

Visvesvaraya Technological University



**Belagavi
PROJECT REPORT
8th semester, B.E.**

“GOUSHALA - A LIVESTOCK MANAGEMENT USING AWS IoT”

A Project Report submitted to Visvesvaraya Technological University in partial fulfillment of the requirement for the award of degree of Bachelor of Engineering in Electronics and Telecommunication Engineering

Submitted by,

Anjan D S	4JN19ET004
Harsha J Nazre	4JN19ET014
R R Shriya	4JN19ET026
Smriti K Vantkar	4JN19ET028

**Under the guidance of:
Mr. Benak Patel M P, B.E., M.E.,
Assistant Professor
Dept. of ETE**



Department of Electronics and Telecommunication Engineering

**Jawaharlal Nehru New College of Engineering
SHIVAMOGGA – 577204
2022-23**

National Educational Society®
Jawaharlal Nehru New College of Engineering
SHIVAMOGGA – 577204



Department of Electronics and Telecommunication Engineering
CERTIFICATE

This is to certify that the project work entitled

“GOUSHALA- A LIVESTOCK MANAGEMENT USING AWS IoT”
Is a bonafide work carried out jointly by

Anjan D S	4JN19ET004
Harsha J Nazre	4JN19ET014
R R Shriya	4JN19ET026
Smriti K Vantkar	4JN19ET028

These students of 8th semester B.E., Electronics and Telecommunication Engineering under our supervision and guidance towards the partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Electronics and Telecommunication Engineering as per the university regulations during the year 2022-23.

.....
Signature of the Guide

Mr. Benak Patel M P
Asst. Prof., E.T.E.

.....
Signature of the HOD

Dr. Surendra S.
HOD, E.T.E.

.....
Signature of Principal

Dr. K. Nagendra Prasad
Principal, JNNCE

Name of Examiners:

Signature of Examiners with date:

1.

.....

2.

.....

Abstract

Livestock management is a crucial component of the agricultural business, and using technology to ensure optimal results is becoming increasingly important. Using linked devices and sensors, Amazon IoT delivers a scalable and secure platform for controlling and monitoring livestock. This abstract focuses on some of the most important aspects and benefits of utilizing Amazon IoT for livestock management, such as real-time data collecting, analytics, and remote monitoring. Farmers and ranchers may use Amazon IoT to acquire insights about animal health, behaviour, and general performance, leading to better decision-making and improved results for their livestock operations.

Using the Raspberry Pi, BLE Beacons, and Amazon IoT core, a solution has been given to this problem. This BLE beacon is used to track the animals. This approach is mostly dependent on radio signal technologies like WIFI and RFID. The RSSI (Received Signal Strength Indication) approach is mostly used in BLE for livestock tracking.

By allowing remote monitoring, data-driven decision-making, and automation, utilizing AWS IoT services in livestock management systems has the potential to revolutionize the sector. Livestock managers may maximize resource use, increase animal welfare, and eventually improve the overall productivity and sustainability of livestock-based sectors by using real-time information and sophisticated analytic.

ACKNOWLEDGEMENT

Firstly, we would like to express our sincere gratitude to the almighty for his solemn presence throughout the project work.

We express our sincere gratitude to **Dr. Surendra S., Professor**, Head of the Electronics and Telecommunication Engineering Department for providing us with adequate facilities, ways and means by which we are able to complete this project work.

We are deeply indebted to our project coordinators **Mrs. Aparna**, Assistant Professor, Department of Electronics and Telecommunication Engineering and **Mr. Benak Patel**, Asst. Professor, ETE department for providing us with valuable advice and guidance during the course of the study. Without their wise counsel and able guidance, it would have been impossible to complete our project work in this manner.

We would like to express a deep sense of gratitude and thanks profusely to our project guide **Mr. Benak Patel**, Asst. Professor, Department of Electronics and Telecommunication engineering for his proper guidance and valuable suggestions. Without his wise counsel and able guidance, it would have been impossible to complete the project work in this manner.

We would also like to express our special thanks to the Dean Academics **Dr. Manjunatha P** for providing an opportunity to carry out this project work.

We would also like to express our special thanks to the Principal **Dr. K. Nagendra Prasad** for providing an opportunity to carry out this project work.

We would like to extend our heartfelt gratitude to the teaching and non-teaching staff of **Department of Electronics and Telecommunication Engineering** for their constructive support and co-operation at each and every juncture of the project work.

Finally, we would also like to express our gratitude to **J N N College of Engineering, Shivamogga** for providing us with all the required facilities without which the project work would not have been possible.

Project associates,

ANJAN D S	4JN19ET004
HARSHA J NAZRE	4JN19ET014
R R SHRIYA	4JN19ET026
SMRITI K VANTKAR	4JN19ET028

Table of contents

Chapter No.	Title	Page No.
	ABSTRACT	i
	ACKNOWLEDGEMENT	ii
	TABLE OF CONTENTS	iii
	LIST OF FIGURES	iv
	LIST OF ABBREVIATIONS	v
1	PREAMBLE	
1.1	Introduction	1
1.2	Literature survey	2-4
1.3	Problem statement	4
1.4	Objectives	5
1.5	Methodology	6
1.6	Scope of the project	6
1.7	Organization of the Report	6
2	THEORETICAL BACKGROUND	
2.1	Management of Livestock	7
2.2	Amazon Web Services using IoT	8

2.3	EMQX Broker	9
3	DESIGN AND IMPLEMENTATION	
3.1	Introduction	10
3.1.1	System Design	11
3.2	Block Diagram	12
3.3	Working Principle	13-15
3.4	System Flowchart	16
3.5	Summary	17
4	SOFTWARE IMPLEMENTATION	
4.1	AWS IoT Core	18
4.2	EMQX	19
4.3	Paho Javascript	20
4.4	MQTT Protocol	20
5	RESULTS AND DISCUSSION	
5.1	Results	21-23
5.2	Societal Benefits	24
6	CONCLUSION AND FUTURE SCOPE	
6.1	Advantages	25
6.2	Limitations	26
6.3	Applications	27
6.4	Conclusion	28
6.5	Future Scope	28
APPENDIX		

List of Figures

- 2.2.1: AWS using IoT
- 2.2.2: EMQX Model
- 3.1.1 AWS IoT core services
- 3.2.1: Block Diagram of Livestock Management
- 3.1.1: AWS IoT core setup
- 3.3.1: Message sent from Raspberry Pi to AWS IoT Core via MQTT Protocol
- 3.4.1: System Flowchart
- 4.1.1: AWS IoT Core
- 4.2.1: EMQX
- 4.4.1: MQTT Protocol
- 4.5.1: Balena.io
- 4.5.2: Cloud IoT Provisioning with AWS
- 5.1.1: Message published on Raspberry Pi
- 5.1.2: Message received by the subscriber AWS management console
- 5.1.3: BLE Beacon connected to Raspberry Pi
- 5.1.4: EMQX MQTT Broker-Client message Pub/Sub
- 5.1.5: BLE Beacon Distance Measuring
- 5.1.6: Histogram representing RSSI values vs distance for near objects
- 5.1.7: Graph representing distance of near objects
- 5.1.8: Graph representing distance of far objects
- 5.1.9: Histogram representing RSSI values vs distance for near objects
- 5.1.10: Table Representation
- 5.1.11: BLE Interface

List of Abbreviations

1. BLE Beacon: Bluetooth Low Energy Beacon
2. AWS: Amazon Web Services
3. WIFI: Wireless Fidelity
4. RFID: Radio Frequency Identification
5. ZigBee: Zonal Intercommunication Global – Standard
6. RSSI: Received Signal Strength Indication
7. MQTT: Message Queuing Telemetry Transport
8. IoT: Internet of Things
9. TCP/IP: Transmission Control Protocol/Internet Protocol
10. HTTPS: Hyper Text Transfer Protocol Secure
11. WSS: Web Socket Secure
12. LoRaWAN: Low Range Wide Area Networking
13. UUID: Universal Unique Identifiers
14. SCM: Supply Chain Management
15. SoC: System on Chip
16. GPU: Graphics Processing Unit
17. ARM: Advanced RISC Machine
18. HDMI: High Definition Multimedia Interface
19. GPIO: General Purpose Inputs and outputs
20. CSI: Camera Interface
21. DIS: Display Interface
22. LNS: LoRaWAN Network Server
23. Amazon SQS: Amazon Simple Queue Service
24. Amazon SNS: Amazon Simple Notification Service
25. Amazon S3: Amazon Simple Storage Service
26. M2M: Machine to Machine
27. IAC: Infrastructure as Code
28. MAC: Media Access Control

CHAPTER 1

Preamble

1.1 Introduction

The care of livestock is a crucial aspect of animal husbandry. Numerous instances of livestock disappearing and then being discovered dead exist. This is primarily due to handlers' ineffective control of the livestock. We have developed a method to address this issue using the RaspberryPi, BLE Beacons, and AWS IoT core. The cattle are tracked using this BLE beacon. Mostly, this solution is designed based on radio signal technologies such as WIFI, and RFID. Recently, BLE has become the centre of interest for positioning methods due to its low power consumption, cost effective, and deployment flexibility.

The RSSI technology is mostly used by BLE for livestock tracking. It measures the RSSI level of signals sent by fixed BLE beacons in order to calculate the distance between each of them and a mobile device that has to be found. Several RSSI-based positioning methods have been presented in the literature. Using these procedures, an accurate distance is determined in a laboratory setting utilising a precise signal measurement. In fact, this technique uses a radio propagation model for linking RSSI measurement to the position of the mobile device. It requires precise description of the radio propagation environment which is difficult to predict. In addition, BLE signal propagation is highly influenced by interference and obstacle on the environment. In this project, we present our livestock tracking solution which combines and analyses the most accurate and relevant techniques in the literature based on RSSI measurement. MQTT is a data transmission protocol that is widely used in Internet of Things designs that are based on TCP/IP. Because of its minimal bandwidth consumption and suitability for adoption by devices with limited resources, it is preferable to other protocols such as HTTP.

It employs a publish-subscribe paradigm (also known as pub-sub) with three entities: publisher, subscriber, and broker. The former is capable of sending messages on a certain topic. Those communications are received by the broker, who passes them to clients registered to the same subject, known as subscribers. The idea of using the MQTT protocol to control an BLE localization system, is quite innovative. AWS IoT Core lets you connect billions of IoT devices and route trillions of messages to AWS services without managing infrastructure. Connect, manage, and scale device fleets easily and reliably without provisioning or managing servers. One can choose preferred communication protocol, including MQTT, HTTPS, MQTT over WSS, and LoRaWAN.

1.2 Literature Survey

1.1.1 Exploring the Suitability of BLE Beacons to Track Poacher Vehicles in Harsh Jungle Terrains, Karan Juj, Charith Perera, Cardiff University, United Kingdom.

Our overall aim is focused on exploring whether we could use Bluetooth Low Energy (BLE) technology to track poacher vehicles in remote and rural areas such as Sabah, in Malaysia, especially deep inside the jungle terrain with little or no communication technologies exists. Tracking technologies are currently limited to relying on satellites or cellular-towers for environments that do not permit access to these signals, very few viable alternatives exist. This paper explores the use of BLE as a method to track vehicles. It works by mounting Bluetooth beacons beside a road and placing a receiver concealed somewhere inside the vehicle. As the vehicle drives past the beacon, the receiver and beacon are momentarily in range, the receiver then stores a unique ID from the beacon, and when the vehicle is then in an area with GSM signal, an SMS is sent containing the unique IDs of the beacons that have been detected. This project is prototyped and tested in collaboration with the Danau Girang Field Centre in Sabah, Malaysia. The results offer insights for how effective BLE beacons are in a tracking situation for where the beacon and receiver are in range for a short period of time as well as how different obstructions will affect the range and strength of the signal. It is important to note that our objective is not to catch the poacher, instead to understand how they move around within jungle terrain, as we can use such information to develop a comprehensive plan against poaching activities.

1.1.2 Propagation Measurements and Modelling for Monitoring and Tracking in Animal Husbandry Applications, J. M. R. de S. Neto, J. J. C. Silva, T. C. M. Cavalcanti, D. P. Rodrigues and J. S. da Rocha Neto.

Transmission loss measurements using IEEE 802.15.4 technology are reported in an outdoor environment for application to monitoring and tracking of small ruminants. The simplest possible propagation model is shown to reflect the general features of the measured propagation data. Its absolute accuracy, however, is probably inadequate for use in a location algorithm based on model inversion without optimization of its parameters. Model calibration to reflect inter-site variation of the propagation environment is suggested as a possible way of realizing a location system with useful accuracy and adequate portability.

1.1.3 BLE Beacon Based Patient Tracking in Smart Care Facilities, Brad Kennedy, Graham W. Taylor, Petros Spachos.

Patient tracking is an important component toward creating smart cities. In this demo we use Bluetooth Low Energy beacons and single board computers to track patients in the emerging field of smart care facilities. Our model utilizes a fixed scanner moving transmitter method for wireless tracking of patients through the facility. The data collected by all scanners is stored within a central database that is designed to be efficiently queried. We show how inexpensive components in conjunction with free open source software can be used to implement a patient tracking system. We focus on the pipeline between acquisition and display of the location data. Additionally, we also discuss the manipulation of the data required for usability, and optional filtering operations that improve accuracy.

1.1.4 Indoor Positioning System for IoT Device based on BLE Technology and MQTT Protocol, Kais Mekki1, Eddy Bajic, Fernand Meyer.

The arrival of Bluetooth Low Energy (BLE) creates opportunities for great innovations. One possible application is indoor localization. This paper presents a system that can track mobile device and help finding their location within a building perimeter. With the help of BLE beacons that can be deployed in different locations, the position of a mobile device can be estimated using RSSI techniques and trilateration methods. The overall system is controlled using MQTT protocol. Along this paper, we describe our system, the techniques we use, and the experiments we conducted along with the results.

1.1.5 A Prototype Air Flow Control System for Home Automation using MQTT over WebSocket in AWS IoT Core, Nadia Imtiaz Jaya

Amazon Web Services have recently developed their IoT platform, AWS IoT Core, which integrates all the necessary functions required for developing an IoT system. Our objective in this paper is to explore some of those functions and their integration in our designed project. In light of this, we develop and present a prototype air flow control system for home automation using AWS IoT Core and MQTT protocol over Websocket server. The advantages obtained in the choice of modules, communication protocols and services are explained in depth. System architecture, its implementation and performance analysis are also presented. Keywords—IoT; AWS IoT Core; MQTT; Websocket server; air flow control; smart home.

1.3 Problem Statement

Our initiative is to solve the issue of inefficient livestock control, which results in cases of cattle going missing and then being found dead. This is mainly because huge livestock herds are challenging to track and manage using conventional techniques. The goal of our project is to create a tracking and management system for herds utilizing the Raspberry Pi, BLE Beacons, and Amazon IoT Core, which can offer a more precise and effective technique for controlling cattle. The management and care of cattle may benefit from this method.

1.4 Objectives

The Objectives and key points of this solution for livestock tracking are:

1. It precisely determines the position of the mobile device by utilizing RSSI and BLE beacon technologies.
2. To increase the precision of the location estimations, it integrates and examines the most precise and pertinent methodologies in the literature.
3. It enables simple administration of sizable herds by utilizing the MQTT protocol for remote control of the livestock tracking system.
4. It makes use of Amazon IoT Core for quick and dependable connectivity, fleet management, and scaling.
5. It offers end-to-end encryption, mutual authentication, and secure device connections for data transmission.

Overall, this solution has the potential to improve the care and management of livestock by providing a more accurate and efficient method for tracking and managing herds.

1.5 Methodology

The methodology for our project will involve the following steps:

1. List the specifications for the livestock tracking system, such as the kind of cattle to be monitored, the environment in which it will be utilized, and the level of precision that should be wanted in the location estimations.
2. Choose and set up the solution's hardware and software components, such as the Raspberry Pi, BLE Beacons, and Amazon IoT Core.
3. To increase the precision of location estimates, create algorithms for estimating the position of mobile devices using RSSI and BLE beacon technologies, and integrate and analyse the most precise and pertinent methodologies in the literature.
4. Develop the MQTT protocol for remote control of the cattle tracking system, and use Amazon IoT Core for simple and dependable device connection, administration, and scalability.
5. In a controlled setting, evaluate the solution's accuracy and dependability and make any required improvements to improve its performance.
6. Install the system in a real-world setting and follow its performance to determine its efficacy in tracking and controlling animals.
7. Based on the findings of the real-world testing, make any necessary adjustments to the solution.

Overall, this methodology will allow us to develop and implement a reliable and effective solution for tracking and managing livestock.

1.6 Scope of the Project

AWS IoT may be used to monitor individual animals' health and well-being, including vital signs, behaviour, and location. This data may be used to diagnose disease early on, track the spread of infectious diseases, and optimise treatment programmes.

1.7 Organization of the report

Chapter 2: Establishes the theoretical background, exploring relevant concepts, principles, and theories as the foundation of the study.

Chapter 3: Focuses on system design, outlining the methodology used, identifying requirements, specifying system components, and justifying design decisions.

Chapter 4: Details the software implementation phase, covering coding, testing, integration, and addressing challenges encountered during development.

Chapter 5: Presents results and outcomes, analyzing system performance, effectiveness, and impact, including data presentation, statistical analysis, and discussing findings in relation to research objectives. It also acknowledges limitations and suggests future research areas.

CHAPTER 2

Theoretical Background

2.1 Management of Livestock

Livestock management is the practice of caring for and raising domesticated animals, such as cows, pigs, sheep, and chickens, in a way that maximizes their health, welfare, and productivity. It is a critical component of agriculture and food production, providing a vital source of protein and other nutrients to human populations around the world. Animal welfare is the term used to describe the mental and emotional well-being of animals under human care. It is a crucial factor to take into account when managing livestock since it impacts not only the well-being and productivity of the animals but also the quality and safety of the food they produce. For the health and production of animals, proper nutrition is crucial. It entails feeding animals a healthy diet that satisfies their nutritional requirements and promotes growth and development.

Genetic selection and breeding may significantly contribute to enhancing the production and well-being of animals. Breeders may create animals that are more suited to their environment and more effective at turning feed into meat, milk, or eggs by choosing animals with desired qualities including disease resistance, fertility, and growth rate.

For the prevention and control of illnesses in cattle, effective health management procedures are essential. This can involve practices like immunization, bio security, frequent animal monitoring, and the care of ill animals. As it has an impact on the size and productivity of herds and flocks, reproduction is a crucial component of livestock management. The goal of efficient reproductive management is to increase an animal's capacity for procreation while lowering its risk of contracting sickness and other health problems. Understanding and using these ideas allows livestock managers to enhance their animals' health and production while also ensuring that they are handled humanely and sustainably. Efficient livestock management strategies are crucial for addressing the world's rising food demand in an era of rapidly expanding human population.

2.2 Amazon Web Services Using IoT

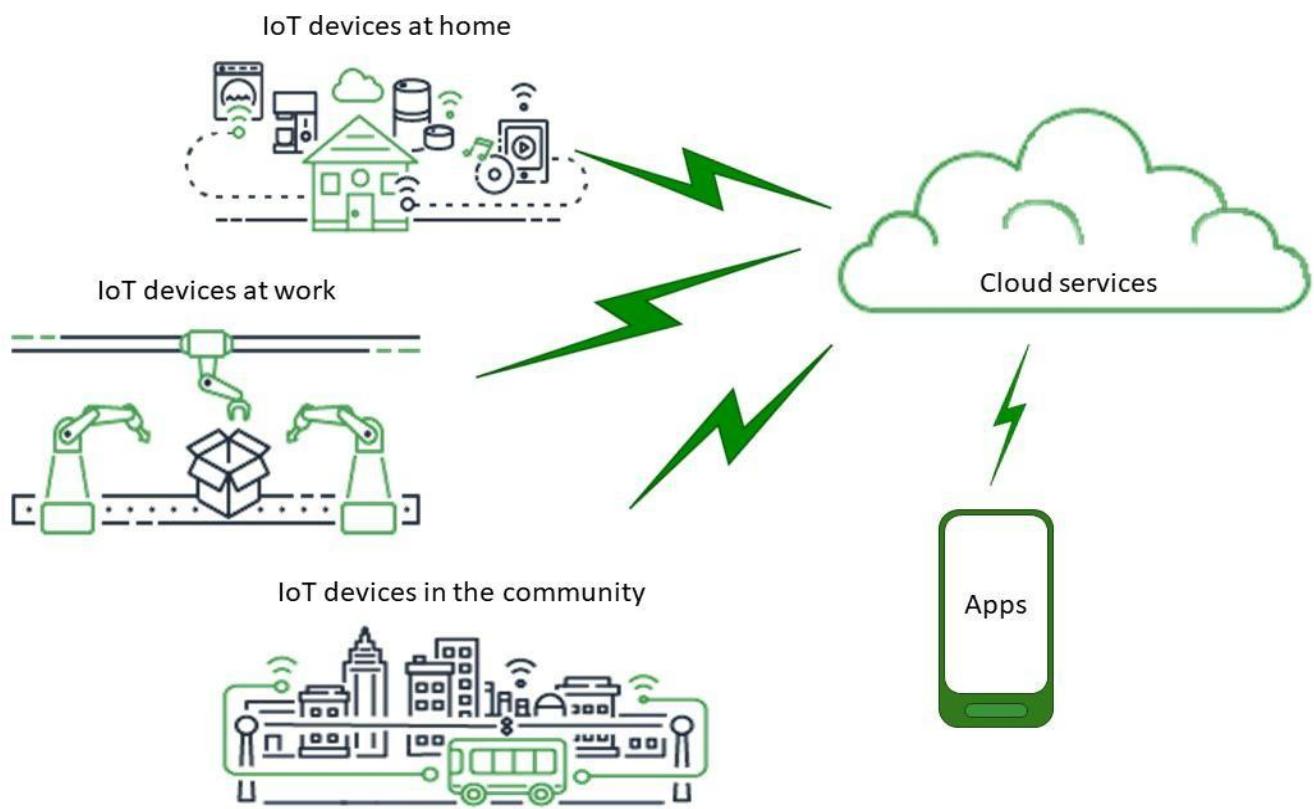


Fig 2.2.1: AWS using IoT

Amazon Web Services (AWS) is a cloud computing platform that offers a variety of services, including Internet of Things (IoT) services. Amazon IoT is a platform that allows devices to connect to the cloud, gather and analyse data, and create applications and services based on that data. The network of physical objects, including machines, automobiles, household appliances, and other objects, is referred to as the "Internet of Things." These objects are equipped with sensors, software, and other technologies that allow them to connect and exchange data across a network. IoT devices may be utilised for a variety of purposes, including environmental monitoring, industrial automation, and home automation. For the development of complete Bluetooth Low Energy (BLE) beacon applications, Amazon offers services. Small, battery-operated BLE beacons send data to neighbouring devices using Bluetooth technology. Many uses for them exist, including proximity marketing, asset monitoring, and interior navigation.

2.3 EMQX Broker

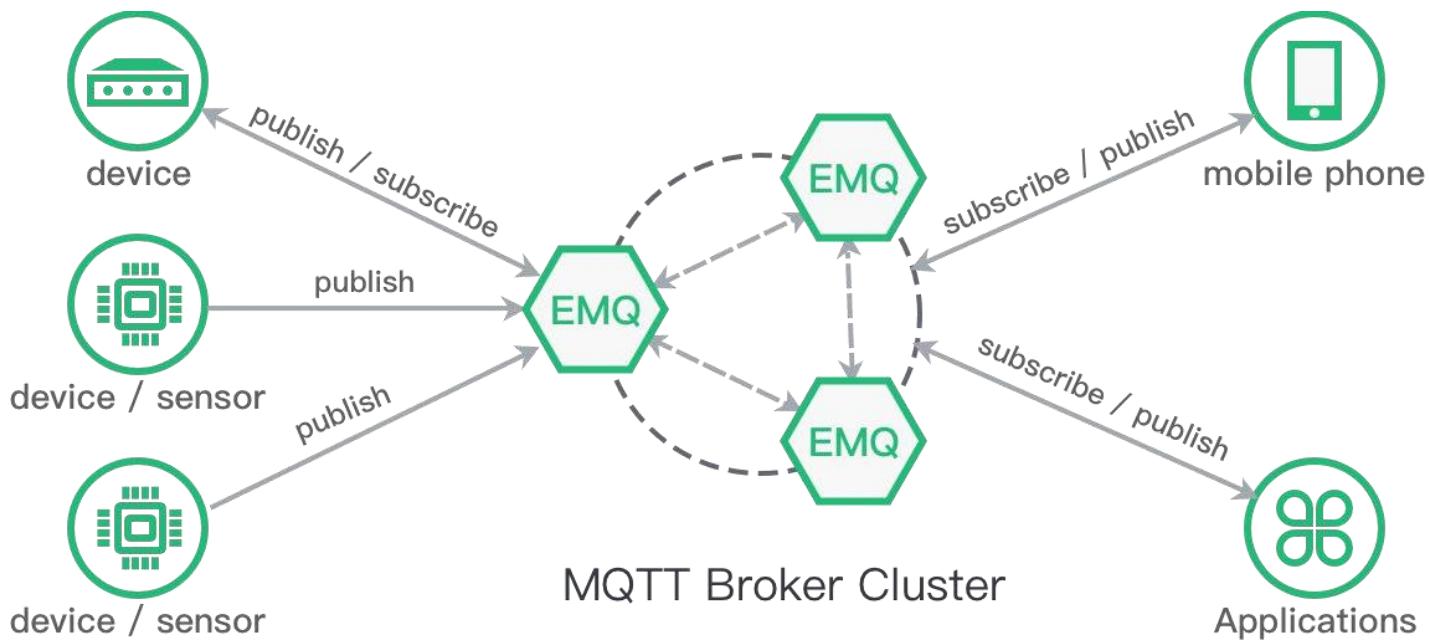


Fig 2.2.2: EMQX Model

EMQ X is an open-source, highly scalable, distributed MQTT message broker for IoT, M2M, and mobile applications. In IoT applications where devices have constrained processing power and battery life, MQTT is a messaging protocol that is created to be lightweight and effective. In order to enable millions of concurrent connections and message transfers per second, EMQ X is built to be very scalable. It may be set up on a solitary system or over a network. EMQ X allows automated fail over and clustering, making sure that MQTT clients may connect to a broker even if one or more nodes fail. EMQ X supports widely used security technologies including SSL/TLS encryption, as well as procedures for authentication and authorization. EMQ X allows bridging between several MQTT brokers as well as MQTT 3.1, MQTT 3.1.1, and MQTT 5.0.

Many IoT applications, including smart cities, industrial automation, and home automation, can make use of EMQ X. It offers an effective, scalable, and secure messaging infrastructure that makes it possible for IoT devices to interact with cloud services as well as one another. Moreover, EMQ X makes it simpler to create and manage IoT applications by offering an intuitive interface for managing and watching MQTT connections and messages.

CHAPTER 3

Design and Implementation

3.1 Introduction

Every living creature on this earth has an important role in the ecosystem. Livestock management is a very essential part of animal husbandry. This project attempts to propose an advanced animal track-&-trace Supply-Chain Management (SCM) network for activities and real-time data collection and storage by employing certain Internet of Things' solutions, such as AWS IoT core using BLE Beacon and Raspberry Pi. Bluetooth beacons are hardware transmitters, a class of Bluetooth Low Energy (BLE) devices that broadcast their identifier to nearby portable electronic devices.

Amazon Web Services has developed AWS IoT Core which is a fully managed IoT cloud platform that lets connected devices to interact with cloud services and other devices securely and conveniently. The Core allows users to easily connect to other AWS services - AWS Lambda, AWS Kinesis, AWS IoT Analytics and many more that provides a range of compute, storage, networking, management, analytics and other capabilities, as shown in Fig 2.1.1. AWS with its huge computing space and variety of processing capabilities provide a low-cost scalable infrastructure as a service model. Its massive global presence with servers scattered widely and pricing policies which include an entry level free tier, charge by usage and literally no fixed charges makes it convenient for start-ups and growing businesses. AWS IoT Core architecture lets users connect to server without the hassle of managing or updating it in addition to providing security and reliability to data transfer. It can support a billion of devices at a time and save its current state as a shadow profile and update it continuously so that users can connect to it even when the device is offline. The Core provides end to end encryption on the data and can control access permission by means of policies. Simple condition matching to control actuators can be done on the data through use of IoT rules and it can also be connected to other services for higher processing. IoT Core supports HTTP, WebSocket, and MQTT-the light weight IoT protocol.

3.1.1 System Design

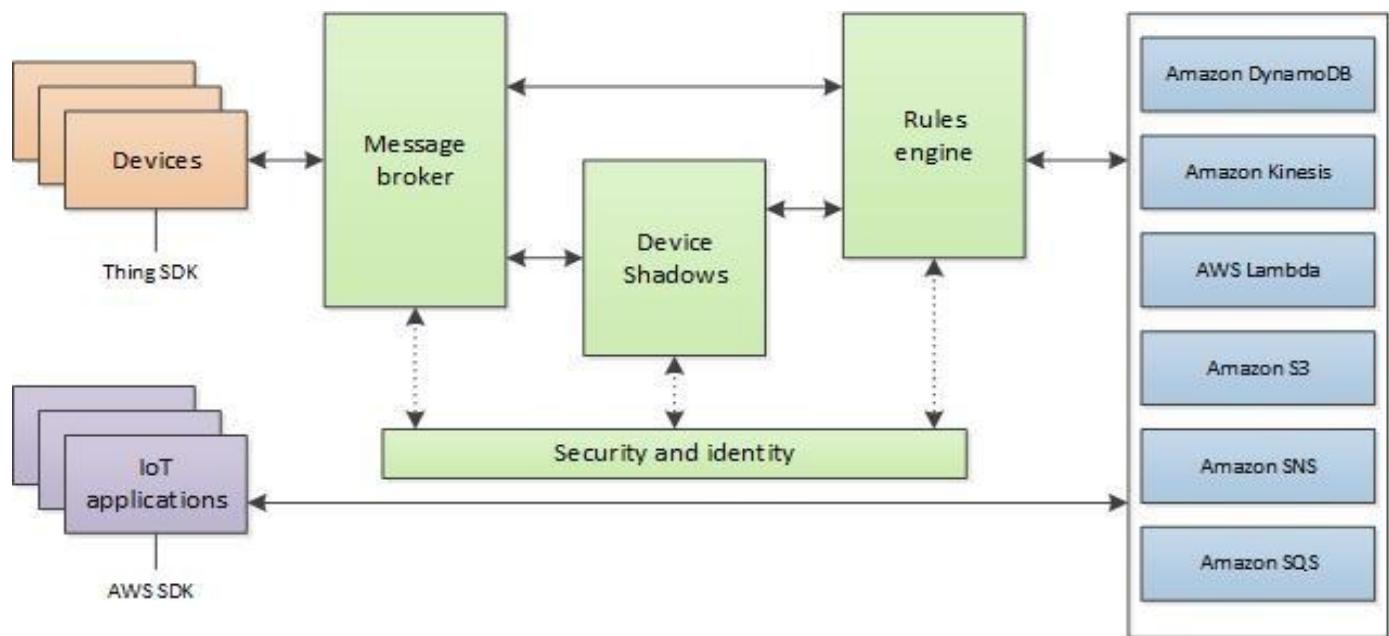


Fig 3.1.1 AWS IoT core services

There are various advantages to the proposed animal track-and-trace SCM network that uses Amazon IoT Core and BLE Beacon. It enables real-time livestock tracking and data collecting, which may be utilised for a variety of objectives such as guaranteeing their health and wellbeing, monitoring their movement, and controlling their supply chain. Because the beacon may be connected to the animal and transmit its identity to adjacent devices, it enables for simple and convenient tracking.

A safe and dependable platform for handling and storing the acquired data is provided through the usage of Amazon IoT Core. Moreover, it provides a variety of computation, storage, and networking capabilities, enabling additional data processing and analysis. The Core supports MQTT, a simple IoT protocol that makes it possible for devices to communicate and transmit data quickly. Using Amazon IoT Core and BLE Beacon, the proposed animal track-and-trace SCM network provides a complete solution for monitoring and tracking animals in a supply chain. It enables the collecting and storing of data in real-time and the connection to other services for additional processing and analysis.

3.2 Block Diagram

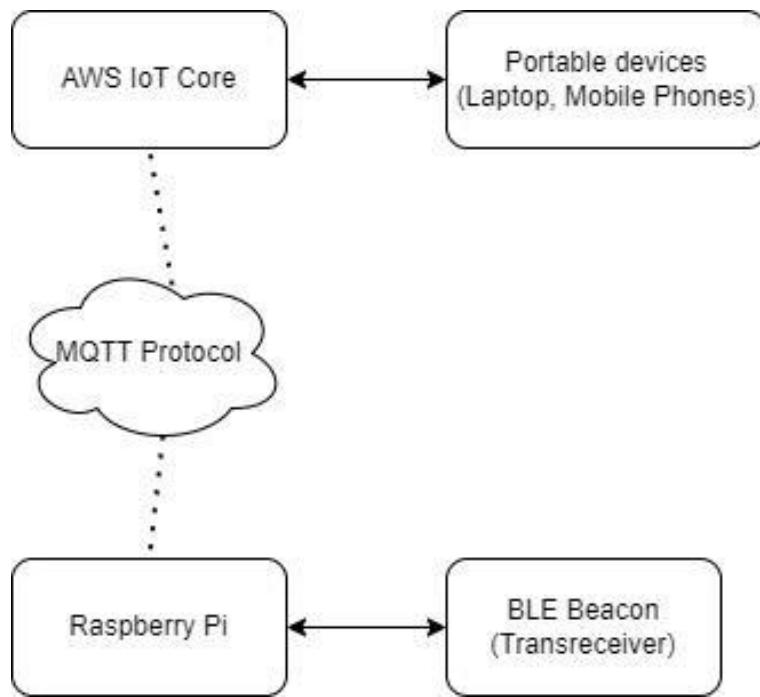


Fig 3.2.1: Block Diagram of Livestock Management

3.3 Working Principle

The Raspberry Pi serves as the system's brain in this project. The AWS IoT core is linked to the Raspberry Pi via MQTT, which allows for bidirectional machine-to-machine communication. Furthermore, the raspberry pi is Bluetooth-connected to the BLE Beacon. The BLE Beacon is used to locate livestock. Bluetooth beacons are a type of Bluetooth Low Energy (BLE) device that broadcasts its identifier to nearby portable electronic devices. When smartphones, tablets, and other devices are in close proximity to a beacon, the technology allows them to perform actions. AWS IoT offers cloud services for connecting your IoT devices to other devices and AWS cloud services. AWS IoT device software assists you in integrating your IoT devices into AWS IoT-based solutions.

1. AWS IoT Setup

AWS IoT Core supports connections with IoT devices, wireless gateways, services, and apps. Devices connect to AWS IoT Core, so they can send data to and receive data from AWS IoT services and other devices. Apps and other services also connect to AWS IoT Core to control and manage the IoT devices and process the data from IoT solution. This section describes how to choose the best way to connect and communicate with AWS IoT Core for each aspect of your IoT solution.

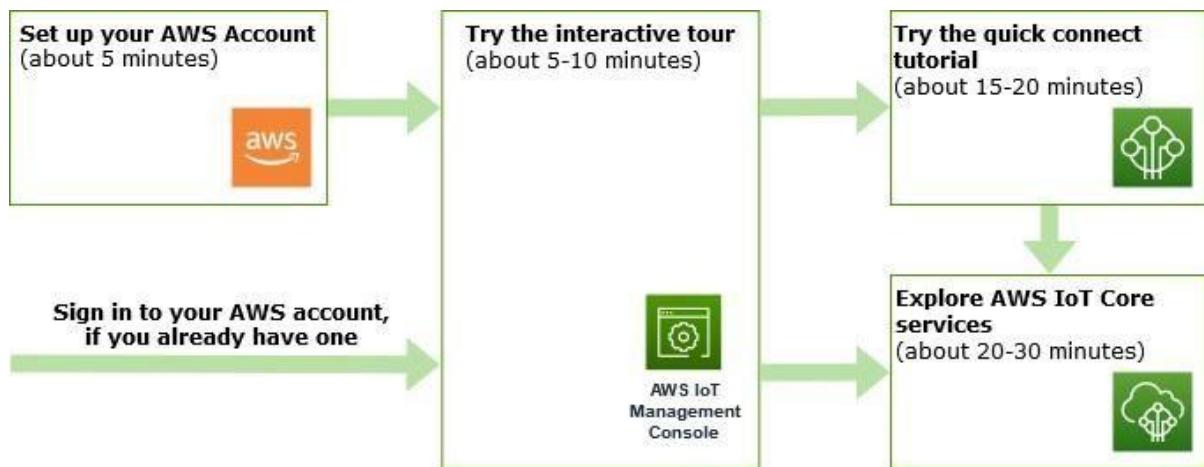


Fig 3.1.1: AWS IoT core setup

2. Connect device to AWS IoT Core

Step 1 :- Prepare IoT Device

How it works



In this wizard, we'll be creating a thing resource in AWS IoT. A thing resource is a digital representation of a physical device or logical entity.



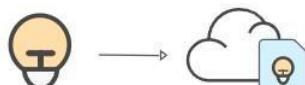
A thing resource uses certificates to secure communication between your device and AWS IoT. AWS IoT policies control access to the AWS IoT resources. This wizard creates the certificate and policy for your device.



When a device connects to AWS IoT, policies enable it to subscribe and publish MQTT messages with AWS IoT message broker.

Step 2 :- Register and Secure IoT device

Represent your device in the cloud



A thing resource is a digital representation of a physical device or logical entity in AWS IoT. A thing resource lets your device use AWS IoT features such as Device Shadows, events, jobs, and other device management features. Certificates authenticate your device, and policies authorize access to other AWS resources and actions.

This wizard helps you create the thing resource, policy, and certificate resources necessary to connect your device to AWS IoT so that it can publish simple messages. After you complete this wizard, you can edit the resources to explore AWS IoT features further.

Step 4 :- Download connection kit

Install the software on your device



We created the AWS IoT resources that your device needs to connect to AWS IoT. We also created a connection kit that includes the resources in a zipped file that you need to install on your device. The resources in the connection kit are listed below. In this step, you'll install them on your device.

Step 5 :- Run Connection Kit

After running the start script, return to this screen to see the messages between your device and AWS IoT. The messages from your device appear in the following list.

```
pi@raspberrypi: /var/iot_try
File Edit Tabs Help
Received message from topic 'sdk/test/Python': b'"Hello World! [33]"'
Publishing message to topic 'sdk/test/Python': Hello World! [34]
Received message from topic 'sdk/test/Python': b'"Hello World! [34]"'
Publishing message to topic 'sdk/test/Python': Hello World! [35]
Received message from topic 'sdk/test/Python': b'"Hello World! [35]"'
Publishing message to topic 'sdk/test/Python': Hello World! [36]
Received message from topic 'sdk/test/Python': b'"Hello World! [36]"'
Publishing message to topic 'sdk/test/Python': Hello World! [37]
Received message from topic 'sdk/test/Python': b'"Hello World! [37]"'
Publishing message to topic 'sdk/test/Python': Hello World! [38]
Received message from topic 'sdk/test/Python': b'"Hello World! [38]"'
Publishing message to topic 'sdk/test/Python': Hello World! [39]
Received message from topic 'sdk/test/Python': b'"Hello World! [39]"'
Publishing message to topic 'sdk/test/Python': Hello World! [40]
Received message from topic 'sdk/test/Python': b'"Hello World! [40]"'
Publishing message to topic 'sdk/test/Python': Hello World! [41]
Received message from topic 'sdk/test/Python': b'"Hello World! [41]"'
Publishing message to topic 'sdk/test/Python': Hello World! [42]
Received message from topic 'sdk/test/Python': b'"Hello World! [42]"'
Publishing message to topic 'sdk/test/Python': Hello World! [43]
Received message from topic 'sdk/test/Python': b'"Hello World! [43]"'
Publishing message to topic 'sdk/test/Python': Hello World! [44]
Received message from topic 'sdk/test/Python': b'"Hello World! [44]"'
```

Fig 3.3.1: Message sent from Raspberry Pi to AWS IoT Core via MQTT Protocol

3.4 System flowchart

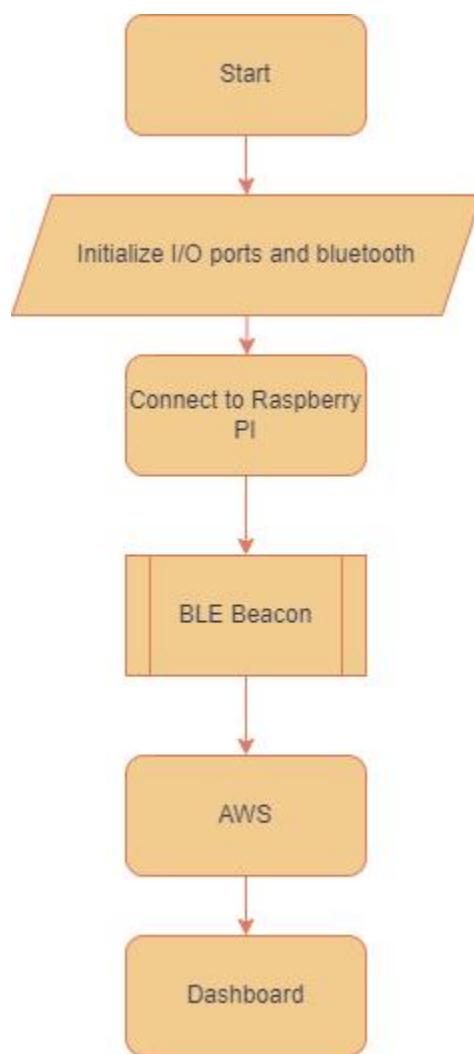
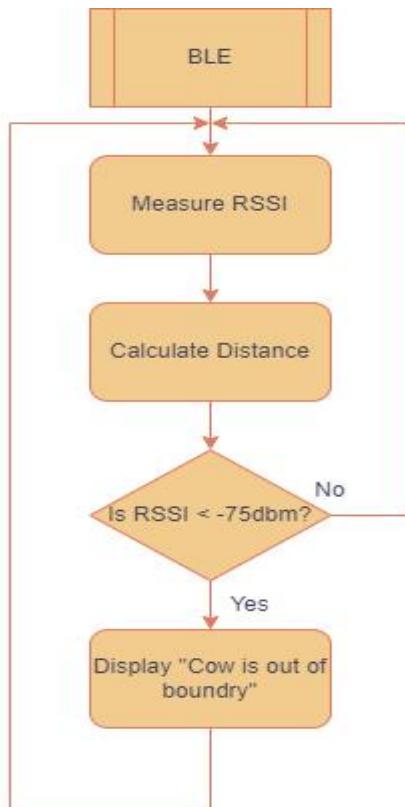
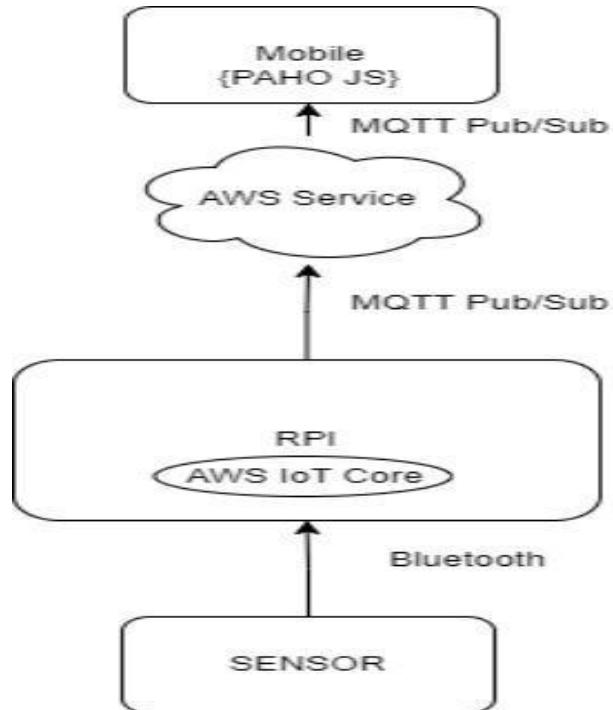


Fig 3.4.1: System Flowchart

Temperature, humidity, and other environmental data are gathered by sensors installed in the animal areas. The EMQ X Broker, which serves as a middleman between the devices and the cloud, receives the data after it has been gathered. The data is sent via the MQTT protocol, a lightweight messaging system appropriate for IoT applications. The EMQ X Rules Engine may be used to process and filter data as it comes in. This enables real-time data analysis and the ability to send alerts or notifications when particular criteria are surpassed. Using a technology like Grafana, a dashboard may be developed to display the data collected in real-time. Farmers and managers can use this dashboard. Based on the findings of the study, farmers or managers can take the required steps to improve livestock management, such as revising feeding schedules, changing the environment, or treating any ill animals.

**Fig 3.4.2: BLE Beacon working flowchart**

3.5 Design flow

**Fig 3.5: Design flow**

3.6 Summary

Using AWS IoT, livestock management involves gathering data from sensors installed in livestock areas, sending the data to AWS IoT Core. The proposed animal track-and-trace SCM network using AWS IoT Core and BLE Beacon offers a comprehensive solution for managing and tracking livestock. It allows for real-time data collection and storage, as well as the ability to connect to other services for further processing and analysis. The use of BLE Beacon allows for easy and convenient tracking, while the use of AWS IoT Core provides a secure and reliable platform for data management.

In order to fulfil the rising demands on farms and enable them to run effectively and precisely using technology, AWS is collaborating with AGCO. Solutions for managing livestock that are IoT-enabled remove uncertainty regarding herd health. Battery-powered sensors that are attached to an animal's collar or tag track its location, temperature, blood pressure, and heart rate while wirelessly transmitting the information to a farmer's device in almost real-time.

CHAPTER 4

Software Implementation

4.1 AWS IoT Core

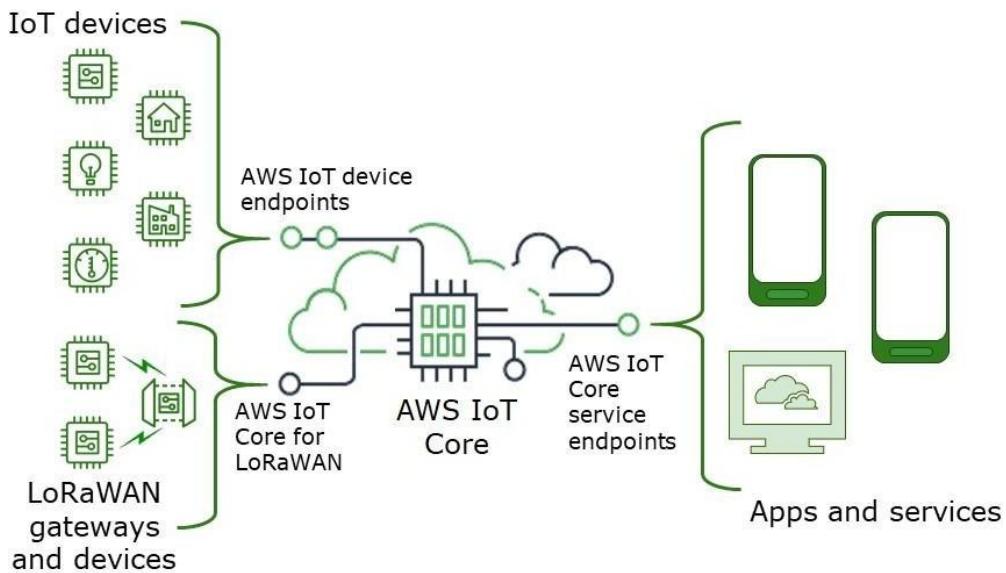


Fig 4.1.1: AWS IoT Core

AWS IoT provides the cloud services that connect your IoT devices to other devices and AWS cloud services. AWS IoT provides device software that can help you integrate your IoT devices into AWS IoT-based solutions. If your devices can connect to AWS IoT, AWS IoT can connect them to the cloud services that AWS provides. AWS IoT lets you select the most appropriate and up-to-date technologies for your solution. To manage and support IoT devices in the field, AWS IoT Core supports these protocols:

- MQTT (Message Queuing and Telemetry Transport)
- MQTT over WSS (WebSocket's Secure)
- HTTPS (Hypertext Transfer Protocol - Secure)
- LoRaWAN (Long Range Wide Area Network)

The AWS IoT Core message broker supports devices and clients that use MQTT and MQTT over WSS protocols to publish and subscribe to messages. It also supports devices and clients that use the HTTPS protocol to publish messages. AWS IoT Core for LoRaWAN helps you connect and manage wireless LoRaWAN (low-power long-range Wide Area Network) devices. AWS IoT Core for LoRaWAN replaces the need for you to develop and operate a LoRaWAN Network Server (LNS).

4.2 EMQX

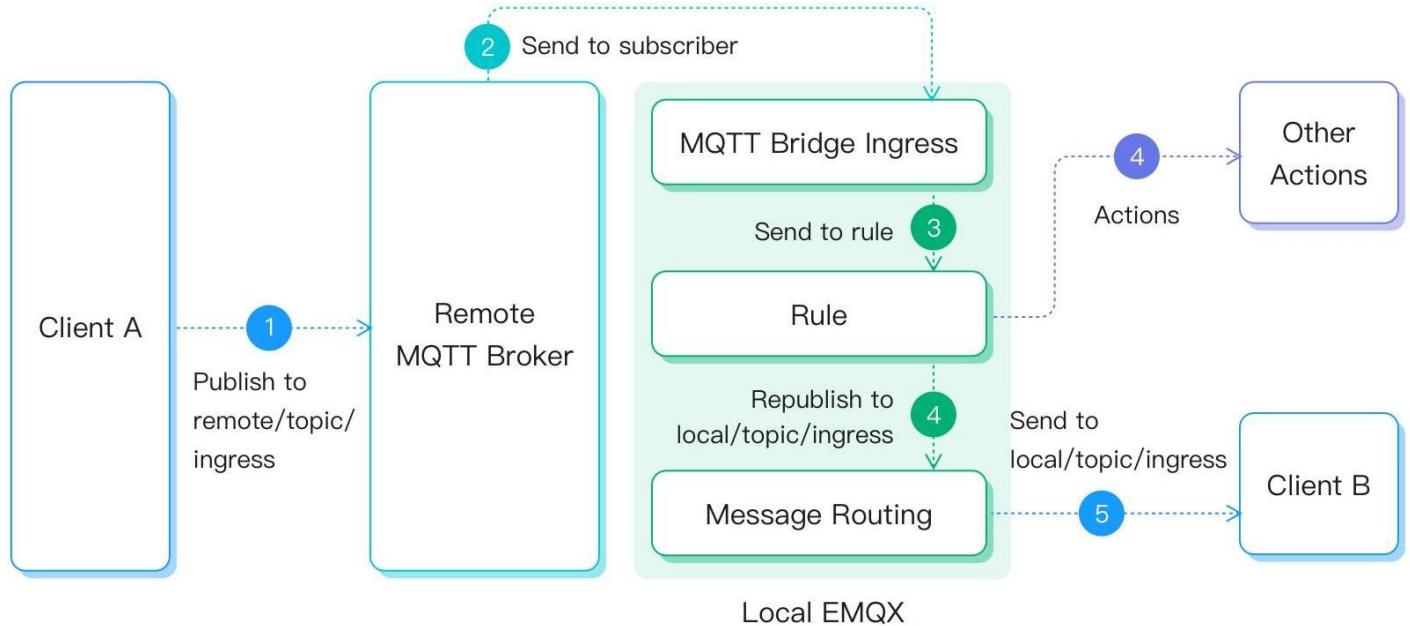


Fig 4.2.1: EMQX

EMQ X provides a web-based dashboard for managing and monitoring the broker. It provides real-time metrics, such as message throughput, connection counts, and broker performance. It also provides detailed logs and error messages for troubleshooting and debugging. A distributed MQTT message broker that is open-source and optimized for handling massive IoT applications is called EMQ X. For IoT applications, it offers excellent performance, stability, and scalability, making it a popular option for developers and businesses. The MQTT 5.0, MQTT 3.1.1, and Web Socket protocols are all supported by EMQ X, along with plugins for adding new features. It may be utilized for a variety of IoT applications, including industrial IoT, smart cities, and smart homes. IoT data management is made flexible and customized with EMQ X, which also enables real-time data processing, filtering, and analytics.

4.3 PAHO Javascript

Paho JavaScript is a JavaScript client library for MQTT, a lightweight messaging protocol developed for Internet of Things (IoT) applications. Paho JavaScript is an Eclipse Paho project that provides a simple and user-friendly API for connecting to MQTT brokers and publishing/subscribing to MQTT topics.

Developers may use Paho JavaScript to create web-based IoT apps that connect with MQTT-enabled devices and servers. The library works with a variety of platforms and frameworks and supports both browser-based and Node.js applications. Paho JavaScript adheres to the most recent MQTT 5.0 protocol, which includes capabilities such as Quality of Service (QoS) levels, Last Will and Testament (LWT), and persistent sessions. SSL/TLS encryption is also supported for secure communication. Paho JavaScript is a popular programming language for developing web-based IoT applications that require real-time data transfer and may be used for a variety of IoT use cases such as home automation, industrial automation, and smart cities.

4.4 MQTT Protocol

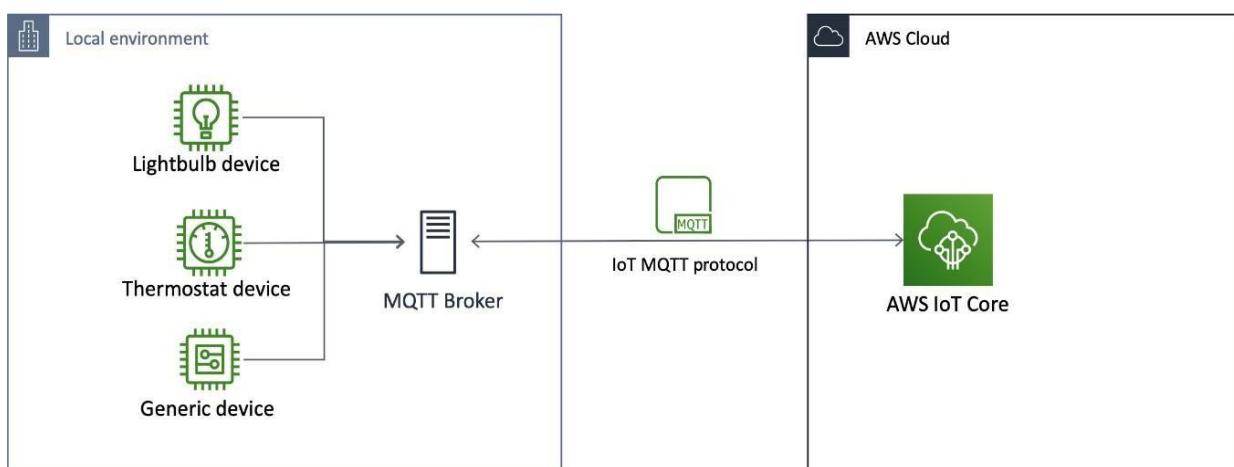


Fig 4.4.1: MQTT Protocol

MQTT is an open source protocol that follows publish - subscribe mechanism for many to many distributions as shown in Fig. 3. It is lightweight and bandwidth efficient as it uses a small header and it also has continuous session awareness. It is designed to minimize use of device resources and ensure reliability and varying degrees of service delivery assurance . Using MQTT, the clients do not communicate directly with the endpoint; publish - subscribe mechanism decouples the client (publisher) who is sending the message and another client (subscriber) who is receiving the message.
 MQTT architecture The MQTT broker connects the publishers and subscribers. It receives all incoming messages from the publishers, filters them and redistributes them accordingly such that only specific clients receive specific messages.

4.5 Balena IoT Dashboard



Fig 4.5.1: Balena.io

The AWS IoT platform provides a valuable suite of services to collect, store, and distribute IoT data and actions. Its IoT Core service is the portal for registration and messaging with Internet-connected things. We want to make it easy for balena devices to register and interact with IoT Core.

IoT provisioning tools automate device registration to AWS IoT Core, and leverage balenaCloud and environment variables to store and access the registration credentials. This guide shows how provisioning works and gets project started with the tools in the aws-iot-provision repository.

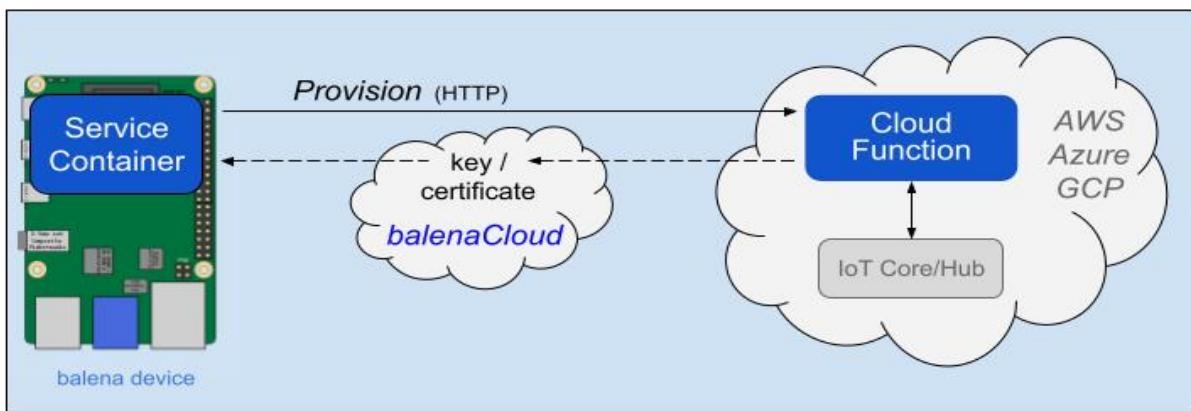


Fig 4.5.2: Cloud IoT Provisioning with AWS

Provisioning includes three components:

- **Service Container** like Cloud Relay block on a device to request the provisioning and use the credential environment variables from balenaCloud
- **Lambda (cloud) function** to securely validate device identity and register the device with IoT Core, triggered by an HTTP request (source code)
- **balenaCloud** to accept and store the generated key/certificate credentials for the device

CHAPTER 5

Results and Discussion

5.1 Results

Steps Involved

- Successfully created AWS account.
- AWS IoT Setup Successfully completed.
- Set up a directory to test MQTT publishing.
- Installed pip and the AWS IoT SDK for Python.
- Created an AWS IoT Core policy.
- Created an AWS IoT thing.
- Created a Python Program file to publish the message via Raspberry Pi and AWS IoT Core through MQTT Protocol.
- Tested the Setup.
- Received the message from Raspberry Pi and got the response in AWS Console.
- Connected BLE Beacon to Raspberry pi and viewed the MAC Address of BLE Beacon.
- Client-side server is connected successfully.

```
pi@raspberrypi:~/Desktop/python_aws_iot_sdk $ python3 publish.py
Connecting to a32mh0ovcqxbg8-ats.iot.ap-south-1.amazonaws.com with client ID 'testDevice'...
Connected!
Begin Publish
Published: '{"message": "Hello World [1]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [2]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [3]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [4]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [5]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [6]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [7]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [8]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [9]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [10]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [11]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [12]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [13]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [14]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [15]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [16]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [17]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [18]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [19]"}' to the topic: 'test/testing'
Published: '{"message": "Hello World [20]"}' to the topic: 'test/testing'
Publish End
```

Fig 5.1.1: Message published on Raspberry Pi



Fig 5.1.2: Message received by the subscriber AWS management console

A terminal window showing command-line output. The user runs several commands related to Bluetooth: "cd bluez-5.50/tools/", "sudo hciconfig hci0 down", "sudo hciconfig hci0 up", and "sudo hcitool -i hci0 leskan". The output includes a "LE Scan ..." message followed by a list of MAC addresses and their descriptions: 61:04:8A:10:6D:AD (unknown), 61:04:8A:10:6D:AD (unknown), DD:34:02:06:D3:30 (unknown), DD:34:02:06:D3:33 (unknown), DD:34:02:06:D3:33 KBPro_215521, and DD:34:02:06:D3:30 KBPro_215518. Below this, it shows two entries for 7A:76:D7:7A:5E:72, both labeled as "(unknown)".

```
pi@raspberrypi:~ $ cd bluez-5.50/tools/
pi@raspberrypi:~/bluez-5.50/tools $ sudo hciconfig hci0 down
pi@raspberrypi:~/bluez-5.50/tools $ sudo hciconfig hci0 up
pi@raspberrypi:~/bluez-5.50/tools $ sudo hcitool -i hci0 leskan
LE Scan ...
61:04:8A:10:6D:AD (unknown)
61:04:8A:10:6D:AD (unknown)
DD:34:02:06:D3:30 (unknown)
DD:34:02:06:D3:33 (unknown)
DD:34:02:06:D3:33 KBPro_215521
DD:34:02:06:D3:30 KBPro_215518
7A:76:D7:7A:5E:72 (unknown)
7A:76:D7:7A:5E:72 (unknown)
```

Fig 5.1.3: BLE Beacon connected to Raspberry Pi

Clients

Client ID		Username				<input type="button" value="Search"/>	<input type="button" value="Reset"/>	More
Client ID	Username	IP Address	Keepalive	Protocol Type	Status	Connected At	Actions	
raspberry-pi-client	rpi	157.45.171.66:36105	60	MQTT	Connected	2023-04-14 14:31:09	<input type="button" value="Kick Out"/>	

Subscriptions

Client ID		Topic		<input type="button" value="Search"/>	<input type="button" value="Reset"/>	More
Client ID	Topic			QoS		
raspberry-pi-client	hello_world			0		

Fig 5.1.4: EMQX MQTT Broker-Client message Pub/Sub**Fig 5.1.5: BLE Beacon Distance Measuring**

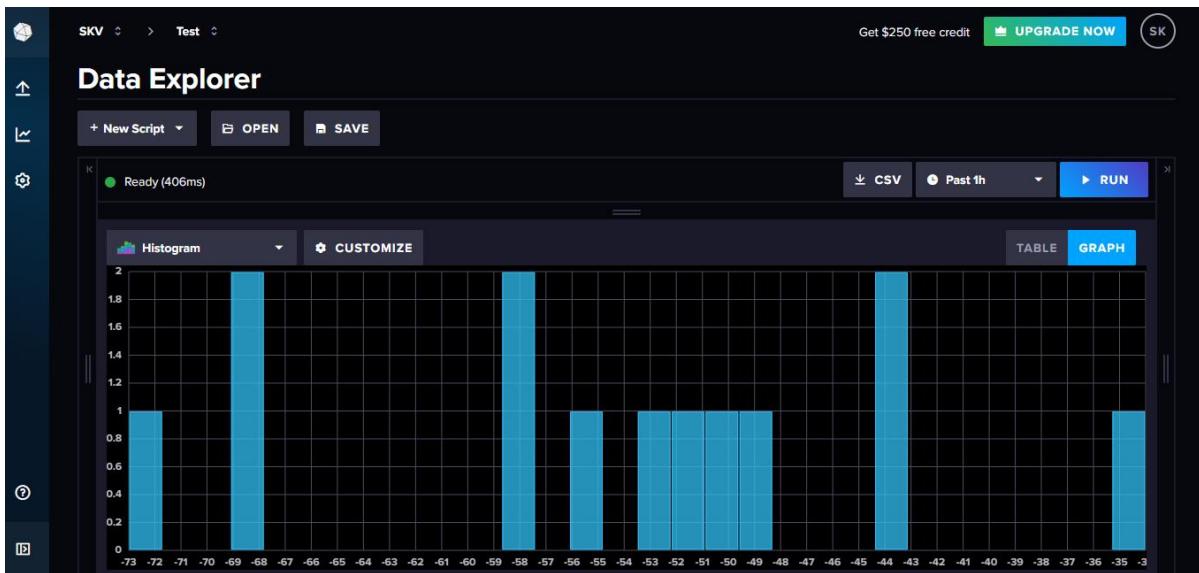


Fig 5.1.6: Histogram representing RSSI values vs distance for near objects

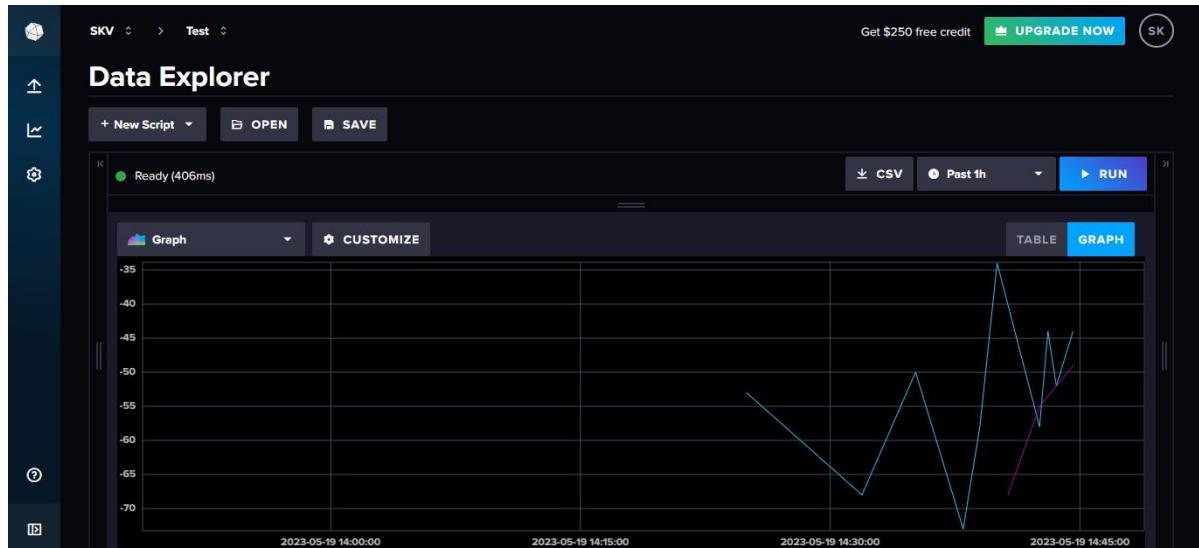


Fig 5.1.7: Graph representing distance of near objects

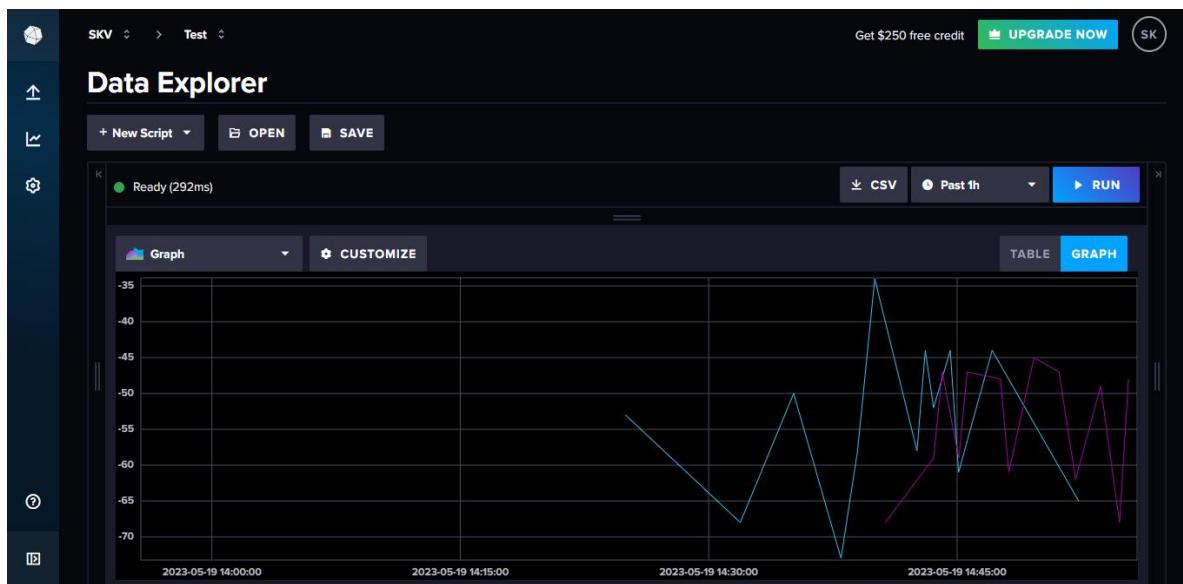


Fig 5.1.8: Graph representing distance of far objects



Fig 5.1.9: Histogram representing RSSI values vs distance for near objects

The screenshot shows the Data Explorer interface from a cloud provider. At the top, there are buttons for 'New Script', 'OPEN', and 'SAVE'. Below this is a status bar with 'Ready (292ms)', a 'CSV' button, a 'Past 1h' time filter, and a 'RUN' button. The main area is titled 'Data Explorer' and contains a table with the following columns: deviceId, deviceName, tagId, _time, and _value. The table displays 10 rows of data, each corresponding to a device named 'jolly-dream' with tag ID '777772E-6B6B-6D63-6E...' and various timestamped values ranging from -68 to -62.

deviceId	deviceName	tagId	_time	_value
8c580bda41d1fea7db0b6...	jolly-dream	777772E-6B6B-6D63-6E...	2023-05-19 02:40:40 GM...	-68
8c580bda41d1fea7db0b6...	jolly-dream	777772E-6B6B-6D63-6E...	2023-05-19 02:43:34 GM...	-59
8c580bda41d1fea7db0b6...	jolly-dream	777772E-6B6B-6D63-6E...	2023-05-19 02:44:05 GM...	-47
8c580bda41d1fea7db0b6...	jolly-dream	777772E-6B6B-6D63-6E...	2023-05-19 02:45:06 GM...	-59
8c580bda41d1fea7db0b6...	jolly-dream	777772E-6B6B-6D63-6E...	2023-05-19 02:45:36 GM...	-47
8c580bda41d1fea7db0b6...	jolly-dream	777772E-6B6B-6D63-6E...	2023-05-19 02:47:37 GMT...	-48
8c580bda41d1fea7db0b6...	jolly-dream	777772E-6B6B-6D63-6E...	2023-05-19 02:48:07 GM...	-61
8c580bda41d1fea7db0b6...	jolly-dream	777772E-6B6B-6D63-6E...	2023-05-19 02:49:38 GM...	-45
8c580bda41d1fea7db0b6...	jolly-dream	777772E-6B6B-6D63-6E...	2023-05-19 02:51:09 GMT...	-47
8c580bda41d1fea7db0b6...	jolly-dream	777772E-6B6B-6D63-6E...	2023-05-19 02:52:09 GM...	-62

Fig 5.1.10: Table Representation

The screenshot shows the BLE Interface from a cloud provider. At the top, there are tabs for 'Goushala- A Livestock Management using AWS IoT' (selected), 'Submissions', and '+ Add Tab'. Below this is a search bar and a 'Filter' button. The main area is a table with columns: Tag ID, Status, and RSSI. The table displays 5 rows of data, each corresponding to a tag with ID '777772E-6B6B-6D63-6E2E-636F6D000001-3-18910' and various status and RSSI values.

	Tag ID	Status	RSSI
1	777772E-6B6B-6D63-6E2E-636F6D000001-3-18910	Within Boundary	-53
2	777772E-6B6B-6D63-6E2E-636F6D000001-3-18913	Out of Boundary	-75
3	777772E-6B6B-6D63-6E2E-636F6D000001-3-18910	Within Boundary when obstacle is present	-73
4	777772E-6B6B-6D63-6E2E-636F6D000001-3-18913	Within Boundary when no obstacle	-34
5	777772E-6B6B-6D63-6E2E-636F6D000001-3-18910	Out of Boundary when obstacle is present	-80

Fig 5.1.11: BLE Interface

Balena.io Setup

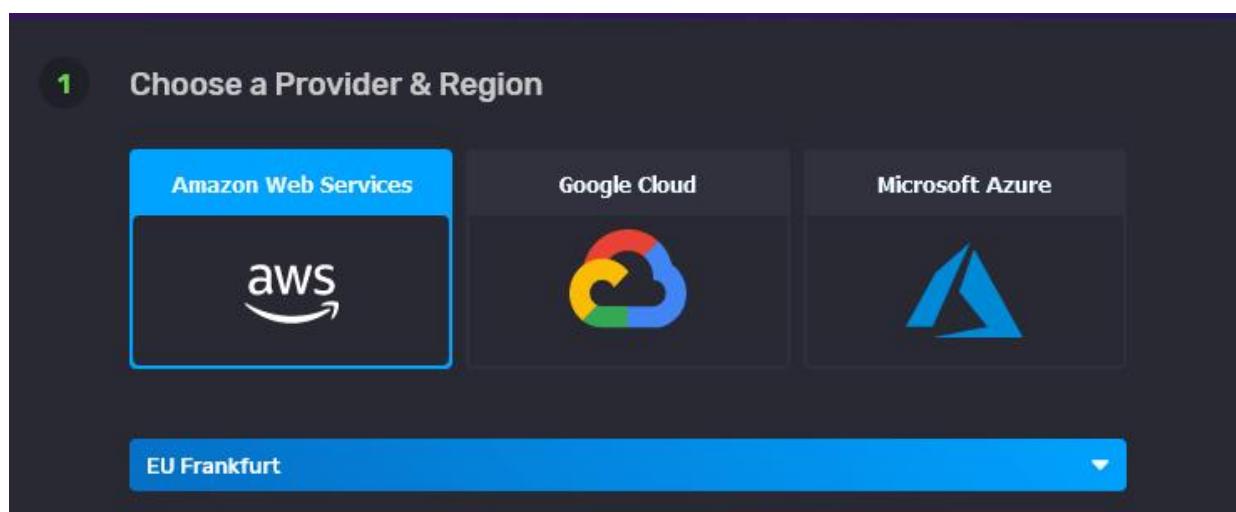
1. Create an InfluxDB Cloud Account

First going to set up the cloud database first, since our project need some information from it later.

InfluxDB Cloud 2.0 (in public beta at the time of writing) is a *Time Series-as-a-Platform* database technology.

2. Set up AWS account

First, create an account on the InfluxDB Cloud 2.0 site, choosing AWS as the cloud provider:



3. Create bucket and token in InfluxDB Cloud

To enable the Raspberry Pi's services to read from and write to the bucket, a token must be created. As long as it is set up to read and write to the bucket that was just established, users can use the same token many times in this situation:

4. Setup the Raspberry Pi

Create and deploy to application

Use an existing application instead

Application name: balena-Locating

Default device type: Raspberry Pi 4

Application type: View docs

Microservices: recommended

GitHub repo URL: https://github.com/balena-io-playground/balenaLocating

Advanced

Application configuration variables

+ Add item

Name: INFLUX_BUCKET	Value: UPDATE_ME
Name: INFLUX_HOST	Value: UPDATE_ME
Name: INFLUX_KEY	Value: UPDATE_ME
Name: INFLUX_ORG	Value: UPDATE_ME
Name: RUN_DASHBOARD	Value: false
Name: DEBUG	Value: false
Name: RSSI_THRESHOLD	Value: -75
Name: SEP_PERIOD	Value: 30

+ Add item

Cancel Create and deploy

The diagram illustrates the mapping between AWS CloudWatch Metrics and the AWS IoT Core application configuration variables. Red arrows point from specific metrics in the CloudWatch Metrics interface to their corresponding configuration variables in the AWS IoT Core application setup.

AWS CloudWatch Metrics Interface:

- Buckets:** A bucket named "balenaLocating" is highlighted. A red arrow points from its "Copy to Clipboard" button to the "INFLUX_BUCKET" variable in the AWS IoT Core application configuration.
- Tokens:** A token named "balenaLocating" is highlighted. A red arrow points from its "Copy to Clipboard" button to the "INFLUX_HOST" variable in the AWS IoT Core application configuration.
- Client Libraries:** A library named "balenaLocating" is highlighted. A red arrow points from its "Copy to Clipboard" button to the "INFLUX_KEY" variable in the AWS IoT Core application configuration.
- Monitoring:** A metric named "balenaLocating" is highlighted. A red arrow points from its "Copy to Clipboard" button to the "INFLUX_ORG" variable in the AWS IoT Core application configuration.
- Tasks:** A task named "balenaLocating" is highlighted. A red arrow points from its "Copy to Clipboard" button to the "RUN_DASHBOARD" variable in the AWS IoT Core application configuration.

AWS IoT Core Application Configuration:

The configuration variables are listed as follows:

- INFLUX_BUCKET: UPDATE_ME
- INFLUX_HOST: UPDATE_ME
- INFLUX_KEY: UPDATE_ME
- INFLUX_ORG: UPDATE_ME
- RUN_DASHBOARD: false
- DEBUG: false
- RSSI_THRESHOLD: -75
- SEP_PERIOD: 30

Click Create and Deploy at this point.

By doing this, an application will be generated with all of the code deployed and the device variables created and set!

5. Add a device and download the balenaOS disk image from the dashboard

 **Add new device**

Select device type

Raspberry Pi 4

Select OS version

v2.46.1+rev3 (recommended)

Select edition

Development Production

ⓘ Recommended for first time users

Development images should be used when you are developing an application and want to use the fast local mode workflow. This variant should never be used in production.

Network Connection

Ethernet only Wifi + Ethernet

Wifi SSID

yourwifi

Wifi Passphrase

***** 

+ Advanced

 [Download balenaOS \(~146 MB\)](#)

This process generates a customized image that is tailored for the application and device type and, if user requested them, includes network configurations.

6. Flash SD card with the balenaOS disk image and boot the device



Service	Status	Release
beaconService	Running	a312a91
dashboard	Running	a312a91

7. View the Dashboard

The dashboard is a simple web app which queries the cloud database to find the last time a beacon was detected, and which sensor detected it. To view it, turn on user device's Public URL:

DEVICE

Office (edit)

STATUS ✓ Online	UUID 8a4ab88 (edit)	TYPE Balena Fin (CM3)
LAST SEEN (VPN) Online (for a day)	HOST OS VERSION balenaOS 2.48.0+rev1 development	SUPERVISOR VERSION 10.8.0
CURRENT RELEASE 28c5a0b (edit)	TARGET RELEASE 28c5a0b	IP ADDRESS 192.168.86.179 (edit) 192.168.86.178 (edit)
TAGS (0) (edit) No tags configured yet	PUBLIC DEVICE URL (edit)	
NOTES	<div style="border: 1px solid #ccc; padding: 5px;">Add device notes...</div>	

SERVICES

Service	Status	Release
beaconService	Running	28c5a0b
dashboard	Running	28c5a0b

The screenshot shows the 'BalenaLocating Dashboard' interface. On the left is a sidebar with 'Home', 'Devices', and 'Tags' buttons. The main area has a heading 'Where's my stuff?' and a sub-instruction 'Here's a list of the iBeacons I've found.' Below this is a table with two rows of data. A button 'Hide Un-Named Tags?' is visible above the table. At the bottom, a message says 'Refreshing in 1 seconds'.

Tag (ID)	Last Seen	Location (UUID)	Strength
7777772E-6B6B-6D63-6E2E-636F6D000001-3-18910	05/02/2023 07:58:12	aged-time	-54
7777772E-6B6B-6D63-6E2E-636F6D000001-3-18913	05/02/2023 07:57:46	aged-time	-48

There's a tag on work with the lengthy ID FDA50693-A4E2-4FB1-AFCF-C6EB07647825-10011-10011. while haven't yet given it a name. likewise find that the website provides information about the location, signal strength, and the most recent time it was seen. Maybe don't need to continuously refreshing the browser to see the most recent information because underneath the table is a countdown till the table changes.

The 'place (UUID)' column displays a mapping between the place that each device in the balenaCloud application represents and its UUID. In reality, this location value is just the device's name; I'll teach how to rename a device shortly. The device(s) will initially have names generated by Balena.

5.2 Societal Benefits

Animal husbandry includes a critical component called livestock care. There have been several occasions where livestock has vanished and then been found dead. This is primarily a result of the cattle being poorly controlled by the handlers. To solve this problem, we have created a technique. The caretaker's obligation is diminished in this situation. When the herd enters a dangerous area, he may react quickly and bring them back to safety. By enhancing animal health and production, AWS IoT may help increase the supply of high-quality meat, dairy, and other livestock products. In underdeveloped countries, in particular, this can help to improve food security and reduce the risk of food shortages. Farmers can get help from AWS IoT to make the most use of resources like grain, water, and pastureland. This may minimise the impact of animal production on the environment, including greenhouse gas emissions and water use. AWS IoT's real-time monitoring of vital signs, location, and behaviour may help identify animals in need of medical care and ensure that they get it quickly. This has the potential to improve animal welfare and reduce mortality rates, two important social objectives.

CHAPTER 6

Conclusion and Future Scope

6.1 Advantages

- **Improved data collection:** AWS IoT allows for the collection of massive volumes of data from sensors and devices put on animals and their environments. This information may be analyzed in real time, giving farmers insights on animal health, behaviour, and productivity.
- **Remote monitoring:** AWS IoT enables farmers to remotely monitor their animals and their surroundings, decreasing the need for physical inspections and enabling for the discovery of concerns like as disease outbreaks to occur more quickly.
- **Predictive analytics:** Using AWS IoT to analyse data and detect trends, farmers may forecast and avoid problems before they arise. Sensors, for example, may be used to track feed intake and anticipate when an animal will become unwell or give birth.
- **Improved efficiency:** AWS IoT can assist farmers in automating processes such as feeding and watering, lowering labour costs and increasing operational efficiency.
- **Improved animal welfare:** By monitoring vital signs, location, and behaviour in real time, AWS IoT can assist identify animals in need of medical attention and guarantee that they receive prompt treatment. This has the potential to increase animal well being and lower death rates.
- **Environmental monitoring:** AWS IoT may be used to monitor and regulate environmental elements such as temperature and humidity, ensuring that animals are housed in the best possible circumstances. This can assist cattle minimize stress and disease, boosting their health and production.

6.2 Limitations

While using AWS IoT in livestock management has various advantages, there are also restrictions and problems to consider. Here are some of the restrictions:

- **Cost:** Implementing AWS IoT systems can be costly, particularly for small and medium-sized farms. The cost of sensors, devices, and infrastructure might be prohibitively expensive, and there may be continuing expenditures for data storage and processing.
- **Technical expertise:** Setting up and maintaining an AWS IoT system involves specialised technical knowledge, which some farmers may not have or find practicable. This can be a hurdle to adoption, limiting the benefits of these systems to bigger, more technologically advanced farms.
- **Data privacy and security:** Collecting and analysing data from cattle might generate privacy and security concerns. Farmers must ensure that sufficient precautions are in place to protect their animals' privacy and prevent unauthorised access to sensitive data.
- **Connectivity:** AWS IoT requires a consistent and dependable internet connection, which may not be accessible in all places. Poor connection can restrict these devices' use and hinder real-time monitoring and analysis.
- **Compatibility:** AWS IoT may not be suitable for all cattle or farming operations. Different types of sensors and devices may be required for different animal species, and certain agricultural operations may not lend themselves to the deployment of IoT technology.

6.3 Applications

AWS IoT may be used for livestock management in a variety of ways, some of which are as follows:

- Animal health and behaviour may be monitored using AWS IoT, which provides real-time data on vital signs, mobility, and other parameters. This information may be used to detect early symptoms of disease or damage, allowing farmers to intervene before the situation worsens.
- AWS IoT can assist farms in optimizing breeding cycles and improving reproductive success. Sensors can identify when animals are in heat, allowing farmers to better schedule insemination and enhance the chance of successful fertilization.
- AWS IoT can assist farmers in optimizing their livestock feeding and nutrition programmes. Sensors may be used to monitor feed and water intake, providing data on consumption rates and assisting farmers in making ration adjustments for optimal development and health.
- Web Application Hosting: AWS provides scalable and reliable infrastructure to host web applications. Services like Amazon EC2 (Elastic Compute Cloud), AWS Elastic Beanstalk, and AWS Lambda allow developers to deploy and manage applications easily.
- Data Storage and Management: AWS offers various storage options, such as Amazon S3 (Simple Storage Service) for object storage, Amazon EBS (Elastic Block Store) for block-level storage, and Amazon RDS (Relational Database Service) for managed databases. These services provide durability, scalability, and ease of data management.
- Internet of Things (IoT): AWS IoT services provide a platform for securely connecting and managing IoT devices. AWS IoT Core allows devices to communicate with the cloud, while services like AWS IoT Analytics and AWS IoT Events enable data analysis and automation based on IoT device data.
- Machine Learning: AWS offers a suite of services for machine learning and artificial intelligence (AI). Amazon SageMaker provides a fully managed platform for building, training, and deploying machine learning models. AWS Rekognition offers image and video analysis, while Amazon Comprehend provides natural language processing capabilities.
- DevOps and Continuous Integration/Deployment (CI/CD): AWS provides tools like AWS CodePipeline and AWS CodeDeploy for building automated workflows and managing the continuous integration, delivery, and deployment of applications. Services like AWS CodeCommit and AWS CodeBuild support source code management and build processes.

6.4 Conclusion

The use of AWS IoT for livestock management has the potential to revolutionize the way farmers manage their cattle. AWS IoT may help farmers optimize their operations, increase animal welfare, and cut costs by giving real-time data on animal health, behaviour, and environmental factors. The benefits of implementing AWS IoT in livestock management include increased production, efficiency, and profitability, as well as decreased environmental impact and increased food security. Implementing AWS IoT in livestock management also has several obstacles and constraints, such as the high cost of installation, the requirement for technical skills, worries about data privacy and security, and the requirement for a dependable internet connection. Despite these obstacles, AWS IoT has enormous potential benefits in livestock management, and the technology is fast expanding. As more farmers utilize these systems and develop experience with them, the benefits are expected to expand and the limits will be addressed. Overall, using AWS IoT in livestock management is a great potential for farmers to better their operations while also contributing to a more sustainable and responsible agricultural industry.

6.5 Future Scope

AWS IoT may be used with predictive analytics and machine learning algorithms to find trends in animal behaviour, find early indicators of sickness, and improve breeding and feeding practise. Robotic systems may be connected to AWS IoT to automate duties like feeding, cleaning, and monitoring, which lowers labour costs and boosts productivity. AWS IoT can be combined with other cutting-edge innovations like block-chain and precision farming to give a more complete picture of the whole livestock supply chain. AWS IoT may be used to monitor livestock in rural locations as internet connectivity spreads, allowing farmers to grow their businesses and access new markets. AWS IoT may be tailored to match the particular requirements of various livestock species, production methods, and geographical locations, enabling farmers to adjust their management practices to their particular situation.

References

- 1.Karan Juj, Charith Perera, "Exploring the Suitability of BLE Beacons to Track Poacher Vehicles in Harsh Jungle Terrains", Cardiff University, United Kingdom, IEEE-2020.
- 2.J. M. R. de S. Neto, J. J. C. Silva, T. C. M. Cavalcanti, D. P. Rodrigues and J. S. da Rocha Neto, Ian A. Glover, "Propagation Measurements and Modeling for Monitoring and Tracking in Animal Husbandry Applications", Department of Electronic & Electrical Engineering University of Strathclyde, Scotland, UK and CAPES Visiting Professor, Universidade Federal de Campina Grande, Campina Grande - PB, Brazil, IEE-2010.
- 3.Brad Kennedy, Graham W. Taylor, Petros Spachos,"BLE Beacon Based Patient Tracking in Smart Care Facilities", School of Engineering, University of Guelph, Guelph, ON, N1G 2W1, Canada, Demonstration on Pervasive Computing and Communications 2018.
- 4.Kais Mekki, Eddy Bajic, Fernand Meyer, "Indoor Positioning System for IoT Device based on BLE Technology and MQTT Protocol", Research Centre for Automatic Control of Nancy, Campus Sciences, BP 70239, Vandoeuvre, 54506, France, IEEE 5th World Forum on Internet of Things (WF-IoT), 2019.
- 5.Nadia Imtiaz Jaya,Md. Farhad Hossain, "A Prototype Air Flow Control System for Home Automation using MQTT over Websocket in AWS IoT Core", Department of Electrical and Electronic Engineering Bangladesh University of Engineering and Technology Dhaka-1205, Bangladesh, International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery-2018

Appendix

Expenditure

Sl No.	Components	No. Of Components	Cost
1.	Raspberry Pi 3 Model B+	1	Rs. 3800
2.	BLE Beacon	2	Rs. 1500
		Total cost	Rs.5300

Discrimination of Grey Cast Iron Morphology using Integrated SVM and ANN Approaches

Amany Khaled, Mostafa R.A. Atia

Mechanical Engineering (Mechatronics) Department
Arab Academy for Science, Technology & Maritime
Transport
Cairo, Egypt
amanykhaled@gmail.com, mrostom1@yahoo.com

Tarek Moussa

Mechanical Design and production Department
Faculty of Engineering, Ain Shams University
Cairo, Egypt
tarekmoussa2001@yahoo.com

Abstract—The internal structure of Grey Cast Iron (GCI) and its microstructure determines the acceptance or rejection of several mechanical parts in the inspection process. This is based on the change of GCI mechanical properties due to the variation of its cooling rate. Visual inspection by metallurgical experts has been the approved method to assess GCI types. However, such method has always been subject to human error, biased categorization, lack of experience and variations in performance level. Even though several commercial software is available for such discrimination approaches, multiple flaws and defects are detected in the way it assesses samples. This research introduces a new software that is capable of distinguishing between GCI and other types of cast irons based on Support Vector Machines (SVM). Moreover, the software can identify the GCI types according to international standards using a well-trained Artificial Neural Network (ANN).

Keywords—Feed forward neural network; Gray Level Co-Occurrence Matrix (GLCM); grey cast iron; image processing; microstructure; support vector machines

I. INTRODUCTION

Cast irons are iron-carbon-silicon alloys that serve a great number of different applications in industry where it's the elementary structure of manufacturing machines like rollers, valves, pump bodies and mechanical gears, among others. The properties of such alloys are influenced by the internal morphology whose characterization governs the material selection criteria in such applications [1]. Based on specific standards, experts used to visually identify and characterize such morphological differences. Such a task is exhausting and time consuming since specialists spend a lot of time subjected to the light through the objective length of the microscope. Consequently, this can cause them real fatigue to exhaustion which can affect the accuracy and quality of their decisions [2]. Moreover, assessment of industrial samples is prone to commercial fraud and misleading acceptance of non-standard samples. Therefore, automation of the inspection process with a suitable classifier will bring the categorization process to a more professional and accurate level illuminating the probability of biased selection of samples.

This research is organized as: First a quick literature review about previous works and standards used is conducted. Then experimental work that consists of samples preparation and microstructural image capture. The next

sections describe how the images are enhanced and what features are captured from the images. Then in the next sections comes the dataset preparation for SVM and ANN training and the results of the decision making techniques. Last section is concerned with the introduced software that's used to assess GCI sample images based on the trained SVM and ANN.

II. LITERATURE REVIEW

Gray Cast Iron (GCI) is the most commonly used materials out of all other cast irons. GCIs are less ductile and more brittle and hence their usage has been constrained [3]. The morphology and the distribution of graphite grains are the decisive factors in judging the properties of the material cast iron [4].

General Motors DAEWOO Auto & Technology Company (GMDAT) has developed the standard EDS-T-7101 to apply microscopic inspection over GCI microstructures and grade them accordingly. The GMDAT standard presents five classes to discriminate different morphologies of graphite particles referred to five classes from A to E [5]. Details of GMDAT standard pattern classification are illustrated through Fig. 1 and summarized in [6].

Statistical feature extraction using Gray Level Co-occurrence Matrix (GLCM) has been widely tested in previous literatures like in [7]-[9] then ANN was used to distinguish different types. Moreover, there has been a wide interest in using support vector machines as a classifier to identify several metallurgical samples categorization tasks like in [2][10]-[12].

This research adopts using image processing techniques associated with a SVM and ANN to distinguish between the images of GCI and other cast iron types and grade them according to the standards mentioned earlier.

Despite the fact that there are several commercial software available for GCI morphology categorization, a number of drawbacks make such software insufficient for such a task. For instance, user interface for such software is complex enough to be used by non-experts. Moreover, most of the discrimination decisions are based on only image processing techniques, which is not sufficient to assess images correctly.

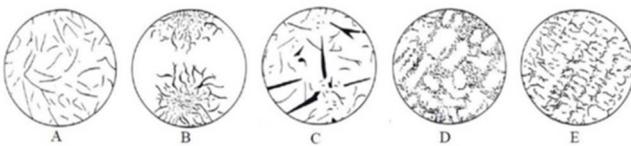


Fig. 1. Different grades of GCI defined by the standards EDS - T - 7101.

Since there's been a wide interest in automation of manual operations that are fully dependent on human's experience and capabilities, microstructure analysis of GCI is held under investigation to introduce a novel software to partially replace human with an intelligent system. Image analysis is one of the major fields that affects the knowledge of materials behavior & performance and has a great contribution in such area. A generic software is to be proposed to include both the manual and the automatic assessment. The manual assessment module is to speed up the manual decision making operation and guarantee unbiased decision. Moreover, the automatic assessment module, based on integrated SVM and ANN decision, is to be used to categorize GCI samples automatically.

III. EXPERIMENTAL WORK

The main objective behind this work is to capture images of different grades of GCI microstructure ready for image analysis. A number of GCI specimens were prepared for microscopic examination. An inverted microscope type INNOVATEST IN-MM600 was used to capture images for further analysis.

Since discrimination of GCI grades is made on unetched surfaces, no etchant was chosen or used. Steps of sample preparation must be professionally handled according to the mentioned standards otherwise; the microstructure of graphite flakes might be affected. This might affect the textural analysis and decision of ANN and SVM later on. Even though the inverted microscope has much more advantages over the light and electron ones, captured images still had several problems that may affect analysis afterwards.

A problem is always initiated in microscopic images where there is remarkable difference in contrast and brightness within the same image. This can be caused by local imperfections within the sample surface. Moreover, distortion in images may occur due to the presence of dust or insufficient illumination. As a result, there was a great need to apply image enhancement techniques to ensure that images are ready for textural analysis afterwards.

IV. IMAGE ANALYSIS AND TEXTURAL ANALYSIS

Since all images were required to be bound by the same levels of contrast and brightness to ensure fair decision about the grade type and categorization. Image enhancement techniques involved contrast adjustment approaches like piecewise linear transformation and logarithmic one. Moreover, Histogram equalization was used as a final enhancement step. An intermediate program (Matlab R2013a) was used to illustrate the image enhancement techniques over captured images.

In order to train a neural network, statistical data was needed to be extracted from images. Textural image analysis was held through a statistical analysis tool; Gray level co-occurrence matrices. Features extracted from such matrices define the texture statistically for further decision making through ANN and SVM as discussed later. The block diagram in Fig. 2 shows the steps used towards feature extraction.

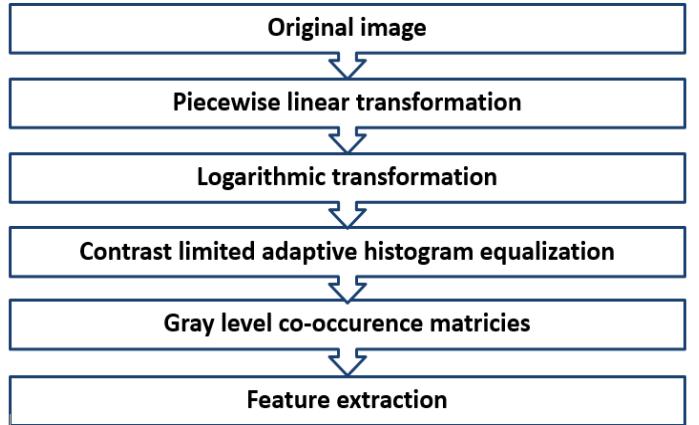


Fig. 2. Block diagram of image enhancement steps to prepare them for feature extraction.

A. Grey Level Co-occurrence Matrices

Gray Level Co-occurrence Matrices (GLCM) algorithm is a statistical one. It studies the relation and effect of grey level value and the probability of occurrence of each pixel and its neighboring one [14]. Through applying some mathematical operations to the different probabilities of occurrences of image intensities, statistical features can be extracted. These features can define and distinguish between textures as introduced by Haralick in [15]. GLCM can be calculated as follows:

$$M_{co} = \sum_{x=1}^k \sum_{y=1}^k \begin{cases} 1 & \text{if } I(x, y) = i \text{ and } I(x + d_x, y + d_y) = j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Where:

K	Image size
I	Image
d	distance between pixels
i	Intensity of pixel (reference)
j	Intensity of pixel (neighbor)

V. FEATURE SELECTION AND EXTRACTION

The variation in grey levels and the probability of its occurrence identifies the nature of the texture. A number of statistical features were introduced by Haralick et al. [15] which can be applied to various image based applications.

A number of previous literatures were investigated in order to obtain a decision on the commonly and widely used features that affects textures in different types of applications. TABLE I shows a summary on the selected research papers and the applied features.

TABLE I. CROSS MAPPING BETWEEN REVIEWED LITERATURES AND THE FREQUENCY OF FEATURES USAGE

<i>F</i> <i>paper</i>	1	2	3	4	5	6	7	8	9	10	11	12	13
[7]													
[16]													
[17]													
[18]													
[14]													
[19]													
[20]													
[21]													
[22]													
[23]													
[24]													
[25]													
[26]													
[27]													
[28]													
[29]													
[30]													
[31]													
[32]													

The features discussed by Harralick and others as follows;
 1) Energy – 2) Contrast – 3) Correlation - 4) Variance - 5) Homogeneity - 6) Average - 7) Sum variance – 8) Sum entropy – 9) Entropy – 10) Difference variance – 11) Difference entropy – 12) Maximum correlation – 13) dissimilarity

Each feature was calculated using GLCM with 1 separating step. Average between features calculated at different orientations was calculated to prepare the dataset used for neural network and support vector machine training afterwards. Commonly used features were a good guide to choosing the most promising features to be applied. TABLE II shows the mathematical relations that governs the chosen features calculation.

TABLE II. SELECTED FEATURES EXTRACTED FROM IMAGES

Feature	Equation
Contrast	$CON = \sum_{i,j}^k i - j ^2 p(i,j)$
Correlation	$COR = \sum_{i,j} \frac{(i - \mu_x)(j - \mu_y)}{\sigma_x \sigma_y} p(i,j)$
Energy	$ENG = \sum_{i,j} p(i,j)^2$
Entropy	$ENR = \sum_{i,j} p(i,j) \log(p(i,j))$
Homogeneity	$HOM = \sum_{i,j} \frac{1}{1 - (i + j)^2} p(i,j)$
Dissimilarity	$DIS = \sum_{i,j} i - j p(i,j)$

Where,

i Intensity of pixel (reference)
j Intensity of pixel (Neighboor)
 σ_x standard deviation horizontally

σ_y standard deviation vertically
 μ_x mean horizontally calculated
 μ_y mean vertically calculated

VI. DATASET PREPARATION

Since both SVM and ANN are supervised learning approaches, they require a relatively large set of data to train the classifier and succeed in categorizing the input images. A carefully prepared dataset was required in order to obtain remarkable results. Two datasets were prepared, one for the SVM while the other for the ANN.

SVM classifier was observed to be able to classify data correctly using only two features. This will reduce processing time and eliminate redundant training data. Out of the six chosen features, dissimilarity and entropy were used for classification after several trials of different combinations between features. Results of dissimilarity and entropy together as input features were the most reliable ones compared to other features.

A dataset for training SVM was created with 300 images mixed between GCI and ductile cast iron (DCI) one ready for training. Testing data was prepared from a different dataset of 40 different GCI images and 200 DCI images (total of 240 images for testing). Inputs for training were dissimilarity and entropy while the decision, output, was the sample is either GCI or ductile cast iron.

As for the ANN, a dataset was prepared with 300 images of different GCI grades and Ductile CI images as well. Inputs for the NN were the six chosen features discussed earlier while the output was grading the percentage of the sample between texture grades from A to E according to experts' eye decision. Since a large dataset was required in order to train a good network, developing a helping tool for experts to submit sample decision was a necessity to facilitate the grading process.

Grey CIMA manual assessment is a module, programmed in MATLAB that helps calculate grades of different GCI images based on the grit method. Single grade Regions of interest are allocated by the user and accordingly its submitted for calculation. This module calculates and displays the percentage of grade appearance based on the GMDAT standards. Validation and details of this manual module are illustrated in details in [33].

VII. SVM TRAINING AND RESULTS

The main objective behind this research was the efficient discrimination of different grades of GCI according to the standards discussed earlier. A drawback was observed in commercial software where it sometimes fails to identify GCI samples from others i.e. it can output a wrong decision on an image of other types of cast irons to be GCI grades. This raised the need of a classifier that distinguishes GCI samples from other cast iron types as an initial step towards GCI grading.

The superiority of the SVM classifier generalization lies in its capability of identifying the maximal margin classifier, that

is, the samples are classified into different categories using the largest boundaries possible. [34].

Since Cover theorem for patterns-separation ability states that “a complex pattern classification problem cast in a high dimensional space non-linearly is more likely to be linearly separable than in a low dimensional space”. Higher order functions are needed to separate non-linear data. The basic SVM classifier deals with data that can be linearly separated. However, there’s a possibility of using it for non-linearly separable data using the Kernel function such as the radial basis or the polynomial functions [34].

The introduced classifier was developed through Matlab statistics and machine learning toolbox. Since the data was not linearly separated, Gaussian radial basis kernel function (RBF) was used and the optimization function was based on quadratic programming.

RBFs have their origins in techniques for performing exact function interpolation where it consists of a feed forward neural network that has one hidden layer and an output layer. This hidden layer is the one responsible for non-linear calculations for the inputs. Moreover, the output layer starts to estimate appropriate values for the targets using linear regression techniques. Although several forms of radial basis may be used, Gaussian kernels are most commonly used [35].

The SVM has shown remarkable results by training over the dataset discussed earlier. Fig. 3 shows the hyper plane surface that separates the training data between GCI and DCI images (0 for GCI and 100 for Ductile CI).

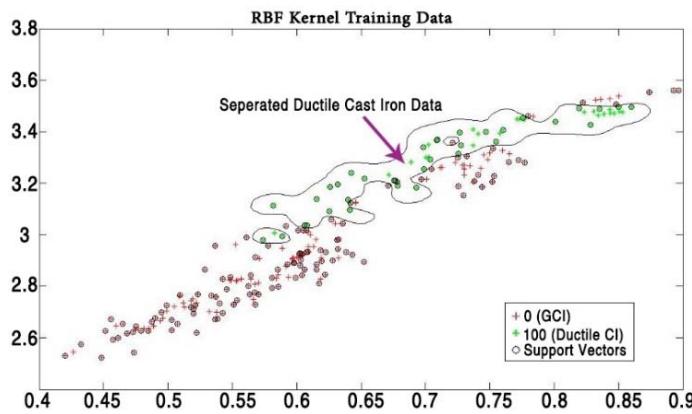


Fig. 3. Training results of SVM classification of ductile and grey cast iron with separating plane.

A. SVM Validation

Testing the SVM has shown 100% efficient classification of ductile cast iron images. As for the GCI, the classifier has failed to identify only one image out of around 40 test images introduced. Fig. 4 shows the test data for both DCI and GCI.

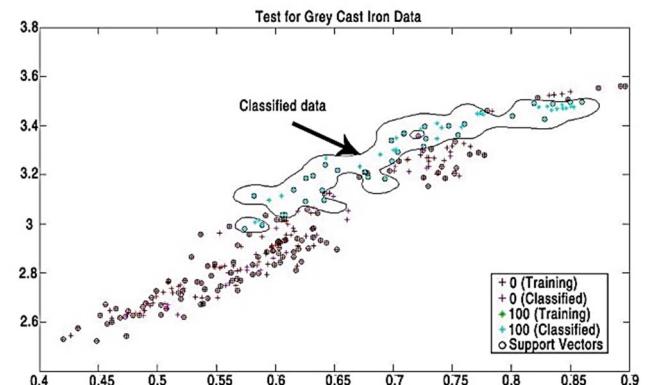


Fig. 4. Test and training results of SVM classification of ductile and grey cast iron.

VIII. ANN TRAINING AND RESULTS

The basic theme behind a neural network design is building a mathematical model by choosing a certain number of neurons that are connected together through different mathematical routes. Each neuron is given a certain weight that identifies the closest existing output. The more the dataset records are, the more learning capability the artificial neural network has. One more advantage of this paradigm is that information is generalized through a number of iterations to guarantee the capability of solving complex problems correctly even if the inputs were not included in the training dataset from the start [36].

Several trials have been conducted over the design and implementation of different neural networks. The best of all these networks was chosen after trying different training functions while varying the number of neurons, hidden layers and nodes.

The trained feed forward neural network had a single hidden layer with 20 neurons supplied. It was based on tan sigmoidal training function for both the hidden and the output layer. The trained ANN had shown regression of around 99.7% in training of samples while in test data, regression became 94.9% as shown in Fig. 5.

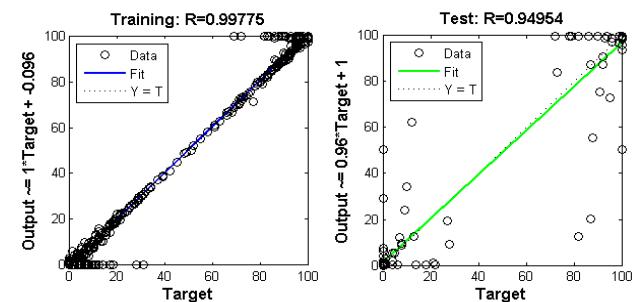


Fig. 5. Regression of both training and test data of ANN.

Errors histogram in Fig. 6 shows remarkable results where only high repetition of errors was found in values of -3.37 and 3.558, which did not affect the decision in a harmful manner. On the other hand, other values of errors occurred in very rare cases that can be neglected.

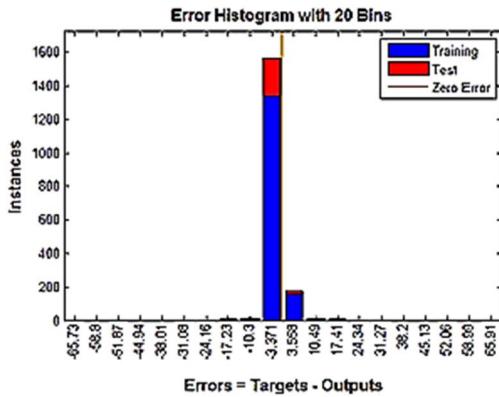


Fig. 6. Errors histogram results of the ANN.

A. ANN Validation

Validation of NN output was carried out using two datasets. One approach was using 15% of the dataset already proposed for training. Every time training was carried out, data was randomized and therefore, the validation dataset in this approach was not fixed. Other approach was a dataset of 40 new images not involved in any of the training steps of the neural network to be under test for neural network validation.

These datasets were both used for results comparison between Gray CIMA manual Module (introduced by the metallurgist's decision), Gray CIMA automatic module (based on the results of the neural networks) and the commercial software associated with the microscopic setup (Metavision).

ANN has an advantage over other decision-making techniques is that it can be retrained on a wider range of data. The more size and variability of the data supplied to the training dataset, the higher the accuracy of results. Therefore, one can retrain same designed networks with a wider range of data and obtain better results.

It was observed that there is a significant difference between the output of Gray CIMA and that of Metavision. Using the validation dataset, results of assessment of both the automatic gray CIMA module and that of Metavision were compared to the manually assessed images by expert using the manual module in gray CIMA. A sample of 13 images is shown with details of decision using Gray CIMA and Metavision in TABLE III. It is observed that Metavision software is always oriented to give a value at grade A (the most desirable in industrial applications) and the rest is a mixture without defining it. On the other hand, the automatic module of Gray CIMA has successfully distinguished the difference between grades with less error rates than that of Metavision.

Since the commercial software is only decisive to cover grade A, comparison was made between the assessment of only grade A using the three modules. Fig. 7 shows a bar chart that maps the decision of the three modules together. Its

observed that Metavision have already assessed 6 samples in a wrong way out of the 13 samples which means that 46% of the decisions were wrong compared to the manual assessment. On the contrary, the automatic gray CIMA has assessed one sample only out of the 13 samples in a wrong way, which means that 7.5% of the decisions were wrong compared to that of the manual assessment.

TABLE III. COMPARISON BETWEEN DECISION OF BOTH MODULES OF GRAY CIMA AND THE COMMERCIAL SOFTWARE METAVISION

Image	Gray CIMA [manual by Expert]					Gray CIMA [Automatic]					Metavision [Commercial software]					
	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	Mix
1418.bmp	0	0	0	73	27	0	0	0	60	40	53.2	0	0	0	0	46.8
1434.bmp	81	0	19	0	0	88	0	12	3	0	87.5	0	0	0	0	12.5
1446.bmp	94	0	6	0	0	96	0	4	0	0	88	0	0	0	0	12
1463.bmp	0	0	18	82	0	9.3	0	8	78.5	3.6	86.5	0	0	0	0	13.5
1611.bmp	0	0	0	100	0	0	0	0	90	10	67.2	0	0	0	0	32.8
1618.bmp	0	0	0	95	5	0	0	0	89	11	67.3	0	0	0	0	32.7
1637.bmp	0	0	0	38	62	0	0	0	36	64	58.3	0	0	0	0	41.7
1667.bmp	0	0	0	23	77	0	0	0	20	80	63.8	0	0	0	0	36.2
1692.bmp	95	0	5	0	0	94	0	5	0	1	87.3	0	0	0	0	12.7
1699.bmp	93	0	7	0	0	95	0	5	0	0	91.8	0	0	0	0	8.2
1710.bmp	100	0	0	0	0	95	0	3	0	2	87.8	0	0	0	0	12.2
1715.bmp	96	0	4	0	0	93	0	7	0	0	92.1	0	0	0	0	7.9
1728.bmp	99	0	1	0	0	96	0	0	0	4	77.5	0	0	0	0	22.5

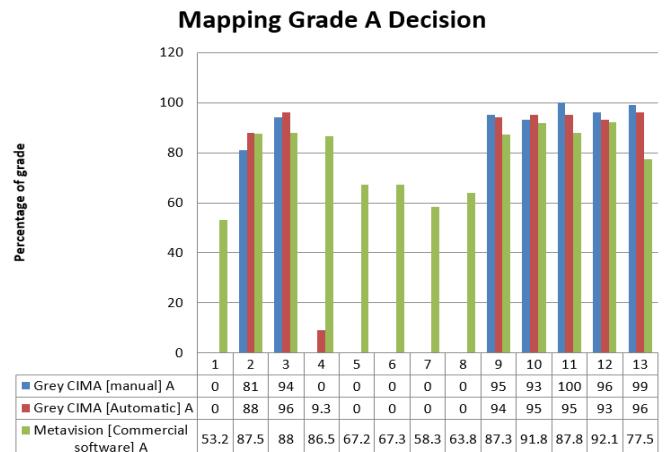


Fig. 7. Mapping decision of grade A between Metavision and both modules of Gray CIMA (Manual and automatic).

IX. GRAY CIMA: GREY CAST IRON MULTI-ASSESSMENT SOFTWARE

Gray CIMA has focused on the three main problems with the commercial software. One is the complexity of its interface where needs full knowledge from user of image enhancement techniques and expressions. The other two drawbacks are concerned with the assessment results. First, the automatic assessment module is always biased to measure the value of "Grade A" presence and therefore, it sometimes judges the grade A existence although it might be absent in the introduced image. This leads to a decision completely far away from reality measured by metallurgist. The other drawback lies in the decision of distinguishing the GCI image from the start. Even though the module is made for graphite flakes assessment, lack of knowledge of the user may let them introduce another type of cast iron image for assessment. The commercial software still observes a percentage of grade A and does not warn the user that this is not GCI. This may mislead the decision of the metallurgist if the analysis decision was based on the automatic software.

A. The user interface

The user interface was developed to be easy for use without having several drop-down menus as most commercial software. The interface contains two modules; the manual one to the left as shown in Fig. 8 and the automatic one to the right. The manual module is the one concerned with helping the metallurgist in measuring the grades areas according to the grid method mentioned in the standards. As discussed earlier, this module was used to generate the decisions in the dataset used in training the ANN. The manual module is discussed in detail in [33]. The automatic module is the one concerned with the trained SVM and ANN decision.

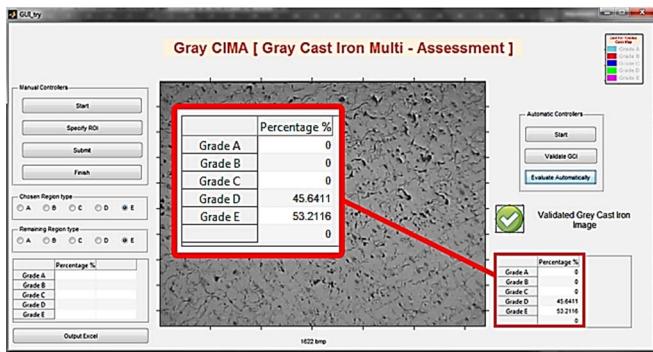


Fig. 8. User interface of Gray CIMA.

B. The Automatic Module

The automatic module has two main concerns. One is to ensure that the image is GCI, which is based on SVM decision while the other is to grade the submitted image between grades A and E, which is based on ANN decision.

The user loads an image by hitting start button. Pressing the “validate GCI” button will give out a reading of either the image is GCI or not. Then hitting the “Evaluate automatically” button will show percentages of grades available in the image loaded. Fig. 8 shows a sample of a loaded image that is validated to be GCI and grading results are shown in details after automatic assessment.

X. CONCLUSION

This research introduces a software that is concerned with automatic discrimination of different grey cast iron grades. The software decides the microstructural category using SVM and ANN approaches. Features distinguishing microstructural types, identified according to GMDAT standards, were based on GLCM statistical approach. Manual identification of these microstructures was held by an expert using the grit method based module. SVM and ANN were trained and tested using separate validation datasets. Tests have successfully shown that the AVM has identified DCI with 100% efficiency while GCI was identified with a percentage of error of about 2.3%. As for the grading section, ANN tests have proven 98% efficient identification of the submitted samples with 3-4% error. The results were put into comparison and found superior to that of the commercial software associated with the microscopic setup; Metavision. The proposed software is a successful add-on to the development routes of automating the process of microstructural characterization of GCI. Such task

can be hard, time consuming, costly and sometimes inaccurate if assessment was only based on manual operators.

REFERENCES

- V. Albuquerque, J. M. R. S. Tavares, and P. Cortez, “Quantification of the Microstructures of Hypoeutectic White Cast Iron using Mathematical Morphology and an Artificial Neuronal Network,” International Journal of Microstructure and Materials Properties, vol. 5, no. 1, pp. 1–19, 2010.
- J. P. D. A. V. H. C. F. A. X. Papa and J. M. R. Tavares, “Fast Automatic Microstructural Segmentation of Ferrous Alloy Samples using Optimum-Path Forest,” Computational Modeling of Objects Represented in Images .Springer Berlin Heidelberg, pp. 210–220, 2010.
- S. N. Gurugubelli, “Experimental Investigations On Structure And Mechanical Properties Of Semi Solid Casting Of Gray Cast Iron,” INTERNATIONAL JOURNAL OF ADVANCED SCIENTIFIC RESEARCH AND TECHNOLOGY, vol. 3, no. 2, pp. 190–193, 2012.
- P. Prakash, V. D. Mytri, and P. S. Hiremath, “Comparative Analysis of Spectral and Spatial Features for Classification of Graphite Grains in Cast Iron,” International Journal of Advanced Science and Technology, vol. 29, no. April, pp. 31–40, 2011.
- O. da F. M. Gomes and S. Paciornik, “Automatic Classification of Graphite in Cast Iron,” Microscopy and Microanalysis, vol. 11, no. 4, pp. 363–371, 2005.
- H. Jiang, Y. Tan, J. Lei, L. Zeng, Z. Zhang, and J. Hu, “Auto-analysis system for graphite morphology of grey cast iron,” Journal of Automated Methods & Management in Chemistry, vol. 25, no. 4, pp. 87–92, 2003.
- P. K. N. Rode, “Analysis of ms using glcm,” Engineering and Technology, vol. 1, no. 5, pp. 121–126, 2012.
- S. Jain, “Brain Cancer Classification Using GLCM Based Feature Extraction in Artificial Neural Network,” Methodology, vol. 4, no. 07, pp. 966–970, 2013.
- S. D. M. Raja and A. Shanmugam, “ANN and SVM Based War Scene Classification Using Invariant Moments and GLCM Features: A Comparative Study,” Machine Learning, vol. 2, no. 6, pp. 869–873, 2012.
- J. P. Papaa, R. Y. M. Nakamuraa, V. H. C. de Albuquerque, A. X. Falc~Aoc, Jo~Tavaresd, and M. R., “Computer techniques towards the automatic characterization of graphite particles in metallographic images of industrial materials,” Expert Systems with Applications, 2012.
- K. Roberts, G. Weikum, and F. M\"ucklich, “Examinations on the Automatic Classification of Lamellar Graphite Using the Support Vector Machine,” Practical Metallography, vol. 42, no. 8, pp. 396–410, 2003.
- E. El-sebakhy, “Extreme learning machine as a new framework in predicting material properties: Methodology and comparison,” The 12th International Conference of International Association for Computer Methods and Advances in Geomechanics (IACMAG), no. 2002, pp. 1–6, 2008.
- C. Tanchotsrinon, S. Phimoltares, and C. Lursinsap, “An autonomic building detection method based on texture analysis, color segmentation, and neural classification,” in 5th International Conference on Knowledge and Smart Technology (KST). IEEE, 2013, pp. 162–167.
- C. N. Rao, S. S. Sastry, K. Mallika, H. S. Tiong, and K. B. Mahalakshmi, “Co-Occurrence Matrix and Its Statistical Features as an Approach for Identification Of Phase Transitions Of Mesogens,” International Journal of Innovative Research in Science, Engineering and Technology, vol. 2, no. 9, pp. 4531–4538, 2013.
- K. S. I. D. Robert M.Harralick, “Textural Features for Image Classification,” IEEE Trans. on Systems, Man and Cybernetics, SMC-3(, vol. 6, pp. 610–621, 1973.
- D. A. Clausi, “An analysis of co-occurrence texture statistics as a function of grey level quantization,” Design, vol. 28, no. 1, pp. 45–62, 2002.
- R. O. K. Reddy, “Classifying Similarity and Defect Fabric Textures based on GLCM and Binary Pattern Schemes,” Engineering, no. November, pp. 25–33, 2013.

- [18] L. B. Mostaço-guidolin, A. C. Ko, F. Wang, B. Xiang, M. Hewko, G. Tian, A. Major, M. Shiomi, and M. G. Sowa, "Collagen morphology and texture analysis: from statistics to classification," pp. 1–10, 2013.
- [19] R. Quevedo, E. Valencia, J. M. Bastías, and S. Cárdenas, "Description of the enzymatic browning in avocado slice using GLCM image texture," Change, 2011.
- [20] D. J. Bermejo, "High definition video quality assessment metric built upon full reference ratios," asasa, 2012.
- [21] D. GADKARI, "IMAGE QUALITY ANALYSIS USING GLCM," 2004.
- [22] A. Ali, X. Jing, and N. Saleem, "GLCM-BASED FINGERPRINT RECOGNITION ALGORITHM," Broadband Network and Multimedia Technology (IC-BNMT), pp. 207–211, 2011.
- [23] I. Texture and F. Extraction, "MSc THESIS Implementing Texture Feature Extraction Algorithms on FPGA," Electrical Engineering, 2009.
- [24] J. M. Flores, "Texture Characterization and Analysis," Analysis, 2010.
- [25] R. W. Conners, M. M. Trivedi, and C. A. Harlow, "Segmentation of a High-Resolution Urban Scene Using Texture Operators *," Image Processing, pp. 273–310, 1984.
- [26] V. S. Thakare, N. N. Patil, and J. S. Sonawane, "Survey On Image Texture Classification Techniques," International Journal of Advancements in Technology, vol. 4, no. 1, pp. 97–104, 2013.
- [27] L. Collini, G. Nicoletto, and R. Kone, "Microstructure and mechanical properties of pearlitic gray cast iron," Materials Science and Engineering A, vol. 488, pp. 529–539, 2008.
- [28] A. Gebejes, E. M. Master, and A. Samples, "Texture Characterization based on Grey-Level Co-occurrence Matrix," Artificial Intelligence, pp. 375–378, 2013.
- [29] A. Rampun, H. StrStrange, and R. Zwigelaar, "Texture Segmentation Using Different Orientations of GLCM Features," Image (Rochester, N.Y.), 2013.
- [30] G. A. B, M. C. Chandrashekhar, and M. Z. Kurian, "Texture Feature Extraction of Video Frames Using GLCM," International Journal of Engineering, vol. 4, no. 6, pp. 2718–2721, 2013.
- [31] D. B. Biswajit Pathak, "Texture Analysis Based on The Gray Level Cooccurrence Matrix Considering Possible Orientations," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, pp. 4206–4212, 2013.
- [32] T. A. Pham, "Optimization of Texture Feature Extraction Algorithm," Electrical Engineering, 2010.
- [33] A. Khaled, M. R. A. Atia, and T. Moussa, "Grey Cast Iron Categorization using Artificial Neural Network," International Journal of Scientific & Engineering Research, vol. 5, no. 3, 2014.
- [34] M. S. Reis and A. Bauer, "Image-based classification of paper surface quality using wavelet texture analysis," Chemical Engineering, vol. 34, no. 2010, pp. 2014–2021, 2014.
- [35] S. H. Paul Yee, "A Dynamic Regularized Radial Basis Function Network for Nonlinear, Nonstationary Time Series Prediction," IEEE TRANSACTIONS ON SIGNAL PROCESSING,, vol. 47, no. 9, pp. 2501 – 2521, 1999.
- [36] V. H. C. D. Albuquerque, P. C. Cortez, and A. R. D. Alexandria, "A New Solution for Automatic Microstructures Analysis from Images Based on a Backpropagation Artificial Neural Network," Non-Destructive testing and evaluation, 2008.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/344028755>

Region of interest-based image retrieval techniques: a review

Article · May 2020

DOI: 10.11591/ijai.v9.i3.pp520-528

CITATIONS

0

READS

701

3 authors, including:



Nasharuddin Zainal
Universiti Kebangsaan Malaysia

77 PUBLICATIONS 491 CITATIONS

[SEE PROFILE](#)



Shahrizan Jamaludin
Universiti Malaysia Terengganu

18 PUBLICATIONS 91 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Speech production of prelingually deaf children with cochlear implants [View project](#)



Human/Pedestrian Detection and Tracking for Surveillance [View project](#)

Region of interest-based image retrieval techniques: a review

Mardhiyah Md Jan¹, Nasharuddin Zainal², Shahrizan Jamaludin³

^{1,2}Department of Electrical, Electronic & Systems Engineering, Universiti Kebangsaan Malaysia, Malaysia

³Department of Maritime Technology, Universiti Malaysia Terengganu, Malaysia

Article Info

Article history:

Received Apr 18, 2020

Revised Jun 10, 2020

Accepted Jun 26, 2020

Keywords:

Image processing

Image retrieval system

Region of interest (ROI)

Region-based image retrieval
(RBIR)

ABSTRACT

This paper presents a review of the region of interest-based (ROI) image retrieval techniques. In this study, the techniques, the performance evaluation parameters, and databases used in image retrieval process are being reviewed. A part of an image that is considered important or a selected certain area of the image is what defines a region of interest. Retrieval performance in large databases can be improved with the application of content-based image retrieval systems which deals with the extraction of global and region features of images. The capability of reflecting users' specific interests with greater accuracy has shown to be more effective when using region-based features compared to global features. Segmentation, feature extraction, indexing, and retrieval of an image are the tasks required in retrieving images that contain similar regions as specified in a query. The idea of the region of interest-based image retrieval concepts is presented in this paper and it is expected to accommodate researchers that are working in the region-based image retrieval system field. This paper reviews the work of image retrieval researchers in the span of twenty years. The main goal of this paper is to provide a comprehensive reference source for scholars involved in image retrieval based on ROI.

This is an open access article under the CC BY-SA license.



Corresponding Author:

Mardhiyah Md Jan,
Department of Electrical, Electronic & Systems Engineering,
Faculty of Engineering & Built Environment,
Universiti Kebangsaan Malaysia,
43600 Bangi, Selangor, Malaysia.
Email: P100114@siswa.ukm.edu.my

1. INTRODUCTION

The rapid expansion of the internet and fast advancement in colour imaging technologies have made digital colour images more readily obtainable to professional and amateur users [1]. The medical field, surveillance system, and digital forensics are among the areas that have been using a vast amount of multimedia data in the form of audio, video, and images in this advance development of internet and multimedia technologies. This situation leads to the need for developing a system that is able to store and efficiently retrieve these digital data [2].

The main goal of an image retrieval system is to search and retrieve images from a variety and large databases in minimal time with high accuracy [3]. Two of the techniques that are generally used to search and retrieve images in the database are the Text-based Image Retrieval (TBIR) and Content-based Image Retrieval (CBIR). TBIR works by using retrieval keys which include classification codes, keywords or subject headings in retrieving images. TBIR is considered a non-standard technique of retrieving images due to the inconsistency of interpretation of keywords used by users.

However, TBIR is the most common retrieval technique that is being used, where the search is based on the explanation of the images. Normally a TBIR works by running a database search for similarity text surrounding the images as given by the query string of any keyword used by users. At present, Google Images is the most frequently used TBIR system. The string matching requires less computation time making the TBIR system work fast. Nevertheless, there is also a disadvantage of TBIR where it is at times difficult to describe the entire graphic content of images in word form. This factor may lead to the TBIR system to produce irrelevant results when querying. Some annotations of the images may also be incorrect and this will consume a lot of time for the TBIR system to yield the desired output. TBIR can be successfully defined as a document retrieval problem [4].

CBIR systems were developed to overcome the limitations met by the TBIR systems. CBIR is one of the instances of information retrieval that applies computer vision techniques to solve searching and managing large image databases related problems [5]. A CBIR uses graphic contents of a certain image defined by colour texture, shape, and spatial location that are identified as the low-level features to represent the image in the database. When a desired image is being used as an input in the system, it will retrieve a similar image matched to the example image provided. By querying using CBIR systems it will eliminate the requirement for expressing the graphic content images in words form that resembles the human perception of visual data. Among the CBIR systems that are commonly available are QBIC [6], Photobook [7], VisualSeek [8], Netra [9] and SIMPLICITY [10].

There have been extensive studies done on CBIR and the progress has been discussed comprehensively in [11-13]. An efficient shape descriptor is essential to identify the features of an image content, including its shape, colour and texture [14-15]. Region-based Image Retrieval (RBIR) system is recognized as one of the categories of the CBIR system. Image segmentation and defining measurement criteria constitute the ROI-based image retrieval system [16]. This system works by producing the representation of the image in the database by utilizing the features extracted from the region or part of the image.

RBIR systems can be categorized into System Designated Region-of-Interest (SDR) [17] and User Designated Region-of-Interest (UDR) [18] approaches. When querying to the database using the SDR approach, dividing the image into significant regions and designating every region as ROI is done by the system automatically. Meanwhile, by using the UDR method, the selection of ROI in the image for query formulation is done manually by the user.

One of the factors to determine the success of an SDR method is the accuracy of the segmentation technique used in dividing an image into many regions. However, unpredicted noise in the output may result in the reduction of retrieval accuracy thus making image segmentation not always reliable. Inquest to identify either the boundaries of the regions of the objects in the image are the two diverse yet complementary perspectives that are usually being used in image segmentation [19]. Furthermore, many existing segmentation techniques fail to extract objects of interest despite their ability to accurately identify specific regions from images.

The SDR method has some limitations in reflecting the user's goal in the process of retrieval due to these reasons. It is complicated to decide beforehand the part of the image that will be chosen when the user opts for the ROI using the UDR approach manually. Extraction of the images' feature values and matching them with the ROI for retrieval is done by dividing the image into a smaller number of blocks is the solution to this problem. Correct selection of blocks overlying the ROI is a crucial decision since the UDR method is able to have a variety of sizes and can have multiple blocks in order to satisfy the user query precisely [20].

ROI placement also plays a significant role in obtaining an effective ROI image retrieval [21]. Fixed location matching will be the result in the case of blocks having the exact placement as the ROI. However, this method could not retrieve the same images when similar regions to ROI lie in other parts of the images. For instance, the system fails to produce the exact images containing an elephant that is situated on the top-left corner of an image when initially the user who queries for an elephant in the bottom-right corner of the block image is being used. Opting for all blocks matching strategy might solve this problem, but using this approach will lead to the time complexity that needed to be compensated as it increases the computational complexity and time as the blocks increase in dimension layout. This paper presents the review of region of interest-based image retrieval with various of retrieval techniques used and the findings obtained.

2. IMAGE PROCESSING METHODS

Image processing generally relies on the specific feature of an image and matching it to any specific object. This method is performed to extract useful information from the images. Digital image processing focuses on improving pictorial information for human interpretation and processing of image data for transmission and storage. Recognition of an object is a very significant challenge in image processing.

Since every object is a part of an image, thus, matching ROI can also be considered as an object of a particular image. There are several operations involved in image processing methods such as representation, segmentation, clustering, feature extraction, and matching. These operations will further be discussed below.

2.1. Image representation

There are a lot of images available in the digital world today, however, the actual representation of an image is denoted as a collection of discrete picture element called as the pixels. Pixel values are most often grey levels in the range of 0 until 255 (black-white). Binary is the image representation format usually used and later will be translated into different image formats such as jpg, bitmap, and png [22-23]. These image representations rely on the pixel size and these images are represented by pixels similar to building blocks. In order to reduce the complexity and computational time of image processing, a high-quality image is required [24].

2.2. Image segmentation

The definition of image segmentation in computer vision is given as the process of segregating a digital image into many segments. Image segmentation partitions the image pixels into groups that strongly correlate with the objects in an image. The aim of segmentation is to simplify the image into a representation that is easier to analyse and more meaningful [19, 25]. In the image retrieval field, the segmentation process is still considered an open problem in computer vision despite some of the systems that can deliberately be very efficient. By only segmenting the region required by the user instead of having to do the segmentation process for the whole image, the ROI concept that is used can reduce the computation time since the segmented region is smaller compared to the entire image.

2.3. Region clustering

In terms of features present, there might be features that are quite similar in many regions that belong to a different set of images [26]. A training data and a codebook size are produced by getting the average of efficacy and accuracy of the retrieval based on the features that are obtained from the regions of the whole database images. All the data and details are computed then stored as a codeword. The cluster is recognized and will be registered in the codebook and the matching index will be stored for each region of an image in the database used while removal will be done to the original feature of that particular region. The nearest entry in the codebook is located and the matching index will be replacing its feature for a new image region. Region clustering rely mainly on the assumption that the pixels of one region is similar with the neighbouring pixels.

2.4. Feature extraction

Feature extraction is important since it is the process of obtaining global image features such as colour or local descriptors like texture and shape of a particular image. In this phase, an identity is given for each character which is represented by a feature vector [27]. When being compared to dimensional histograms, these representations are considered more efficient in terms of retrievals. The most prominent features of the image only will be extracted after feature extraction analysis has been done by the system [28]. One of the key attributes of an image region that has been segmented is the shape of the region and it is a crucial role in representing the effectiveness of a robust image retrieval system [29]. The aim of feature extraction is to extract some sets of features in order to minimize the image retrieval rate.

2.5. The general framework of RBIR

User-dependent retrieval and system-dependent retrieval are two of the most commonly used methods that exist in region-based image retrieval. In the UDR approach, features of the blocks in the layout that the user selects will be the region's features that the system will extract. Meanwhile, by using the SDR approach, regions or objects of the image will be the potential ROIs based on the segmentation algorithm. The obtained features will then be utilized for matching purposes to produce results of the image retrieval and producing database images of the region selection [25]. The RBIR works by filtering out the candidate image and calculating the similarity with the query using the image database which will lead to significant improvement in retrieval speed [30].

3. RELATED WORKS

Most of the literature on ROI-based image retrieval considers a variety of issues that can be solved. Many authors have contributed their works in the ROI area and some of them are listed below:

Chen et al. [31], discussed a new approach to index image in content-based image retrieval by including Kohonen's Self Organizing Map (SOM), one of the neural network model available. They stated that the significant contribution of this approach is that it is interactive, where the user is able to choose an ROI desired from the sample images before the system focuses on the query matching result of the corresponding regional colour features, thus producing the region that contains the same regions as required by the user. The capability of the system to partition every image into many standardize regions for indexing and representing the image adaptively is possible using the SOM algorithm. Figure 1 depicts the output of the system.

The users are the more suitable person to specify the content of the query and not the computer itself, thus Moghaddam et al. [32] presented another alternative image retrieval system based on this opinion. The users are able to choose multiple ROI and can specify the relevance of their spatial layout in the retrieval process by using this proposed system. The similarity bounds on histogram distances for reducing the database search is also derived in this discussion. An example of multiple ROI queries using an MRI database can be seen in Figure 2.

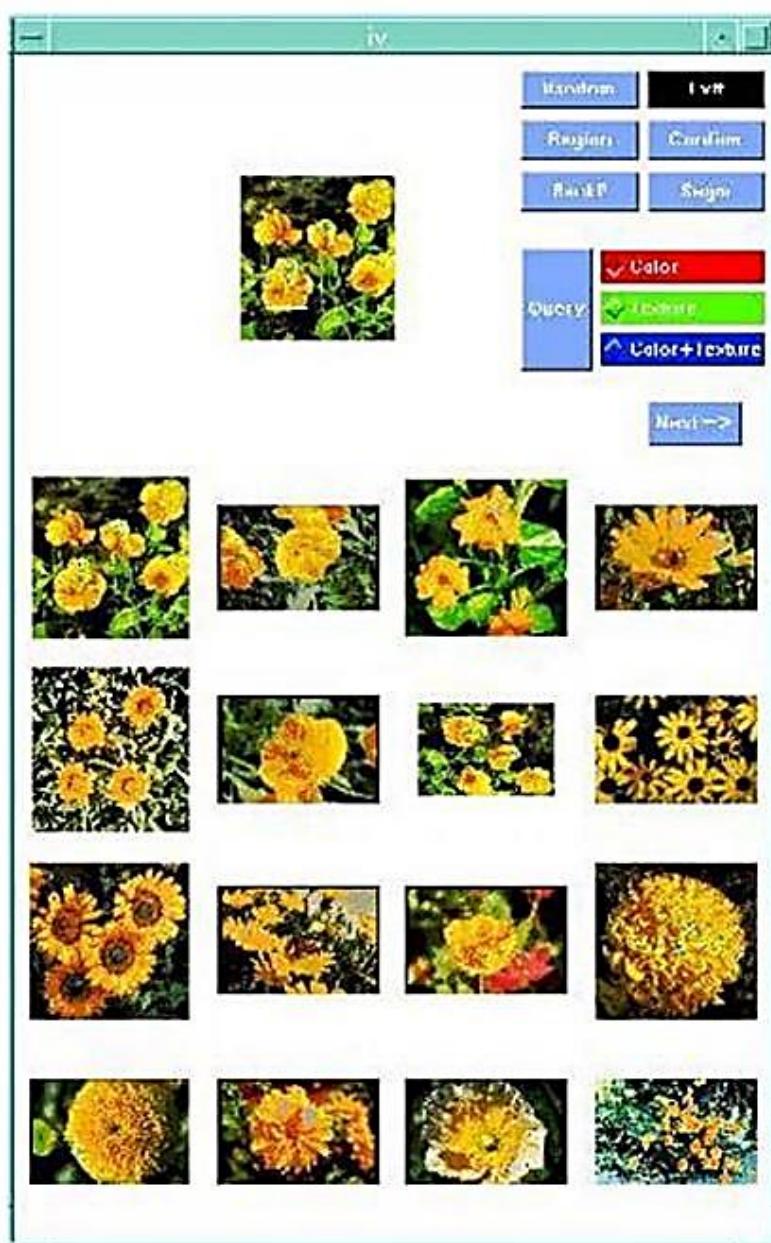


Figure 1. An example of retrieved images containing flowers with yellow colour from the database using [31]

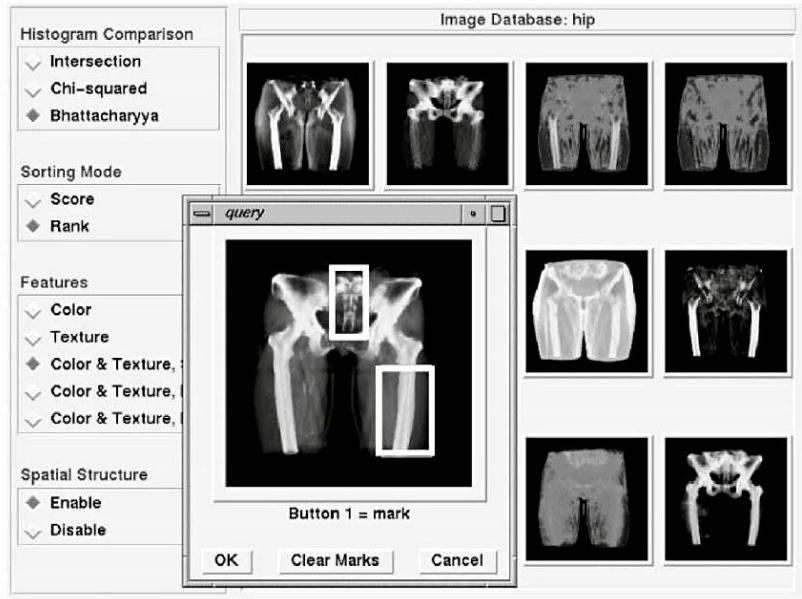


Figure 2. An example of an MRI imagery database using multiple ROI query using the method by [32]

Another technique is the region-based image retrieval system using a high-level semantic learning idea that has been suggested by [33]. The significant feature of this system is its ability to process both query by ROI and query by keyword. The image will be partitioned into several regions and the system will extract the regions with low-level features. High-level features are then obtained using the suggested decision tree-based learning algorithm named decision tree-based image semantic learning algorithm, DT-ST. The examples of query and dominant region using the proposed system are portrayed in Figure 3.

Combining the text-based retrieval system using a unique region-based inverted file indexing method with the semantic image retrieval model is the proposed idea discussed by [34]. By using this method, images were interpreted into textual documents which are then being indexed and retrieved, similar to the conventional text-based query process. Two examples of the output of this proposed retrieval system are shown in Figure 4.

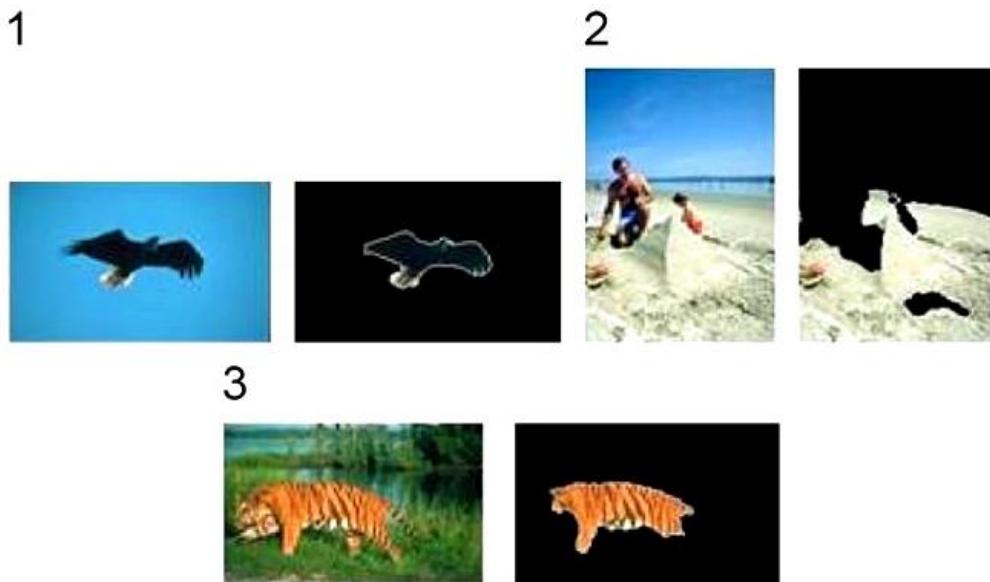


Figure 3. Dominant region query image examples obtained from [33]



Figure 4. The examples of co-occurring objects in images being retrieved using [34]

4. SUMMARIZED LITERATURE REVIEW

The properties used in 15 articles related to region-based Image Retrieval systems reviews in this paper are listed in Table 1. The properties are classified into the concept, performance evaluation parameter, the database used, and findings by concerned authors.

Table 1. Summarized literature survey of region-based image retrieval techniques (*continue*)

Authors/Year	Technique	Performance Evaluation Parameter	Databases	Findings
T. Chen, L.-H. Chen, K.-K. Ma [31] (1999)	Region-of-interest (ROI) image retrieval based on colour image indexing	Precision of image retrieval	500 various contents images	High precision in results of image retrieval. Rotated and scaled versions of images can also be detected by using the proposed method.
Qi Tian, Ying Wu, Thomas S. Huang [35] (2000)	Combining user-defined region of interest (ROI) and spatial layout Content-based Image Retrieval	Accuracy of image retrieval	COREL database images	More capable of capturing image details compared to the global approach.
Baback Moghaddam, Henning Biermann, Dimitris Margaritis [32] (2001)	Multiple region-of-interest (ROI) and spatial layout for Content-based Image Retrieval	Accuracy and speed of image retrieval	VisTex and GIS orthophoto Imagery database images	The proposed method can reduce computational time and speed-up the retrieving possibility based on partial spatial configuration.
Feng Jing, Ming Jing Li, Lei Zhang, Hong-Jiang Zhang, Bo Zhang [36] (2003)	Learning in region-based image retrieval	Accuracy of image retrieval	10 000 COREL database images	Shows the effectiveness of retrieval after comparing results using the learning schemes for region-based image retrieval with database images.
Khanh Vu, Kien A. Hua, Wallapak Tavanapong [37] (2003)	Region-of-interest (ROI) queries based on sampling matching	Robustness and speed of image retrieval	Art Explosion commercial database images	The user-defined region query can overcome irrelevant region retrieved results. Fast retrieval when the indexing technique is used.

Table 1. Summarized literature survey of region-based image retrieval techniques

Authors/Year	Technique	Performance Evaluation Parameter	Databases	Findings
B. G. Prasad, K. K. Biswas, S. K. Gupta [38] (2004)	Region-based Image Retrieval using integrated colour, shape and location index	Efficacy of image retrieval	MPEG-7 database images	The proposed method is robust towards scaling and rotation invariance
Feng Jing, Ming Jing Li, Hong-Jiang Zhang, Bo Zhang [39] (2004)	Region-based image retrieval using relevance feedback	Precision of image retrieval	10 000 COREL Stock Photo Library	Integrating the relevance feedback algorithm into a region-based image retrieval system results in a more precise output of the image retrieved.
Byoung Chul Ko, Hyeran Byun [40] (2005)	Semantic image segmentation using adaptive circular filters based on Bayes' theorem and texture distribution	Region precision of image retrieval	COREL-photo CD images	The proposed method able to accommodate region scaling, rotation, and translation.
Qiang Zhou, Limin Ma, Mehmet Celenk, David Chelberg [41] (2005)	Content-based image retrieval based on region-of-interest (ROI) detection and relevance feedback	Accuracy of image retrieval	2500 natural colour images	The proposed method shows promising results compared to the global feature-based image description.
Jing Zhang, Choong-Woong Yoo, Seok-Wun Ha [42] (2007)	Region-of-interest (ROI) based image retrieval by combining colour and texture feature	Precision and speed of image retrieval	300 nature images (http://www.cs.washington.edu/research/imagedatabase/)	The proposed method improves in accuracy and speed of image retrieval.
Ying Liu, Dengsheng Zhang, Guojun Lu [33] (2008)	Region-based image retrieval using decision tree learning	Accuracy of the image retrieval method	5000 COREL database images	The proposed method displays a high accuracy of retrieval.
Yung-Kuan Chan, Yu-An Ho, Yi-Tung Liu, Rung-Ching Chen [43] (2008)	Colour Variances among Adjacent Objects (CVAAO)	Precision of image retrieval Robustness in resisting image invariance	Full-colour images	CVAAO feature can describe pixel colour and colour complexity of an image. Able to precisely sever the target region image The method is insensitive towards scale, shift, rotation and distortion variance.
Dengsheng Zhang, Md Monirul Islam, Guojun Lu, Jin Hou [34] (2009)	Semantic image retrieval using region base inverted file	Accuracy of image retrieval	7600 COREL database images	The proposed method presents higher retrieval performance compared to traditional content-based image retrieval.
Chaobing Huang, Quan Liu, Shengsheng Yu [44] (2011)	Region-of-interest (ROI) extraction based on visual saliency	Accuracy of ROI extraction	200 SIMPLICITY database images	Discrete Moment Transform (DMT)-based saliency can determine large ROI.
Zhe Wang, Guizhong Liu, Yang Yang [45] (2012)	Region-of-interest (ROI)-based image retrieval system using an auxiliary Gaussian weighting scheme (AGW)	Precision of retrieval system	100 000 Google Images	High precision retrieval performance However, the proposed method shows inaccuracies towards scaling and rotating image invariances.

5. CONCLUSION AND FUTURE WORKS

This paper categorizes the various techniques and examples used in region of interest-base image retrieval. A collection of papers was studied and numerous concepts and methods are classified in the given table. Several results of the proposed methods by authors are being presented in the illustrated figures. In the image retrieval field, the development of inexpensive, rapid and robust retrieval systems are among the important factors to be considered. Hence, an immense range of applications can gain benefits from these image retrieval technologies. There are countless field that uses these RBIR techniques applications such as medical imaging, archaeology, zoology, and criminal investigation. Overall, the objective of this study is to provide comprehensive reference for researchers in similar research area has been achieved. In conclusion, there is a lot of research yet to be done to improve the performance used in the current technology of RBIR and to expand the application in many more fields.

ACKNOWLEDGEMENTS

This research activity had been conducted in the Vision@UKM laboratory and is supported by the Ministry of Education (MOE) Grant fund: FRGS/1/2018/TK04/UKM/02/16.

REFERENCES

- [1] W. Zhou, H. Li, and Q. Tian, "Recent Advance in Content-based Image Retrieval: A Literature Survey," arXiv Prepr. arXiv1706.06064, vol. 2, pp. 1-22, 2017.
- [2] G. Qiu, "Color image indexing using BTC," *IEEE Trans. Image Process.*, vol. 12, no. 1, pp. 93-101, 2003.
- [3] V. Tyagi, "Content-Based Image Retrieval Techniques : A Review," in *Content-Based Image Retrieval*, Singapore: Springer, 2017, pp. 29-48.
- [4] J. Yu, X. Yang, F. Gao, and D. Tao, "Deep Multimodal Distance Metric Learning Using Click Constraints for Image Ranking," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4014-4024, 2017.
- [5] N. T. L. Mai, S. S. Bin Ahmad Ridzuan, and Z. Bin Omar, "Content-based image retrieval system for an image gallery search application," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 3, pp. 1903-1912, 2018.
- [6] C. Faloutsos et al., "Efficient and effective Querying by Image Content," *J. Intell. Inf. Syst.*, vol. 3, no. 3-4, pp. 231-262, 1994.
- [7] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook-Pentland.Pdf," *Int. J. Comput. Vis.*, vol. 18, no. 3, pp. 233-254, 1996.
- [8] J. R. Smith and S. F. Chang, "VisualSEEk: A fully automated content-based image query system," Proc. 4th ACM Int. Conf. Multimedia, Multimed. 1996, pp. 87-98, 1997.
- [9] W. Y. Ma and B. S. Manjunath, "NeTra: A toolbox for navigating large image databases," *Multimed. Syst.*, vol. 7, no. 3, pp. 184-198, 1999.
- [10] J. Z. Wang, J. Li, and G. Wiederholdy, "SIMPLIcity: Semantics-sensitive integrated matching for picture libraries?," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1929, no. 9, pp. 360-371, 2000.
- [11] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 2, no. 1, pp. 1-19, 2006.
- [12] Y. Liu, D. Zhang, G. Lu, and W. Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognit.*, vol. 40, no. 1, pp. 262-282, 2007.
- [13] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349-1380, 2000.
- [14] G. Sucharitha and R. K. Senapati, "Shape based image retrieval using lower order zernike moments," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 3, pp. 1651-1660, 2017.
- [15] T. G. Ngo, Q. T. Ngo, and D. D. Nguyen, "Image Retrieval with relevance feedback using SVM active learning," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 6, pp. 3238-3246, 2016.
- [16] Y. Wang, K. Bin Jia, and P. Y. Liu, "A novel ROI based image retrieval algorithm," Second Int. Conf. Innov. Comput. Inf. Control. ICICIC 2007, 2008.
- [17] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld : Image Segmentation Using Expectation-Maximization and Its Application to Image Querying," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1026-1038, 2002.
- [18] K. Wong, K. Cheung, and L. Po, "MIRROR: an interactive content based image retrieval system," *2005 IEEE Int. Symp. Circuits Syst.*, pp. 1541-1544 Vol. 2, 2005.
- [19] Q. Wu and K. R. Castleman, "Image Segmentation," in *Microscope Image Processing*, 2007, pp. 159-194.
- [20] N. Shrivastava and V. Tyagi, "A Review of ROI Image Retrieval Techniques," *Adv. Intell. Syst. Comput.*, vol. 328, pp. 509-518, 2015.
- [21] E. R. Vimina and K. P. Jacob, "An Evaluation of Image Matching Algorithms for Region Based Image Retrieval," *Int. J. Adv. Comput. Technol.*, vol. 6, no. 6, pp. 75-85, 2014.
- [22] Y. Rui, T. Huang, and S. Chang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues," *J. of Visual Commun. Image Represent.*, vol. 62, no. 10, pp. 39-62, 1999.
- [23] M. Jourlin and J. C. Pinoli, "Logarithmic image processing," *Adv. Imaging Electron Phys.*, vol. 115, no. C, pp. 129-196, 2001.
- [24] C. J. Du and D. W. Sun, "Recent developments in the applications of image processing techniques for food quality evaluation," *Trends Food Sci. Technol.*, vol. 15, no. 5, pp. 230-249, 2004.
- [25] S. Bansod, "A Brief Review On Image Retrieval Techniques and its Scope," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 5, no. 3, pp. 455-459, 2017.
- [26] B. H. Kang, "A review on image and video processing," *Int. J. Multimed. Ubiquitous Eng.*, vol. 2, no. 2, pp. 49-64, 2007.
- [27] G. Kumar, "A Detailed Review of Feature Extraction in Image Processing Systems," 2014 Fourth Int. Conf. Adv. Comput. Commun. Technol., pp. 5-12, 2014.
- [28] S. Leandrou, S. Petroudi, P. A. Kyriacou, C. C. Reyes-Aldasoro, and C. S. Pattichis, "Quantitative MRI Brain Studies in Mild Cognitive Impairment and Alzheimer's Disease: A Methodological Review," *IEEE Rev. Biomed. Eng.*, vol. 11, pp. 97-111, 2018.
- [29] M. S. Meharban and D. S. Priya, "A Review on Image Retrieval Techniques," *Bonfring Int. J. Adv. Image Process.*, vol. 6, no. 2, pp. 07-10, 2016.

- [30] F. Jing and B. Zhang, "An Effective Region-Based Image Retrieval Framework," *Multimed.* '02 Proc. tenth ACM Int. Conf. Multimed., no. 60135010, pp. 456-465, 2002.
- [31] T. Chen, L. H. Chen, and K. K. Ma, "Colour image indexing using SOM for region-of-interest retrieval," *Pattern Anal. Appl.*, vol. 2, no. 2, pp. 164-171, 1999.
- [32] B. Moghaddam, H. Biermann, and D. Margaritis, "Regions-of-interest and spatial layout for content-based image retrieval," *Multimed. Tools Appl.*, vol. 14, no. 2, pp. 201-210, 2001.
- [33] Y. Liu, D. Zhang, and G. Lu, "Region-based image retrieval with high-level semantics using decision tree learning," *Pattern Recognit.*, vol. 41, no. 8, pp. 2554-2570, 2008.
- [34] D. Zhang, M. M. Islam, G. Lu, and J. Hou, "Semantic image retrieval using region based inverted file," *DICTA 2009 - Digit. Image Comput. Tech. Appl.*, pp. 242-249, 2009.
- [35] Q. Tian, Y. Wu, and T. S. Huang, "Combine user defined region-of-interest and spatial layout for image retrieval," *IEEE Int. Conf. Image Process.*, vol. 3, pp. 746-749, 2000.
- [36] F. Jing, M. Li, L. Zhang, H. J. Zhang, and B. Zhang, "Learning in region-based image retrieval," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2728, pp. 206-215, 2003.
- [37] K. Vu, K. A. Hua, and W. Tavanapong, "Image quality assessment based on regions of interest," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 1045-1049, 2003.
- [38] B. G. Prasad, K. K. Biswas, and S. K. Gupta, "Region-based image retrieval using integrated color, shape, and location index," *Comput. Vis. Image Underst.*, vol. 94, no. 1-3, pp. 193-233, 2004.
- [39] F. Jing, M. Li, H. J. Zhang, and B. Zhang, "Relevance feedback in region-based image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 672-681, 2004.
- [40] B. Ko and H. Byun, "FRIP: A region-based image retrieval tool using automatic image segmentation and stepwise boolean and matching," *IEEE Trans. Multimed.*, vol. 7, no. 1, pp. 105-113, 2005.
- [41] Q. Zhou, L. Ma, M. Celenk, and D. Chelberg, "Content-based image retrieval based on ROI detection and relevance feedback," *Multimed. Tools Appl.*, vol. 27, no. 2, pp. 251-281, 2005.
- [42] J. Zhang, C. W. Yoo, and S. W. Ha, "ROI based natural image retrieval using color and texture feature," *Proc. - Fourth Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2007*, vol. 4, no. Fskd, pp. 740-744, 2007.
- [43] Y. K. Chan, Y. A. Ho, Y. T. Liu, and R. C. Chen, "A ROI image retrieval method based on CVAAO," *Image Vis. Comput.*, vol. 26, no. 11, pp. 1540-1549, 2008.
- [44] C. Huang, Q. Liu, and S. Yu, "Regions of interest extraction from color image based on visual saliency," *J. Supercomput.*, vol. 58, no. 1, pp. 20-33, 2011.
- [45] Z. Wang, G. Liu, and Y. Yang, "A new ROI based image retrieval system using an auxiliary Gaussian weighting scheme," *Multimed. Tools Appl.*, vol. 67, no. 3, pp. 549-569, 2012.

BIOGRAPHIES OF AUTHORS



Mardhiyah Md Jan received her BEng in Electronics Computer & Information Engineering from International Islamic University Malaysia (IIUM), Malaysia, in 2013 and MEng in Communication and Computer Engineering from Universiti Kebangsaan Malaysia (UKM), Malaysia, in 2015. Currently she is pursuing her PhD studies at Department of Electrical, Electronics & Systems Engineering, Faculty of Engineering & Built Environment, Universiti Kebangsaan Malaysia, Bangi, Malaysia. Her main research interests include computer engineering, image processing and projection mapping.



Dr. Nasharuddin Zainal received his BEng in Computer Science from Tokyo Institute of Technology (Japan), in 1998, MEng in Communication and Computer Engineering from Universiti Kebangsaan Malaysia (UKM), Malaysia, in 2003, and DrEng degree from Tokyo Institute of Technology (Japan) in 2010. He is currently an Associate Professor at the Department of Electrical, Electronics & Systems Engineering, Faculty of Engineering & Built Environment, Universiti Kebangsaan Malaysia, Bangi, Malaysia. His main research interests include computer engineering, image and video processing and pattern recognition.



Dr. Shahrizan Jamaludin received his BEng in Electronic (Computer) Engineering from Universiti Teknikal Malaysia Melaka (UTeM) and both his MEng in Communication and Computer Engineering and PhD in Electrical, Electronic and Systems Engineering from Universiti Kebangsaan Malaysia (UKM). He is a graduate engineer certified by the Board of Engineers Malaysia and a graduate technologist certified by the Malaysia Board of Technologists. Currently, he is a senior lecturer at the Department of Maritime Technology, Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu (UMT). His research interests include Image processing, computer vision, signal processing, pattern recognition, embedded system, computer engineering, software engineering, control system, automation, instrumentation, robotic, maritime technology and occupational safety and health.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/226359900>

Contour and Texture Analysis for Image Segmentation

Article in International Journal of Computer Vision · June 2001

DOI: 10.1023/A:1011174803800 · Source: CiteSeer

CITATIONS

1,103

READS

1,666

4 authors, including:



Jitendra Malik
University of California, Berkeley

250 PUBLICATIONS 91,746 CITATIONS

[SEE PROFILE](#)



Serge Belongie
Cornell University

313 PUBLICATIONS 79,672 CITATIONS

[SEE PROFILE](#)



Thomas Leung
Google Inc.

42 PUBLICATIONS 16,652 CITATIONS

[SEE PROFILE](#)



Contour and Texture Analysis for Image Segmentation

JITENDRA MALIK, SERGE BELONGIE, THOMAS LEUNG* AND JIANBO SHI[†]
Computer Science Division, University of California at Berkeley, Berkeley, CA 94720-1776, USA

Received December 28, 1999; Revised February 23, 2001; Accepted February 23, 2001

Abstract. This paper provides an algorithm for partitioning grayscale images into disjoint regions of coherent brightness and texture. Natural images contain both textured and untextured regions, so the cues of contour and texture differences are exploited simultaneously. Contours are treated in the *intervening contour* framework, while texture is analyzed using *textons*. Each of these cues has a domain of applicability, so to facilitate cue combination we introduce a gating operator based on the texturedness of the neighborhood at a pixel. Having obtained a local measure of how likely two nearby pixels are to belong to the same region, we use the spectral graph theoretic framework of normalized cuts to find partitions of the image into regions of coherent texture and brightness. Experimental results on a wide range of images are shown.

Keywords: segmentation, texture, grouping, cue integration, texton, normalized cut

1. Introduction

To humans, an image is not just a random collection of pixels; it is a meaningful arrangement of regions and objects. Figure 1 shows a variety of images. Despite the large variations of these images, humans have no problem interpreting them. We can agree about the different regions in the images and recognize the different objects. Human visual grouping was studied extensively by the Gestalt psychologists in the early part of the 20th century (Wertheimer, 1938). They identified several factors that lead to human perceptual grouping: similarity, proximity, continuity, symmetry, parallelism, closure and familiarity. In computer vision, these factors have been used as guidelines for many grouping algorithms.

The most studied version of grouping in computer vision is image segmentation. Image segmentation techniques can be classified into two broad families—(1) region-based, and (2) contour-based approaches. Region-based approaches try to find partitions of the image pixels into sets corresponding to coherent im-

age properties such as brightness, color and texture. Contour-based approaches usually start with a first stage of edge detection, followed by a linking process that seeks to exploit curvilinear continuity.

These two approaches need not be that different from each other. Boundaries of regions can be defined to be contours. If one enforces closure in a contour-based framework (Elder and Zucker, 1996; Jacobs, 1996) then one can get regions from a contour-based approach. The difference is more one of emphasis and what grouping factor is coded more naturally in a given framework.

A second dimension on which approaches can be compared is local vs. global. Early techniques, in both contour and region frameworks, made local decisions—in the contour framework this might be declaring an edge at a pixel with high gradient, in the region framework this might be making a merge/split decision based on a local, greedy strategy.

Region-based techniques lend themselves more readily to defining a global objective function (for example, Markov random fields (Geman and Geman, 1984) or variational formulations (Mumford and Shah, 1989)). The advantage of having a global objective function is that decisions are made only when

*Present address: Compaq Cambridge Research Laboratory.

[†]Present address: Robotics Institute, Carnegie Mellon University.

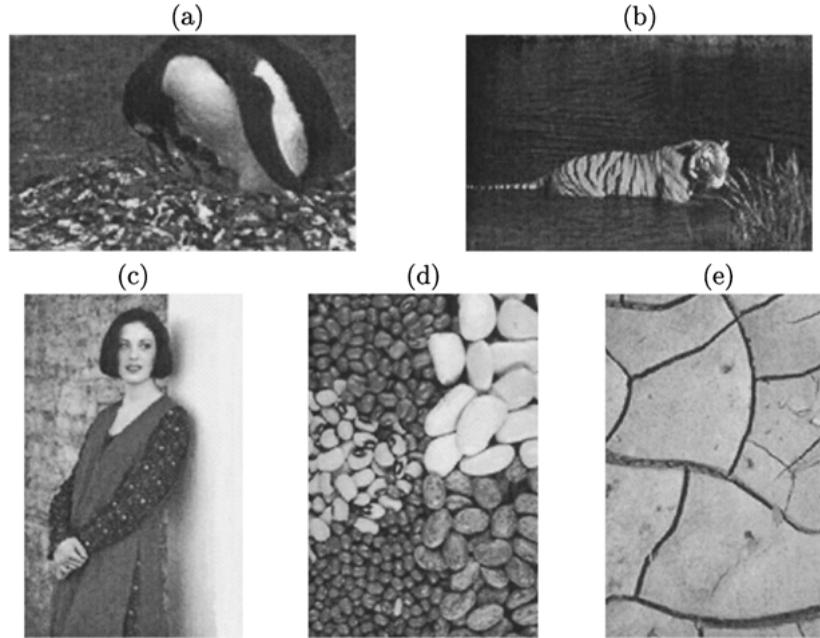


Figure 1. Some challenging images for a segmentation algorithm. Our goal is to develop a single grouping procedure which can deal with all these types of images.

information from the whole image is taken into account at the same time.

In contour-based approaches, often the first step of edge detection is done locally. Subsequently efforts are made to improve results by a global linking process that seeks to exploit curvilinear continuity. Examples include dynamic programming (Montanari, 1971), relaxation approaches (Parent and Zucker, 1989), saliency networks (Sha'ashua and Ullman, 1988), stochastic completion (Williams and Jacobs, 1995). A criticism of this approach is that the edge/no edge decision is made prematurely. To detect extended contours of very low contrast, a very low threshold has to be set for the edge detector. This will cause random edge segments to be found everywhere in the image, making the task of the curvilinear linking process unnecessarily harder than if the raw contrast information was used.

A third dimension on which various segmentation schemes can be compared is the class of images for which they are applicable. As suggested by Fig. 1, we have to deal with images which have both textured and untextured regions. Here boundaries must be found using *both* contour and texture analysis. However what we find in the literature are approaches which concentrate on one or the other.

Contour analysis (e.g. edge detection) may be adequate for untextured images, but in a textured region

it results in a meaningless tangled web of contours. Think for instance of what an edge detector would return on the snow and rock region in Fig. 1(a). The traditional “solution” for this problem in edge detection is to use a high threshold so as to minimize the number of edges found in the texture area. This is obviously a non-solution—such an approach means that low-contrast extended contours will be missed as well. This problem is illustrated in Fig. 2. There is no recognition of the fact that extended contours, even weak in contrast, are perceptually significant.

While the perils of using edge detection in textured regions have been noted before (see e.g. Binford, 1981), a complementary problem of contours constituting a problem for texture analysis does not seem to have been recognized before. Typical approaches are based on measuring texture descriptors over local windows, and then computing differences between window descriptors centered at different locations. Boundaries can then give rise to thin strip-like regions, as in Fig. 3. For specificity, assume that the texture descriptor is a histogram of linear filter outputs computed over a window. Any histogram window near the boundary of the two regions will contain large filter responses from filters oriented along the direction of the edge. However, on both sides of the boundary, the histogram will indicate a featureless region. A segmentation algorithm based on, say, χ^2

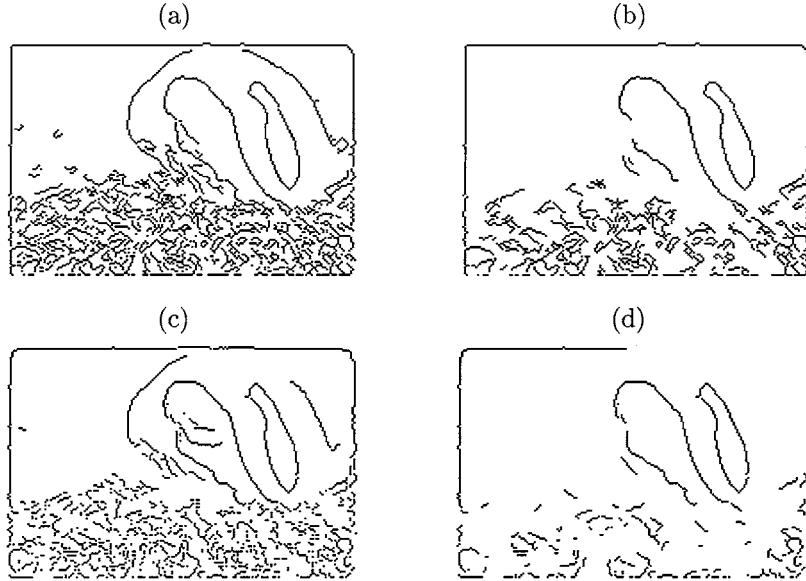


Figure 2. Demonstration of texture as a problem for the contour process. Each image shows the edges found with a Canny edge detector for the penguin image using different scales and thresholds: (a) fine scale, low threshold, (b) fine scale, high threshold, (c) coarse scale, low threshold, (d) coarse scale, high threshold. A parameter setting that preserves the correct edges while suppressing spurious detections in the textured area is not possible.



Figure 3. Demonstration of the “contour-as-a-texture” problem using a real image. (a) Original image of a bald eagle. (b) The groups found by an EM-based algorithm (Belongie et al., 1998).

distances between histograms, will inevitably partition the boundary as a group of its own. As is evident, the problem is not confined to the use of a histogram of filter outputs as texture descriptor. Figure 3(b) shows the actual groups found by an EM-based algorithm using an alternative color/texture descriptor (Belongie et al., 1998).

1.1. Desiderata of a Theory of Image Segmentation

At this stage, we are ready to summarize our desired attributes for a theory of image segmentation.

1. It should deal with general images. Regions with or without texture should be processed in the same

framework, so that the cues of contour and texture differences can be simultaneously exploited.

2. In terms of contour, the approach should be able to deal with boundaries defined by brightness step edges as well as lines (as in a cartoon sketch).
3. Image regions could contain texture which could be regular such as the polka dots in Fig. 1(c), stochastic as in the snow and rock region in (a) or anywhere in between such as the tiger stripes in (b). A key question here is that one needs an automatic procedure for scale selection. Whatever one’s choice of texture descriptor, it has to be computed over a local window whose size and shape need to be determined adaptively. What makes scale selection a challenge is that the technique must deal with the

wide range of textures—regular, stochastic, or intermediate cases—in a seamless way.

1.2. Introducing Textons

Julesz introduced the term *texton*, analogous to a phoneme in speech recognition, nearly 20 years ago (Julesz, 1981) as the putative units of preattentive human texture perception. He described them qualitatively for simple binary line segment stimuli—oriented segments, crossings and terminators—but did not provide an operational definition for gray-level images. Subsequently, texton theory fell into disfavor as a model of human texture discrimination as accounts based on spatial filtering with orientation and scale-selective mechanisms that could be applied to arbitrary gray-level images became popular.

There is a fundamental, well recognized, problem with linear filters. Generically, they respond to any stimulus. Just because you have a response to an oriented odd-symmetric filter doesn't mean there is an edge at that location. It could be that there is a higher contrast bar at some other location in a different orientation which has caused this response. Tokens such as edges or bars or corners can not be associated with the output of a single filter. Rather it is the signature of the outputs over scales, orientations and order of the filter that is more revealing.

Here we introduce a further step by focussing on the *outputs* of these filters considered as points in a high dimensional space (on the order of 40 filters are used). We perform vector quantization, or clustering, in this high-dimensional space to find prototypes. Call these prototypes *textons*—we will find empirically that these tend to correspond to oriented bars, terminators and so on. One can construct a universal texton vocabulary by processing a large number of natural images, or we could find them adaptively in windows of images. In each case the K -means technique can be used. By mapping each pixel to the texton nearest to its vector of filter responses, the image can be analyzed into texton channels, each of which is a point set.

It is our opinion that the analysis of an image into textons will prove useful for a wide variety of visual processing tasks. For instance, in Leung and Malik (1999) we use the related notion of 3D textons for recognition of textured materials. In the present paper, our objective is to develop an algorithm for the segmentation of an image into regions of coherent brightness and texture—we will find that the texton representation will

enable us to address the key problems in a very natural fashion.

1.3. Summary of Our Approach

We pursue image segmentation in the framework of Normalized Cuts introduced by Shi and Malik (1997, 2000). The image is considered to be a weighted graph where the nodes i and j are pixels and edge weights, W_{ij} , denote a local measure of similarity between the two pixels. Grouping is performed by finding eigenvectors of the Normalized Laplacian of this graph (§3). The fundamental issue then is that of specifying the edge weights W_{ij} ; we rely on normalized cuts to go from these local measures to a globally optimal partition of the image.

The algorithm analyzes the image using the two cues of contour and texture. The local similarity measure between pixels i and j due to the contour cue, W_{ij}^{IC} , is computed in the *intervening contour* framework of Leung and Malik (1998) using peaks in contour orientation energy (§2 and §4.1). Texture is analysed using textons (§2.1). Appropriate local scale is estimated from the texton labels. A histogram of texton densities is used as the texture descriptor. Similarity, W_{ij}^{TX} , is measured using the χ^2 test on the histograms (§4.2). The edge weights W_{ij} combining both contour and texture information are specified by gating each of the two cues with a texturedness measure (§4.3).

In (§5), we present the practical details of going from the eigenvectors of the normalized Laplacian matrix of the graph to a partition of the image. Results from the algorithm are presented in (§6). Some of the results presented here were published in Malik et al. (1999).

2. Filters, Composite Edgels, and Textons

Since the 1980s, many approaches have been proposed in the computer vision literature that start by convolving the image with a bank of linear spatial filters f_i tuned to various orientation and spatial frequencies (Knutsson and Granlund, 1983; Koenderink and van Doorn, 1987; Fogel and Sagi, 1989; Malik and Perona, 1990). (See Fig. 4 for an example of such a filter set.)

These approaches were inspired by models of processing in the early stages of the primate visual system (e.g. DeValois and DeValois, 1988). The filter kernels f_i are models of receptive fields of simple cells in visual

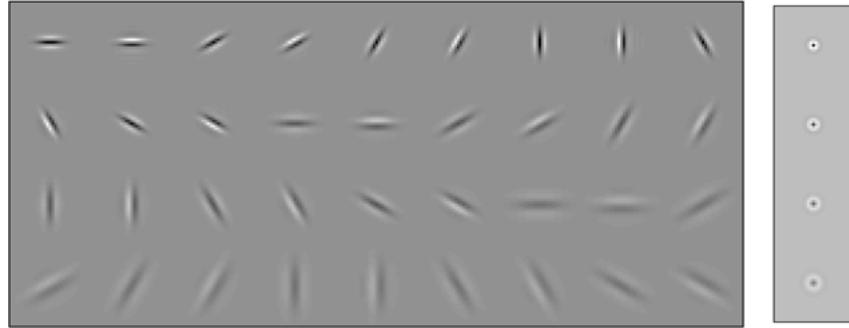


Figure 4. Left: Filter set f_i consisting of 2 phases (even and odd), 3 scales (spaced by half-octaves), and 6 orientations (equally spaced from 0 to π). The basic filter is a difference-of-Gaussian quadrature pair with 3 : 1 elongation. Right: 4 scales of center-surround filters. Each filter is L_1 -normalized for scale invariance.

cortex. To a first approximation, we can classify them into three categories:

1. Cells with radially symmetric receptive fields. The usual choice of f_i is a Difference of Gaussians (DOG) with the two Gaussians having different values of σ . Alternatively, these receptive fields can also be modeled as the Laplacian of Gaussian.
2. Oriented odd-symmetric cells whose receptive fields can be modeled as rotated copies of a horizontal oddsymmetric receptive field. A suitable point spread function for such a receptive field is $f(x, y) = G'_{\sigma_1}(y)G_{\sigma_2}(x)$ where $G_\sigma(x)$ represents a Gaussian with standard deviation σ . The ratio $\sigma_2 : \sigma_1$ is a measure of the elongation of the filter.
3. Oriented even-symmetric cells whose receptive fields can be modeled as rotated copies of a horizontal evensymmetric receptive field. A suitable point spread function for such a receptive field is

$$f(x, y) = G''_{\sigma_1}(y)G_{\sigma_2}(x)$$

The use of Gaussian derivatives (or equivalently, differences of offset Gaussians) for modeling receptive fields of simple cells is due to Young (1985). One could equivalently use Gabor functions. Our preference for Gaussian derivatives is based on their computational simplicity and their natural interpretation as ‘blurred derivatives’ (Koenderink and van Doorn, 1987, 1988).

The oriented filterbank used in this work, depicted in Fig. 4, is based on rotated copies of a Gaussian derivative and its Hilbert transform. More precisely, let $f_1(x, y) = G''_{\sigma_1}(y)G_{\sigma_2}(x)$ and $f_2(x, y)$ equal the

Hilbert transform of $f_1(x, y)$ along the y axis:

$$\begin{aligned} f_1(x, y) &= \frac{d^2}{dy^2} \left(\frac{1}{C} \exp\left(\frac{y^2}{\sigma^2}\right) \exp\left(\frac{x^2}{\ell^2\sigma^2}\right) \right) \\ f_2(x, y) &= \text{Hilbert}(f_1(x, y)) \end{aligned}$$

where σ is the scale, ℓ is the aspect ratio of the filter, and C is a normalization constant. (The use of the Hilbert transform instead of a first derivative makes f_1 and f_2 an exact quadrature pair.) The radially symmetric portion of the filterbank consists of Difference-of-Gaussian kernels. Each filter is zero-mean and L_1 normalized for scale invariance (Malik and Perona, 1990).

Now suppose that the image is convolved with such a bank of linear filters. We will refer to the collection of response images $I * f_i$ as the *hypercolumn transform* of the image.

Why is this useful from a computational point of view? The vector of filter outputs $I * f_i(x_0, y_0)$ characterizes the image *patch* centered at x_0, y_0 by a set of values at a *point*. This is similar to characterizing an analytic function by its derivatives at a point—one can use a Taylor series approximation to find the values of the function at neighboring points. As pointed out by Koenderink and van Doorn (1987), this is more than an analogy, because of the commutativity of the operations of differentiation and convolution, the receptive fields described above are in fact computing ‘blurred derivatives’. We recommend Koenderink and van Doorn (1987, 1988), Jones and Malik (1992), and Malik and Perona (1992) for a discussion of other advantages of such a representation.

The hypercolumn transform provides a convenient front end for contour and texture analysis:

- *Contour.* In computational vision, it is customary to model brightness edges as step edges and to detect them by marking locations corresponding to the maxima of the outputs of odd-symmetric filters (e.g. Canny, 1986) at appropriate scales. However, it should be noted that step edges are an inadequate model for the discontinuities in the image that result from the projection of depth or orientation discontinuities in physical scene. Mutual illumination and specularities are quite common and their effects are particularly significant in the neighborhood of convex or concave object edges. In addition, there will typically be a shading gradient on the image regions bordering the edge. As a consequence of these effects, real image edges are not step functions but more typically a combination of steps, peak and roof profiles. As was pointed out in Perona and Malik (1990), the oriented energy approach (Knutsson and Granlund, 1983; Morrone and Owens, 1987; Morrone and Burr, 1988) can be used to detect and localize correctly these composite edges.

The oriented energy, also known as the “quadrature energy,” at angle 0° is defined as:

$$OE_{0^\circ} = (I * f_1)^2 + (I * f_2)^2$$

OE_{0° has maximum response for horizontal contours. Rotated copies of the two filter kernels are able to pick up composite edge contrast at various orientations.

Given OE_θ , we can proceed to localize the composite edge elements (edgels) using oriented non-maximal suppression. This is done for each scale in the following way. At a generic pixel q , let $\theta^* = \arg \max OE_\theta$ denote the dominant orientation and OE^* the corresponding energy. Now look at the two neighboring values of OE^* on either side of q along the line through q perpendicular to the dominant orientation. The value OE^* is kept at the location of q only if it is greater than or equal to each of the neighboring values. Otherwise it is replaced with a value of zero.

Noting that OE^* ranges between 0 and infinity, we convert it to a probability-like number between 0 and 1 as follows:

$$p_{con} = 1 - \exp(-OE^*/\sigma_{IC}) \quad (1)$$

σ_{IC} is related to oriented energy response purely due to image noise. We use $\sigma = 0.02$ in this paper.

The idea is that for any contour with $OE^* \gg \sigma_{IC}$, $p_{con} \approx 1$.

- *Texture.* As the hypercolumn transform provides a good local descriptor of image patches, the boundary between differently textured regions may be found by detecting curves across which there is a significant gradient in one or more of the components of the hypercolumn transform. For an elaboration of this approach, see Malik and Perona (1990).

Malik and Perona relied on averaging with large kernels to smooth away spatial variation for filter responses within regions of texture. This process loses a lot of information about the distribution of filter responses; a much better method is to represent the neighborhood around a pixel by a histogram of filter outputs (Heeger and Bergen, 1995; Puzicha et al., 1997). While this has been shown to be a powerful technique, it leaves open two important questions. Firstly, there is the matter of what size window to use for pooling the histogram—the integration scale. Secondly, these approaches only make use of marginal binning, thereby missing out on the informative characteristics that joint assemblies of filter outputs exhibit at points of interest. We address each of these questions in the following section.

2.1. Textons

Though the representation of textures using filter responses is extremely versatile, one might say that it is overly redundant (each pixel value is represented by N_{fil} real-valued filter responses, where N_{fil} is 40 for our particular filter set). Moreover, it should be noted that we are characterizing textures, entities with some spatially repeating properties by definition. Therefore, we do not expect the filter responses to be totally different at each pixel over the texture. Thus, there should be several distinct filter response vectors and all others are noisy variations of them.

This observation leads to our proposal of clustering the filter responses into a small set of prototype response vectors. We call these prototypes *textons*. Algorithmically, each texture is analyzed using the filter bank shown in Fig. 4. Each pixel is now transformed to a N_{fil} dimensional vector of filter responses. These vectors are clustered using K -means. The criterion for this algorithm is to find K “centers” such that after assigning each data vector to the nearest center, the sum

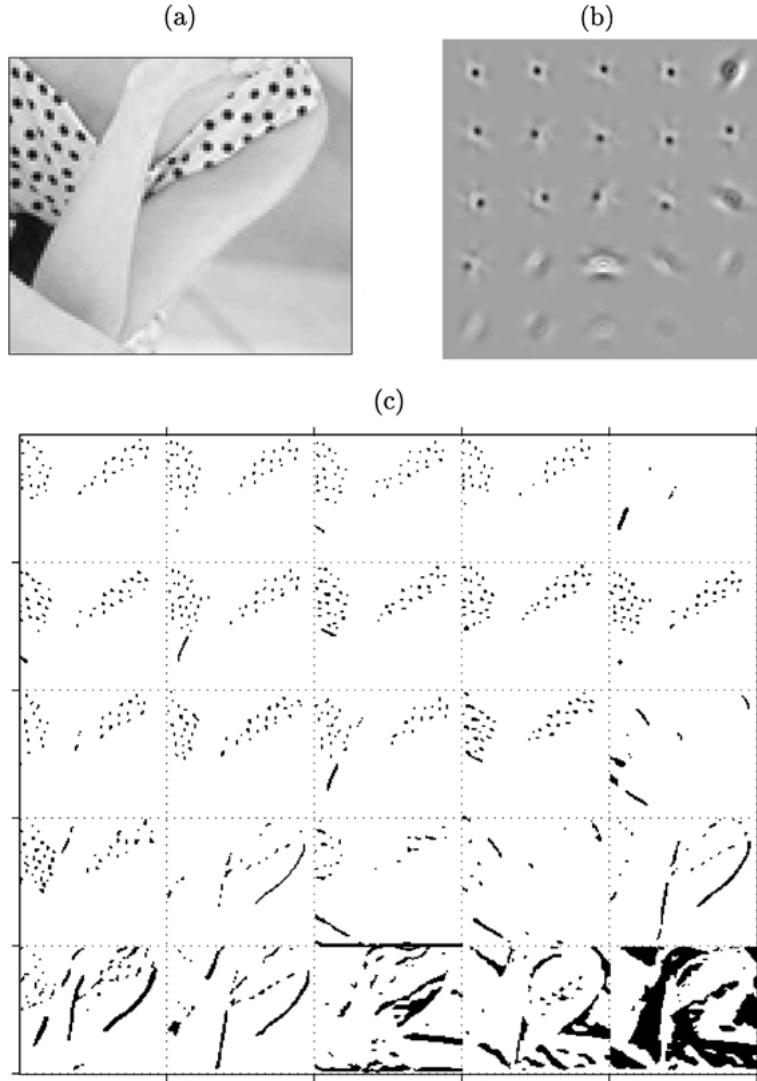


Figure 5. (a) Polka-dot image. (b) Textons found via K -means with $K = 25$, sorted in decreasing order by norm. (c) Mapping of pixels to the texton channels. The dominant structures captured by the textons are translated versions of the dark spots. We also see textons corresponding to faint oriented edge and bar elements. Notice that some channels contain activity inside a textured region or along an oriented contour and nowhere else.

of the squared distance from the centers is minimized. K -means is a greedy algorithm that finds a local minimum of this criterion.¹

It is useful to visualize the resulting cluster centers in terms of the original filter kernels. To do this, recall that each cluster center represents a set of projections of each filter onto a particular image patch. We can solve for the image patch corresponding to each cluster center in a least squares sense by premultiplying the vectors representing the cluster centers by the pseudoinverse of the filterbank (Jones and Malik, 1992). The matrix rep-

resenting the filterbank is formed by concatenating the filter kernels into columns and placing these columns side by side. The set of synthesized image patches for two test images are shown in Figs. 5(b) and 6(b). These are our textons. The textons represent assemblies of filter outputs that are characteristic of the local image structure present in the image.

Looking at the polka-dot example, we find that many of the textons correspond to translated versions of dark spots.² Also included are a number of oriented edge elements of low contrast and two textons representing

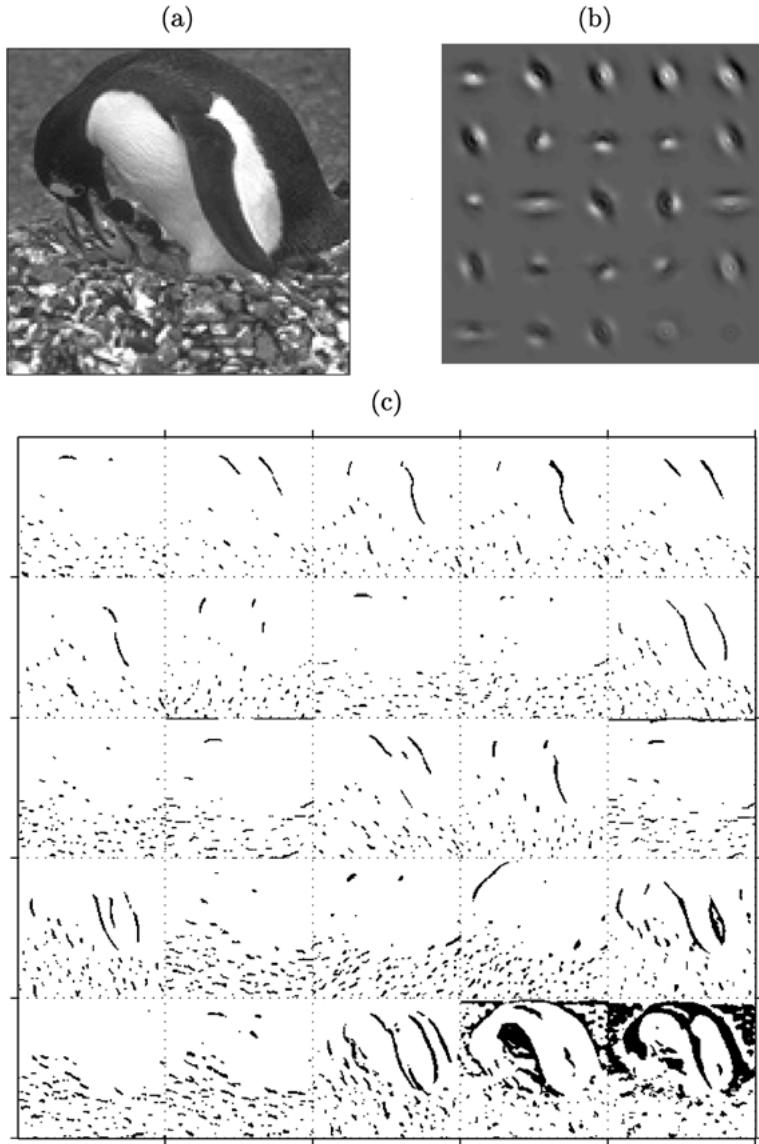


Figure 6. (a) Penguin image. (b) Textons found via K -means with $K = 25$, sorted in decreasing order by norm. (c) Mapping of pixels to the texton channels. Among the textons we see edge elements of varying orientation and contrast along with elements of the stochastic texture in the rocks.

nearly uniform brightness. The pixel-to-texton mapping is shown in Fig. 5(c). Each subimage shows the pixels in the image that are mapped to the corresponding texton in Fig. 5(b). We refer to this collection of discrete point sets as the texton *channels*. Since each pixel is mapped to exactly one texton, the texton channels constitute a partition of the image.

Textons and texton channels are also shown for the penguin image in Fig. 6. Notice in the two examples how much the texton set can change from one image

to the next. The spatial characteristics of both the deterministic polka dot texture and the stochastic rocks texture are captured across several texton channels. In general, the texture boundaries emerge as point density changes across the different texton channels. In some cases, a texton channel contains activity inside a particular textured region and nowhere else. By comparison, vectors of filter outputs generically respond with some value at every pixel—a considerably less clean alternative.

We have not been particularly sophisticated in the choice of K , the number of different textons for a given image. How to choose an optimal value of K in K -means has been the subject of much research in the model selection and clustering literature; we used a fixed choice $K = 36$ to obtain the segmentation results in this paper. Clearly, if the images vary considerably in complexity and number of objects in them, an adaptive choice may give better results.

The mapping from pixel to texton channel provides us with a number of discrete point sets where before we had continuous-valued filter vectors. Such a representation is well suited to the application of techniques from computational geometry and point process statistics. With these tools, one can approach questions such as, “what is the neighborhood of a texture element?” and “how similar are two pixels inside a textured region?”

Several previous researchers have employed clustering using K -means or vector quantization as a stage in their approach to texture classification—two representative examples are McLean (1993) and Raghunathan et al. (1997). What is novel about our approach is the identification of clusters of vectors of filter outputs with the Julesz notion of textons. Then first order statistics of textons are used for texture characterization, and the spatial structure within texton channels enables scale estimation. Vector quantization becomes much more than just a data compression or coding step. The next subsection should make this point clear.

2.1.1. Local Scale and Neighborhood Selection. The texton channel representation provides us a natural way to define texture scale. If the texture is composed of discrete elements (“texels”), we might want to define a notion of texel neighbors and consider the mean distance

between them to be a measure of scale. Of course, many textures are stochastic and detecting texels reliably is hard even for regular textures.

With textons we have a “soft” way to define neighbors. For a given pixel in a texton channel, first consider it as a “thickened point”—a disk centered at it.³ The idea is that while textons are being associated with pixels, since they correspond to assemblies of filter outputs, it is better to think of them as corresponding to a small image disk defined by the scale used in the Gaussian derivative filters. Recall Koenderink’s aphorism about a point in image analysis being a Gaussian blob of small σ !

Now consider the Delaunay neighbors of all the pixels in the thickened point of a pixel i which lie closer than some outer scale.⁴ The intuition is that these will be pixels in spatially neighboring texels. Compute the distances of all these pixels to i ; the median of these constitutes a robust local measure of inter-texel distance. We define the local scale $\alpha(i)$ to be 1.5 times this median distance.

In Fig. 7(a), the Delaunay triangulation of a zoomed-in portion of one of the texton channels in the polka-dot dress of Fig. 5(a) is shown atop a brightened version of the image. Here the nodes represent points that are similar in the image while the edges provide proximity information.

The local scale $\alpha(i)$ is based just on the texton channel for the texton at i . Since neighboring pixels should have similar scale and could be drawn from other texton channels, we can improve the estimate of scale by median filtering of the scale image.

2.1.2. Computing Windowed Texton Histograms. Pairwise texture similarities will be computed by comparing windowed texton histograms. We define the

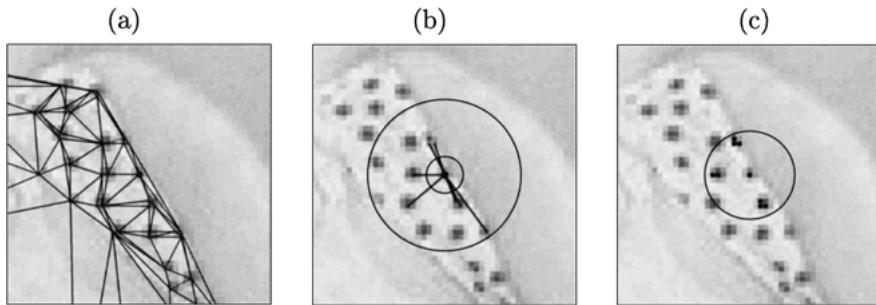


Figure 7. Illustration of scale selection. (a) Closeup of Delaunay triangulation of pixels in a particular texton channel for polka dot image. (b) Neighbors of thickened point for pixel at center. The thickened point lies within inner circle. Neighbors are restricted to lie within outer circle. (c) Selected scale based on median of neighbor edge lengths, shown by circle, with all pixels falling inside circle marked with dots.

window $\mathcal{W}(i)$ for a generic pixel i as the axis-aligned square of radius $\alpha(i)$ centered on pixel i .

Each histogram has K bins, one for each texton channel. The value of the k th histogram bin for a pixel i is found by counting how many pixels in texton channel k fall inside the window $\mathcal{W}(i)$. Thus the histogram represents texton frequencies in a local neighborhood. We can write this as

$$h_i(k) = \sum_{j \in \mathcal{W}(i)} I[T(j) = k] \quad (2)$$

where $I[\cdot]$ is the indicator function and $T(j)$ returns the texton assigned to pixel j .

3. The Normalized Cut Framework

In the Normalized Cut framework (Shi and Malik, 1997, 2000), which is inspired by spectral graph theory (Chung, 1997), Shi and Malik formulate visual grouping as a graph partitioning problem. The nodes of the graph are the entities that we want to partition; for example, in image segmentation, they are the pixels. The edges between two nodes correspond to the *strength* with which these two nodes belong to one group; again, in image segmentation, the edges of the graph correspond to how much two pixels agree in brightness, color, etc. Intuitively, the criterion for partitioning the graph will be to minimize the sum of weights of connections *across* the groups and maximize the sum of weights of connections *within* the groups.

Let $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$ be a weighted undirected graph, where \mathbf{V} are the nodes and \mathbf{E} are the edges. Let \mathbf{A}, \mathbf{B} be a partition of the graph: $\mathbf{A} \cup \mathbf{B} = \mathbf{V}, \mathbf{A} \cap \mathbf{B} = \emptyset$. In graph theoretic language, the similarity between these two groups is called the *cut*:

$$\text{cut}(\mathbf{A}, \mathbf{B}) = \sum_{i \in \mathbf{A}, j \in \mathbf{B}} W_{ij}$$

where W_{ij} is the weight on the edge between nodes i and j . Shi and Malik proposed to use a *normalized* similarity criterion to evaluate a partition. They call it the *normalized cut*:

$$N \text{cut}(\mathbf{A}, \mathbf{B}) = \frac{\text{cut}(\mathbf{A}, \mathbf{B})}{\text{assoc}(\mathbf{A}, \mathbf{V})} + \frac{\text{cut}(\mathbf{B}, \mathbf{A})}{\text{assoc}(\mathbf{B}, \mathbf{V})}$$

where $\text{assoc}(\mathbf{A}, \mathbf{V}) = \sum_{i \in \mathbf{A}, k \in \mathbf{V}} W_{ik}$ is the total connection from nodes in \mathbf{A} to all the nodes in the graph.

For more discussion of this criterion, please refer to Shi and Malik (2000).

One key advantage of using the normalized cut is that a good approximation to the optimal partition can be computed very efficiently.⁵ Let \mathbf{W} be the association matrix, i.e. W_{ij} is the weight between nodes i and j in the graph. Let \mathbf{D} be the diagonal matrix such that $D_{ii} = \sum_j W_{ij}$, i.e. D_{ii} is the sum of the weights of all the connections to node i . Shi and Malik showed that the optimal partition can be found by computing:

$$\begin{aligned} \mathbf{y} &= \arg \min N\text{cut} \\ &= \arg \min_{\mathbf{y}} \frac{\mathbf{y}^T (\mathbf{D} - \mathbf{W}) \mathbf{y}}{\mathbf{y}^T \mathbf{D} \mathbf{y}} \end{aligned} \quad (3)$$

where $\mathbf{y} = \{a, b\}^N$ is a binary indicator vector specifying the group identity for each pixel, i.e. $y_i = a$ if pixel i belongs to group \mathbf{A} and $y_j = b$ if pixel j belongs to \mathbf{B} . N is the number of pixels. Notice that the above expression is a Rayleigh quotient. If we relax \mathbf{y} to take on real values (instead of two discrete values), we can optimize Eq. (3) by solving a generalized eigenvalue system. Efficient algorithms with polynomial running time are well-known for solving such problems.

The process of transforming the vector \mathbf{y} into a discrete bipartition and the generalization to more than two groups is discussed in (§5).

4. Defining the Weights

The quality of a segmentation based on Normalized Cuts or any other algorithm based on pairwise similarities fundamentally depends on the weights—the W_{ij} 's—that are provided as input. The weights should be large for pixels that should belong together and small otherwise. We now discuss our method for computing the W_{ij} 's. Since we seek to combine evidence from two cues, we will first discuss the computation of the weights for each cue in isolation, and then describe how the two weights can be combined in a meaningful fashion.

4.1. Images Without Texture

Consider for the moment the “cracked earth” image in Fig. 1(e). Such an image contains no texture and may be treated in a framework based solely on contour features. The definition of the weights in this case, which we denote W_{ij}^{IC} , is adopted from the *intervening contour* method introduced in Leung and Malik (1998).

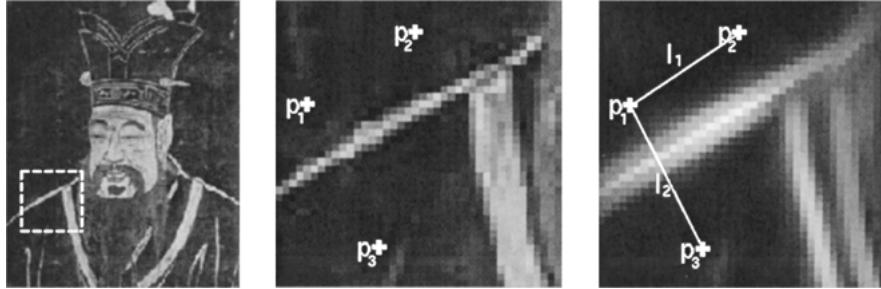


Figure 8. Left: the original image. Middle: part of the image marked by the box. The intensity values at pixels p_1 , p_2 and p_3 are similar. However, there is a contour in the middle, which suggests that p_1 and p_2 belong to one group while p_3 belongs to another. Just comparing intensity values at these three locations will mistakenly suggest that they belong to the same group. Right: orientation energy. Somewhere along l_2 , the orientation energy is strong which correctly proposes that p_1 and p_3 belong to two different partitions, while orientation energy along l_1 is weak throughout, which will support the hypothesis that p_1 and p_2 belong to the same group.

Figure 8 illustrates the intuition behind this idea. On the left is an image. The middle figure shows a magnified part of the original image. On the right is the orientation energy. There is an extended contour separating p_3 from p_1 and p_2 . Thus, we expect p_1 to be much more strongly related to p_2 than p_3 . This intuition carries over in our definition of dissimilarity between two pixels: if the orientation energy along the line between two pixels is strong, the dissimilarity between these pixels should be high (and W_{ij} should be low).

Contour information in an image is computed “softly” through orientation energy (OE) from elongated quadrature filter pairs. We introduce a slight modification here to allow for exact sub-pixel localization of the contour by finding the local maxima in the orientation energy perpendicular to the contour orientation (Perona and Malik, 1990). The orientation energy gives the confidence of this contour. W_{ij}^{IC} is then defined as follows:

$$W_{ij}^{IC} = 1 - \max_{x \in M_{ij}} p_{con}(x)$$

where M_{ij} is the set of local maxima along the line joining pixels i and j . Recall from (§2) that $p_{con}(x)$, $0 < p_{con} < 1$, is nearly 1 whenever the orientated energy maximum at x is sufficiently above the noise level. In words, two pixels will have a weak link between them if there is a strong local maximum of orientation energy along the line joining the two pixels. On the contrary, if there is little energy, for example in a constant brightness region, the link between the two pixels will be strong. Contours measured at different scales can be taken into account by computing the orientation energy maxima at various scales and setting p_{con} to be the maximum over all the scales at each pixel.

4.2. Images that are Texture Mosaics

Now consider the case of images wherein all of the boundaries arise from neighboring patches of different texture (e.g. Fig. 1(d)). We compute pairwise texture similarities by comparing windowed texton histograms computed using the technique described previously (§2.1.2). A number of methods are available for comparing histograms. We use the χ^2 test, defined as

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}$$

where h_i and h_j are the two histograms. For an empirical comparison of the χ^2 test versus other texture similarity measures, see Puzicha et al. (1997).

W_{ij}^{TX} is then defined as follows:

$$W_{ij}^{TX} = \exp(-\chi^2(h_i, h_j)/\sigma_{TX}) \quad (4)$$

If histograms h_i and h_j are very different, χ^2 is large, and the weight W_{ij}^{TX} is small.

4.3. General Images

Finally we consider the general case of images that contain boundaries of both kinds. This presents us with the problem of *cue integration*. The obvious approach to cue integration is to define the weight between pixels i and j as the product of the contribution from each cue: $W_{ij} = W_{ij}^{IC} \times W_{ij}^{TX}$. The idea is that if either of the cues suggests that i and j should be separated, the composite weight, W_{ij} , should be small. We must be careful, however, to avoid the problems listed in the

Introduction (§1) by suitably gating the cues. The spirit of the gating method is to make each cue “harmless” in locations where the other cue should be operating.

4.3.1. Estimating Texturedness. As illustrated in Fig. 2, the fact that a pixel survives the non-maximum suppression step does not necessarily mean that that pixel lies on a region boundary. Consider a pixel inside a patch of uniform texture: its oriented energy is large but it does not lie on the boundary of a region. Conversely, consider a pixel lying between two uniform patches of just slightly different brightness: it does lie on a region boundary but its oriented energy is small. In order to estimate the “probability” that a pixel lies on a boundary, it is necessary to take more surrounding information into account. Clearly the true value of this probability is only determined after the final correct segmentation, which is what we seek to find. At this stage our goal is to formulate a local estimate of the texturedness of the region surrounding a pixel. Since this is a local estimate, it will be noisy but its objective will be to bootstrap the global segmentation procedure.

Our method of computing this value is based on a simple comparison of texton distributions on either side of a pixel relative to its dominant orientation. Consider a generic pixel q at an oriented energy maximum. Let the dominant orientation be θ . Consider a circle of radius $\alpha(q)$ (the selected scale) centered on q . We first divide this circle in two along the diameter with orientation θ . Note that the contour passing through q is tangent to the diameter, which is its best straight line approximation. The pixels in the disk can be partitioned into three sets D_0 , D_- , D_+ which are the pixels in the strip along the diameter, the pixels to the left of D_0 , and the pixels to the right of D_0 , respectively. To compute our measure of texturedness, we consider two half window comparisons with D_0 assigned to each side. Assume without loss of generality that D_0 is first assigned to the “left” half. Denote the K -bin histograms of $D_0 \cup D_-$ by h_L and D_+ by h_R respectively. Now consider the χ^2 statistic between the two histograms:

$$\chi^2(h_L, h_R) = \frac{1}{2} \sum_{k=1}^K \frac{[h_L(k) - h_R(k)]^2}{h_L(k) + h_R(k)}$$

We repeat the test with the histograms of D_- and $D_0 \cup D_+$ and retain the maximum of the two resulting values, which we denote χ_{LR}^2 . We can convert this



Figure 9. Illustration of half windows used for the estimation of texturedness. The texturedness of a label is based on a χ^2 test on the textons in the two sides of a box as shown above for two sample pixels. The size and orientation of the box is determined by the selected scale and dominant orientation for the pixel at center. Within the rocky area, the texton statistics are very similar, leading to a low χ^2 value. On the edge of the wing, the χ^2 value is relatively high due to the dissimilarity of the textons that fire on either side of a step edge. Since in the case of the contour the contour itself can lie along the diameter of the circle, we consider two half-window partitions: one where the thin strip around the diameter is assigned to the left side, and one where it is assigned to the other. We consider both possibilities and retain the maximum of the two resulting χ^2 values.

to a probability-like value using a sigmoid as follows:

$$p_{texture} = 1 - \frac{1}{1 + \exp [- (\chi_{LR}^2 - \tau) / \beta]} \quad (5)$$

This value, which ranges between 0 and 1, is small if the distributions on the two sides are very different and large otherwise. Note that in the case of untextured regions, such as a brightness step edge, the textons lying along and parallel to the boundary make the statistics of the two sides different. This is illustrated in Fig. 9. Roughly, $p_{texture} \approx 1$ for oriented energy maxima in texture and $p_{texture} \approx 0$ for contours. $p_{texture}$ is defined to be 0 at pixels which are not oriented energy maxima.

4.3.2. Gating the Contour Cue. The contour cue is gated by means of suppressing contour energy according to the value of $p_{texture}$. The gated value, p_B , is defined as

$$p_B = (1 - p_{texture}) p_{con} \quad (6)$$

In principle, this value can be computed and dealt with independently at each filter scale. For our purposes, we found it sufficient simply to keep the maximum value

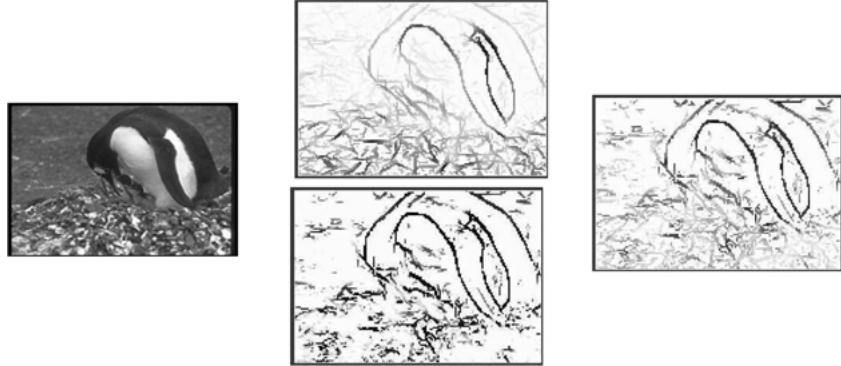


Figure 10. Gating the contour cue. Left: original image. Top: oriented energy after nonmaximal suppression, OE^* . Bottom: $1 - p_{texture}$. Right: p_B , the product of $1 - p_{texture}$ and $p_{con} = 1 - \exp(-OE^*/\sigma_{IC})$. Note that this can be thought of as a “soft” edge detector which has been modified to no longer fire on texture regions.

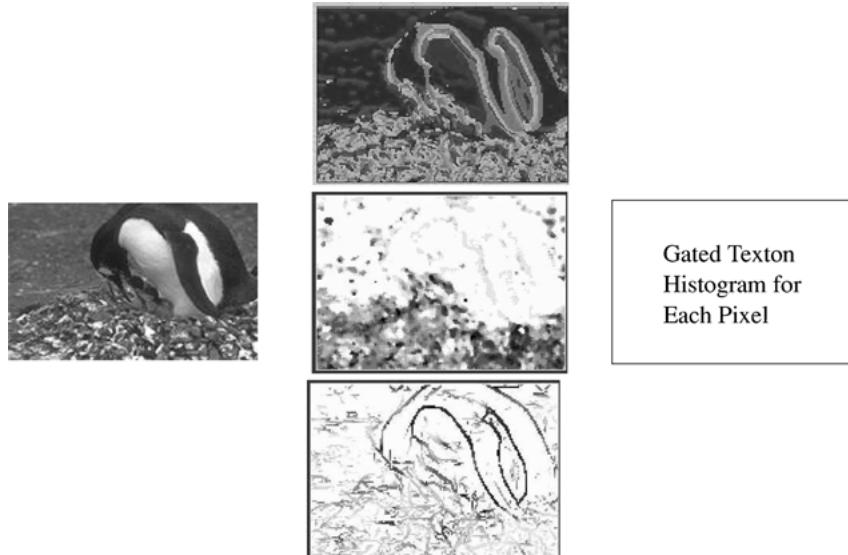


Figure 11. Gating the texture cue. Left: original image. Top: Textons label, shown in pseudocolor. Middle: local scale estimate $\alpha(i)$. Bottom: $1 - p_{texture}$. Darker grayscale indicates larger values. Right: Local texton histograms at scale $\alpha(i)$ are gated using $p_{texture}$ as explained in 4.3.3.

of p_B with respect to σ . The gated contour energy is illustrated in Fig. 10, right. The corresponding weight is then given by

$$W_{ij}^{IC} = 1 - \max_{x \in M_{ij}} p_B(x)$$

4.3.3. Gating the Texture Cue. The texture cue is gated by computing a texton histogram at each pixel which takes into account the texturedness measure $p_{texture}$ (see Fig. 11). Let h_i be the K -bin texton histogram computed using Eq. (2). We define a $(K + 1)$ -bin histogram \hat{h}_i by introducing a 0th bin. The intuition

is that the 0th bin will keep a count of the number of pixels which do not correspond to texture. These pixels arise in two forms: (1) pixels which are not oriented energy maxima; (2) pixels which are oriented energy maxima, but correspond to boundaries between two regions, thus should not take part in texture processing to avoid the problems discussed in (§1). More precisely, \hat{h}_i is defined as follows:

$$\begin{aligned} \hat{h}_i(k) &= \sum_{j \in N(i)} p_{texture}(j) \cdot I[T(j) = k] \quad \forall k = 1 \dots K \\ \hat{h}_i(0) &= N_B + \sum_{j \in N(i)} (1 - p_{texture}(j)) \end{aligned}$$

where $\mathcal{N}(i)$ denotes all the oriented energy maxima lying inside the window $\mathcal{W}(i)$ and N_B is the number of pixels which are not oriented energy maxima.

4.3.4. Combining the Weights. After each cue has been gated by the above procedure, we are free to perform simple multiplication of the weights. More specifically, we first obtain W^{IC} using Eq. (6). Then we obtain W^{TX} using Eq. (4) with the gated versions of the histograms. Then we simply define the combined weight as

$$W_{ij} = W_{ij}^{IC} \times W_{ij}^{TX}$$

4.3.5. Implementation Details. The weight matrix is defined between any pair of pixels i and j . Naively, one might connect every pair of pixels in the image. However, this is not necessary. Pixels very far away from the image have very small likelihood of belonging to the same region. Moreover, dense connectivity means that we need to solve for the eigenvectors of a matrix of size $N_{pix} \times N_{pix}$, where N_{pix} is close to a million for a typical image. In practice, a sparse and short-ranged connection pattern does a very good job. In our experiments, all the images are of size 128×192 . Each pixel is connected to pixels within a radius of 30. Furthermore, a sparse sampling is implemented such that the number of connections is approximately constant at each radius. The number of non-zero connections per pixel is 1000 in our experiments. For images of different sizes, the connection radius can be scaled appropriately.

The parameters for the various formulae are given here:

1. The image brightness lies in the range $[0, 1]$.
2. $\sigma_{IC} = 0.02$ (Eq. (1)).
3. The number of textons computed using K -means: $K = 36$.
4. The textons are computed following a contrast normalization step, motivated by Weber’s law. Let $|F(x)|$ be the L_2 norm of the filter responses at pixel x . We normalize the filter responses by the following equation:

$$F(x) \leftarrow F(x) \times \frac{\log\left(1 + \frac{|F(x)|}{0.03}\right)}{|F(x)|}$$

5. $\sigma_{TX} = 0.025$ (Eq. (4)).
6. $\tau = 0.3$ and $\beta = 0.04$ (Eq. (5))

Note that these parameters are the same for all the results shown in (§6).

5. Computing the Segmentation

With a properly defined weight matrix, the normalized cut formulation discussed in (§3) can be used to compute the segmentation. However, the weight matrix defined in the previous section is computed using only local information, and is thus not perfect. The ideal weight should be computed in such a way that region boundaries are respected. More precisely, (1) texton histograms should be collected from pixels in a window residing exclusively in one and only one region. If instead, an isotropic window is used, pixels near a texture boundary will have a histogram computed from textons in both regions, thus “polluting” the histogram. (2) Intervening contours should only be considered at region boundaries. Any responses to the filters inside a region are either caused by texture or are simply mistakes. However, these two criteria mean that we need a segmentation of the image, which is exactly the reason why we compute the weights in the first place! This chicken-and-egg problem suggests an iterative framework for computing the segmentation. First, use the local estimation of the weights to compute a segmentation. This segmentation is done so that no region boundaries are missed, i.e. it is an over-segmentation. Next, use this initial segmentation to update the weights. Since the initial segmentation does not miss any region boundaries, we can coarsen the graph by merging all the nodes inside a region into one *super-node*. We can then use these *super-nodes* to define a much simpler segmentation problem. Of course, we can continue this iteration several times. However, we elect to stop after 1 iteration.

The procedure consists of the following 4 steps:

1. Compute an initial segmentation from the locally estimated weight matrix.
2. Update the weights using the initial segmentation.
3. Coarsen the graph with the updated weights to reduce the segmentation to a much simpler problem.
4. Compute a final segmentation using the coarsened graph.

5.1. Computing the Initial Segmentation

Computing a segmentation of the image amounts to computing the eigenvectors of the generalized

eigensystem: $(\mathbf{D} - \mathbf{W})\mathbf{v} = \lambda \mathbf{D}\mathbf{v}$ (Eq. (3)). The eigenvectors can be thought of as a transformation of the image into a new feature vector space. In other words, each pixel in the original image is now represented by a vector with the components coming from the corresponding pixel across the different eigenvectors. Finding a partition of the image is done by finding the clusters in this eigenvector representation. This is a much simpler problem because the eigenvectors have essentially put regions of coherent descriptors according to our cue of texture and contour into very tight clusters. Simple techniques such as K -means can do a very good job in finding these clusters. The following procedure is taken:

1. Compute the eigenvectors corresponding to the second smallest to the twelfth smallest eigenvalues of the generalized eigensystem $((\mathbf{D} - \mathbf{W})\mathbf{v} = \lambda \mathbf{D}\mathbf{v})$.⁶ Call these 11 eigenvectors $v_i, i = 2, \dots, 12$. The corresponding eigenvalues are $\lambda_i, i = 2, \dots, 12$.
2. Weight⁷ the eigenvectors according to the eigenvalues: $\hat{v}_i = \frac{1}{\sqrt{\lambda_i}} v_i, i = 2, \dots, 12$. The eigenvalues indicate the “goodness” of the corresponding eigenvectors. Now each pixel is transformed to an 11 dimensional vector represented by the weighted eigenvectors.
3. Perform vector quantization on the 11 eigenvectors using K -means. Start with $K^* = 30$ centers. Let the corresponding RMS error for the quantization be e^* . Greedily delete one center at a time such that the increase in quantization error is the smallest. Continue this process until we arrive at K centers when the error e is just greater than $1.1 \times e^*$.

This partitioning strategy provides us with an initial segmentation of the image. This is usually an over-segmentation. The main goal here is simply to provide an initial guess for us to modify the weights. Call this initial segmentation of the image S_0 . Let the number of segments be N_0 . A typical number for N_0 is 10–100.

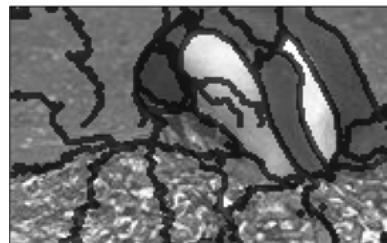


Figure 12. p_B is allowed to be non-zero only at pixels marked.

It should be noted that this strategy for using multiple eigenvectors to provide an initial oversegmentation is merely one of a set of possibilities. Alternatives include recursive splitting using the second eigenvector or first converting the eigenvectors into binary valued vectors and using those simultaneously as in Shi and Malik (2000). Yet another hybrid strategy is suggested in Weiss (1999). We hope that improved theoretical insight into spectral graph partitioning will give us a better way to make this, presently somewhat *ad hoc* choice.

5.2. Updating Weights

The initial segmentation S_0 found in the previous step can provide a good approximation to modify the weight as we have discussed earlier. With S_0 , we modify the weight matrix as follows:

- To compute the texton histograms for a pixel in R_k , textons are collected only from the intersection of R_k and the isotropic window of size determined by the scale, α .
- p_B is set to zero for pixels that are not in the region boundaries of S_0 .

The modified weight matrix is an improvement over the original local estimation of weights.

5.3. Coarsening the Graph

By hypothesis, since S_0 is an over-segmentation of the image, there are no boundaries missed. We do not need to recompute a segmentation for the original problem of N pixels. We can coarsen the graph, where each node of the new graph is a segment in S_0 . The weight between two nodes in this new graph is computed as follows:

$$\hat{W}_{kl} = \sum_{i \in R_k} \sum_{j \in R_l} W_{ij} \quad (7)$$

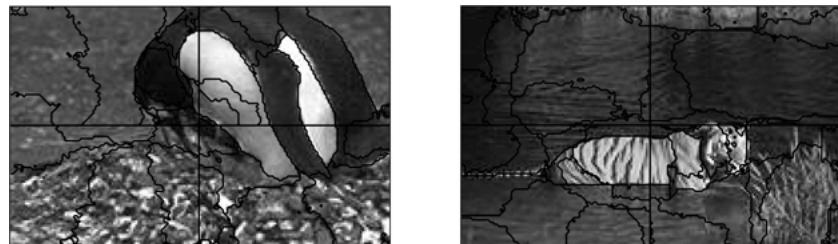


Figure 13. Initial segmentation of the image used for coarsening the graph and computing final segmentation.

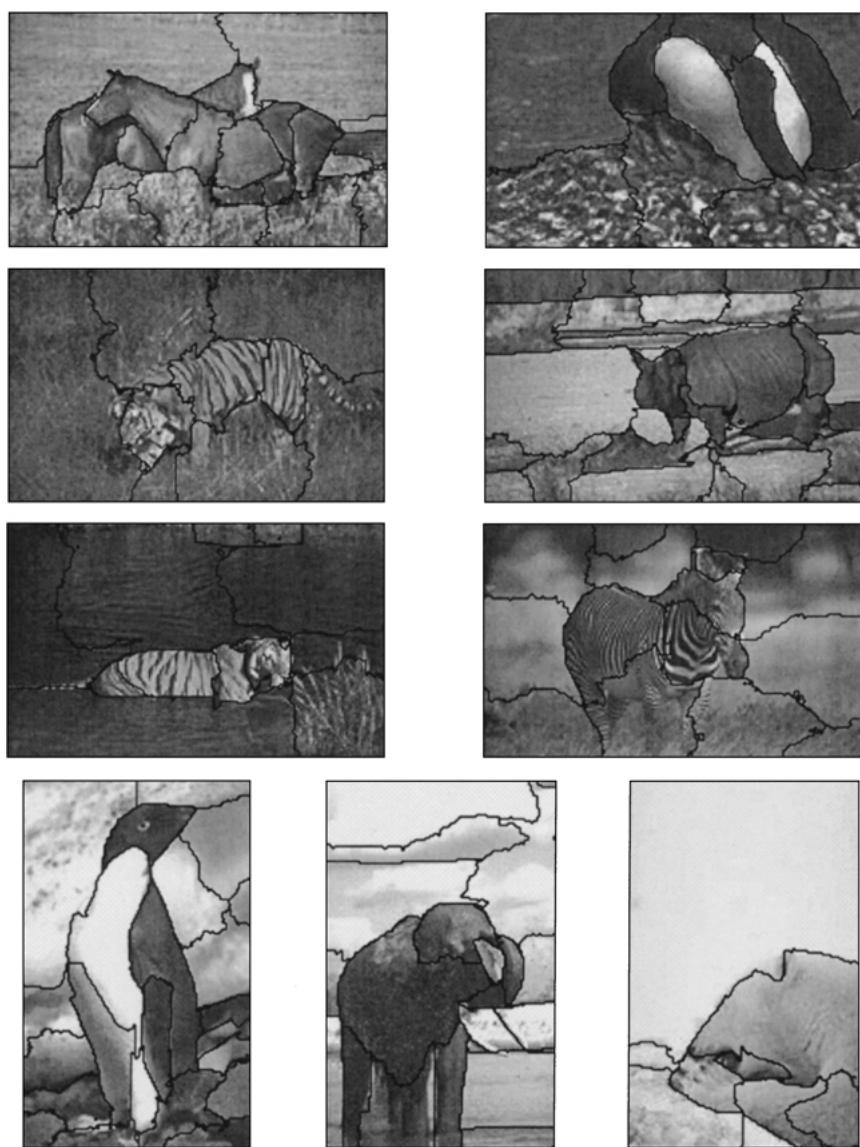


Figure 14. Segmentation of images with animals.



Figure 15. Segmentation of images with people.

where R_k and R_l indicate segments in S_0 (k and $l \in \{1, \dots, N_0\}$); \hat{W} is the weight matrix of the coarsened graph and W is the weight matrix of the original graph. This coarsening strategy is just an instance of graph contraction (Chung, 1997). Now, we have reduced the original segmentation problem with an $N \times N$ weight matrix to a much simpler and faster segmentation problem of $N_0 \times N_0$ without losing in performance.

5.4. Computing the Final Segmentation

After coarsening the graph, we have turned the segmentation problem into a very simple graph partitioning

problem of very small size. We compute the final segmentation using the following procedure:

1. Compute the second smallest eigenvector for the generalized eigensystem using \hat{W} .
2. Threshold the eigenvector to produce a bipartitioning of the image. 30 different values uniformly spaced within the range of the eigenvector are tried as the threshold. The one producing a partition which minimizes the normalized cut value is chosen. The corresponding partition is the best way to segment the image into two regions.
3. Recursively repeat steps 1 and 2 for each of the partitions until the normalized cut value is larger than 0.1.

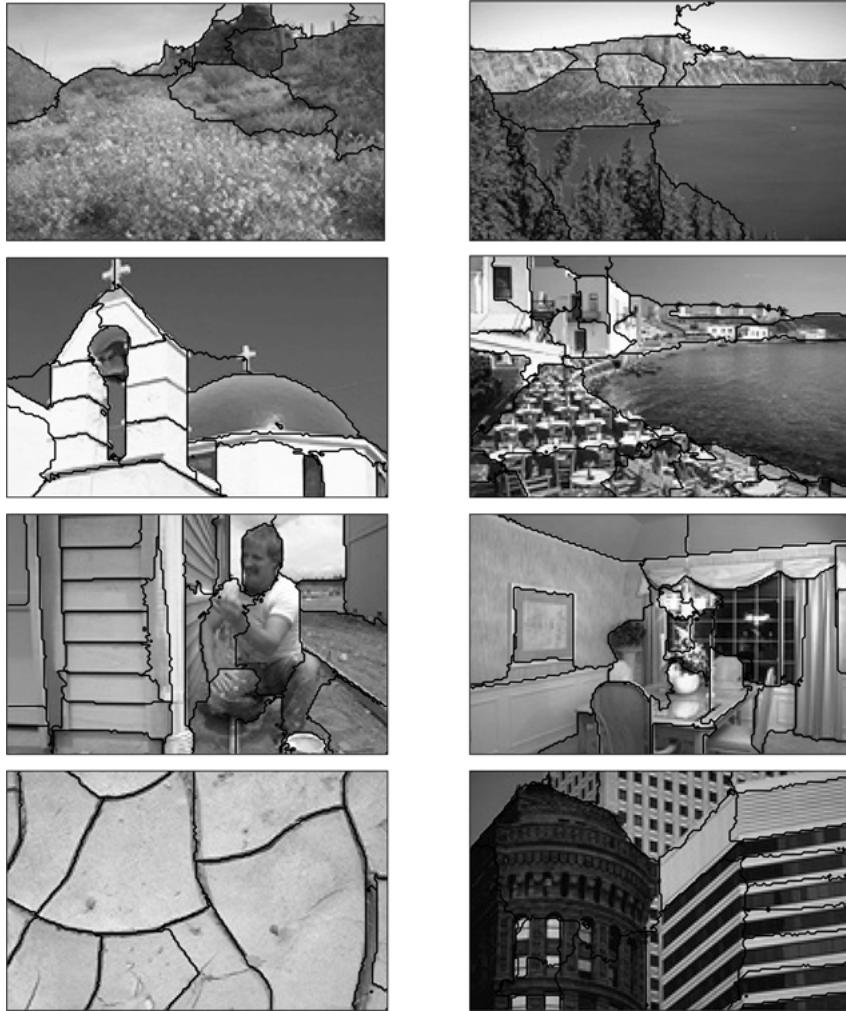


Figure 16. Segmentation of images of natural and man-made scenes.

5.5. Segmentation in Windows

The above procedure performs very well in images with a small number of groups. However, in complicated images, smaller regions can be missed. This problem is intrinsic for global segmentation techniques, where the goal is find a big-picture interpretation of the image. This problem can be dealt with very easily by performing the segmentation in windows.

Consider the case of breaking up the image into quadrants. Define Q^i to be the set of pixels in the i th quadrant. $Q^i \cap Q^j = \emptyset$ and $\cup_{i=1}^4 Q^i = \text{Image}$. Extend each quadrant by including all the pixels which are less than a distance r from any pixels in Q^i , with r being the maximum texture scale, $\alpha(i)$, over the whole

image. Call these enlarged windows \hat{Q}^i . Note that these windows now overlap each other.

Corresponding to each \hat{Q}^i , a weight matrix \hat{W}^i is defined by pulling out from the original weight matrix W the edges whose end-points are nodes in \hat{Q}^i . For each \hat{W}^i , an initial segmentation \hat{S}_0^i is obtained, according to the procedure in (§5.1). The weights are updated as in (§5.2). The extension of each quadrant makes sure that the arbitrary boundaries created by the windowing do not affect this procedure:

Texton histogram upgrade For each pixel in Q^i , the largest possible histogram window (a $(2\alpha + 1)^2$ box) is entirely contained in \hat{Q}^i by virtue of the extension.



Figure 17. Segmentation of paintings.

This means the texton histograms are computed from all the relevant pixels.

Contour upgrade The boundaries in Q_i are a proper subset of the boundaries in \hat{Q}_i . So, we can set the values of p_B at a pixel in Q_i to be zero if it lies on a region boundary in \hat{Q}_i . This enables the correct computation of W_{ij}^{IC} . Two example contour update maps are shown in Fig. 12.

Initial segmentations can be computed for each \hat{Q}_i to give S_0^i . They are restricted to Q^i to produce S_0^i . These segmentations are merged to form an initial segmentation $S_0 = \cup_{i=1}^4 S_0^i$. At this stage, fake boundaries from the windowing effect can occur. Two examples are shown in Fig. 13. The graph is then coarsened and

the final segmentation is computed as in (§5.3) and (§5.4).

6. Results

We have run our algorithm on a variety of natural images. Figures 14–17 show typical segmentation results. In all the cases, the regions are cleanly separated from each other using combined texture and contour cues. Notice that for all these images, a single set of parameters are used. Color is not used in any of these examples and can readily be included to further improve the performance of our algorithm.⁸ Figure 14 shows results for animal images. Results for images containing people are shown in Fig. 15 while natural and

man-made scenes appear in Fig. 16. Segmentation results for paintings are shown in Fig. 17. A set of more than 1000 images from the commercially available Corel Stock Photos database have been segmented using our algorithm.⁹

Evaluating the results against ground truth—What is the correct segmentation of the image?—is a challenging problem. This is because there may not be a single correct segmentation and segmentations can be to varying levels of granularity. We do not address this problem here; a start has been made in recent work in our group (Martin et al., 2000).

Computing times for a C++ implementation of the entire system are under two minutes for images of size 108×176 pixels on a 750 MHz Pentium III machine. There is some variability from one image to another because the eigensolver can take more or less time to converge depending on the image.

7. Conclusion

In this paper we have developed a general algorithm for partitioning grayscale images into disjoint regions of coherent brightness and texture. The novel contribution of the work is in cue integration for image segmentation—the cues of contour and texture differences are exploited simultaneously. We regard the experimental results as promising and hope that the paper will spark renewed research activity in image segmentation, one of the central problems of computer vision.

Acknowledgments

The authors would like to thank the Berkeley vision group, especially Chad Carson, Alyosha Efros, David Forsyth, and Yair Weiss for useful discussions during the development of the algorithm. We thank Doron Tal for implementing the algorithm in C++. This research was supported by (ARO) DAAH04-96-1-0341, the Digital Library Grant IRI-9411334, NSF Graduate Fellowships to SB and JS and a Berkeley Fellowship to TL.

Notes

1. For more discussions and variations of the K -means algorithm, the reader is referred to Duda and Hart (1973) and Gersho and Gray (1992).
2. It is straightforward to develop a method for merging translated versions of the same basic texton, though we have not found it

necessary. Merging in this manner decreases the number of channels needed but necessitates the use of phase-shift information.

3. This is set to 3% of the image dimension in our experiments. This is tied to the intermediate scale of the filters in the filter set.
4. This is set to 10% of the image dimension in our experiments.
5. Finding the true optimal partition is an NP-hard problem.
6. The eigenvector corresponding to the smallest eigenvalue is constant, thus useless.
7. Since normalized cut can be interpreted as a spring-mass system (Shi and Malik, 2000), this normalization comes from the *equipartition theorem* in classical statistical mechanics which states that if a system is in equilibrium, then it has equal energy in each mode (Belongie and Malik, 1998).
8. When color information is available, the similarity W_{ij} becomes a product of 3 terms: $W_{ij} = W_{ij}^{IC} \times W_{ij}^{TX} \times W_{ij}^{COLOR}$. Color similarity, W_{ij}^{COLOR} , is computed using χ^2 differences over color histograms, similar to texture measured using texture histograms. Moreover, color can cluster into “colorons”, analogous to textures.
9. These results are available at the following web page: <http://www.cs.berkeley.edu/projects/vision/Grouping/overview.html>

References

- Belongie, S., Carson, C., Greenspan, H., and Malik, J. 1998. Color- and texture-based image segmentation using EM and its application to content-based image retrieval. In *Proc. 6th Int. Conf. Computer Vision*, Bombay, India, pp. 675–682.
- Belongie, S. and Malik, J. 1998. Finding boundaries in natural images: A new method using point descriptors and area completion. In *Proc. 5th Euro. Conf. Computer Vision*, Freiburg, Germany, pp. 751–766.
- Binford, T. 1981. Inferring surfaces from images. *Artificial Intelligence*, 17(1–3):205–244.
- Canny, J. 1986. A computational approach to edge detection. *IEEE Trans. Pat. Anal. Mach. Intell.*, 8(6):679–698.
- Chung, F. 1997. *Spectral Graph Theory*, AMS. Providence, RI.
- DeValois, R. and DeValois, K. 1988. *Spatial Vision*. Oxford University Press. New York, N.Y.
- Duda, R. and Hart, P. 1973. *Pattern Classification and Scene Analysis*, John Wiley & Sons. New York, N.Y.
- Elder, J. and Zucker, S. 1996. Computing contour closures. In *Proc. Euro. Conf. Computer Vision*, Vol. I, Cambridge, England, pp. 399–412.
- Fogel, I. and Sagi, D. 1989. Gabor filters as texture discriminator. *Biological Cybernetics*, 61:103–113.
- Geman, S. and Geman, D. 1984. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741.
- Gersho, A. and Gray, R. 1992. *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, MA.
- Heeger, D.J. and Bergen, J.R. 1995. Pyramid-based texture analysis/synthesis. In *Proceedings of SIGGRAPH '95*, pp. 229–238.
- Jacobs, D. 1996. Robust and efficient detection of salient convex groups. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(1):23–37.
- Jones, D. and Malik, J. 1992. Computational framework to determine stereo correspondence from a set of linear spatial filters. *Image and Vision Computing*, 10(10):699–708.

- Julesz, B. 1981. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97.
- Knutsson, H. and Granlund, G. 1983. Texture analysis using two-dimensional quadrature filters. In *Workshop on Computer Architecture for Pattern Analysis and Image Database Management*, pp. 206–213.
- Koenderink, J. and van Doorn, A. 1987. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375.
- Koenderink, J. and van Doorn, A. 1988. Operational significance of receptive field assemblies. *Biological Cybernetics*, 58:163–171.
- Leung, T. and Malik, J. 1998. Contour continuity in region-based image segmentation. In *Proc. Euro. Conf. Computer Vision*, Vol. 1, H. Burkhardt and B. Neumann (Eds.), Freiburg, Germany, pp. 544–559.
- Leung, T. and Malik, J. 1999. Recognizing surfaces using three-dimensional textons. In *Proc. Int. Conf. Computer Vision*, Corfu, Greece, pp. 1010–1017.
- Malik, J., Belongie, S., Shi, J., and Leung, T. 1999. Textons, contours and regions: Cue integration in image segmentation. In *Proc. IEEE Intl. Conf. Computer Vision*, Vol. 2, Corfu, Greece, pp. 918–925.
- Malik, J. and Perona, P. 1990. Preattentive texture discrimination with early vision mechanisms. *J. Optical Society of America*, 7(2):923–932.
- Malik, J. and Perona, P. 1992. Finding boundaries in images. In *Neural Networks for Perception*, Vol. 1, H. Wechsler (Ed.). Academic Press, pp. 315–344.
- Martin, D., Fowlkes, C., Tal, D., and Malik, J. 2000. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Technical Report UCB CSD-01-1133, University of California at Berkeley. <http://http.cs.berkeley.edu/projects/vision/Grouping/overview.html>.
- McLean, G. 1993. Vector quantization for texture classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):637–649.
- Montanari, U. 1971. On the optimal detection of curves in noisy pictures. *Comm. Ass. Comput.*, 14:335–345.
- Morrone, M. and Burr, D. 1988. Feature detection in human vision: A phase dependent energy model. *Proc. R. Soc. Lond. B*, 235:221–245.
- Morrone, M. and Owens, R. 1987. Feature detection from local energy. *Pattern Recognition Letters*, 6:303–313.
- Mumford, D. and Shah, J. 1989. Optimal approximations by piecewise smooth functions, and associated variational problems. *Comm. Pure Math.*, 42:577–684.
- Parent, P. and Zucker, S. 1989. Trace inference, curvature consistency, and curve detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(8):823–839.
- Perona, P. and Malik, J. 1990. Detecting and localizing edges composed of steps, peaks and roofs. In *Proc. 3rd Int. Conf. Computer Vision*, Osaka, Japan, pp. 52–57.
- Puzicha, J., Hofmann, T., and Buhmann, J. 1997. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pp. 267–272.
- Raghav, P., Poongodi, R., and Yegnanarayana, B. 1997. Unsupervised texture classification using vector quantization and deterministic relaxation neural network. *IEEE Transactions on Image Processing*, 6(10):1376–1387.
- Sha'ashua, A. and Ullman, S. 1988. ‘Structural saliency: The detection of globally salient structures using a locally connected network. In *Proc. 2nd Int. Conf. Computer Vision*, Tampa, FL, USA, pp. 321–327.
- Shi, J. and Malik, J. 1997. Normalized cuts and image segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pp. 731–737.
- Shi, J. and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905.
- Weiss, Y. 1999. Segmentation using eigenvectors: A unifying view. In *Proc. IEEE Intl. Conf. Computer Vision*, Vol. 2, Corfu, Greece, pp. 975–982.
- Wertheimer, M. 1938. Laws of organization in perceptual forms (partial translation). In *A Sourcebook of Gestalt Psychology*, W. Ellis (Ed.) Harcourt Brace and Company, pp. 71–88.
- Williams, L. and Jacobs, D. 1995. Stochastic completion fields: A neural model of illusory contour shape and salience. In *Proc. 5th Int. Conf. Computer Vision*, Cambridge, MA, pp. 408–415.
- Young, R.A. 1985. The Gaussian derivative theory of spatial vision: Analysis of cortical cell receptive field line-weighting profiles. Technical Report GMR-4920, General Motors Research.

Detection Of Shapes And Counting In Toy Manufacturing Industry With Help Of Python

J. Uma

P. Yuvarani

Department of Electronics and Instrumentation Engineering,

M. Kumarasamy College of Engineering, Karur – 639113, Tamil Nadu.

uma.jaga96@gmail.com

Abstract: The proposed work focused on detection of shapes of toy and its count in the particular area to segregate the items manufactured in the toy industry for packing. The identification of shapes using Ramer –Douglas-Peucker algorithm in Python language is a technique implemented with the help of open CV image processing tool. The prototype of the model is developed by giving a sample input image with different shapes as file. The screen in the python language shows the name of the shapes with its count in the input sample file. The proposed idea may extend in tool manufacturing industry to identify the different shapes of part which is difficult to segregate and consume more man power for the particular process.

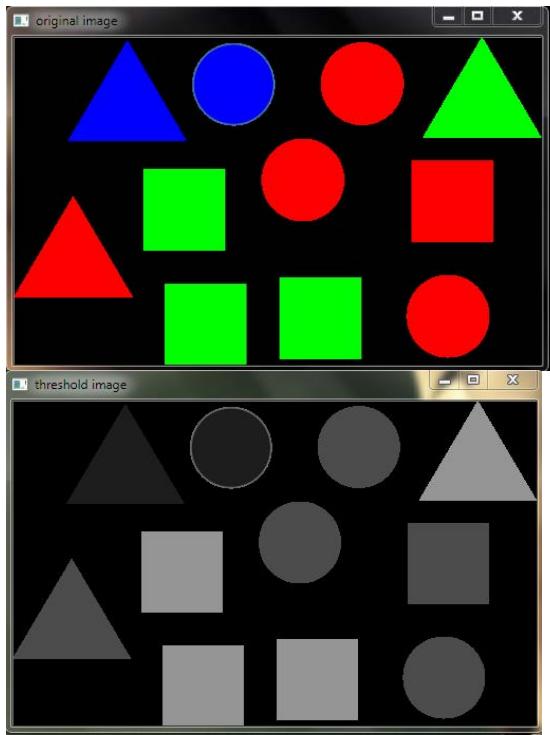
1. Introduction:

Computer Vision is a rapid developing platform for identifying, analyzing and understanding of images. The process of images and its results provides a list of opportunities for the industry and reduce the man power work. Open CV is an image processing library created by Intel and it is available as a tool for programming in the Python language. The requirement of open CV to analyze the image should be of BGR or gray scale in order to process or output the file as per the user requirement. In the paper,Gupta et al,the algorithm for vision based detection if vehicle in the traffic signal point using camera and sequencing the monocular images is described.Daniel et al, uses subtraction and modeling technique to estimate the speed of vehicle movement and image sequencing.Cherg and Kamath developed the model to compare the performance of an sequencing in the large data set.

Daigavane P.M,Bajaj P.R,describes the real time vehicle detection and counting method for controlling the traffic on Highways using Matlab software. Belongie S,Malik J,Puzicha J,the author described the shape matching and its detection algorithm by comparing the image segmentation technique. Tobias O.J,Seara R,used the histogram threshold technique to identify the edge corner in the object and its count by using Fuzzy sets in Image processing tool in Matlab.Hussin et al., describes the identification of mango from a mango tree with the help of colouring technique.By using Circular Hough Transform with given radius as the input the edge detection technique is implemented to count the number of mango in a tree. Hussin et al,describes the image segmentation and cascading techinique using 2-D CT slices by the linear iterative clustering approach.The above super-pixel segmentation was done in 80 manually segmented CT volumes using six-fold cross-validation method.Dollar et al, the author proved the detection of image very fastly using a technique called structured Forest. The technique is mainly implemented in image processing related to medical area for fast and accurate detection with sophisticated component.

Gray scale image carries only intensity information in which the value of each pixel is a single sample unit. Images of the sort consists of black and white colours, the shades of gray images varies from black at the weakest intensity and white at the strongest intensity. Gray scale images are black and white images in the context of imaging also called as bi-level or binary images.

2. GRAY SCALE IMAGE:



3. SIMPLE THRESHOLDING:

Threshold value is assigned as one value if pixel value is greater than a threshold list; otherwise it is assigned as another value.

```
cv2.threshold (gray, 0, 255, 3)
```

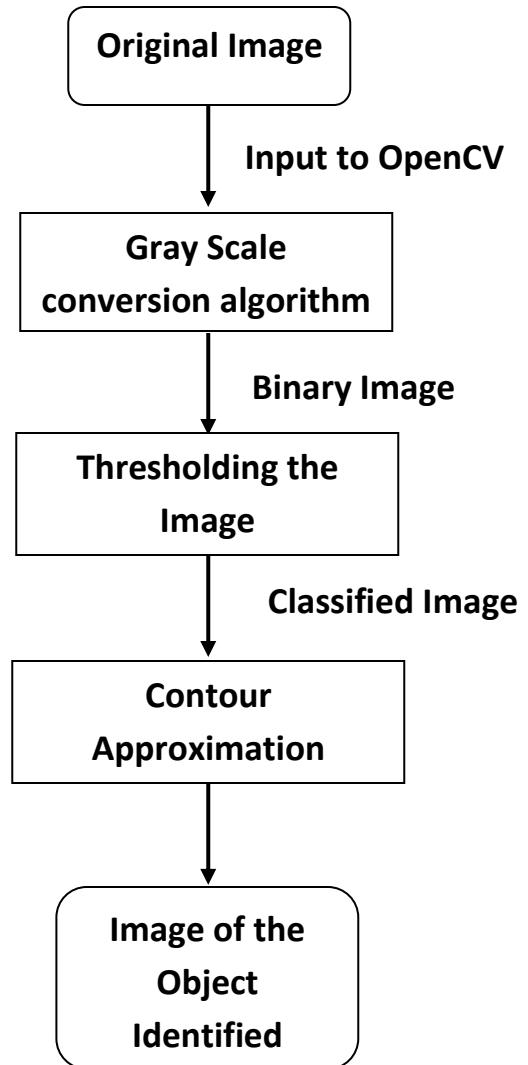
In the instruction given the first word represents a type of image i.e., gray in colour. The second one represent a threshold value maximum limit to classify the pixel values and the third one represents the value which is going to given in pixel of more than the threshold value.

4. CONTOURS:

Contour is a curve which joining all the points in a continuous manner having same intensity or colour. Contour is a best tool for object detection and recognition with the help of shape analysis. Contour algorithm is used to detect edge detection with threshold value for best results use binary images. In open CV, the object need to found should be in white colour and background images should be of black in colour after finding contour points.

5. SHAPE DETECTION LOGIC:

In shape detection Ramer-Douglas-Peucker algorithm or split and merge algorithm. The contour approximation is used since it reduces the number of points in the curve and it is continuous manner. The curve can be detected as short line segments with approximated curve consists of subset points which is defined by the original curve. Contour approximation is implemented in OpenCV via the cv2.approxPolyDP method.



The flowchart describes the image identification algorithm with the help of Python language in Open CV tool. First the original image is converted into gray scale image or Black and White image. The second level is Thresholding the image to classify the image according to the corner edge detection and its

threshold value. The contour approximation is used to detect the image of the object. This is final stage of the algorithm and by using counting loop in the program the number of object in the particular shape is the output of the Open CV tool.

6. Results and Its Discussion :

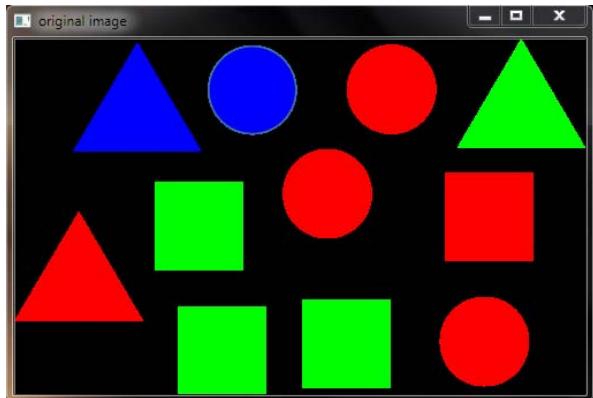


Figure 3 : Input Image 1 for Detection

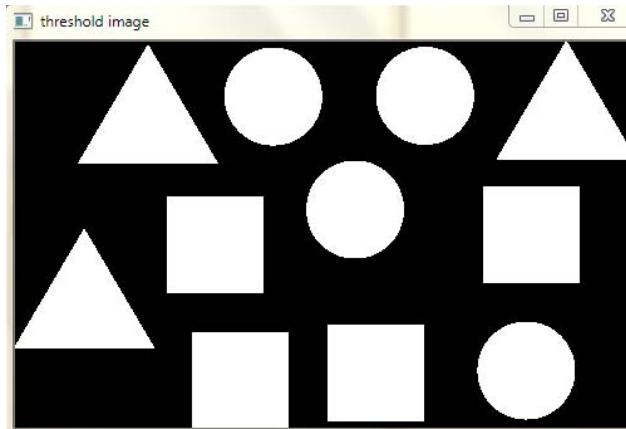


Figure 4 : Gray Scale Image 1

The figure 3 represents the image 1 for detecting the shape and its number in the particular area. The same kind of image may be captured by the camera in the toy industry of particular area to detect and count the number of objects. Then the image is converted into gray scale and threshold it to give as the input for the approximation algorithm Ramer –Douglas-Peucker. With the help of Open CV tool and Python 2.7.12.Shell, the output screen of the image is shown in figure 5.

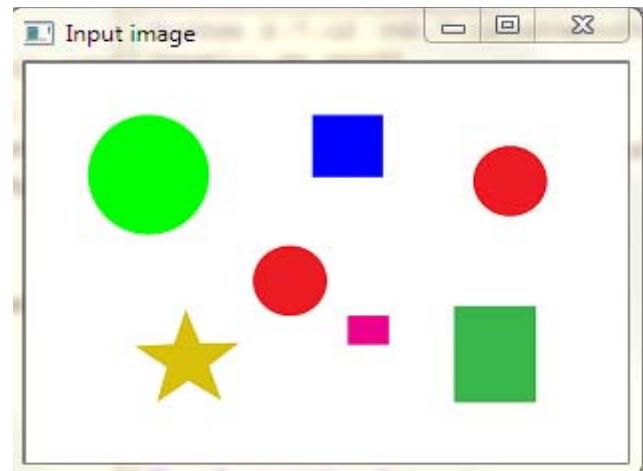


Figure 5 : Identified and count of the Image1

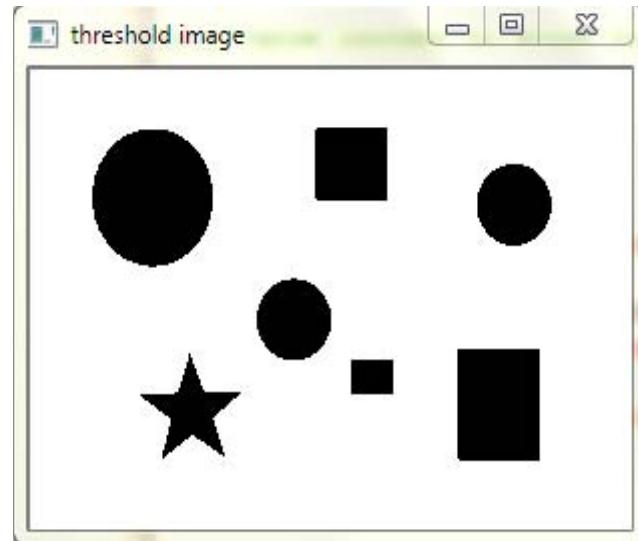


Figure 6 : Input Image 2 for Detection

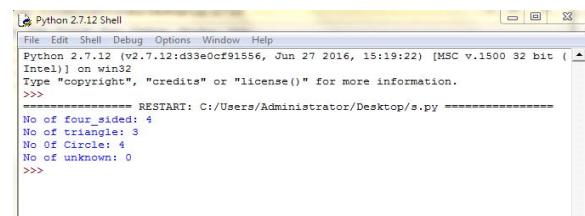


Figure 7 : Gray Scale Image 2

```

Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:19:22) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Users/Administrator/Desktop/s.py =====
No of four_sided: 3
No of triangle: 0
No Of Circle: 3
No of unknown: 1
>>>

```

Figure 8: Identified and count of the Image2

The figure 6 is the image 2 to detect the image of the object with unknown object identification in the input .The figure 7 represents the gray scale and threshold image2 and the figure 8 represents the output of the image 2.

7. CONCLUSION:

The present paper proved the usage of Open CV tool and python language to detect the shape of the image and its number of occurrence in the particular area of the jpeg file. The main focus of the algorithm is used to identify and to detect the number of shapes in the manufacturing industry of the toy shapes. With the help of contour approximation algorithm in Open CV tool used for computer vision development and applications. The Open CV tool is fastest compile tool and its mostly available as open source tool.

REFERENCES:

1. Gupta S.,Masoud O.,Martin R.F.K and Papanikolopoulos N.P., “Detection and Classification of Vehicles”,in IEEE Transaction on Intelligent Transportation Systems,vol.3,no.1,March,2002,pp. 37-47.
2. Dailey D.J.,Cathey F. and Pumrin S., “An Algorithm to Estimate Mean Traffic Speed Using Uncalibrated Cameras”,In IEEE Transactions on Intelligent Transportations Systems,vol.1,no.2,June 2000,pp.98-107.
3. S.Cheung and C.Kamth, “Robust Techniques for Background Subtraction in Urban Traffic Video”,In Video Communications and Image Processing,SPIE Electronic Imaging,San Jose,January ,2004.
4. Daigavane P.M,Bajaj P.R, “Real Time Vehicle Detection and Counting Method for Unsupervised Traffic Video on Highways”,IJCNS International Journal of Computer Science and Network Security,vol.10.no.8,August 2010,pp.112-117.
5. Belongie S,Malik J,Puzicha J, “Shape Matching and Object Recognition using Shape Contexts”,IEEE Transactions on Pattern Analysis and Machine Intelligence,vol.24,no 4,2002,pp.509-512.
6. Tobias O.J,Seara R, “Image Segmentation by Histogram Thresholding Using Fuzzy Sets”,IEEE Transactions on Image Processing,vol.11,no.12,2002,pp.1457-1465.
7. Hussin R,Rizon Jubari M,Ng Wei Kang,Ismail R.C,Kamarudin A, “Digital Image Processing Technique for Object detection from Complex Background Image”,Procedia Engineering,vol.41,2012,pp.340-344.

8. Amal Farag,Li Lu,Holger R,Roth Jaiwin Liu,Evrin Turkbey and Ronald M.Summer, “A Bottom Up Approach for Pancreas Segmentation Using Cascaded Superpixels and Deep Image Patch Labeling”,IEEE Transactions on Image Processing,vol.26,Januray 2017,pp.386-399.
9. Dollar P and Zitnick L, “Fast Edge Detection Using Structured Forests”,IEEE Transactions on Pattern Analysis and Machine Intelligence”,vol.37,no.8,August 2015,pp.1558-1570.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/224382156>

Detection of steel defect using the image processing algorithms

Conference Paper · January 2009

DOI: 10.1109/INMIC.2008.4777721 · Source: IEEE Xplore

CITATIONS
33

READS
4,824

4 authors, including:



Mahrokh Sharifzadeh
Alzahra University

3 PUBLICATIONS 62 CITATIONS

[SEE PROFILE](#)



S. Alirezaee
University of Windsor

51 PUBLICATIONS 301 CITATIONS

[SEE PROFILE](#)



Rassoul Amirfattahi
Isfahan University of Technology

74 PUBLICATIONS 765 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Event Related Potentials and Time Perception Mechanism in Human Beings [View project](#)



Motion Artifact Reduction in Digital Subtraction Angiography Using Image Registration [View project](#)

Detection of Steel Defect Using the Image Processing Algorithms

M. Sharifzadeh
Isfahan University of
Technology,
Isfahan, Iran

S. Alirezaee
Islamic Azad University,
Abhar Branch,
Abhar, Iran

R. Amirfattahi
Isfahan University of
Technology,
Isfahan, Iran

S. Sadri
Isfahan University of Technology,
Isfahan, Iran

Abstract—In this paper, detection and classification of steel surface defects is investigated. Image processing algorithms are applied for detecting four popular kind of steel defects, i.e., hole, scratch, Coil break and rust. The results show that the applied algorithms have a good performance on steel defect detection. Numerical results indicate that the implemented image processing algorithms have 88.4%, 78%, 90.4%, 90.3 % accuracy respectively on the hole, scratch, Coil break and rust defect.

Keywords- Automatic inspection; Image processing; Steel defect.

I. INTRODUCTION

Automatic metal surface inspection is a well known problem and is being considered for more than two decades [1, 2]. The Steel quality control is currently done mainly by human visual inspection. Human inspectors classify the defects according to their cause and origin because the inspection results are used as feedback to correct the manufacturing process. The experience of the inspector is essential, because there are no fixed defect criteria. The inspector's pass/reject decisions seem to be based on the types of defects and their extent, the maximum number of defects per unit of surface area and the total number of defects on the entire inspected strip. In addition, the inspector's knowledge of the customer and the use of the strip have a great impact on the decisions.

As the human visual inspection is an unreliable, tedious and boring task, automation of the visual inspection can provide a reliable quality control system for steel manufacturers. We are aiming through this research to detect steel defects by the image processing algorithm.

This paper is organized as follows. In the next section steel defect types is described. Steel defect detection will be presented in the third section. The numerical results present in the fourth section. Finally, conclusion of the work is discussed in the fourth section.

II. STEEL DEFECTS

It has been determined about 210 defects on the steel surface in the Mobarake steel complex which is the biggest steel manufacturer in the Iran as well as in the Middle East. However, the most important and the most occurred defects are

hole, scratch, Coil break and rust. This research aims to detect these defects on the steel sheet images.

For this research, we have collected an image database of steel defect images. Image database consists of 93 images of real defects which 59 images has been collected from the university of Kanpur (India) [3,4] and the rest has been collected from the Mobarake Steel Complex(Isfahan, Iran). Because of the limited access to the factory, we have generated 157 synthesized images by photo shop software and added to the database. A sample of real and synthesized images of hole defect has been presented in the Fig. 1 and 2.

III. DEFECT DETECTION

Some of the common operations for defect detecting are: Theresholding, Noise removal, Edge detection and Segmentation [5].



Figure 1. A sample of hole defect



Figure 2. A synthesized hole defect using the photo shop software

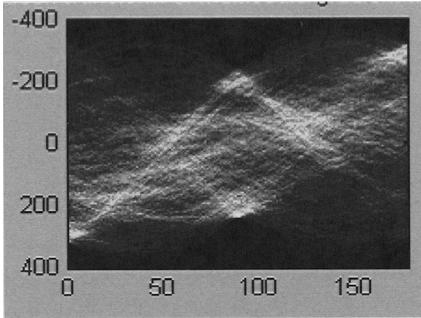


Figure 3. Hough transform of a sample hole defect

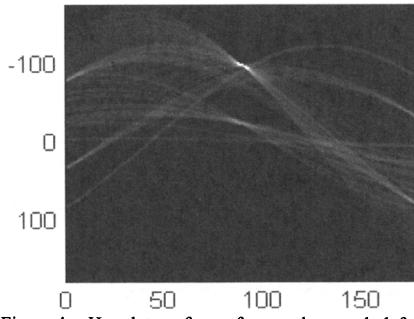


Figure 4. Hough transform of a sample scratch defect

We have tested several image processing algorithm and selected the high performance method. In the following, defect detection will be described with details.

A. Hole and scratch detection

Thresholding is the first step in the hole and scratch detection. The second step is the hough transform [5]. Experimental results show that the hough transform of the hole defect has a Gaussian function with a large σ (Fig. 3) and the scratch defect has a small σ (Fig. 4). The algorithm has 88.4% accuracy on the hole detection and 78% accuracy on the scratch detection.

B. Coil break detection

Pixels of this defect type have been distributed over the wide range of the steel sheet. Experimental results show an evident difference between the histogram of this defect image and the other defects (Fig. 5). For finding this defect, we have defined two thresholds T1 and T2 respectively named up and down thresholds. A defect will be the coil break if two below conditions are satisfied:

1. more than n1 percents of the image pixels have gray level of more than T1 threshold.
2. more than n2 percents of the image pixels have gray level of less than T2 threshold.

For finding parameters i.e. T1, T2, n1, n2, we applied the algorithm for different parameter values and evaluated it on the coil break detection. As can be seen in the Fig. 5, the best parameters are:

$$n1=1.8 \quad n2=2.2$$

$$T1=.9 \quad T2=.35$$

We applied the algorithm on the 250 images of different defects; we achieved 90.4% for the coil break detection.

C. Rust detection

The first step in finding the rust defects is segmentation. For segmentation, image has been thresholded. For thresholding, many methods such as Maximum Entropy Sum Method, Entropic Correlation Method and Renyi Entropy are reported [15]. However, in this research we have chosen Renyi Entropy.

Based on the nature of this defect, it uniformly covers major parts of the steel sheet. Therefore, after binerization (Fig .6), we should count the ones in the image and compare with a threshold. We have defined T1 and T2 respectively for the boundary of rust and the rust defect level. We experimentally found T1=22 and T2= 55 and achieved 90.3% accuracy on the rust detection (Fig. 7).

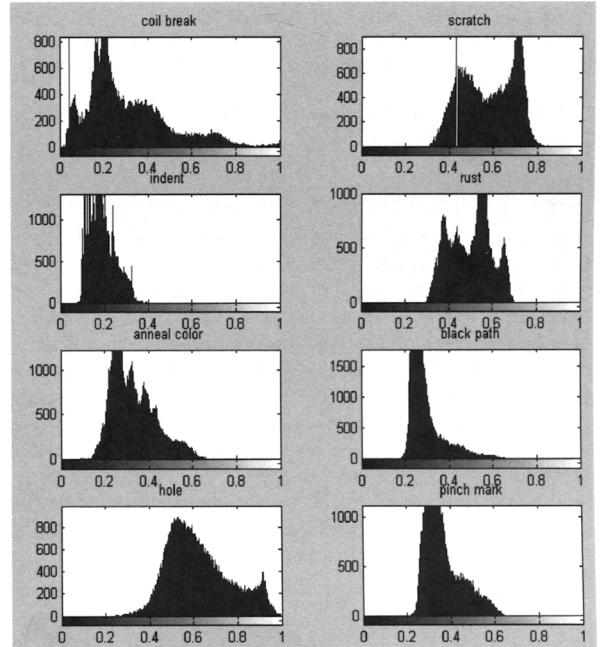
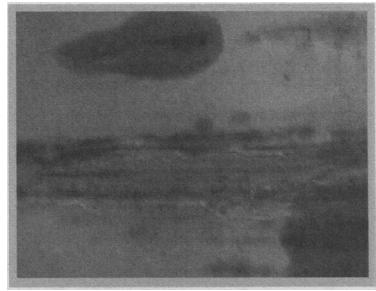
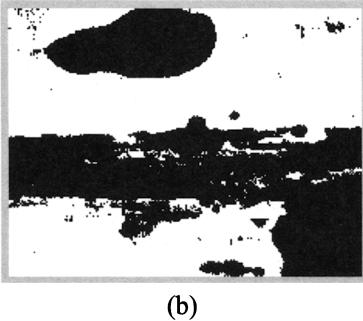


Figure 5. Comparisons of the coil break defect image histogram with the other defects



(a)



(b)

Figure 6. (a)original image of a rust defect (b)Global thresholded image of the rust defect

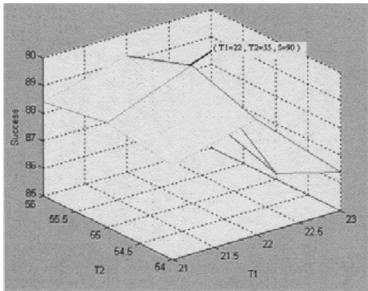


Figure 7. Performance of the rust detection method based on the different T1 and T2

IV. CONCLUSION

In this paper, detection and classification of steel surface defects were investigated. Image processing algorithms are applied for detecting four popular kind of steel defects, i.e., hole, scratch, Coil break and rust. A set of 250 steel defect images were used for testing the proposed method. The results show that the applied algorithms have a good performance on steel defect detection. Numerical results indicate that the implemented image processing algorithms have 88.4%, 78%, 90.4%, 90.3 % accuracy respectively on the hole, scratch, Coil break and rust defect.

ACKNOWLEDGEMENT

The author would like to thank the Mobarake Steel Complex for supporting this research.

REFERENCES

- [1] Chalasani, "Segmentation and performance evaluation of steel defect images", Master's thesis, Department of Mechanical Engineering, Indian institute of Technology, Kanpur, 2000.
- [2] P. Guha, "Automated visual inspection of steel surface, texture segmentation and development of a perceptual similarity measure", Department of Mechanical Engineering, Indian institute of Technology, kanpure, Master's thesis, April 2001.
- [3] J. Jarvinen, "Real-time surface inspection of steel strips", Machine Vision News, vol. 7, pp. 1-5 ,2002.
- [4] A. Kumar, "Real-time based dsp identification system using content-based imaging techniques", Department of electrical Engineering, Indian Institute of Technology, kanpur, Master's thesis, 1999.