:

A. Vehicle Image Captured By Camera

The image of the vehicle whose number plate is to be identified is captured using digital camera.

B.EXTRACTION OF NUMBER PLATE LOCATION

In this step the number plate is extracted by firstly converting RGB Image i.e., the captured image to Gray Scale Image. Here mathematical morphology is used to detect the region and Canny operator are used to calculate the threshold value. After this we get a dilated image.

C. SEGMENTATION AND CHARACTER RECOGNITION

Here contour area technique is used for segmentation. The contour is used to measure the properties of the image region. The basic step in recognition of vehicle number plate is to detect the plate size. Here the segmented image is multiplied with gray scale image so that we only get the number plate of the vehicle.

D. DISPLAY THE VEHICLE NUMBER

After undergoing the above steps the number plate is displayed in PYTHON window

CHAPTER: 4

Open CV

4.1 Introduction to Open CV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. Open CV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, Open CV makes it easy for businesses to utilize and modify the code.

Fig. 4.1 OpenCV logo

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. Open CV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many start-ups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of Open CV. Open CV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. Open CV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. Open CV is written natively in C++ and has a template interface that works seamlessly with STL containers.

4.2 Advantages of OpenCV over MATLAB

- ✓ Speed: Matlab is built on Java, and Java is built upon C. So when you run a Matlab program, your computer is busy trying to interpret all that Matlab code. Then it turns it into Java, and then finally executes the code. OpenCV, on the other hand, is basically a library of functions written in C/C++. You are closer to directly provide machine language code to the computer to get executed. So ultimately you get more image processing done for your computers processing cycles, and not more interpreting. As a result of this, programs written in OpenCV run much faster than similar programs written in Matlab. So, conclusion? OpenCV is damn fast when it comes to speed of execution. For example, we might write a small program to detect people's smiles in a sequence of video frames. In Matlab, we would typically get 3-4 frames analyzed per second. In OpenCV, we would get at least 30 frames per second, resulting in real-time detection.

- ✓ Resources needed: Due to the high level nature of Matlab, it uses a lot of your systems resources. And I mean A LOT! Matlab code requires over a gig of RAM to run through video. In comparison, typical OpenCV programs only require ~70mb of RAM to run in real-time.

- ✓ Cost: List price for the base (no toolboxes) MATLAB (commercial, single user License) is around USD 2150. OpenCV (BSD license) is *free!* Now, how do you beat that?

- ✓ Portability: MATLAB and OpenCV run equally well on Windows, Linux and MacOS. However, when it comes to OpenCV, any device that can run C, can, in all probability, run OpenCV.

- ✓ Specific: OpenCV was made for image processing. Each function and data structure was designed with the Image Processing coder in mind. Matlab, on the other hand, is quite generic. You get almost anything in the world in the form of toolboxes. All the way from financial toolboxes to highly specialized DNA toolboxes.

4.3 Despite all these amazing features, OpenCV does lose out over MATLAB on some points:

- ✓ Ease of use: Matlab is a relatively easy language to get to grips with. Matlab is a pretty high-level scripting language, meaning that you don't have to worry about libraries, declaring variables, memory management or other lower-level programming issues. As such, it can be very easy to throw together some code to prototype your image processing idea. Say for example I want to read in an image from file and display it.

- ✓ Memory Management: OpenCV is based on C. As such, every time you allocate a chunk of memory you will have to release it again. If you have a loop in your code where you allocate a chunk of memory in that loop and forget release it afterwards, you will get what is called a "leak". This is where the program will

use a growing amount of memory until it crashes from no remaining memory. Due to the high-level nature of
Matlab, it is “smart” enough to automatically allocate and release memory in the background.
-Matlabs memory management is pretty good. Unless you’re careful with your OpenCV memory allocation and releasing, you can still be frustrated beyond belief.

- ✓ Development Environment: Matlab comes with its own development environment. For OpenCV, there is no particular IDE that you have to use. Instead, you have a choice of any C programming IDE depending on whether you are using Windows, Linux, or OS X. For Windows, Microsoft Visual Studio or NetBeans is the typical IDE used for OpenCV. In Linux, its Eclipse or NetBeans, and in OSX, we use Apple’s Xcode.

- ✓ Debugging: Many of the standard dedugging operations can be used with both Matlab and OpenCV: breakpoints can be added to code, the execution of lines can be stepped through, variable values can be viewed during code execution etc. Matlab however, offers a number of additional debugging options over OpenCV. One great feature is that if you need to quickly see the output of a line of code, the semi-colon at the end can be omitted. Also, as Matlab is a scripting language, when execution is stopped at a particular line, the user can type and execute their own lines of code on the fly and view the resulting output without having to recompile and link again. Added to this is are Matlab’s powerful functions for displaying data and images, resulting in Matlab being our choice for the easiest development environment for debugging code.

CHAPTER: 5

Google Cloud Platform

Google Cloud Platform (GCP), offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. Registration requires a credit card or bank account details.

Google Cloud Platform provides infrastructure as a service, platform as a service, and server less computing environments.

In April 2008, Google announced App Engine, a platform for developing and hosting web applications in Google-managed data centres, which was the first cloud computing service from the company. The service became generally available in November 2011. Since the announcement of App Engine, Google added multiple cloud services to the platform.

Google Cloud Platform is a part of Google Cloud, which includes the Google Cloud Platform public cloud infrastructure, as well as Google Workspace (G Suite), enterprise versions of Android and Chrome OS, and application programming interfaces (APIs) for machine learning and enterprise mapping services.

5.1 Google lists over 100 products under the Google Cloud brand. Some of the key services are listed below.

Fig. 5.1 Conference presentation on Google Cloud

5.1.1 Compute

- App Engine – Platform as a Service to deploy Java, PHP, Node.js, Python, C#, .Net, Ruby and Go applications.
- Compute Engine - Infrastructure as a Service to run Microsoft Windows and Linux virtual machines.

- Google Kubernetes Engine (GKE) or GKE on-prem offered as part of Anthos platform - Containers as a Service based on Kubernetes.
- Cloud Functions - Functions as a Service to run event-driven code written in Node.js, Java, Python, or Go.
- Cloud Run - Compute execution environment based on Knative. Offered as Cloud Run (fully managed) or as Cloud Run for Anthos. Currently supports GCP, AWS and VMware management.

5.1.2 Storage & Databases

- Cloud Storage - Object storage with integrated edge caching to store unstructured data.
- Cloud SQL - Database as a Service based on MySQL, PostgreSQL and Microsoft SQL Server.
- Cloud Bigtable - Managed NoSQL database service.
- Cloud Spanner - Horizontally scalable, strongly consistent, relational database service.
- Cloud Datastore - NoSQL database for web and mobile applications.
- Persistent Disk - Block storage for Compute Engine virtual machines.
- Cloud Memorystore - Managed in-memory data store based on Redis and Memcached.
- Local SSD: High-performance, transient, local block storage.
- Filestore: High-performance file storage for Google Cloud users.

5.1.3 Networking

- VPC - Virtual Private Cloud for managing the software defined network of cloud resources.
- Cloud Load Balancing - Software-defined, managed service for load balancing the traffic.
- Cloud Armor - Web application firewall to protect workloads from DDoS attacks.
- Cloud CDN - Content Delivery Network based on Google's globally distributed edge points of presence.
- Cloud Interconnect - Service to connect a data center with Google Cloud Platform
- Cloud DNS - Managed, authoritative DNS service running on the same infrastructure as Google.
- Network Service Tiers - Option to choose Premium vs Standard network tier for higher-performing network.

5.1.4 Big Data

- BigQuery - Scalable, managed enterprise data warehouse for analytics.
- Cloud Dataflow - Managed service based on Apache Beam for stream and batch data processing.
- Cloud Dataproc - Big data platform for running Apache Hadoop and Apache Spark jobs.

- Cloud Composer - Managed workflow orchestration service built on Apache Airflow.
- Cloud Datalab - Tool for data exploration, analysis, visualization and machine learning. This is a fully managed Jupyter Notebook service.
- Cloud Dataprep - Data service based on Trifacta to visually explore, clean, and prepare data for analysis.
- Cloud Pub/Sub - Scalable event ingestion service based on message queues.
- Cloud Data Studio - Business intelligence tool to visualize data through dashboards and reports.

5.1.5 Cloud AI

- Cloud AutoML - Service to train and deploy custom machine, learning models. As of September 2018, the service is in Beta.
- Cloud TPU - Accelerators used by Google to train machine learning models.
- Cloud Machine Learning Engine - Managed service for training and building machine learning models based on mainstream frameworks.
- Cloud Job Discovery - Service based on Google's search and machine learning capabilities for the recruiting ecosystem.
- Dialogflow Enterprise - Development environment based on Google's machine learning for building conversational interfaces.
- Cloud Natural Language - Text analysis service based on Google Deep Learning models.
- Cloud Speech-to-Text - Speech to text conversion service based on machine learning.
- Cloud Text-to-Speech - Text to speech conversion service based on machine learning.
- Cloud Translation API - Service to dynamically translate between thousands of available language pairs
- Cloud Vision API - Image analysis service based on machine learning
- Cloud Video Intelligence - Video analysis service based on machine learning

5.1.6 Management Tools

- Operations suite (formerly Stack driver) - Monitoring, logging, and diagnostics for applications on Google Cloud Platform and AWS.
- Cloud Deployment Manager - Tool to deploy Google Cloud Platform resources defined in templates created in YAML, Python or Jinja2.
- Cloud Console - Web interface to manage Google Cloud Platform resources.
- Cloud Shell - Browser-based shell command-line access to manage Google Cloud Platform resources.
- Cloud Console Mobile App - Android and iOS application to manage Google Cloud Platform resources.
- Cloud APIs - APIs to programmatically access Google Cloud Platform resources

5.1.7 Identity & Security

- Cloud Identity - Single sign-on (SSO) service based on SAML 2.0 and OpenID.

- Cloud IAM - Identity & Access Management (IAM) service for defining policies based on role-based access control.
- Cloud Identity-Aware Proxy - Service to control access to cloud applications running on Google Cloud Platform without using a VPN.
- Cloud Data Loss Prevention API - Service to automatically discover, classify, and redact sensitive data.
- Security Key Enforcement - Two-step verification service based on a security key.
- Cloud Key Management Service - Cloud-hosted key management service integrated with IAM and audit logging.
- Cloud Resource Manager - Service to manage resources by project, folder, and organization based on the hierarchy.
- Cloud Security Command Centre - Security and data risk platform for data and services running in Google Cloud Platform.
- Cloud Security Scanner - Automated vulnerability scanning service for applications deployed in App Engine.
- Access Transparency - Near real-time audit logs providing visibility to Google Cloud Platform administrators.
- VPC Service Controls - Service to manage security perimeters for sensitive data in Google Cloud Platform services.

5.1.8 IoT

- Cloud IoT Core - Secure device connection and management service for Internet of Things.
- Edge TPU - Purpose-built ASIC designed to run inference at the edge. As of September 2018, this product is in private beta.
- Cloud IoT Edge - Brings AI to the edge computing layer.

5.1.9 API Platform

- Maps Platform - APIs for maps, routes, and places based on Google Maps.
- Apigee API Platform - Lifecycle management platform to design, secure, deploy, monitor, and scale APIs.
- API Monetization - Tool for API providers to create revenue models, reports, payment gateways, and developer portal integrations.
- Developer Portal - Self-service platform for developers to publish and manage APIs.
- API Analytics - Service to analyse API-driven programs through monitoring, measuring, and managing APIs.
- Apigee Sense - Enables API security by identifying and alerting administrators to suspicious API behaviours.
- Cloud Endpoints - An NGINX-based proxy to deploy and manage APIs.
- Service Infrastructure - A set of foundational services for building Google Cloud products.

5.2 API

An **application programming interface** (API) is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software. A document or standard that describes how to build such a connection or interface is called an API specification. A computer system that meets this standard is said to implement or expose an API. The term API may refer either to the specification or to the implementation.

In contrast to a user interface, which connects a computer to a person, an application programming interface connects computers or pieces of software to each other. It is not intended to be used directly by a person other than a computer programmer who is incorporating it into software. An API is often made up of different parts which act as tools or services that are available to the programmer. A program or a programmer that uses one of these parts is said to call that portion of the API. The calls that make up the API are also known as subroutines, methods, requests, or endpoints. An API specification defines these calls, meaning that it explains how to use or implement them.

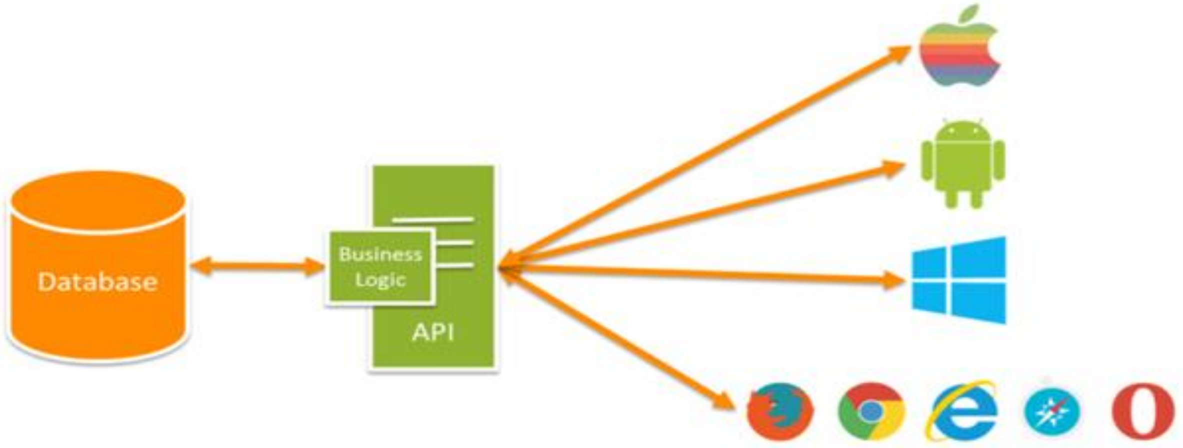


Fig. 5.1 Application Programming Interface (API)

One purpose of APIs is to hide the internal details of how a system works, exposing only those parts a programmer will find useful and keeping them consistent even if the internal details later change. An API may be custom-built for a particular pair of systems, or it may be a shared standard allowing interoperability among many systems.

Web APIs allow communication between computers that are joined by the internet. This is the most common meaning of the term API today. There are also APIs for programming languages, software libraries, computer operating systems, and computer hardware. APIs originated in the 1940s, though the term did not emerge until the 1960s and 70s.

CHAPTER: 6

Cloud Firestore

Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud. Like Firebase Realtime Database, it keeps your data in sync across client apps through realtime listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity. Cloud Firestore also offers seamless integration with other Firebase and Google Cloud products, including Cloud Functions.

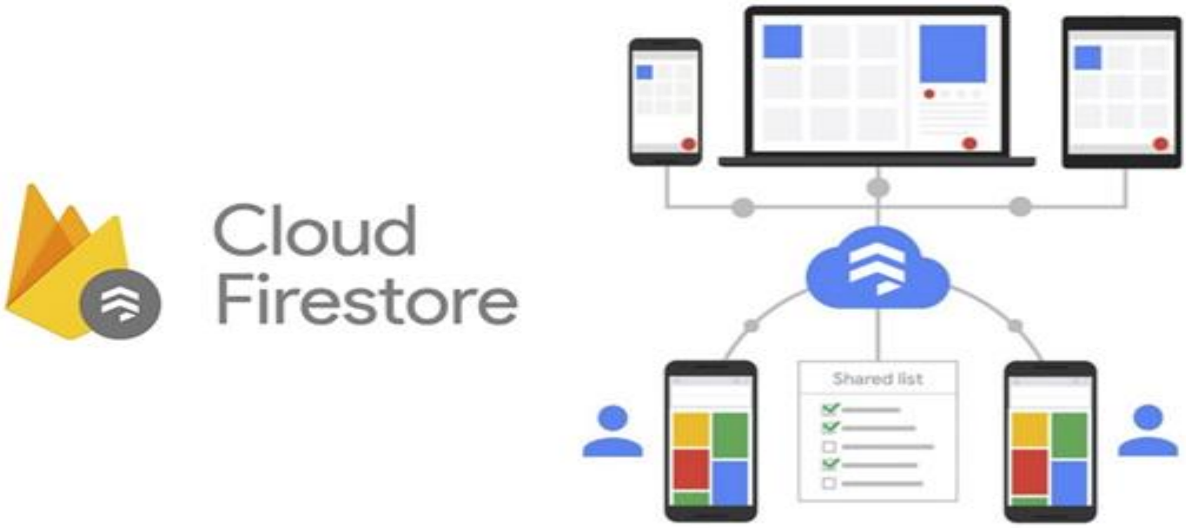


Fig. 6.1 Cloud Firestore

6.1 Key capabilities

Flexibility→

The Cloud Firestore data model supports flexible, hierarchical data structures. Store your data in documents, organized into collections. Documents can contain complex nested objects in addition to subcollections.

Expressive querying→

In Cloud Firestore, you can use queries to retrieve individual, specific documents or to retrieve all the documents in a collection that match your query parameters. Your queries can include multiple, chained filters and combine filtering and sorting. They're also indexed by default, so query performance is proportional to the size of your result set, not your data set.

Realtime updates→

Like Realtime Database, Cloud Firestore uses data synchronization to update data on any connected device. However, it's also designed to make simple, one-time fetch queries efficiently.

Offline support→

Cloud Firestore caches data that your app is actively using, so the app can write, read, listen to, and query data even if the device is offline. When the device comes back online, Cloud Firestore synchronizes any local changes back to Cloud Firestore.

Designed to scale→

Cloud Firestore brings you the best of Google Cloud's powerful infrastructure: automatic multi-region data replication, strong consistency guarantees, atomic batch operations, and real transaction support.

6.2 How does it work?

Cloud Firestore is a cloud-hosted, NoSQL database that your iOS, Android, and web apps can access directly via native SDKs. Cloud Firestore is also available in native Node.js, Java, Python, Unity, C++ and Go SDKs, in addition to REST and RPC APIs.

Following Cloud Firestore's NoSQL data model, you store data in documents that contain fields mapping to values. These documents are stored in collections, which are containers for your documents that you can use to organize your data and build queries. Documents support many different data types, from simple strings and numbers, to complex, nested objects. You can also create subcollections within documents and build hierarchical data structures that scale as your database grows. The Cloud Firestore data model supports whatever data structure works best for your app.

Fig. 6.3

Additionally, querying in Cloud Firestore is expressive, efficient, and flexible. Create shallow queries to retrieve data at the document level without needing to retrieve the entire collection, or any nested subcollections. Add sorting, filtering, and limits to your queries or cursors to paginate your results. To keep data in your apps current, without retrieving your entire database each time an update happens, add realtime listeners. Adding realtime listeners to your app notifies you with a data snapshot whenever the data your client apps are listening to changes, retrieving only the new changes.

Protect access to your data in Cloud Firestore with Firebase Authentication and Cloud Firestore Security Rules for Android, iOS, and JavaScript, or Identity and Access Management (IAM) for server-side languages.

6.3 Implementation path

Integrate the Cloud Firestore SDKs→

Quickly include clients via Gradle, CocoaPods, or a script include.

Secure your data→

Use Cloud Firestore Security Rules or Identity and Access Management

(IAM) to secure your data for mobile/web and server development, respectively.

Add Data→

Create documents and collections in your database.

Get Data→

Create queries or use realtime listeners to retrieve data from the database.

6.4 Cloud Firestore Data model

Cloud Firestore is a NoSQL, document-oriented database. Unlike a SQL database, there are no tables or rows. Instead, you store data in *documents*, which are organized into *collections*.

Each *document* contains a set of key-value pairs. Cloud Firestore is optimized for storing large collections of small documents.

All documents must be stored in collections. Documents can contain *subcollections* and nested objects, both of which can include primitive fields like strings or complex objects like lists.

Collections and documents are created implicitly in Cloud Firestore. Simply assign data to a document within a collection. If either the collection or document does not exist, Cloud Firestore creates it.

6.5 Firebase Features

- Real-time Database − Firebase supports JSON data and all users connected to it receive live updates after every change.
- Authentication – We can use anonymous, password or different social authentications.
- Hosting – The applications can be deployed over secured connection to Firebase servers.

6.6 Firebase Advantages

- It is simple and user friendly. No need for complicated configuration.
- The data is real-time, which means that every change will automatically update connected clients.
- Firebase offers simple control dashboard.
- There are a number of useful services to choose.

6.7 Firebase Limitations

- Firebase free plan is limited to 50 Connections and 100 MB of storage

CHAPTER: 7

Google Sheets

7.1 What is Google Sheets?

Google Sheets is a free, web-based spreadsheet application that is provided by Google within the Google Drive service. The application is also available as a desktop application on ChromeOS, and as a mobile app on Android, Windows, iOS, and BlackBerry. The Google Drive service also hosts other Google products such as Google Docs, Slides, and Forms.

Google Sheets allows users to edit, organize, and analyze different types of information. It allows collaborations, and multiple users can edit and format files in real-time, and any changes made to the spreadsheet can be tracked by a revision history.

7.2 Features of Google Sheets

1. Editing

One of the key features of Google Sheets is that it allows collaborative editing of spreadsheets in real-time. Rather than emailing one document to multiple people, a single document can be opened and edited by multiple users simultaneously. Users can see every change made by other collaborators, and all changes are automatically saved to Google servers.

Google Sheets also includes a sidebar chat feature that allows collaborators to discuss edits in real-time and make recommendations on certain changes. Any changes that the collaborators make can be tracked using the Revision History feature. An editor can review past edits and revert any unwanted changes.

2. Explore

The Explore feature in Google Sheets was first introduced in September 2016, and it uses machine learning to bring additional functionalities. This feature provides a lot of information based on the data added to the spreadsheet, and it can auto-update itself depending on the selected data.

With the Explore feature, users can ask questions, build charts, visualize data, create pivot tables, and format the spreadsheet with different colors. For example, if you are preparing a monthly budget and you've added all the expenses to the spreadsheet, you can use the Explore feature to get the cost of specific expenses such as food, travel, clothing, etc.

On the sidebar, there is a box where you can type the question, and it will return the answer. When you scroll down further in the Explore panel, there is a list of suggested graphs that are representative of the data entered in the spreadsheet, and you can choose between a pivot table, pie chart, or bar chart.

3. Offline editing

Google Sheets supports offline editing, and users can edit the spreadsheet offline either on desktop or mobile apps. On the desktop, users need to use the Chrome browser and install the “Google Docs Offline” Chrome extension to enable offline editing for Google Sheets and other Google applications. When using mobile, users need to use the Google Sheets mobile app for Android and iOS, which support offline editing.

4. Supported file formats

Google Sheets supports multiple spreadsheet file formats and file types. Users can open, edit, save or export spreadsheets and document files into Google Sheets. Some of the formats that can be viewed and converted to Google Sheets include:

.xlsx

.xls

.xlsm

.xlt

.xltx

.xltxm

.ods

.csv

.tsv

5. Integration with other Google products

Google Sheets can be integrated with other Google products such as Google Form, Google Finance, Google Translate, and Google Drawings. For example, if you want to create a poll or questionnaire, you can input the questions in Google Forms, and then import the Google Forms into Google Sheets.

7.3 Google Sheets API

The Google Sheets API is a RESTful interface that lets you read and modify a spreadsheet's data. The most common uses of this API include the following tasks:

- Create spreadsheets
- Read and write spreadsheet cell values
- Update spreadsheet formatting
- Manage Connected Sheets

Below is a list of common terms used in the Sheets API:

Spreadsheet

The primary object in Google Sheets that can contain multiple sheets, each with structured information contained in cells. Every spreadsheet is represented by a Spreadsheet resource and has a unique spreadsheetId value, containing letters, numbers, hyphens, or underscores.

Sheet

A page or tab within a spreadsheet. Each sheet is represented by a Sheet resource and has a unique title and numeric sheetId value.

Cell

An individual field of text or data within a sheet. Cells are arranged in rows and columns, and can be grouped together as a range of cells. Each cell is represented by a CellData resource, but doesn't have a unique ID value. Instead, cells are identified by their row and column coordinates.

A1 notation

A syntax used to define a cell or range of cells with a string that contains the sheet name and starting and ending cell coordinates using column letters and row numbers. This method is most common and useful when referencing an absolute range of cells.

R1C1 notation

A syntax used to define a cell or range of cells with a string that contains the sheet name and starting and ending cell coordinates using row numbers and column numbers. This method is less common, but can be useful when referencing a range of cells relative to a given cell's position.

Named range

A defined cell or range of cells with a custom name to simplify references throughout an application. A named range is represented by a FilterView resource.

Protected range

A defined cell or range of cells that cannot be modified. A protected range is represented by a ProtectedRange resource.

CHAPTER: 8

Image Manipulation and Object detection in Python

8.1 Grayscaling of Images using OpenCV

Grayscaling is the process of converting an image from other color spaces e.g RGB, CMYK, HSV, etc. to shades of gray. It varies between complete black and complete white.

Importance of grayscaling –

- **Dimension reduction:** For e.g. In RGB images there are three color channels and has three dimensions while grayscaled images are single dimensional.
- **Reduces model complexity:** Consider training neural article on RGB images of 10x10x3 pixel.The input layer will have 300 input nodes. On the other hand, the same neural network will need only 100 input node for grayscaled images.
- **For other algorithms to work:** There are many algorithms that are customized to work only on grayscaled images e.g. Canny edge detection function pre-implemented in OpenCV library works on Grayscaled images only.

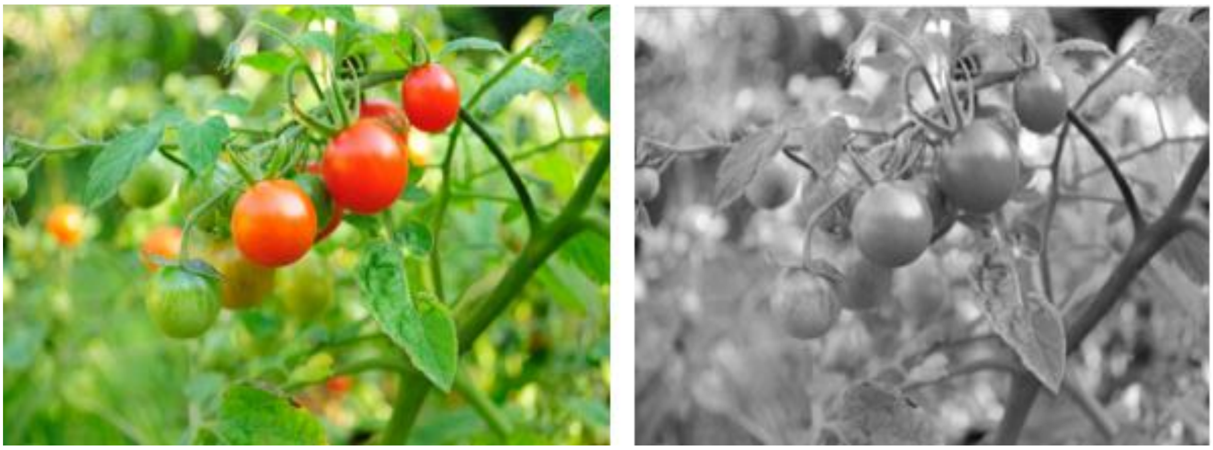


Fig. 8.1 rgb to gray conversion

8.2 Bilateral Filtering

A bilateral filter is used for smoothening images and reducing noise, while **preserving edges**.

However, these convolutions often result in a loss of important edge information, since they blur out everything, irrespective of it being noise or an edge. To counter this problem, the non-linear bilateral filter was introduced.

Bilateral Filter: an Additional Edge Term

The bilateral filter can be formulated as follows:

$$BF[I]_p = \underbrace{\frac{1}{W_p}}_{\text{Normalization Factor}} \sum_{q \epsilon S} \underbrace{G_{\sigma_s}(||p-q||)}_{\text{Space Weight}} \underbrace{G_{\sigma_r}(|I_p - I_q|)}_{\text{Range Weight}} I_q$$

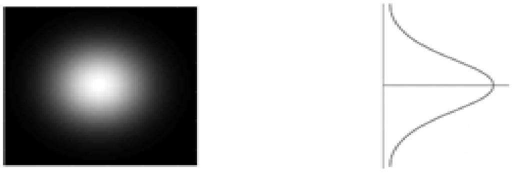


Fig. 8.2 Bilateral Filter Formula

Here, the normalization factor and the range weight are new terms added to the previous equation. σ denotes the spatial extent of the kernel, i.e. the size of the neighbourhood, and σr denotes the minimum amplitude of an edge. It ensures that only those pixels with intensity values similar to that of the central pixel are considered for blurring, while sharp intensity changes are maintained. The smaller the value of σrt, the sharper the edge. As σr tends to infinity, the equation tends to a Gaussian blur.

OpenCV has a function called **bilateralFilter()** with the following arguments:

1. **d:** Diameter of each pixel neighborhood.
2. **sigmaColor:** Value of σ in the color space. The greater the value, the colors farther to each other will start to get mixed.
3. **sigmaColor:** Value of σ in the coordinate space. The greater its value, the further pixels will mix together, given that their colors lie within the sigmaColor range.

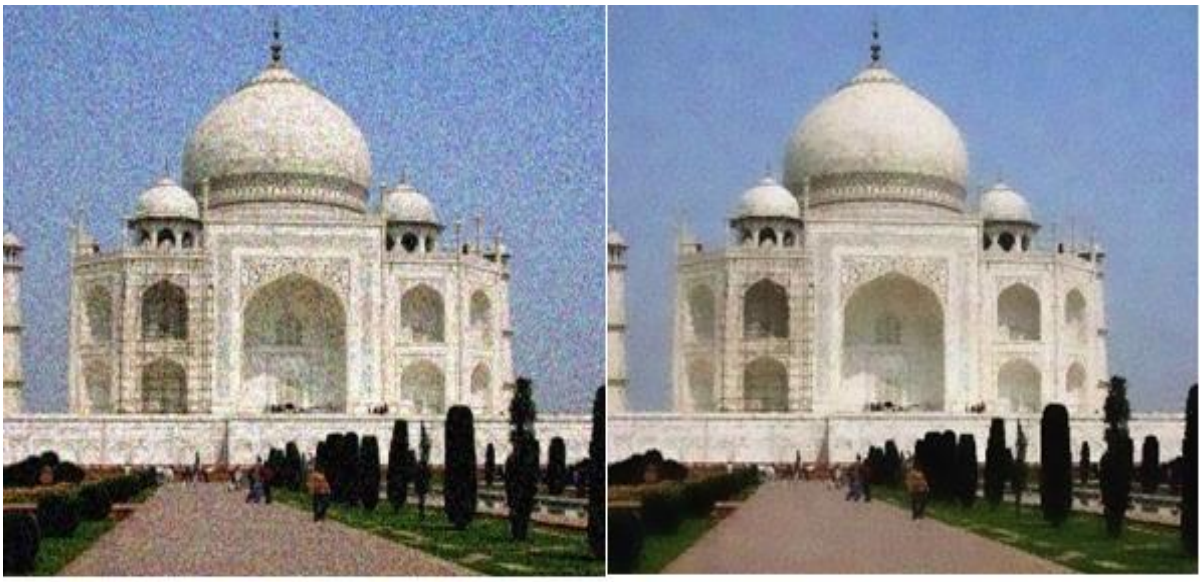


Fig. 8.3 Bilateral Filter

8.3 EDGE DETECTION AND IMAGE GRADIENTS

Edge detection is very important area in computer vision, especially when dealing with contours.

Edges can be defined as boundaries of image, actually they are edges which define object in images they preserve a lot of information about the image. Formally Edges can be defined as sudden changes (discontinuities) in an image and they can encode as much information as pixels.

8.3.1 Canny edge detector

This algorithm was developed by John.F.Canny in 1986.It applies Gaussian blur, finds intensity gradient of image, applies non-maximum suppression (i.e. removes pixels that are not edges) and Hysteresis applies threshold (i.e. if pixel is within the upper and lower threshold, it is considered as an edge).Canny edge detection algorithm uses gradient values as thresholds.

In canny we need to provide two values: threshold1 and threshold2. Any gradient larger than threshold 2 is an edge. Any gradient larger than threshold 1 is considered not to be an edge. Values in between threshold 1 and threshold 2 are either as edge or non-edge on how their intensities are connected, in this case any value below 60 are considered non edges whereas any value above 120 are considered as edges.

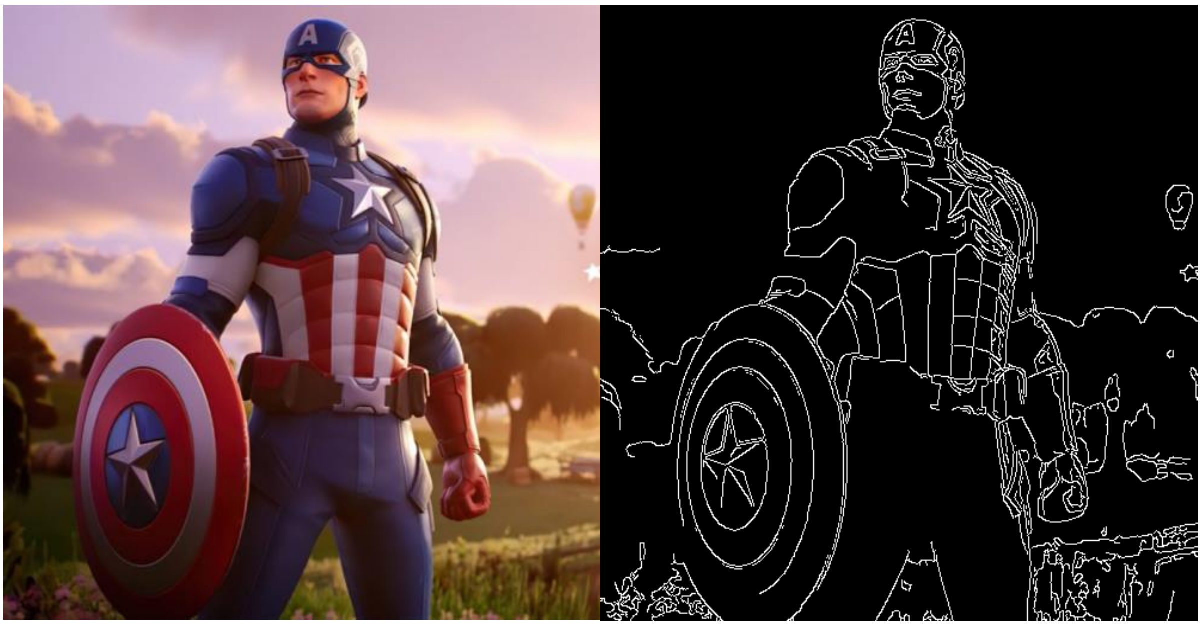


Fig 8.4 Canny Edge Detection

8.4 Find and Draw Contours

Contours are defined as the line joining all the points along the boundary of an image that are having the same intensity. Contours come handy in shape analysis, finding the size of the object of interest, and object detection.

OpenCV has findContour () function that helps in extracting the contours from the image. It works best on binary images, so we should first apply thresholding techniques, Sobel edges, etc.

Contours Approximation Method –

Above, we see that contours are the boundaries of a shape with the same intensity. It stores the (x, y) coordinates of the boundary of a shape. But does it store all the coordinates? That is specified by this contour approximation method. If we pass cv2.CHAIN_APPROX_NONE, all the boundary points are stored. But actually, do we need all the points? For eg, if we have to find the contour of a straight line. We need just two endpoints of that line. This is what cv2.CHAIN_APPROX_SIMPLE does. It removes all redundant points and compresses the contour, thereby saving memory.

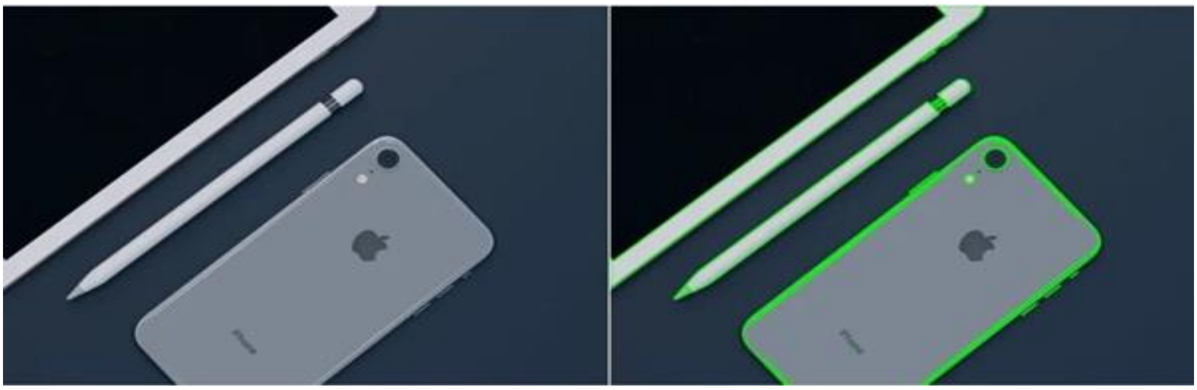


Fig 8.5 Comparative image, input image and output with contours overlaid

8.5 OCR

Optical Character Recognition (OCR) recognizes texts inside images, such as scanned documents and photos, then it converts any kind of images containing written text into machine-readable text data.

Early versions needed to be trained with images of each character and worked on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common, and with support for a variety of digital image file format inputs.

There are many optical character recognition software available. Some of them are Tesseract, OCRopus, Ocular, and SwiftOCR. There are no vast quality differences between them.

8.5.1 Tesseract OCR

Tesseract was originally developed at Hewlett-Packard Laboratories Bristol and at Hewlett-Packard Co, Greeley Colorado between 1985 and 1994, with some more changes made in 1996 to port to Windows, and some C++izing in 1998. In 2005 Tesseract was open sourced by HP. From 2006 until November 2018 it was developed by Google.

Tesseract is an open source text recognition (OCR) Engine, available under the Apache 2.0 license. It can be used directly or by using an API to extract text from images. It supports a wide variety of languages. Tesseract is compatible with many programming languages and frameworks through wrappers. It can be used with the existing layout analysis to recognize text within a large document, or it can be used in conjunction with an external text detector to recognize text from an image of a single text line.

Tesseract 4.00 includes a new neural network subsystem configured as a text line recognizer. The neural network system in Tesseract pre-dates TensorFlow but is compatible with it, as there is a network description language called Variable Graph Specification Language (VGSL) that is also available for TensorFlow.

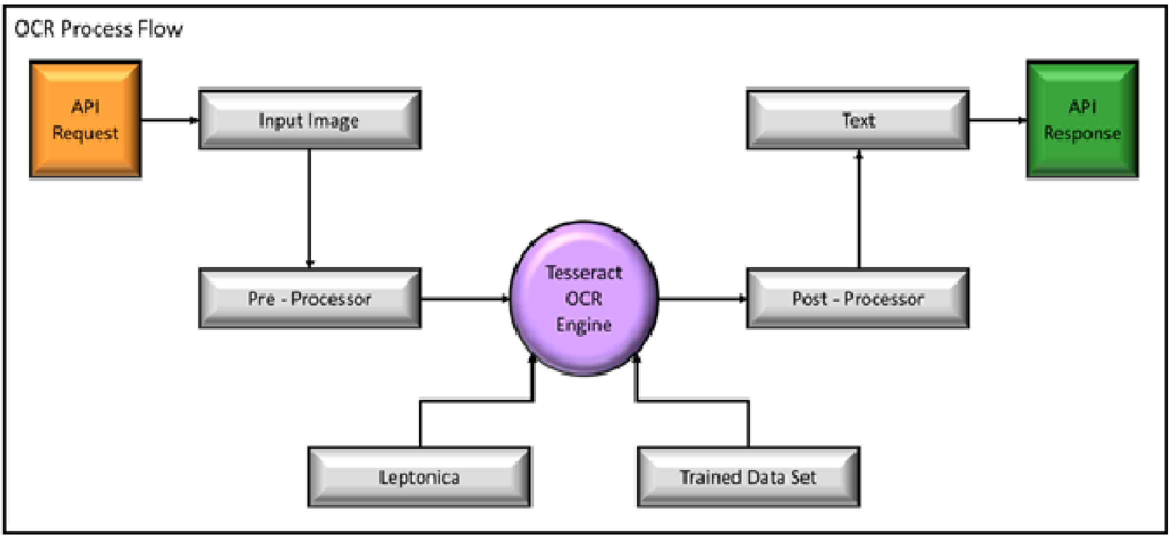


Fig 8.6 Tesseract OCR

CHAPTER: 9

IMPLEMENTATION OF AUTOMATIC VEHICLE NUMBER PLATE DETECTOR

9.1 WORKING OF THE SYSTEM

• IMPORTING THE NECESSARY LIBRARIES

```
import numpy as np
import cv2
import imutils
import pytesseract as tess
tess.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
import time

import firebase_admin
from firebase_admin import credentials
from firebase_admin import firestore

from googleapiclient.discovery import build
from google.oauth2 import service_account
```

- **Image Manipulation**

# Read the image file

```
image = cv2.imread('Car_Image_1.jpg')
```

# Resize the image - change width to 500

```
image = imutils.resize(image, width=500)
```

# Display the original image

```
cv2.imshow("Original Image", image)
```

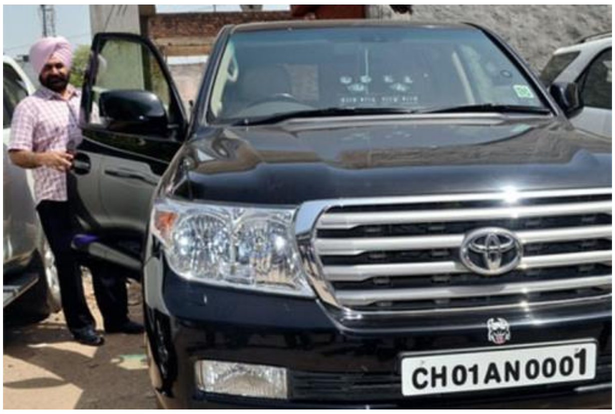


Fig. 9.1 Original Image

# RGB to Gray scale conversion

```
gray = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)
cv2.imshow("1 - Grayscale Conversion", gray)
```

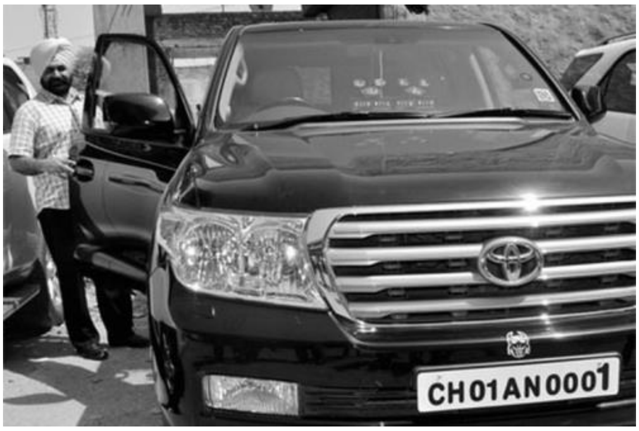


Fig. 9.2 Gray Image

# Noise removal with iterative bilateral filter(removes noise while preserving edges)

```
gray = cv2.bilateralFilter(gray, 11, 17, 17)
cv2.imshow("2 - Bilateral Filter", gray)
```

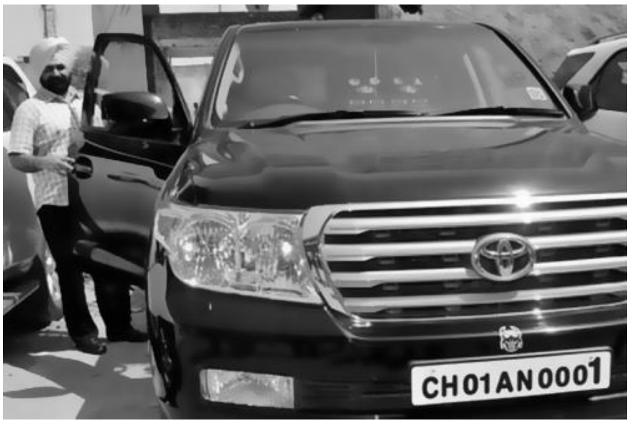


Fig. 9.3 Bilateral Filter

```
# Find Edges of the grayscale image
edged = cv2.Canny(gray, 170, 200)
cv2.imshow("4 - Canny Edges", edged)
```

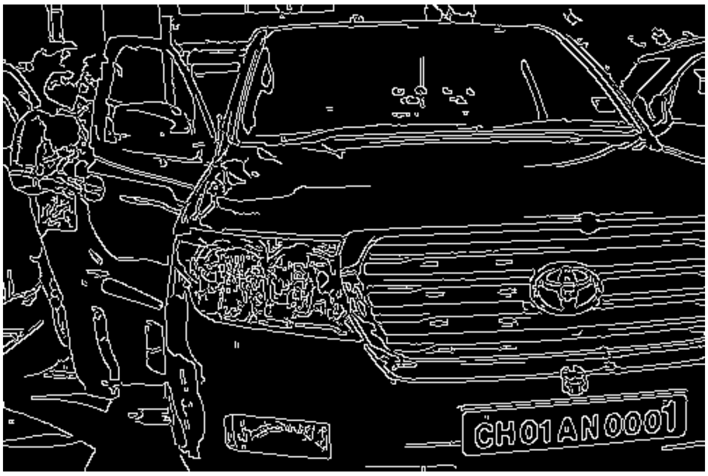


Fig. 9.4 Canny Edge Detection

```
# Find contours based on Edges
(new, cnts, _) = cv2.findContours(edged.copy(), cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)
cnts=sorted(cnts, key = cv2.contourArea, reverse = True)[:30]
#sort contours based on their area keeping minimum required area as '30'
(anything smaller than this will not be considered)

NumberPlateCnt = None

# loop over our contours to find the best possible approximate contour of
number plate
count = 0
for c in cnts:
peri = cv2.arcLength(c, True)
approx = cv2.approxPolyDP(c, 0.02 * peri, True)
if len(approx) == 4:  # Select the contour with 4 corners
NumberPlateCnt = approx   #This is our approx Number Plate Contour
break
```

```
# Drawing the selected contour on the original image
cv2.drawContours(image, [NumberPlateCnt], -1, (0,255,0), 3)
cv2.imshow("Final Image With Number Plate Detected", image)
```

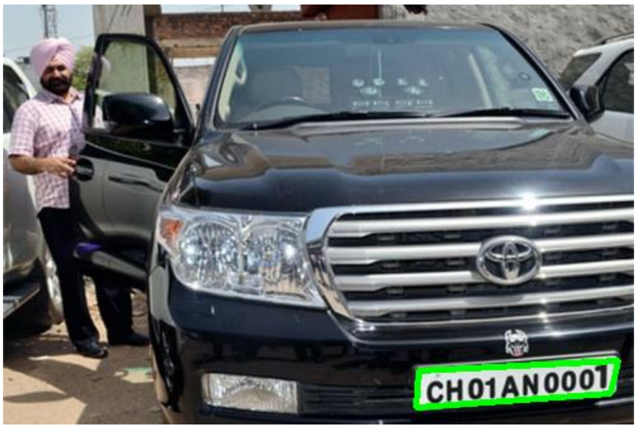


Fig. 9.5 Original Image with detected number plate

```
# Masking the part other than the number plate
mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [NumberPlateCnt], 0, 255, -1)
new_image = cv2.bitwise_and(image, image, mask=mask)
cv2.imshow("Final_image",new_image)
```

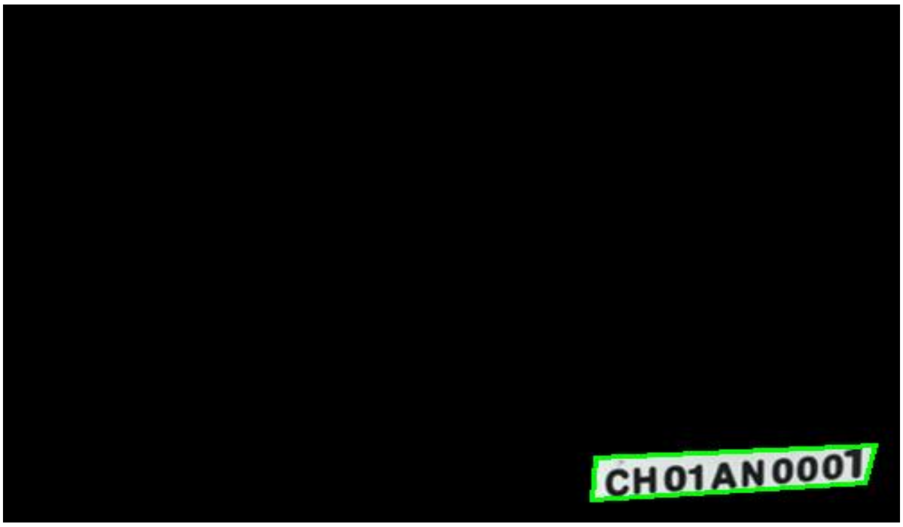


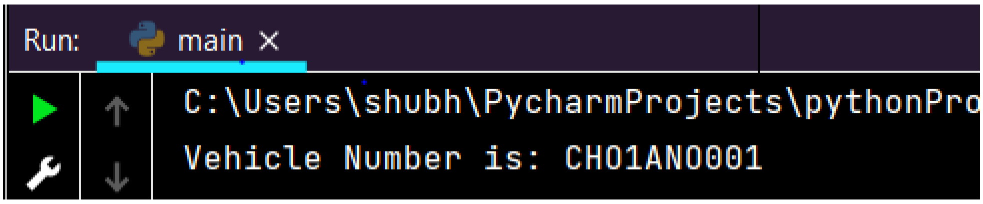
Fig. 9.6 Masking other part except number plate

```
# Configuration for tesseract
config = ('-l eng --oem 1 --psm 3')
```

```
# Run tesseract OCR on image
text = tess.image_to_string(new_image, config=config)

text = text.split('\n')
text = text[0]

print('Vehicle Number is:', text)
```



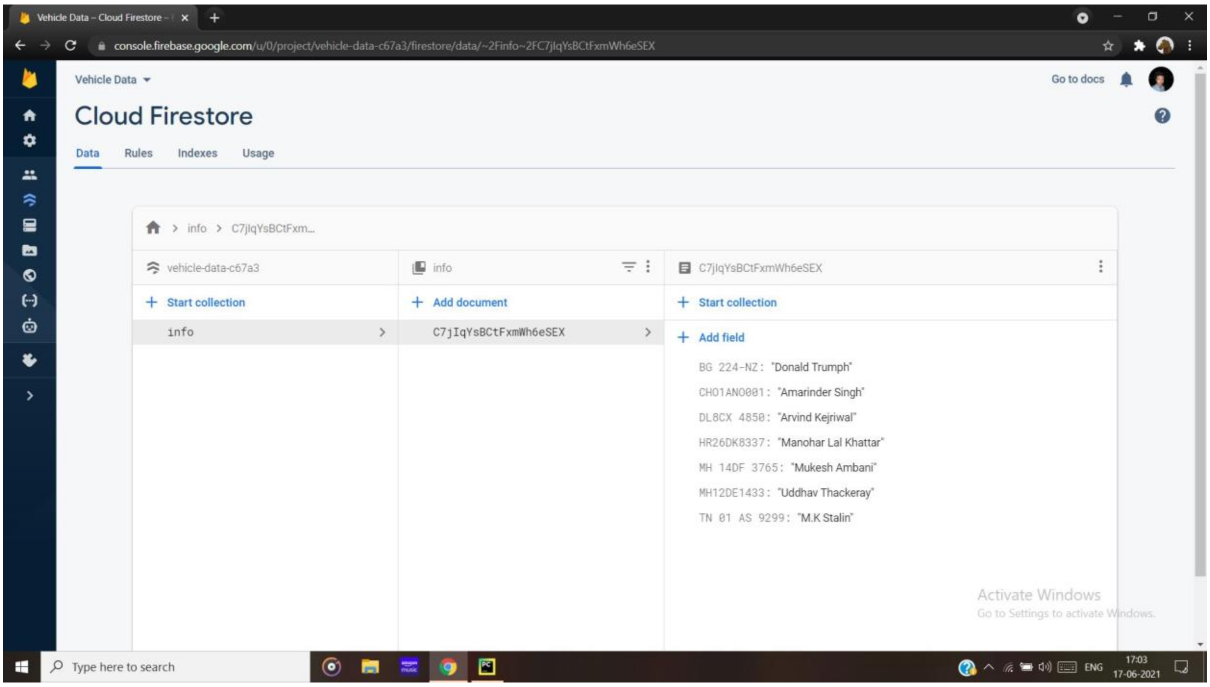
```
C:\Users\shubh\PycharmProjects\pythonPro
Vehicle Number is: CH01AN0001
```

Fig. 9.7 Detected Number Plate Result

- **Working in Firebase Cloud**

```
cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred)

db = firestore.client()
```

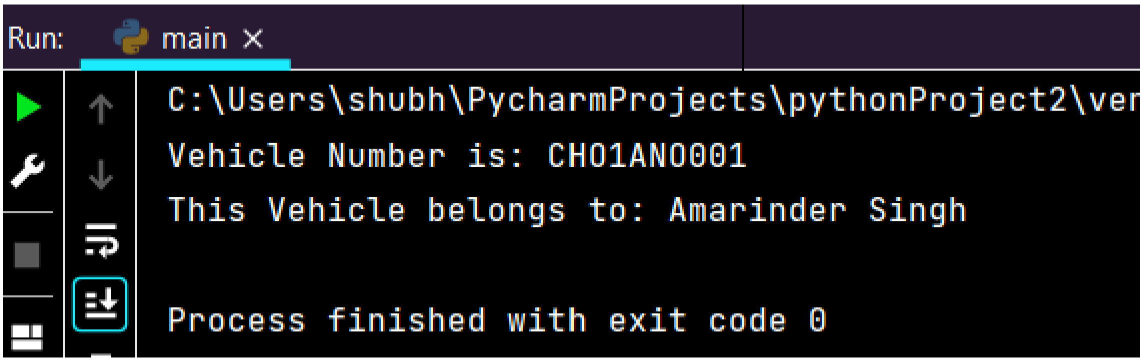


```
BG 224-NZ: "Donald Trumph"
CH01AN0001: "Amarinder Singh"
DL8CX 4850: "Arvind Kejriwal"
HR26DK8337: "Manohar Lal Khattar"
MH 14DF 3765: "Mukesh Ambani"
MH12DE1433: "Uddhav Thackeray"
TN 01 AS 9299: "M.K Stalin"
```

Fig. 9.8 Cloud Firestore Database

result = db.collection('info').document('C7jIqYsBCtFxmWh6eSEX').get()
result = result.to_dict()
result = result[text]

print('This Vehicle belongs to:', result)



```
C:\Users\shubh\PycharmProjects\pythonProject2\ver
Vehicle Number is: CH01AN0001
This Vehicle belongs to: Amarinder Singh

Process finished with exit code 0
```

Fig. 9.9 Name of matched from firestore database cloud

date = time.asctime(time.localtime(time.time()))

- **Google Spreadsheet**

SCOPES = ['https://www.googleapis.com/auth/spreadsheets']

SERVICE_ACCOUNT_FILE = 'keys.json'

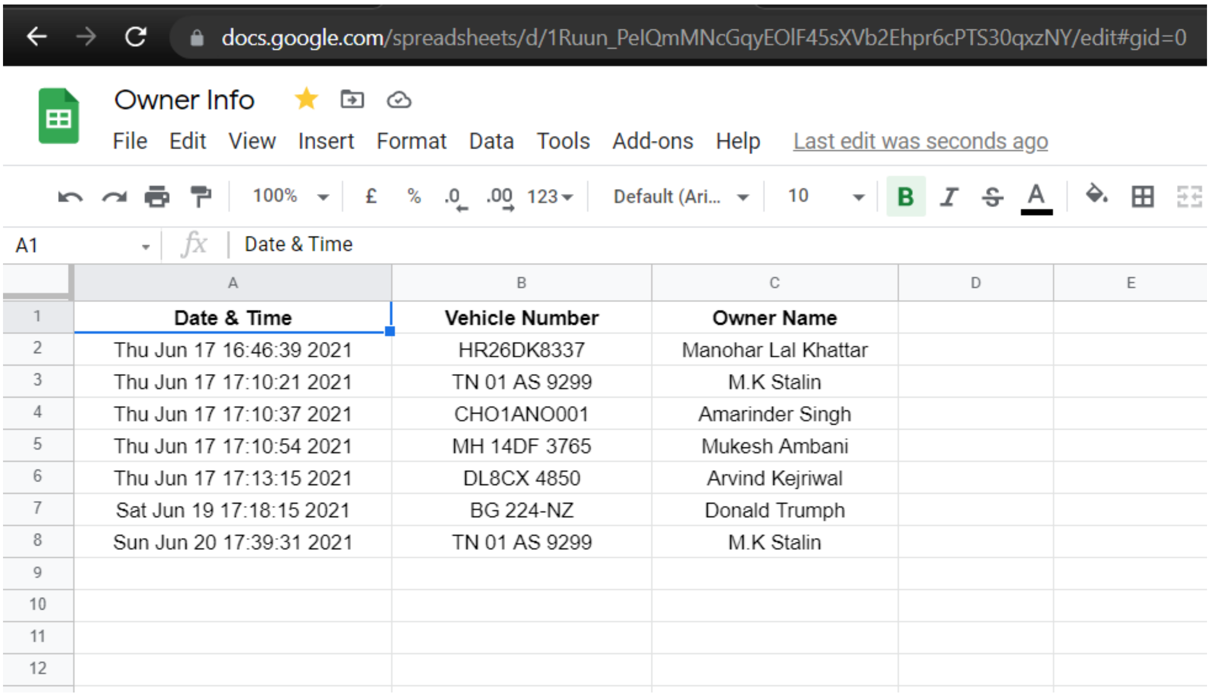
creds = service_account.Credentials.from_service_account_file(
        SERVICE_ACCOUNT_FILE, scopes=SCOPES)

SPREADSHEET_ID =
'1Ruun_PelQmMNcGqyEOlF45sXVb2Ehpr6cPTS30qxzNY'

service = build('sheets', 'v4', credentials=creds)

```
# Call the Sheets API

sheet = service.spreadsheets()
```


docs.google.com/spreadsheets/d/1Ruun_PelQmMNcGqyEOlF45sXVb2Ehpr6cPTS30qxzNY/edit#gid=0
Owner Info
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
100%
£ % .0 .00 123
Default (Ari...
10
B I S A
A1
fx
Date & Time

	A	B	C	D	E
1	Date & Time	Vehicle Number	Owner Name		
2	Thu Jun 17 16:46:39 2021	HR26DK8337	Manohar Lal Khattar		
3	Thu Jun 17 17:10:21 2021	TN 01 AS 9299	M.K Stalin		
4	Thu Jun 17 17:10:37 2021	CHO1ANO001	Amarinder Singh		
5	Thu Jun 17 17:10:54 2021	MH 14DF 3765	Mukesh Ambani		
6	Thu Jun 17 17:13:15 2021	DL8CX 4850	Arvind Kejriwal		
7	Sat Jun 19 17:18:15 2021	BG 224-NZ	Donald Trumph		
8	Sun Jun 20 17:39:31 2021	TN 01 AS 9299	M.K Stalin		
9					
10					
11					
12					

Fig. 9.10 Created Google Spreadsheet

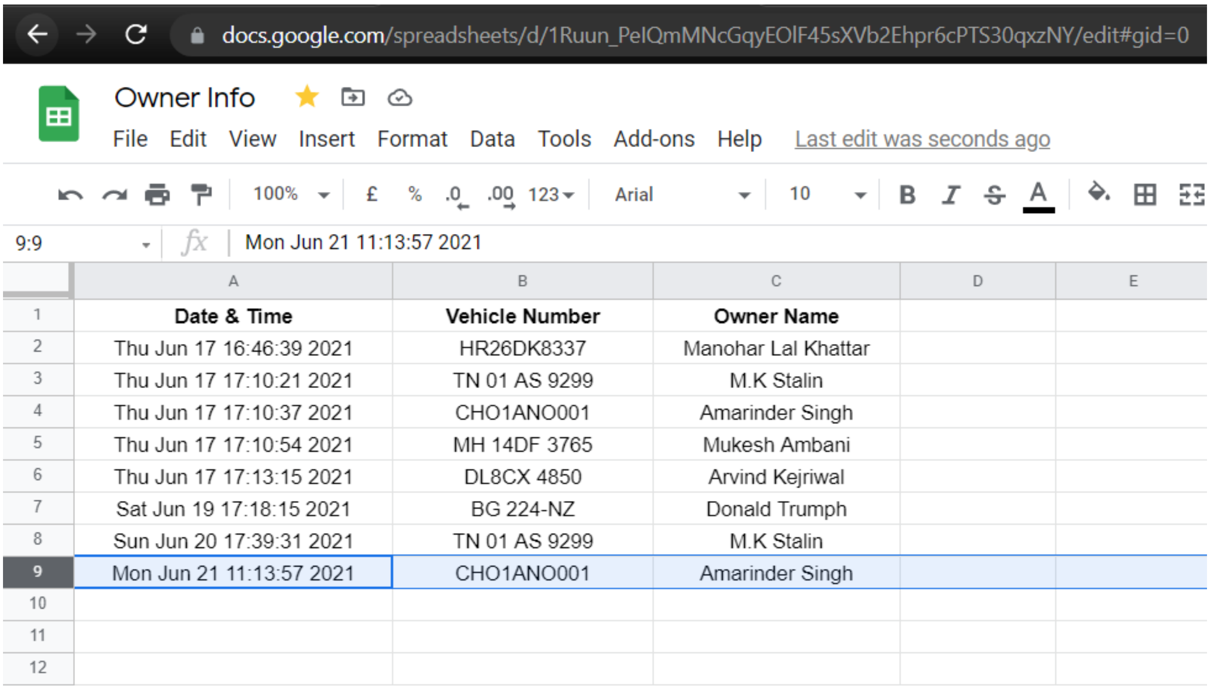
[**This sheet can be viewed online through this link**:- https://docs.google.com/spreadsheets/d/1Ruun_PelQmMNcGqyEOlF45sXVb2Ehpr6cPTS30qxzNY/edit?usp=sharing]

```
i = 1
k = str(i)
loc = "info!A" + k
result1 = sheet.values().get(spreadsheetId=SPREADSHEET_ID,
range=loc).execute()
values = result1.get('values')

while values is not None:
    i = i+1
    k = str(i)
    loc = "info!A" + k
    result11 = sheet.values().get(spreadsheetId=SPREADSHEET_ID,
range=loc).execute()
    values = result11.get('values')

dtr = [[date, text, result]]
```

```
#Update Sheet with recent data
result2 = sheet.values().update(spreadsheetId=SPREADSHEET_ID,
                              range=loc, valueInputOption="USER_ENTERED",
body={"values": dtr}).execute()
```


docs.google.com/spreadsheets/d/1Ruun_PelQmMNcGqyEOlF45sXVb2Ehpr6cPTS30qxzNY/edit#gid=0
Owner Info
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
100% £ % .0 .00 123 Arial 10 B I S A
9:9
fx
Mon Jun 21 11:13:57 2021

	A	B	C	D	E
1	Date & Time	Vehicle Number	Owner Name		
2	Thu Jun 17 16:46:39 2021	HR26DK8337	Manohar Lal Khattar		
3	Thu Jun 17 17:10:21 2021	TN 01 AS 9299	M.K Stalin		
4	Thu Jun 17 17:10:37 2021	CHO1ANO001	Amarinder Singh		
5	Thu Jun 17 17:10:54 2021	MH 14DF 3765	Mukesh Ambani		
6	Thu Jun 17 17:13:15 2021	DL8CX 4850	Arvind Kejriwal		
7	Sat Jun 19 17:18:15 2021	BG 224-NZ	Donald Trumph		
8	Sun Jun 20 17:39:31 2021	TN 01 AS 9299	M.K Stalin		
9	Mon Jun 21 11:13:57 2021	CHO1ANO001	Amarinder Singh		
10					
11					
12					

Fig. 9.11 Result Updated at Google Sheets cloud online

```
cv2.waitKey(0)
```

9.2 PHASES OF THE PROJECT

The entire project work has been divided into 2 phases, PHASE1 (completed in the 7th Semester) and PHASE2 (is completed in 8th Semester)

9.3 PLAN OF THE WORK

In PHASE 1, the following were done-

- Collection of different number plate images from the Internet
- Understanding and being able to work with the MATLAB programming environment.
- Understanding the basics of image processing using MATLAB
- Develop a program using MATLAB image processing toolbox.
- Being able to use the program to detect numbers from static images of no plates

In PHASE 2, the following were planned-

- Become familiar with Python programming environment in Windows Base
- Become familiar with Python programming environment in Linux Base
- Develop a Python program for automatic detection of licence plate.
- Be familiar with Raspberry Pi hardware.
- Implement the Python code in dedicated standalone microcomputer device such as Raspberry Pi.
- Develop and link the program with a database in order to get the info of the owner of the vehicle.

9.4 FINAL STATUS OF THE PROJECT WORK

- ✓ Successful installation of Python3, OpenCV, py-tesseract ocr and other required libraries and software.
- ✓ Successful development of algorithm for basic image processing like edge detection, dilation, contours etc
- ✓ Successfully detect vehicle number plate in Python Environment.
- ✓ Successful setup of Google Cloud Account.
- ✓ Successful creation of database at Google Firebase.
- ✓ Successfully find information of owner of the given Vehicle.
- ✓ Successfully upload the real time data in Google Sheets.

CHAPTER: 10
Conclusion and Future Scope

10.1 CONCLUSION

In this system, an application software is designed for the detection of number plate of vehicles using their number plate. At first plate location is extracted using image processing tools in python then characters are detected using tesseract OCR. Finally the number is matched by cloud database and then real-time data is uploaded in cloud.

10.2 SCOPE OF FUTURE WORK

- The python code can be implemented in Standalone Device to get results from dynamic images.
- Face recognition program can be implemented in Standalone Device.
- Facial data can be stored and matched through could.

REFERNCES

[1] Paper by J.M Mossi, L Sanchis, and A. Albiol, "The Detection of the Parked Vehicles Using the Spatiotemporal Maps," Te IEEE Transactions on the Intelligent Transportation Systems, vol.12, no.4, pp. 1277-1291, 2011.

[2] Convolutional Neural Networks (CNNs): http://cs231n.github.io/apr 2020

[3] Anton SatriaPrabuwono and Ariff Idris, "A Study of Car Park Control System Using Optical Character Recognition," in International Conference on Computer and Electrical Engineering, 2008, pp. 866-870.

[4] Automatic License Plate Detection & Recognition using deep learning https://github.com/GuiltyNeuron/ANPR

[5]WIKIPEDIA https://en.wikipedia.org/wiki/Automatic_number-plate_recognition/apr 2020

[6] Shiqiu Liu, Yiyu He,Zhenyu Xu, YifanZhu, and Han Huang, "The Chinese-style Plate Recognition Based on the Artificial Neural Network and Statistics," The Procedia Engineering, vol. 15, pp. 3556-3561, 2011.

[7]Kumar, Ravi, et al. "Reverse Engineering A Generic Software Exploration Environment Is Made Of Object Oriented Frame Work And Set Of Customizable Tools." International Journal of Advanced Research in Computer Science 2.5 (2011).

[8] Guoliang Fan and Xin Fan, "The Graphical Models for the Joint Segmentation and the Recognition of License Plate Characters," IEEE Signal Processing Letters, vol. 16, no. 1, pp. 10-13, 2009.

[9] You Shyang Chen and ChingHsue Cheng, "A Delphi-based rough sets fusion model for extracting payment rules of vehiclelicense tax in the government sector," Expert Systems with Applications, vol. 37, no. 3, pp. 2161-2174, 2010.

[10] Kumar, Ravi, et al. "System Testability Assessment and testing with Micro architectures." International Journal of Advanced Research in Computer Science 2.6 (2011).

[11] Paper by H. ErdincKocer and K KursatCevik named "Artificial neural networks-based vehicle license plate recognition," at Procedia Computer Science, vol no. 3, pp. 1033-1037, 2011.

[12]Estharakula, Suresh, and JMSV Ravi Kumar. "EBPH-MAC: Emergency Based Priority Hybrid Medium Access Control for Mobility Aware Cooperative WSN's In Indoor Industrial Monitoring." International Journal of Research 5.12 (2018): 1456-1465.

[13] ESTHARAKULA, SURESH, and DR JMSV RAVI KUMAR. "SECRBAC: Secure Data In The Clouds." International Journal of Research 5.15 (2018): 95-106.

[14] Kumar, J. M. S. V., et al. "Analyzing the Modern Tool-Supported UML-Based Static Reverse Engineering." International Journal of Advanced Research in Computer Science 3.4 (2012).

[15] Dara, Suresh, VaishnavaiPasari, NageswaraRaoBanoth, and JMSV Ravi Kumar. "Artificial Bee Colony Algorithm: A Survey and Recent Applications." International Journal of Pure and Applied Mathematics 120, no. 6 (2018): 313-321.

[16] ESTHARAKULA, SURESH, and DR JMSV RAVI KUMAR. "SECRBAC: Secure Data In The Clouds." International Journal of Research 5, no. 15 (2018): 95-106.

[17] RaviKumar JM, Rao KS. International Journal of Sciences & Applied Research.

[18] Parvathi, D. S. L., Leelavathi, N., Ravikumar, J. M. S. V., & Sujatha, B. (2020, July). Emotion Analysis Using Deep Learning. In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 593-598). IEEE.