AWS REKOGNITION
Face recognition case study

Clustrex Data Private Limited
https://www.clustrex.com/

What is AWS Rekognition?

AWS Rekognition is a cloud-based computer vision service offered by Amazon Web Services (AWS) that allows developers to easily integrate image and video analysis capabilities into their applications. It leverages **machine learning (ML)** to detect, identify, and analyze objects, people, scenes, and activities in visual data.

AWS Rekognition offers powerful image and video analysis without requiring expertise in computer vision or deep learning. Its easy-to-use APIs simplify adding visual recognition capabilities to web, mobile, and enterprise applications.

Key Features:

- **Custom Models:** Use **Rekognition Custom Labels** to create custom ML models with your own data, without needing ML experience.
- **Seamless Integration:** Rekognition integrates with **Amazon S3** and **AWS Lambda** for scalable and cost-effective visual analysis. It can analyze data directly from S3 or perform real-time video analysis via **Kinesis Video Streams**.

This makes it ideal for building automated, reliable, and scalable solutions across various industries.

Core Capabilities

1. **Face Detection and Recognition**
   - **Face Detection**: Identifies human faces in images or videos and provides details like facial landmarks (eyes, nose, mouth), emotional expressions (happy, sad, etc.), and demographic estimates (age range and gender).
   - **Face Recognition**: Matches detected faces with a pre-indexed collection of known faces for identity verification, attendance tracking, or personalized user experiences. It supports **face indexing**, allowing fast searches across large datasets.
2. **Object and Scene Detection**
   - Recognizes objects (e.g., laptops, cars, pets) and scenes (e.g., urban areas, nature landscapes) in images and videos. This helps categorize media automatically, facilitating **content management** for streaming services, retail analytics, or automated tagging systems. Multiple objects can be detected within a single frame or image.
3. **Text Recognition (OCR)**
   - Extracts text from images or video frames, including **handwritten** or **printed text** in signs, documents, or license plates. AWS Rekognition supports **multi-language OCR**, making it suitable for **invoice processing**, **digital record-keeping**, and **translation** workflows across regions.

4. **Moderation (Explicit Content Detection)**
    - Automatically detects **explicit content** (e.g., nudity, violence, suggestive imagery) to ensure compliance with regulations and protect brand integrity. It is widely used by **social media platforms** and **advertisers** to filter out inappropriate or offensive content.
5. **Celebrity Recognition**
    - Identifies **well-known public figures** in both images and videos. This feature helps media companies **tag celebrities** during events or sports coverage and is frequently used in **broadcast production** or **automated news curation.**
6. **Video Analysis (Tracking Objects and Faces Across Frames)**
    - Analyzes video streams to detect and **track faces, objects, or activities** across frames. It provides metadata (like timestamps) for significant events, making it useful in **surveillance systems**, **sports broadcasting**, or **content editing** for detecting highlights.
7. **Bulk Processing of Images**
    - AWS Rekognition supports **batch processing** for large datasets of images, which is ideal for companies dealing with **massive media collections** or archives. By processing images in bulk, businesses can **automatically tag, filter, or index** thousands of files efficiently, reducing manual workload. Bulk analysis can be combined with **S3 storage** to trigger Rekognition on new media uploads, enabling **automated workflows**

for content moderation, identity verification, or archival cataloging.

Service Features

1. **Real-Time Processing**
    - Rekognition allows for **low-latency analysis** of images and video streams, enabling real-time responses for applications like **security monitoring** or **live event analysis**. Immediate processing ensures that users are alerted as soon as a critical event is detected, such as identifying a known person or suspicious object in a surveillance feed.
2. **Integration with Other AWS Services**
    - Rekognition integrates seamlessly with other AWS services:
        - **S3**: Store images and videos that Rekognition can analyze upon upload.
        - **Lambda**: Trigger custom actions (like updating a database or sending notifications) based on Rekognition events.
        - **SNS**: Send **real-time alerts** or notifications (e.g., when a specific person or object is detected). These integrations allow businesses to build **scalable, event-driven workflows** for media processing, identity verification, or surveillance.

Use Cases and Applications

AWS Rekognition is widely used across industries:

- **Security and Surveillance**: Monitor cameras in real-time to detect faces, suspicious objects, or activities.
- **Media and Entertainment**: Automate content tagging and celebrity recognition in videos and images.
- **E-commerce and Retail**: Use image analysis for product categorization, inventory management, or customer behavior insights.
- **Compliance and Moderation**: Filter explicit content on social media or in advertisements to ensure compliance with local laws.
- **Identity Verification**: Implement face recognition for **secure access** to applications, facilities, or services.

With its ability to handle both **real-time** and **bulk processing** tasks, Rekognition helps businesses improve operational efficiency, reduce manual effort, and automate critical workflows.

Cost Optimization Strategies for AWS Rekognition:

Here, we'll explore effective strategies for cost optimization when using AWS Rekognition, ensuring you get the most value out of your investment.

1. Choose the Right Rekognition API

Understanding the specific capabilities of AWS Rekognition can help reduce costs significantly.

- **Label Detection vs. Custom Labels:**
  Opt for standard Rekognition features whenever possible. Custom models may offer tailored solutions, but they come at a higher price. If the default capabilities meet your needs, stick with them.
- **Face Search vs. Face Detection:**
  Use **`DetectFaces()`** if you only need to identify faces in images without matching them against a collection. This simple choice can save you money.
- **Video vs. Image Analysis:**
  Analyzing static images is often more economical than processing videos. If feasible, leverage still images to minimize expenses.

2. Use the Right S3 Data Storage Class

The way you store input images and videos can impact your costs:

- **S3 Storage Classes:**
  Store less frequently accessed data in classes like S3 Standard-IA or Glacier Deep Archive to save costs.
- **Lifecycle Policies:**
  Implement S3 Lifecycle Policies to transition older data to cheaper storage automatically, keeping costs in check over time.

3. Implement Request Throttling & Batch Processing

Avoid unnecessary calls to AWS Rekognition:

- **Avoid Redundant Requests:**
  Throttle your requests to ensure you're not making excessive calls for the same analysis.

- **Batch Processing:**
  Utilize batch processing where possible. For instance, you can use `StartLabelDetection()` to analyze entire videos in one go instead of processing individual frames.

4. Monitor Usage with AWS Cost Explorer & Budgets

Keeping an eye on your costs is vital:

- **AWS Cost Explorer:**
  Leverage this tool to identify usage patterns and areas where expenses are climbing.
- **Set Budgets:**
  Configure AWS Budgets and Alerts to notify you when your usage surpasses set thresholds, helping you stay within budget.

5. Optimize Lambda Execution for Rekognition Tasks

If you're using AWS Lambda for Rekognition tasks, consider these tips:

- **Parallelization:**
  Use AWS Step Functions and SQS to distribute workloads and prevent bottlenecks in processing.
- **Adjust Lambda Timeout Limits:**
  Set reasonable timeout limits to avoid unnecessary costs

during execution, particularly for larger video processing tasks.

6. Use Custom Labels Only When Necessary

Custom models can be resource-intensive:

- **Avoid Unnecessary Training:**
  Only train and deploy custom models if the standard Rekognition services fail to meet your requirements.
- **Optimize Training Datasets:**
  Ensure your training datasets are well-structured to reduce costs associated with overfitting and large model sizes.

7. Face Collection Management Best Practices

Managing face collections efficiently can lead to significant savings:

- **Limit Collection Size:**
  Keep the size of your face collections manageable. Larger collections incur higher costs during face searches.
- **Archive Inactive Data:**
  Move inactive face data to lower-cost storage options when not in immediate use.

8. Take Advantage of Free Tier and Discounts

Make the most of AWS offerings:

- **AWS Free Tier:**
  AWS Rekognition provides a Free Tier, allowing you to analyze 5,000 images with `DetectLabels()` monthly for the first 12 months.
- **Explore Savings Plans:**
  If your usage is predictable and substantial, investigate Savings Plans or Enterprise Discount Programs (EDP) to further reduce costs.

9. Compress Images and Videos Before Uploading

Smaller files mean lower processing costs:

- **Optimize Input Files:**
  Before uploading images and videos, compress or downscale them to cut down on processing expenses.

10. Evaluate ROI of Rekognition Features

Finally, continually assess the return on investment (ROI) of the features you're using:

- **Analyze Feature Value:**
  Regularly evaluate which features of AWS Rekognition provide the most value for your business needs. For example, if you're using `DetectText()` infrequently, it might be more cost-effective to explore alternative solutions like using OCR in AWS Lambda with Tesseract.

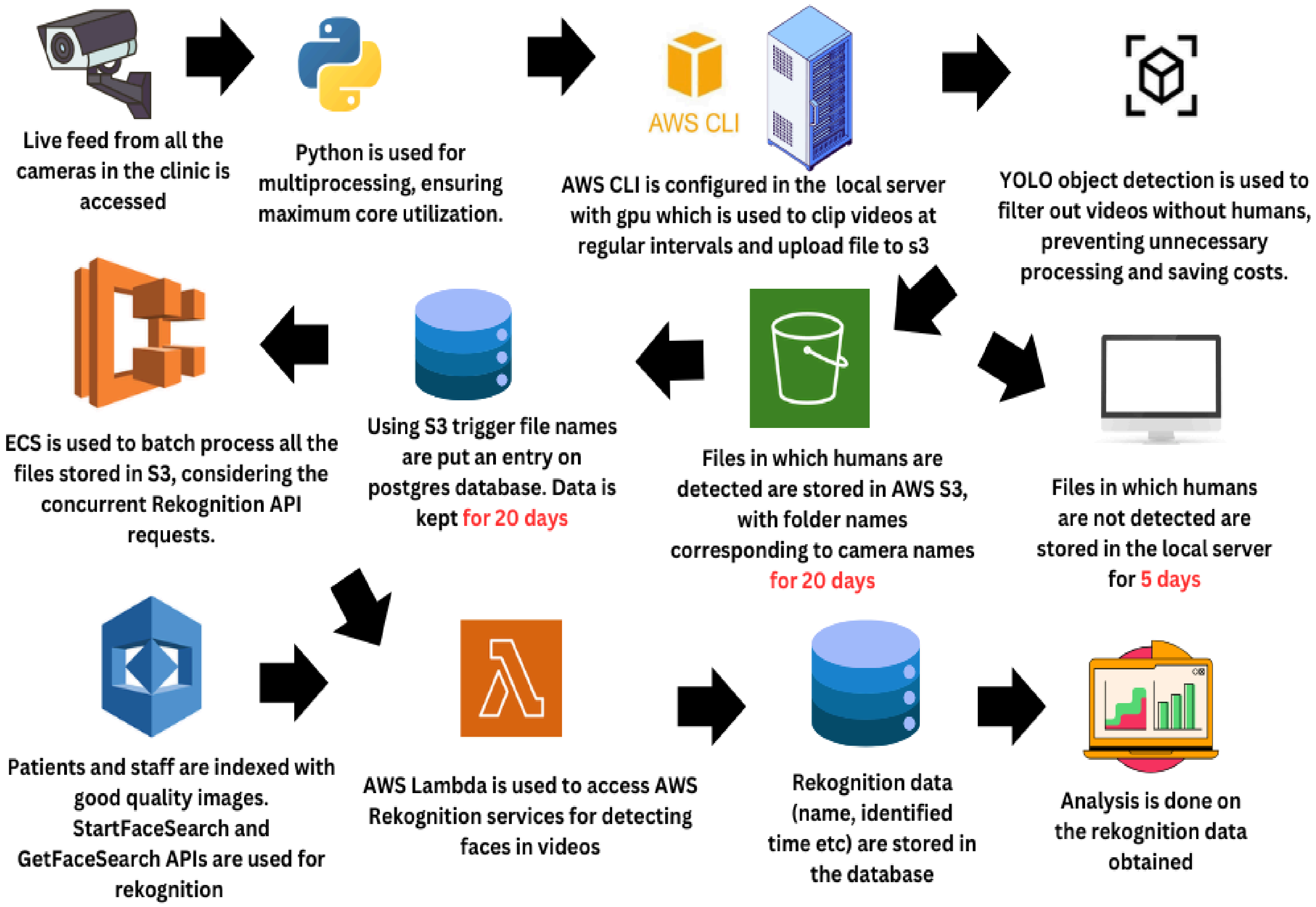
11. Choose Cost-Effective Regions:

- **Analyze pricing differences across AWS regions for Rekognition services:** Some regions may offer lower costs for certain features. By strategically selecting regions for deployment, you can leverage these differences to reduce your overall expenses.

- **Latency Considerations:** While cost is essential, also evaluate latency and performance needs. Sometimes, a slightly higher cost in a region closer to your user base may provide better performance and user experience, potentially offsetting the cost differences.

Face Rekognition Success Story:

Title: Implementation of AWS Rekognition for Patient and Staff Tracking in a Clinic



Live feed from all the cameras in the clinic is accessed

Python is used for multiprocessing, ensuring maximum core utilization.

AWS CLI is configured in the local server with gpu which is used to clip videos at regular intervals and upload file to s3

YOLO object detection is used to filter out videos without humans, preventing unnecessary processing and saving costs.

Files in which humans are not detected are stored in the local server for 5 days

Files in which humans are detected are stored in AWS S3, with folder names corresponding to camera names for 20 days

Using S3 trigger file names are put an entry on postgres database. Data is kept for 20 days

ECS is used to batch process all the files stored in S3, considering the concurrent Rekognition API requests.

Patients and staff are indexed with good quality images. StartFaceSearch and GetFaceSearch APIs are used for rekognition

AWS Lambda is used to access AWS Rekognition services for detecting faces in videos

Rekognition data (name, identified time etc) are stored in the database

Analysis is done on the rekognition data obtained

Live feed data collection:

- Cameras have been installed in all rooms at strategically chosen locations to maximize coverage and accurately track individuals, ensuring precise monitoring of their presence and duration.
- The placement of the cameras has been carefully planned to capture optimal angles, minimizing blind spots and increasing the likelihood of detecting multiple faces.
- This strategic positioning allows for enhanced visibility, ensuring that no critical moments are missed.
- Additionally, the quality of the video feed has been prioritized, providing clear visuals that facilitate accurate identification and monitoring.
- The combination of high-quality feed and well-thought-out camera placement ensures comprehensive tracking and reliable data collection.

Multiprocessing using python:

- We manage footage from 25 cameras, capturing key moments by slicing 0.5-second clips every 5 seconds for efficient analysis.
- These clips are processed using Python's multiprocessing to maximize performance and ensure parallel handling of data, significantly speeding up the workflow.
- This approach enables seamless integration with AWS Rekognition, ensuring accurate and timely face detection and tracking.

- The combination of optimized slicing intervals and concurrent processing ensures that we can extract meaningful insights from large volumes of video data in real-time without compromising accuracy or performance.

Local server with GPU:

- We utilize a local server with 12 cores and GPU capabilities, located within the clinic, to efficiently handle the processing of video footage from 25 cameras.
- This setup minimizes network latency, ensuring real-time analysis and high-quality data management while significantly reducing bandwidth requirements.
- Leveraging the GPU's power, we employ YOLO (You Only Look Once) to detect the presence of humans in each sliced 0.5-second clip. Only clips with potential human activity are sent to AWS Rekognition, while footage of empty rooms or irrelevant data is filtered out.
- This targeted approach not only ensures efficient use of resources but also reduces processing costs by minimizing unnecessary calls to Rekognition.

YOLO11N:

- YOLO11N is the latest iteration in the Ultralytics YOLO series of real-time object detectors
- . Building on the success of its predecessors, YOLO11N introduced significant improvements in both architecture and training methods

- . This model is designed to be fast, accurate, and efficient, making it a versatile choice for a wide range of computer vision tasks
- . With enhanced feature extraction capabilities and optimized training pipelines, YOLO11N delivers faster processing speeds while maintaining a balance between accuracy and performance
- . Whether it's for object detection, tracking, or other complex tasks, YOLO11N redefines what's possible in the realm of computer vision.

Good Quality Images for Indexing with AWS Rekognition Collections:

- High-quality images are essential for indexing patients and staff in AWS Rekognition to ensure accurate and reliable identification. Clear, well-lit images with proper resolution significantly improve facial detection and matching accuracy, reducing the chances of false positives or missed identifications. The better the image quality, the more precise the facial features extracted by Rekognition, enhancing the system’s performance in real-world scenarios.
- The indexed images are stored in **collections**, a structured repository used by AWS Rekognition to manage and search for faces. Each collection acts as a database for facial features, allowing quick comparisons and matches during real-time or batch searches. When new video frames or images are analyzed, the system compares detected faces against these collections to identify known individuals.

- While high-quality images come with associated storage and processing costs, they are a worthwhile investment, as they minimize reprocessing and errors that could arise from low-quality inputs. Furthermore, maintaining well-organized collections ensures faster and more accurate search results, making the system reliable and efficient for applications such as patient tracking, security, or staff identification.

Advantages of using video input for rekognition:

- **Continuous Detection and Tracking:**
  Video allows Rekognition to analyze multiple frames per second, increasing the chances of detecting faces from various angles, even if individuals are moving. This reduces the likelihood of missed detections that might occur in a single image with unfavorable angles or occlusions.
- **Multiple Opportunities for Recognition:**
  As the system analyzes every frame over time, it captures faces in different lighting conditions, facial expressions, and orientations. This ensures more robust and reliable recognition, as it can select the best frame for indexing or comparison.
- **Enhanced Context Awareness:**
  Video provides temporal context, such as detecting when a person enters or leaves a room, how long they stay, and their movement patterns. This is useful for tracking individuals over time, which static images cannot achieve.
- **Improved Accuracy in Real-World Scenarios:**
  Video helps identify individuals even in challenging

situations, such as partially obscured faces or fast movement, by analyzing multiple frames. With images, you only get one chance per capture, which might not be ideal.

- **Optimized Cost and Data Usage:**
  By analyzing continuous footage and only sending relevant clips, we avoid unnecessary uploads and focus on meaningful data. This ensures more efficient use of AWS Rekognition's capabilities while keeping costs under control.

StartFaceSearch and GetFaceSearch APIs:

1. StartFaceSearch API

The **StartFaceSearch** API initiates a face search on a video stored in **Amazon S3**. It compares faces detected in the video frames against a **collection** of indexed faces (such as patient and staff databases) to find potential matches.

- **How it works:**
  - You provide the S3 URI of the video and the target collection ID.
  - The API processes the video asynchronously, extracting faces frame-by-frame.
  - It starts a job, returning a **Job ID** to track the operation's status..

2. GetFaceSearch API

The **GetFaceSearch** API retrieves the results of an ongoing or completed face search initiated by **StartFaceSearch**.

- **How it works:**
    - You pass the **Job ID** from the **StartFaceSearch** call.
    - The API returns details about detected faces, such as:
        - **Matched faces** from the collection with similarity scores.
        - Information about unidentified faces, if no match is found.

Advantages of Using Both APIs Together

- **Efficient Processing:** Video is processed asynchronously, reducing wait times for results.
- **Continuous Monitoring:** Matches are retrieved in near real-time, ensuring timely responses for security or access control.

Together, **StartFaceSearch** and **GetFaceSearch** enable precise, automated face recognition from video streams, providing comprehensive monitoring with reliable identification capabilities.

Data Analysis with Timestamps, Names, and Patient IDs:

Using AWS Rekognition, we collect detailed insights by analyzing video footage with **timestamps**, **names**, and **patient IDs**. This information helps us distinguish between patients and staff, enabling comprehensive tracking and monitoring.

Our analysis provides the following key insights:

1. **Patient Arrival and Check-in Time:**
   The system detects when each patient arrives at the clinic, allowing accurate tracking of their entry time.
2. **Room-wise Duration Monitoring:**
   By tracking patients through different rooms using timestamps, we determine how much time they spend in each room, providing a detailed breakdown of their journey within the facility.
3. **Staff-Patient Interaction:**
   We match patient IDs with the corresponding doctors or technicians who attended them, offering insights into who interacted with which patient and for how long.
4. **Exit Time:**
   The system logs when patients leave, enabling calculation of total time spent in the clinic and individual departments.

This data allows for deeper analysis, such as identifying peak service hours, bottlenecks, or delays in patient flow, improving operational efficiency. By distinguishing between staff and patients through names and IDs, we ensure precise monitoring and reporting, leading to better management and enhanced patient care.