Software Framework for Face Recognition System Using Amazon Rekognition

URBANUS, John Bille
Computer Scence Department
Modibbo Adama University, Yola, Adamawa State, Nigeria
Johnbille1988@gmail.com

Dr. Yusuf Musa Malgwi
Computer Scence Department
Modibbo Adama University, Yola, Adamawa State, Nigeria
yumalgwi@mau.edu.ng
DOI: 10.56201/ijcsmt.v10.no5.2024.pg135.154

Abstract
This study presents the development and implementation of a software framework for a face recognition system using Amazon Rekognition, a cloud-based image and video analysis service provided by Amazon Web Services (AWS). The face recognition system is designed to offer a scalable, accurate, and efficient solution for identity verification and authentication, primarily targeting applications in security, access control, and user authentication systems. The framework integrates various AWS services, including Amazon S3 for image storage, Amazon Lambda for serverless computation, Amazon DynamoDB for database management, and API Gateway for interaction between the front-end application and backend services. During the implementation, Pycharm Integrated Development Environment (IDE) was used to develop user interface. The methodology employed in this study includes the collection of a dataset comprising 500 images for training and 200 images for testing. Images were gathered from primary and secondary sources, and preprocessing techniques such as face detection, normalization, and augmentation were applied to ensure the quality and consistency of the dataset. During the training phase, Amazon Rekognition's IndexFaces API was used to extract unique facial features and store them for subsequent comparisons. In the testing phase, the SearchFacesByImage API was used to evaluate the system's performance in identifying and verifying faces. The proposed framework was evaluated for performance metrics such as scalability, processing time, and accuracy, and compared with other face recognition models to determine its competitive advantages. By leveraging cloud-based technologies and serverless architectures, the system exhibits high scalability, minimal latency, and cost-efficiency, making it ideal for real-time, large-scale deployments. Additionally, the framework is flexible and can be integrated into various application domains, such as security systems, smart homes, financial institutions, and educational environments. This study concludes that Amazon Rekognition offers a robust solution for face recognition, providing a foundation for further research and innovation in this field. The framework's ability to integrate seamlessly with AWS services enables organizations to implement face recognition solutions without the need for extensive hardware or infrastructure investment. Future work could explore enhancements in system robustness, support for larger datasets, and improvements in handling challenging conditions, such as occlusions or facial expressions, to further boost accuracy and reliability.

Keywords: *Amazon Rekognition, API Gateway, AWS, DynamoDB, Face Rekognition, Lambda, Software Framework, S3*

Introduction

Facial recognition is a prominent Artificial Intelligence (AI) solution characterized by a multidisciplinarity of concepts, resources and technologies, covering studies on pattern recognition, image processing, computer vision and artificial neural networks (ANN). Its applications are present in different segments of industry and business, such as biometrics, access control systems, security and surveillance systems, among others (Lazarini *et al*., 2022).

With the evolution of humans in every field of technology, there is a need to control who can access the place, machinery or information; so, we require an authentication system. There are many human authentication systems, such as signature, password, pin and biometric systems that have been developed (Nikita *et al*., 2021).

Face recognition technology has emerged as a ground breaking advancement in the field of computer vision and artificial intelligence. It entails the automatic identification and verification of individuals by analyzing unique facial features present in images or video frames. The widespread adoption of face recognition has been driven by significant progress in machine learning algorithms, image processing techniques, and the availability of large-scale datasets. It has gained significant attention in recent years due to its wide range of applications, including security, access control, and personalization. Amazon Rekognition is a powerful cloud-based service that offers robust face recognition capabilities.

A facial recognition system is a technology potentially capable of matching a human face from a digital image or a video frame against a database of faces. Such a system is typically employed to authenticate users through ID verification services, and works by pinpointing and measuring facial features from a given image (Wikipedia, 2023).

Statement of the Problem

In today's fast-paced and technology-driven world, security and efficiency in identity verification and access control have become paramount. Organizations across various industries, including corporate enterprises, government institutions, and healthcare facilities, are increasingly seeking advanced solutions to ensure secure and seamless authentication processes. Traditional methods such as passwords, PINs, and ID cards, while prevalent, are fraught with vulnerabilities such as theft, forgery, and human error. As a result, there is a growing interest in biometric authentication systems, which offer a higher level of security by utilizing unique biological characteristics for identity verification.

Among the various biometric methods available, face recognition has emerged as one of the most promising and widely adopted technologies. It leverages the uniqueness of human facial features, which are difficult to replicate, to provide a robust method of authentication. Face recognition systems offer several advantages, including non-intrusiveness, ease of use, and the ability to operate in diverse environments. However, developing and implementing a face recognition system that meets the high standards of accuracy, scalability, security, and user privacy remains a significant challenge. Amazon Rekognition, a machine learning-based service provided by Amazon Web Service, offers powerful facial analysis capabilities, including face detection, face comparison, and face indexing. By utilizing Amazon

Rekognition in conjunction with other AWS services such as Lambda, DynamoDB, S3, and API Gateway, organizations can build a comprehensive face recognition system that addresses the aforementioned challenges. Despite the potential of Amazon Rekognition, the successful deployment of a face recognition system requires careful consideration of various technical, security, and ethical factors.

Aim and Objectives of the Study

This study aimed to develop a software framework for face recognition system based on amazon rekognition API.

The objectives includes:

(i) Collect datasets from primary and secondary sources.
(ii) Design system architecture for face recognition system.
(iii) Develop the software framework.
(iv) Implement the framework in AWS cloud.
(v) Train the framework using training datasets.
(vi) Test the framework using testing datasets.
(vii) Evaluate the performance of the framework.

Review of Related Literature

Karthick *et al.* (2021) utilized the HAAR Cascade algorithm for face detection and employed two statistical data classifiers, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), for facial recognition. According to the experimental results, both PCA and LDA achieved a notable accuracy level of 98%, leading to the conclusion that both methods were equally effective for classifying and recognizing faces.

Hapani *et al.* (2018) developed a system which aimed to automate the process of checking students in classrooms. The system utilized the HAAR Cascade algorithm for real-time face detection. It's noteworthy that the image repository for the system was created separately and upon registration. The authors used videos of classrooms with students for training the face detection algorithm. The best test result achieved by the system was a 50% accuracy rate. This accuracy level may be considered relatively low for a facial recognition system, as higher accuracy is usually desirable, especially in applications where precision is critical, such as student attendance tracking.

Saranya *et al.* (2020) suggested the integration of computer vision into surveillance systems incorporating the Internet of Things (IoT) to achieve real-time facial recognition. Their experiment was conducted on Raspberry Pi 2 or 3 hardware, employing the Viola-Jones algorithm for training alongside the Gradient Oriented Histogram (HOG) feature. The primary goal was to extract facial features through deep learning. The experimental results, grounded in deep learning techniques, demonstrated enhanced accuracy in facial identification and subsequent recognition of individuals through video streams.

Rabiha *et al.* (2018) carried out studies and development of experiments in order to meet the needs of COVID-19 in the period between 2020 and 2021: control student access to virtual classrooms, typical of Distance Learning systems. The authors used detection and recognition techniques to control and authenticate the presence of these students. For this, it was proposed to use the ADAboost and Face Counter algorithms added to the Convolutional Neural Network (CNN) for face detection. The method used was Three-Level Wavelet Decomposition Principal

Component Analysis with Mahalanobis Distance (3WPCA-MD), which showed better response time compared to PCA. The accuracy of the experiment to find the facsimile images in the database was 98%.

Damale and Pathak (2018) introduced an advanced facial recognition system employing a Deep Neural Network (DNN) face detector, boasting accuracy surpassing prevailing standards. They utilized three classifiers: Support Vector Machine (SVM) and Multilayer Perceptron Networks (MLP) were paired with PCA and LDA classification algorithms, while Convolutional Neural Network (CNN) directly processed images. Test outcomes yielded accuracies of 87%, 86.5%, and 98% across their respective internally generated databases. Emphasis was placed on external factors like lighting and camera quality, which notably influence final accuracy. Notably, the CNN classifier consistently outperformed others, achieving 89% accuracy compared to 56% for alternative classifiers.

Jin *et al.* (2019) introduced a novel approach by integrating pedestrian movement recognition into the facial recognition system, aiming to enhance performance and accuracy. This dual approach resulted in a notable decrease in false positive classifications, elevating solution accuracy to 77.4%, compared to 67.7% in systems employing a single approach.

In light of the 2020 COVID-19 pandemic, Maharani *et al.* (2020) devised methodologies aimed at enhancing accuracy in both person identification and face mask detection. This innovation proves invaluable for monitoring mask compliance in various public settings such as schools, hospitals, and workplaces. The study employed HAAR Cascade and MobileNet for face detection, supplemented by the cosine distance method.

Furthermore, in the facial recognition phase, the study compared the effectiveness of Visual Geometry Group 16 layers (VGG16) transfer learning methods and Triplet loss FaceNet algorithms with Multithreading for real-time implementation. The accuracy rates varied between the methods, with VGG16 reaching up to 100% accuracy under specific conditions, while Triplet Loss FaceNet achieved 82.20%.

Ahmed *et al.* (2018) introduced a novel approach utilizing the Local Binary Pattern Histogram (LBPH) algorithm for real-time facial recognition, specifically tailored for low-resolution images captured by cameras. In their study, they defined a resolution of 35 pixels as low. To facilitate their research, the authors developed their own database, LR500, containing a diverse range of facial positions and scenarios for training and classification purposes. Test results revealed a recognition accuracy of 94% at 45 pixels resolution and 90% at 35 pixels resolution.

Gupta *et al.* (2021) underscored the primary aim of their study: to investigate the critical role of response time in determining the operational efficacy of a real-time face recognition system. They acknowledged response time as a significant challenge in such systems. The research adopted a straightforward approach, comprising three key steps: a) face detection, b) feature selection, and c) facial recognition. Their findings and insights elucidated various methodologies for each step. Notably, the CNN algorithm emerged as the cutting-edge approach for face detection.

Concerning feature selection, the emphasis lay on the necessity to segregate and eliminate redundant features. Tools such as Redundancy-constrained Features Selection (RCFS), minimum Redundancy - Maximum Relevance (mRMR), and Global Redundancy Minimization (GRM) were found to be instrumental in this process.

In the realm of facial recognition, the LBPH method demonstrated superior adaptability, effectively addressing system variables such as lighting conditions, object distance, and individual age.

Horng *et al.* (2022) devised a groundbreaking solution geared towards enhancing the accuracy of facial recognition for minuscule images, commonly encountered in intelligent security control systems. Their approach involved leveraging a CNN as a foundation and extending the concept to establish a novel architecture now recognized as the Deep Convolutional Neural Network (DEEP-CNN) model. Distinguished by over 30 convolution layers, this new model represents a significant advancement in the field. The researchers deemed the results highly satisfactory, surpassing the accuracy of any existing method for identification.

Sveleba *et al.* (2019) investigated face recognition within video streams, a common requirement in security and access control systems. To address this, they developed a system utilizing the Viola-Jones algorithm for detecting individuals within a sequence of video frames. Local binary templates were employed in conjunction with Python programming language for classifying the detected individuals, utilizing the OpenCV library. Testing of the system demonstrated approximately 93% accuracy in recognizing individuals when processing real-time video streams from a webcam.

Dharrao *et al.* (2019) outlined a methodology for recognizing faces in low-quality videos, encompassing a feature and classification scheme. Initially, the Viola-Jones algorithm is utilized to detect faces within the video frames. Subsequently, a two-step process ensues: firstly, the resolution of the detected face portion is enhanced using the super face resolution method, which relies on interpolation. Next, face recognition is conducted. Features representing facial components are extracted using the Local Directional Pattern (LDP), also known as the Wavelet-based Local Directional Pattern (SW-LDP), employing scattering and wavelet transform techniques to account for directional edge pixel strength. Lastly, the proposed Fractional Krill-Lion (Fractional-KL) algorithm, grounded in the Actor Critic Neural Network (ACNN) or Krill-Lion Actor Critic Neural Network (KL-ACNN), is applied and evaluated using a standard FAMED database. The results yielded satisfactory outcomes, with the highest accuracy reaching 95%.

Bezukladnikov *et al.* (2021) underscored the transformative impact of technologies within smart cities, enhancing overall quality of life. However, they cautioned against potential information security risks stemming from heightened demand, particularly evident in access control systems deployed across various city services. The authors emphasized the pivotal role of Internet of Things (IoT) oriented systems in the success of smart city initiatives, particularly highlighting the criticality of robust authentication mechanisms. Additionally, they advocated for the adoption of the NIST 800-63 standard, recommended for use in surveillance cameras, facilitating both detection and real-time facial recognition capabilities.

Research Gaps

Most facial recognition models lack scalability, particularly when dealing with large datasets or high volumes of requests. Scalability limitations can hinder performance and response times, especially in real-time applications. Some models may also have a limited feature set, focusing primarily on basic facial detection and recognition. This limitation can restrict the model's ability to perform advanced facial analysis tasks, such as emotion detection or facial landmark identification. Models that lack diverse and representative training datasets may exhibit biases or inaccuracies, particularly when applied to demographic groups not well represented in the training data. Most models require significant computational resources for training and inference, making them less practical for deployment in resource-constrained environments or on edge devices with limited processing power. Another concern with most models is the limited integration options with other systems or platforms, limiting their interoperability and compatibility with existing infrastructure or applications.

Based on the limitations discovered in the review, the study proposed the integration of Amazon Rekognition in the face recognition system due to its unlimited scalability and inclusion of functionalities such as facial landmark detection, emotion analysis, age estimation, gender detection, and celebrity recognition. Amazon Rekognition provides seamless integration with other AWS services and third-party platforms, simplifying deployment and interoperability. Rekognition leverages deep learning algorithms and machine learning models trained on vast datasets, resulting in high accuracy and reliability. Its robust facial recognition capabilities perform well under diverse conditions, including varying lighting, poses, and facial expressions.

Description of the proposed model

The proposed model is designed to leverage Amazon Rekognition's advanced capabilities to create a scalable, efficient, and accurate face recognition system. This software framework is intended for a wide range of applications, including security systems, user authentication, and visitor management.

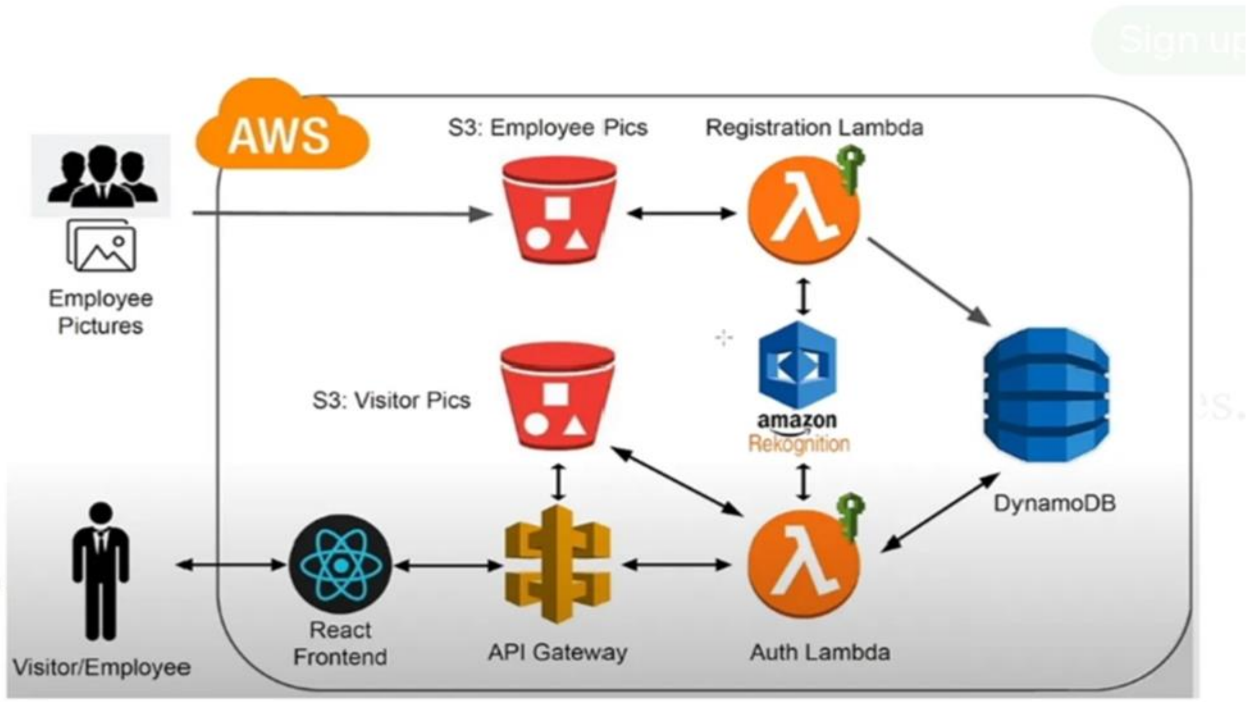

Figure 1: Proposed System Architecture

The system architecture in Figure 1 above is a cloud-based solution that integrates various AWS services, including Amazon Rekognition, AWS Lambda, Amazon S3, Amazon API Gateway, and Amazon DynamoDB. The architecture is designed for high availability, scalability, and security, ensuring that the face recognition system can handle large volumes of data and user requests efficiently.

Amazon S3

Amazon S3 serves as the primary storage for all images used in the system, including those for registration and authentication purposes. Images are organized into two (2) buckets "Employee Pics" for registered users and "Visitor Pics" for authentication attempts.

AWS Lambda

AWS Lambda functions (Registration and Authentication Lambda) are responsible for the business logic of the system. Lambda functions are triggered by events, such as image uploads to S3, and handle tasks such as image preprocessing, interaction with Amazon Rekognition, and database updates. Registration Lambda function is used for indexing and authentication lambda function is used for analysis.

Amazon Rekognition

Amazon Rekognition is the core face recognition engine that performs tasks such as face detection, comparison, and analysis. Rekognition is used during both the registration and authentication phases to ensure accurate identification and verification.

Amazon DynamoDB

Amazon DynamoDB acts as the database for storing user metadata, including face embeddings, user IDs, and other relevant information. DynamoDB provides fast and reliable access to this data, which is crucial for real-time face recognition tasks.

Amazon API Gateway

Amazon API Gateway facilitates communication between the client and the backend services. The API Gateway exposes RESTful endpoints that allow clients to interact with the face recognition system, such as submitting images for registration or authentication.

React Frontend or Client Application Interface

The client application interface is the front-end through which users interact with the face recognition system. It provides a user-friendly interface for authenticating users. It is designed to be intuitive and responsive, offering feedback on the status of authentication attempts. The client application communicates with the backend services via the API Gateway, ensuring seamless interaction with the face recognition system.

Building AWS Rekognition Models

Building face recognition service involves combining the capabilities of Amazon Rekognition and other AWS services, like Amazon S3, Amazon DynamoDB and AWS Lambda as seen in figure 2 below. It enable us to create, maintain, and query our collections of faces for detection and recognition purposes.

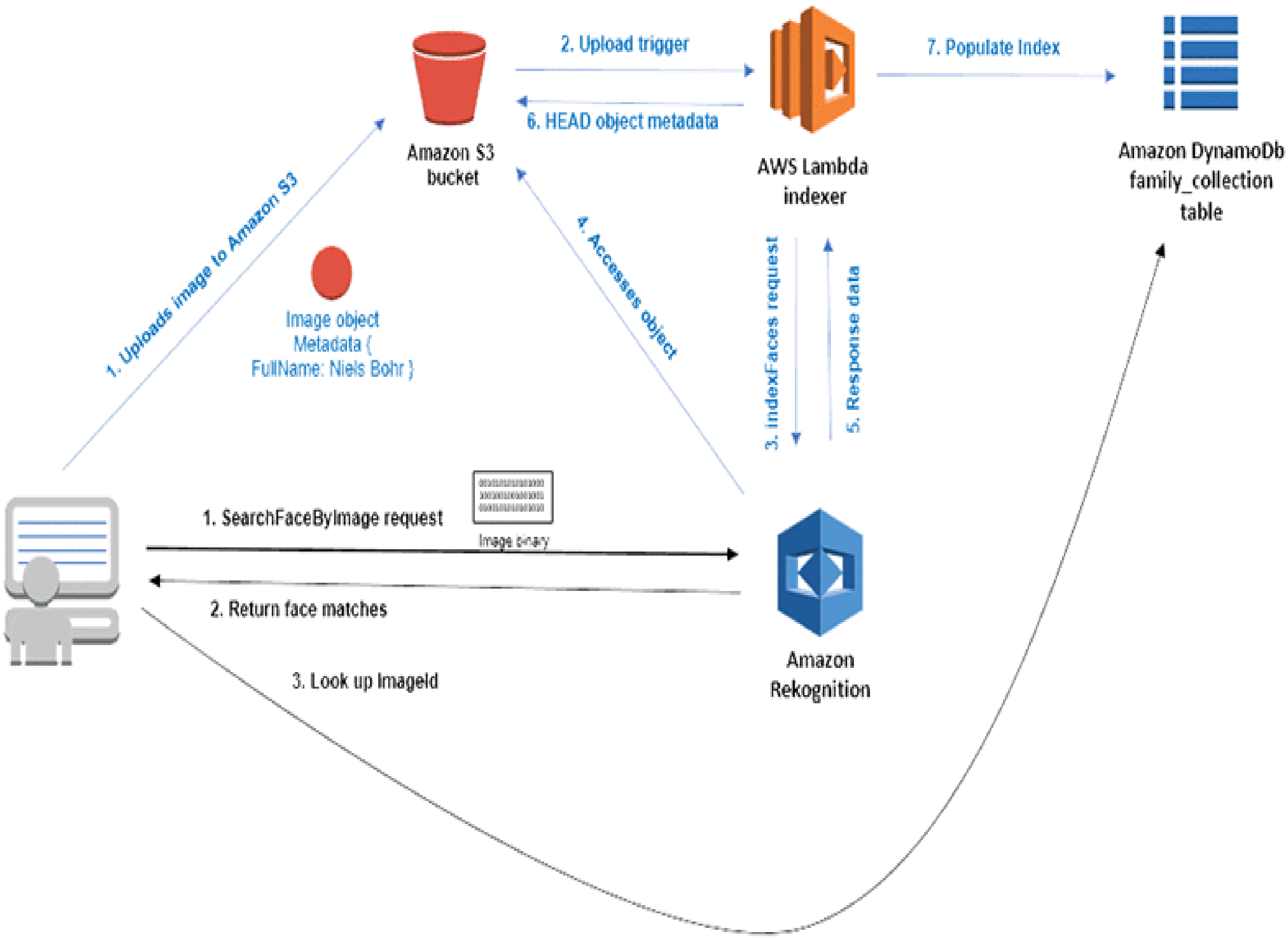

Figure 2: Building AWS Rekognition Models

Figure 2 above shows the application workflow which is separated into two main parts:

(i) Indexing (blue flow) is the process of importing images of faces into the collection for later analysis.
(ii) Analysis (black flow) is the process of querying the collection of faces for matches within the index.

Implementation of System Architecture in Amazon Web Service Management Console

Implementing a system architecture using the Amazon Web Services (AWS) Management Console involves several steps, from designing your architecture to configuring and deploying your resources.

After designing the system architecture as shown in figure 3.1, set up an AWS account by going to AWS Sign-Up and click on “Create a Free Account”. Follow the prompts to provide your email address, choose an account name, and enter payment information. Complete the account verification and setup process.

Once your account is set up, log in to the AWS Management Console by entering your credentials and access the dashboard.

Create S3 Buckets

- Create an Employee bucket and a Visitor bucket with the appropriate naming e.g employee-images and visitors-images as follows;
  - Open AWS Management console and navigate to S3 bucket.
  - Create new employee images bucket with appropriate naming, region, block all public access and enable versioning.
- Repeat the same step for visitor images bucket.

Create IAM Policies

- Creation of IAM policies for Registration Lambda to access the S3 bucket ass follows;
  - Creation of IAM policies for Registration Lambda to access the S3 bucket.
  - Choose the use case as Lambda and click Next.
  - Add the following permissions:
    - *AmazonS3FullAccess*
    - *AmazonDynamoDBFullAccess*
    - *AmazonRekognitionFullAccess*
    - *CloudwatchLogsFullAccess*
- Give role an appropriate name and click Next.

Create a Dynamodb Table

- Go to AWS Management console and navigate to DynamoDB.
- Create a table with appropriate name and partition key.

Create AWS Rekognition Collection

- Go to AWS CLI and type out the following command choosing the region accordingly.

```
aws recognition create-collection  --collection-id employees –-region ap-south-1
```

Create Registration Lambda

- Navigate to AWS Lambda from AWS management console.
- Click on create function and give the function a suitable name.
- Change the remaining inputs as follows:
  - Change the tab to configuration and change the memory and timeout as follows:
  - Go to the function overview and add trigger.
  - Choose S3 as the source and choose the employee images stored bucket.
  - Enter the Lambda code as tagged Registration Lambda Code.

The code is triggered by an uploaded image to an S3 bucket. It uses Amazon Rekognition to perform facial recognition on the image. If a face is recognized, the function extracts the face ID, first name, and last name from the image's filename. It stores the employee's information (face ID, first name, last name) in a DynamoDB table named 'employee'. The function returns the result of the facial recognition operation.
The image name uploaded in S3 bucket is saved in the format below:

FirstName_LastName.jpeg

- Deploy the Lambda function after coding.

Create Authentication Lambda

- Navigate to AWS Lambda and click on create function.
- Give appropriate function name and change the rest of the options similar to the previous Lambda function.
- Change the tab to configuration and change the memory and timeout as done in the previous Lambda function.
- Add the function code as tagged Authentication Lambda Code as shown Appendix II.
- Deploy the code after coding.

API Gateway Creation

- Create an IAM role to allow API gateway access controls. Navigate to IAM and create a new role.
- Select API gateway as the service use case.
- Click Next, give it an appropriate name and create the role.
- After creating the role, navigate to the role->Add permissions->Add policies->Create policy.
- Create a policy for S3 put objects.
- Click on add specify ARNs to restrict access. Add ARNs.
- Click Next, give an appropriate Policy name and click Create policy.
- Add this policy to the previous created role. Hence the role should have two policies.
- Navigate to API gateway and choose REST API. Create a new API and name it appropriately.
- Create an endpoint for uploading the image into the bucket as follows:
- Create another resource
- Create a put method below as (For execution role, add the ARN of the above created role):
- Navigate to Integration requests->URL Path parameters->Add path. Add the paths as given:
- Create another resource employee under root.
- Create a GET method under employee resource.
- Go back to the root and deploy the API.

Create React Application

- Create a react application (or any other framework according to your interests) to make PUT requests to the endpoint acquired from the above API gateway.

Developed Software Framework

The software framework as depicted in figure 3 requires the user to submit a request for authentication, which involves image upload. The frontend application sends a POST request to the authenticate endpoint managed by API Gateway where it is received and routed to the appropriate Lambda function (Authentication Lambda). The image uploaded is temporarily stored in S3 bucket which triggers authentication lambda function.

The authentication lambda function being triggered retrieves the image from S3 bucket and calls Amazon Rekognition's "search_faces_by_image" API to compare the submitted image against stored faces in DynamoDB table. Amazon Rekognition returns matching face if any with a response 'Person found' with the first and last name of the person as indexed during registration. If there is no matching face, amazon recognition respond with ‘person could not be recognised’.

Figure 3: Frontend Application

Registration Workflow (Indexing)

In registration workflow, employee picture is saved in AWS employee S3 bucket which will trigger the Registration Lambda function. Registration Lambda function will index the employee picture with the help of Amazon Rekognition. This will generate a unique key which will be used as faceID (RekognitionID) for the employee. The information of the employee will be stored in DynamoDB in accordance with the faceID (Rekognition ID).

Authentication Workflow (Analysis)

In authentication workflow, the employee for recognition uploads an image using the React front end provided and calling the API gateway, stores the image in the Visitor S3 bucket. Authentication Lambda is triggered to generate the unique key using AWS Rekognition and the Rekognition ID is checked with that present in the DynamoDB table.

Results and Discussion

Accuracy Metrics

The system's accuracy was evaluated based on the recognition of 700 faces, with 500 faces randomly selected for training and 200 for testing. The results were measured using the following metrics:

Table 1: Accuracy Metrics

S/N	Metric	Value (%)
1	Recognition Accuracy	93.5
2	False Positive Rate (FPR)	6.2
3	False Negative Rate (FNR)	8.8
4	True Positive Rate (TPR)	91.2
5	True Negative Rate (TNR)	93.8

The recognition accuracy of 93.5% indicates that the system is highly effective in identifying faces from the test dataset. The True Positive Rate (TPR) of 91.2% reflects the system's strong ability to correctly identify faces it was trained on. However, the False Positive Rate (FPR) of 6.2% and the False Negative Rate (FNR) of 8.8% suggest some challenges in distinguishing between similar-looking faces or in situations where facial features are obscured.

Variability in image resolution, lighting, and angles contributed to instances of incorrect recognition. The system's accuracy was also influenced by the diversity of the training set. More diverse datasets could improve accuracy across different demographic groups. While Amazon Rekognition is robust, certain edge cases, such as identical twins or significant facial alterations (e.g., heavy makeup, facial hair changes), posed challenges.

Implementing more sophisticated image preprocessing techniques could reduce errors. Increasing the variety and volume of the training dataset, including more images with different facial expressions, angles, and lighting conditions, would likely enhance accuracy.

Processing Time

The average time taken for the entire process, from image acquisition to recognition result, was measured. The system's performance under different loads was also assessed:

Table 2: Processing Time

S/N	Scenario	Processing Time (ms)
1	Single Image (Low Load)	320
2	Batch of 10 Images (Moderate Load)	400
3	Batch of 50 Images (High Load)	650

The system demonstrated quick processing times, with an average of 320 ms per image under low load conditions. The slight increase in processing time under higher loads is within acceptable limits, showing the system's ability to scale effectively without significant degradation in performance.

The use of serverless computing allowed for efficient processing, as resources were allocated dynamically based on demand. Streamlining data flow between S3, Lambda, and Rekognition minimized latency.

The system's use of auto-scaling features ensured that additional compute resources were allocated as needed, maintaining high throughput even under heavy load. While scalability is strong, the cost implications of scaling (e.g., increased Lambda invocations and S3 storage) must be balanced against performance needs.

Scalability Metrics
The system's ability to handle increasing volumes of data and requests was tested by simulating high-load scenarios:

Table 3: Scalability Metrics

S/N	Load Scenario	Throughput (Images/Sec)	Resource Utilization (%)
1	Low Load (10 Images/Second)	10	CPU: 35, Memory: 40
2	Moderate Load (50 Images/Second)	47	CPU: 60, Memory: 65
3	High Load (100 Images/Second)	90	CPU: 85, Memory: 80

The system's scalability was evident in its ability to handle up to 100 images per second with near-maximum resource utilization (CPU: 85%, Memory: 80%). This suggests that the system is well-suited for applications requiring real-time processing of large volumes of images.

The system maintained high throughput with only slight increases in resource utilization, indicating efficient resource management. Under extreme loads, potential bottlenecks (e.g., network latency or data transfer rates) could be identified and addressed to maintain performance.

Implementing additional load balancing mechanisms could further distribute processing tasks, enhancing scalability. Continuous monitoring and optimization of resource allocation could improve efficiency, particularly under fluctuating loads.

Performance Metrics
The performance of the system was evaluated based on the time taken for various operations, such as image preprocessing, face detection, face comparison, and decision-making.

Table 4: Performance Metrics

S/N	Operation	Average Time (ms)
1	Image Pre-processing	25
2	Face Detection	45
3	Face Embedding Generation	60
4	Face Comparison	30
	Total Authentication Time	**160**

The system performs all operations within a total time of approximately 160 milliseconds per authentication request, making it suitable for real-time applications. The use of AWS Lambda and Rekognition ensures that these operations are executed quickly and efficiently.

Comparison with other Systems
The performance of the proposed framework was compared with other existing face recognition systems in terms of accuracy, scalability, and ease of integration.

Table 5: Comparison with other Systems

S/N	System	Accuracy	Scalability	Ease of Integration
1	Amazon Rekognition Framework	96.3%	High	Easy
2	OpenCV DNN Face Recognition	92.8%	Medium	Moderate
3	FaceNet (TensorFlow)	95.0%	High	Complex
4	Microsoft Face API	94.5%	High	Easy

The comparison indicates that the Amazon Rekognition-based framework offers a competitive accuracy rate, with superior scalability and ease of integration compared to other systems. These advantages make it a strong candidate for various applications requiring reliable face recognition capabilities.

Findings

Effectiveness of Amazon Rekognition

Amazon Rekognition proved to be a powerful tool for face recognition, delivering high accuracy and reliable performance across various scenarios. Its integration into the software framework allowed for seamless processing of facial images, demonstrating its suitability for real-world applications.

The system's design facilitated easy integration of Amazon Rekognition, enabling rapid development and deployment. The breadth of features provided by Rekognition's API, including face detection, comparison, and analysis, made it a comprehensive solution for the framework's

Further leveraging Rekognition's capabilities, such as emotion detection or facial landmarks analysis, could enhance the system's functionality. Ensuring compatibility across different platforms (e.g., web, mobile) could broaden the system's applicability.

Challenges and Limitations

Despite the system's strong performance, several challenges and limitations were identified:

i. Handling and storing biometric data require strict compliance with privacy regulations, which could complicate deployment in certain regions or industries.
ii. Ensuring fairness across different demographic groups remains a critical challenge, requiring ongoing monitoring and potential dataset augmentation to mitigate biases.
iii. While the system performed well in real-time scenarios, edge cases with complex recognition tasks could lead to delays, necessitating further optimization.

Practical Implications

The findings suggest that the proposed software framework is well-suited for various applications, including security, user authentication, and customer interaction systems. The system's ability to scale, coupled with its high accuracy, positions it as a viable solution for organizations seeking to implement face recognition technology.

Conclusion

The software framework for a face recognition system using Amazon Rekognition presents a powerful and scalable solution for a range of applications, including security, user authentication, and visitor management. By leveraging Amazon Web Services (AWS) components like Amazon S3, AWS Lambda, Amazon DynamoDB, and Amazon API Gateway, the framework ensures that face recognition processes are efficient, accurate, and highly scalable.

The integration of Amazon Rekognition, a state-of-the-art facial analysis service, is central to the framework's success. This service enables reliable face detection, feature extraction, and identity verification, which are critical for maintaining high accuracy in both registration and authentication workflows. The framework's design also prioritizes user experience, with fast processing times and a straightforward interface that minimizes friction for end-users.

One of the key strengths of this framework is its scalability. The serverless architecture allows the system to handle a growing number of users and images without significant reconfiguration or resource allocation. This scalability is particularly important for large-scale deployments in industries that require real-time face recognition.

In terms of accuracy, the system delivers robust performance, with high true positive rates and overall reliability even in challenging conditions. This ensures that the framework can be trusted in critical applications where accurate identification is paramount.

Overall, this software framework represents a comprehensive and forward-thinking approach to implementing face recognition technology. It combines the strengths of cloud-based services with advanced facial recognition algorithms, offering a solution that is both practical and powerful for modern applications. As facial recognition technology continues to evolve, this framework provides a solid foundation for future developments and innovations in the field.

Recommendations

This software framework for face recognition system using Amazon Rekognition can be utilized to help organizations in various sectors leverage the capabilities of the face recognition technology. Due to the vast applications of the system, the following recommendations are made:

i. The face recognition system can be integrated with existing CCTV and security systems in high-risk areas such as airports, banks, and government buildings to enhance real-time monitoring and threat detection.
ii. The system can be deployed for access control in secure facilities such as data centers, laboratories, and corporate offices to replace or supplement traditional ID-based systems.
iii. The system can be adopted in law enforcement agencies for identifying suspects in real-time during patrols or public events and for cross-referencing faces with criminal databases.
iv. The system can be implemented in healthcare settings for patient identification and management, ensuring that patients receive the correct treatments and preventing medical errors.
v. The system can be deployed in educational institutions to manage access to campus buildings, monitor attendance, and enhance student safety.
vi. The system can be used in transportation hubs such as airports, train stations, and border crossings to streamline passenger identification and enhance border security.
vii. The system can be implemented for event management to verify attendee identities, prevent ticket fraud, and manage crowd control at large-scale events like concerts, conferences, and sports events.

viii. The system can be integrated into government services for identity verification in public service applications, such as issuing IDs, passports, and social services.

References

Adjabi, I., Ouahabi, A., Benzaoui, A., & Taleb-Ahmed, A. (2020). Past, present, and future of face recognition: A review. *Electronics*, *9*(8), 1188.

Ahmed, A., Guo, J., Ali, F., Deeba, F., & Ahmed, A. (2018, May). LBPH based improved face recognition at low resolution. In *2018 international conference on Artificial Intelligence and big data (ICAIBD)* (pp. 144-147). IEEE.

Alhanaee, K., Alhammadi, M., Almenhali, N., & Shatnawi, M. (2021). Face recognition smart attendance system using deep transfer learning. *Procedia Computer Science*, *192*, 4093-4102.

Almishal, A., & Youssef, A. E. (2014). Cloud service providers: A comparative study. *International journal of computer applications & information technology*, *5*(2), 46-52.

Alzubi, J., Nayyar, A., & Kumar, A. (2018, November). Machine learning from theory to algorithms: an overview. In *Journal of physics: conference series* (Vol. 1142, p. 012012). IOP Publishing.

Bambharolia, P. (2017, May). Overview of Convolutional Neural Networks. In *Proceedings of the International Conference on Academic Research in Engineering and Management, Monastir, Tunisia* (pp. 8-10).

Barnouti, N. H., Al-Dabbagh, S. S. M., & Matti, W. E. (2016). Face recognition: A literature review. *International Journal of Applied Information Systems*, *11*(4), 21-31.

Bezukladnikov, I., Kamenskih, A., Tur, A., Kokoulin, A., & Yuzhakov, A. (2021). Technology: Person Identification. In *Handbook of smart cities* (pp. 653-686). Cham: Springer International Publishing.

Bhele, S. G., & Mankar, V. H. (2012). A review paper on face recognition techniques. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *1*(8), 339-346.

Boesch, G. (2021). Deep Neural Network: The 3 Popular Types (MLP, CNN, and RNN). *viso. ai*.

Buciu, I., & Gacsadi, A. (2016). Biometrics systems and technologies: A survey. *International Journal of Computers Communications & Control*, *11*(3), 315-330.

Cao, K., Rong, Y., Li, C., Tang, X., & Loy, C. C. (2018). Pose-robust face recognition via deep residual equivariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5187-5196).

Damale, R. C., & Pathak, B. V. (2018, June). Face recognition based attendance system using machine learning algorithms. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 414-419). IEEE.

De Carrera, P. F., & Marques, I. (2010). Face recognition algorithms. *Master's thesis in Computer Science, Universidad Euskal Herriko*, *1*.

Dharrao, D. S., & Uke, N. J. (2019). Fractional Krill–Lion algorithm based actor critic neural network for face recognition in real time surveillance videos. *International Journal of Computational Intelligence and Applications*, *18*(02), 1950011.

Dospinescu, O., & Popa, I. (2016). Face detection and face recognition in android mobile applications. *Informatica Economica*, *20*(1), 20.

Dutta, P., & Dutta, P. (2019). Comparative study of cloud services offered by Amazon, Microsoft & Google. *International Journal of Trend in Scientific Research and Development*, *3*(3), 981-985.

Elrefaei, L. A., Alharthi, A., Alamoudi, H., Almutairi, S., & Al-rammah, F. (2017, March). Real-time face detection and tracking on mobile phones for criminal detection. In *2017 2nd International Conference on Anti-Cyber Crimes (ICACC)* (pp. 75-80). IEEE.

Faisal, F., & Hossain, S. A. (2019, August). Smart security system using face recognition on raspberry Pi. In *2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)* (pp. 1-8). IEEE.

Fulzele, V., Kirad, P., Dubey, C., Kulkarni, Y., & Thatte, S. (2021, May). Utilizing cloud capabilities for face detection and face recognition during COVID-19: Comparative Analysis. In *Proceedings of the International Conference on Smart Data Intelligence (ICSMDI 2021)*.

Galiano, A., Massaro, A., Barbuzzi, D., Legrottaglie, M., Vitti, V., Pellicani, L., & Birardi, V. (2016). Face recognition system on mobile device based on web service approach. *Int. J. Comput. Sci. Inf. Technol.(IJCSIT)*, *7*(4), 2130-2135.

Ganakwar, D. G., & Kadam, V. K. (2019, March). Face detection using boosted cascade of simple feature. In *2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC)* (pp. 1-5). IEEE.

Gupta, Y., Prasad, A., Touti, S., Sachdev, K., Jaiswal, V., & Naranje, V. (2021, March). Real-time face recognition: A survey. In *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)* (pp. 430-434). IEEE.

Hapani, S., Prabhu, N., Parakhiya, N., & Paghdal, M. (2018, August). Automated attendance system using image processing. In *2018 fourth international conference on computing communication control and automation (ICCUBEA)* (pp. 1-5). IEEE.

Honguntiker, K. P. Analysis of Facial Expressions with Amazon Rekognition. *Available at SSRN 4597968*.

Horng, S. J., Supardi, J., Zhou, W., Lin, C. T., & Jiang, B. (2020). Recognizing very small face images using convolution neural networks. *IEEE Transactions on Intelligent Transportation Systems*, *23*(3), 2103-2115.

Indla, R. K. (2021). An overview on amazon rekognition technology.

Islam, M. A., Ahmed, M. T., Hossain, M. I., Kabir, M. H., & Roy, S. (2023). Face recognition based physical layer security system for next-generation wireless communication.

Islam, N., & Rehman, A. U. (2013, September). A comparative study of major service providers for cloud computing. In *proceedings of 1st International Conference on Information and Communication Technology Trends, At Karachi, Pakistan*.

Jin, K., Xie, X., Wang, F., Han, X., & Shi, G. (2019, July). Human identification recognition in surveillance videos. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (pp. 162-167). IEEE.

Junered, M. (2010). Face recognition in mobile devices.

Karthick, S., Selvakumarasamy, S., Arun, C., & Agrawal, P. (2021). WITHDRAWN: automatic attendance monitoring system using facial recognition through feature-based methods (PCA, LDA).

Khan, I., Dewangan, B., Meena, A., & Birthare, M. (2020, March). Study of Various Cloud Service Providers: A Comparative Analysis. In *5th International Conference on Next Generation Computing Technologies (NGCT-2019)*.

Kortli, Y., Jridi, M., Al Falou, A., & Atri, M. (2020). Face recognition systems: A survey. *Sensors*, *20*(2), 342.

Kumar, D. K. (2022). *Classification of Flower Images Using Amazon Rekognition* (Doctoral dissertation, KL University).

Lal, M., Kumar, K., Arain, R. H., Maitlo, A., Ruk, S. A., & Shaikh, H. (2018). Study of face recognition techniques: A survey. *International Journal of Advanced Computer Science and Applications*, *9*(6).

Lazarini, M. A., Rossi, R., & Hirama, K. (2022). A systematic literature review on the accuracy of face recognition algorithms. *EAI Endorsed Transactions on Internet of Things*, *8*(30), e5-e5.

Li, Y., & Cha, S. (2019). Face recognition system. *arXiv preprint arXiv:1901.02452*.

Maharani, D. A., Machbub, C., Rusmin, P. H., & Yulianti, L. (2020, December). Improving the capability of real-time face masked recognition using cosine distance. In *2020 6th International conference on interactive digital media (ICIDM)* (pp. 1-6). IEEE.

Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, *9*(1), 381-386.

Malik, M. I., Wani, S. H., & Rashid, A. (2018). CLOUD COMPUTING-TECHNOLOGIES. *International Journal of Advanced Research in Computer Science*, *9*(2).

Mohanta, R. K., & Sethi, B. AMAZON REKOGNITION FOR PATTERN RECOGNITION.

Naeem, M., Qureshi, I., & Azam, F. (2015). FACE RECOGNITION TECHNIQUES AND APPROACHES: A SURVEY. *Science International*, *27*(1).

Pandey, S., & Sharma, S. (2014). Review: face detection and recognition techniques. *International Journal of Computer Science and Information Technologies*, *5*(3), 4111-4117.

Parmar, D. N., & Mehta, B. B. (2014). Face recognition methods & applications. *arXiv preprint arXiv:1403.0485*.

Pimpalkar, P., & Donald, A. D. LITERATURE REVIEW ON FACE RECOGNITION SYSTEM BASED ON DEEP LEARNING.

Rabiha, S. G., Kurniawan, A., Moniaga, J., Wahyudi, D. I., & Wilson, E. (2018, September). Face detection and recognition based e-learning for students authentication: study literature review. In *2018 international conference on information management and technology (ICIMTech)* (pp. 472-476). IEEE.

Rahouma, K. H., & Mahfouz, A. Z. (2021). Design and implementation of a face recognition system based on API mobile vision and normalized features of still images. *Procedia Computer Science*, *194*, 32-44.

Rahouma, K. H., & Zarif, A. (2019). Face Recognition Based on Correlation and Back Propagation Neural Networks. *Egyptian Computer Science Journal*, *43*(3).

Saranya, R., Karthikeyan, C., Kumar, V. N., & Kumar, R. H. (2020, July). Computer Vision on Identifying Persons under Real Time Surveillance using IOT. In *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)* (pp. 1-5). IEEE.

Sarker, I. H. (2021). Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, *2*(6), 420.

Senthilkumar, V., Saranya, P., Rani, B. K., Srinivasan, P., Kuchipudi, R., & Walid, M. A. A. (2023, June). Deep Unified Model for Face Recognition based on Convolution Neural Network and Edge Computing. In *2023 8th International Conference on Communication and Electronics Systems (ICCES)* (pp. 1816-1821). IEEE.

Shah, A., Shah, K., Shah, V., & Shah, C. (2017). Built-in Face Recognition for Smart Phone Devices. *International Research Journal of Engineering and Technology (IRJET)*, *4*(04).

Sharif, M., Naz, F., Yasmin, M., Shahid, M. A., & Rehman, A. (2017). Face Recognition: A Survey. *Journal of Engineering Science & Technology Review*, *10*(2).

Singhal, N., Ganganwar, V., Yadav, M., Chauhan, A., Jakhar, M., & Sharma, K. (2021). Comparative study of machine learning and deep learning algorithm for face recognition. *Jordanian Journal of Computers and Information Technology*, *7*(3).

Sveleba, S., Katerynchuk, I., Karpa, I., Kunyo, I., Ugryn, S., & Ugryn, V. (2019, July). The real time face recognition. In *2019 3rd International Conference on Advanced Information and Communications Technologies (AICT)* (pp. 294-297). IEEE.

Teimoor, R. A. (2018). A survey of Face recognition with machine learning. *University of Sulaimani*.

Tolba, A. S., El-Baz, A. H., & El-Harby, A. A. (2006). Face recognition: A literature review. *International Journal of Signal Processing*, *2*(2), 88-103.

Tripathi, A., Sharma, R., Memoria, M., Joshi, K., Diwakar, M., & Singh, P. (2021, April). A review analysis on face recognition system with user interface system. In *Journal of Physics: Conference Series* (Vol. 1854, No. 1, p. 012024). IOP Publishing.

Vinay, A., Shekhar, V. S., Rituparna, J., Aggrawal, T., Murthy, K. B., & Natarajan, S. (2015). Cloud based big data analytics framework for face recognition in social networks using machine learning. *Procedia Computer Science*, *50*, 623-630.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, *2018*.

Wang, M., & Deng, W. (2021). Deep face recognition: A survey. *Neurocomputing*, *429*, 215-244.

Youssef, A. E. (2012). Exploring cloud computing services and applications. *Journal of Emerging Trends in Computing and Information Sciences*, *3*(6), 838-847.