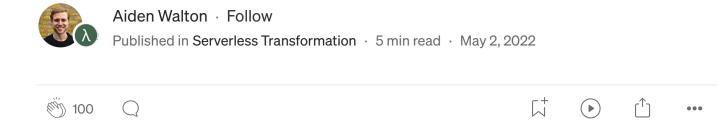
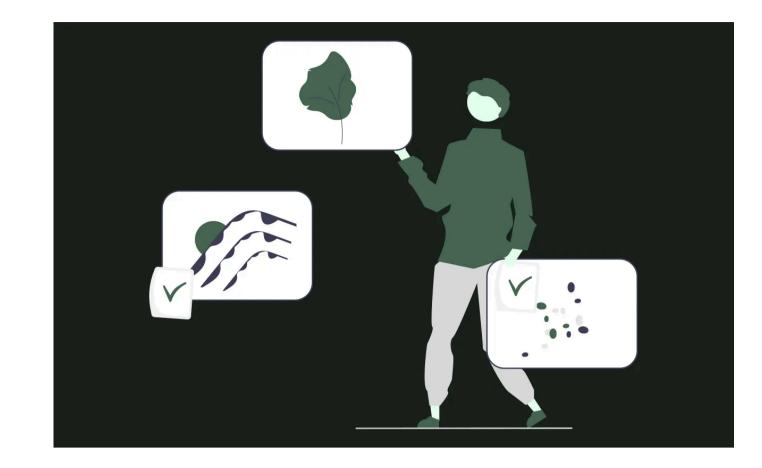
Write \bigcirc

Serverless Image Object Detection at a Social Media Startup



Using Amazon Rekognition to rapidly build an image object detection feature at a high-scale social media startup.



The ability to gather information from images has profound business potential. And, well, it can also just be fun. In this article I'll outline how we used <u>Amazon Rekognition</u> to rapidly build an image object detection feature for a social media startup.

What is Image Object Detection?

Simply put, image object detection is the process of detecting and extracting information about entities in a given image. This involves detecting objects, activities, places, animals, products, etc.

Image object detection has a wide array of use cases across a variety of industries. Major sectors such as banks, insurance, social media and dating apps, news agencies and FinTech use object detection in some form or another.

Our Use Case

Recently, we were tasked with building an image object detection feature for a social media startup. The use case was simple — users should be able to select some of their favourite photos and submit them to be featured on one of the startup's social media pages.

The social media marketing team needed a way to search through the image submissions for certain themes — such as photos of the ocean, popular landmarks, animals, music concerts, etc.

Image Analysis using Deep Learning

Analysing images and classifying them based on scenery and objects within the image is no simple task. Human sight is nothing short of remarkable and building an application that's able to replicate the brain's ability to detect objects is immensely complex. There is an entire Computer Vision industry devoted to doing just that.

Performing object detection from scratch is typically a multi-step process that involves:

- Collecting images and labelling the objects within
- Training the ML models
- Using the trained models to perform the analysis
- Performance tuning
- And, repeat...

Our aim for this feature, like all others on the project, was to build it quickly and test its efficacy in a production environment as soon as possible. Furthermore, we didn't want to devote development resources to building a solution from the ground up when we could leverage existing cloud services.

Cue Serverless — the startup's entire backend is fully Serverless and eventdriven. With this architecture we're able to have teams of developers that only need to focus on features that differentiated the social media app from others. Serverless also enables us to build highly-scalable services whilst also only paying for exactly what we use — an important consideration for a scaling startup.

So to achieve this feature, we used Amazon Rekognition — a fully Serverless image and video analysis service. Using Rekognition, we were able to develop this complex and critical workflow in a matter of hours. Let's dive into it.

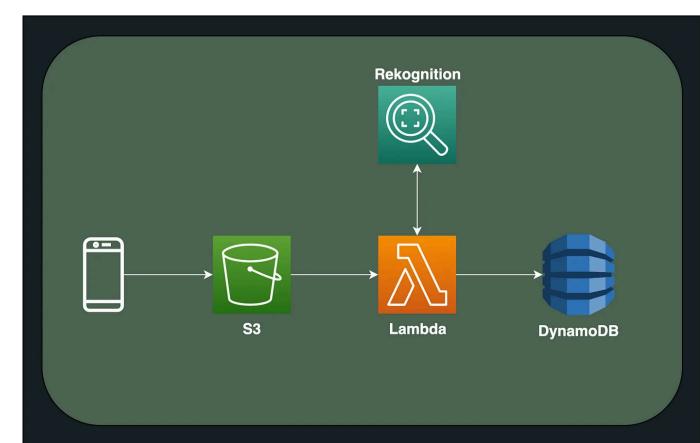
What is Amazon Rekognition?

Amazon Rekognition is an AWS Serverless offering that uses deep learning to perform image and video analysis. Being fully Serverless means that with Rekognition we don't need to worry about the complexity of the underlying infrastructure, we pay only for what we use and it provides us with prewritten software for image and video analysis tasks. Rekognition offers a range of features, including image label detection, face detection, celebrity detection, content moderation and text detection.

The best part? Rekognition abstracts away the heavy lifting of building, training and analysing deep learning models. Image and video analysis is quick and simple, with minimal set-up necessary. We didn't need to worry about building and training our own datasets and provisioning server capacity so that our service would scale. All we needed to worry about was integrating.

Architecture

The architecture is straight-forward. Our mobile app uploads images from users' phones into an S3 bucket. The upload to S3 then triggers a Lambda function which in turn calls the Rekognition API, and stores the results in DynamoDB for querying.



Serverless image object detection architecture diagram using AWS

The Code

Writing code is fun right? Well, writing less code is even more fun.

Rekognition exposes a set of APIs that you send image data to, which perform the analysis and return the results. For our use case, we used the <u>detect-labels</u> API.

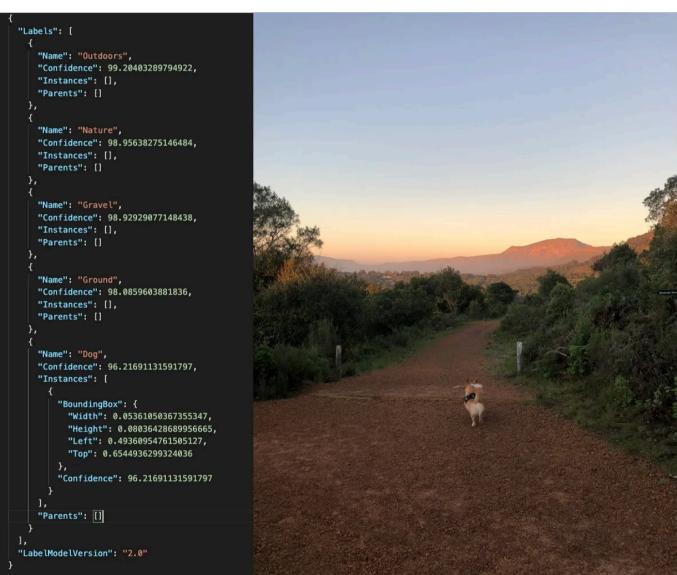
A simplified version of our Serverless framework Infrastructure as Code file looks like this:

```
//serverless.yaml
functions:
  imageLabelDetection:
   handler: image-label-detection.handler
     events:
         bucket: my-image-bucket
         event: s3:ObjectCreated:*
         existing: true
      iamRoleStatements:
       - Effect: Allow
         Action: rekognition:DetectLabels
         Resource: "*"
       - Effect: Allow
         Action: s3:GetObject
         Resource: arn:aws:s3:::my-image-bucket
```

Our Lambda code simply calls the Rekognition API and stores the results in DynamoDB, but you can use whatever makes sense for your use case. We obtain the S3 bucket name and the image's object name from the S3 event and pass those in to the *detectLabels* function of the <u>Rekognition SDK</u>. We also pass in two optional parameters (MaxLabels and MinConfidence) to specify the confidence level threshold and maximum number of labels that we want returned. In the example below, we will only get up to 20 labels in the response and all labels will have a confidence level of more than 80%.

```
//image-label-detecion.js
const AWS = require("aws-sdk");
const rekognition = new AWS.Rekognition();
exports.handler = async (event) => {
 const imageDetails = event.Records[0].s3;
 const bucketName = imageDetails.bucket.name;
  const objectKey = imageDetails.object.key;
 const rekognitionResp = await rekognition
    .detectLabels({
      Image: {
       S30bject: {
         Bucket: bucketName,
         Name: objectKey,
      MaxLabels: 20,
      MinConfidence: 80,
    .promise();
  // Send to data store, e.g. DynamoDB
```

Who doesn't love a picture of a dog? Below is a response for an image that we uploaded to our S3 bucket. As you can see, Rekognition correctly determines that it's an image of a dog on an outdoor, gravel path (and tells us where in the image the dog is!).



Rekognition response (left), uploaded image (right)

Final thoughts

So what are my thoughts after using Rekognition in production for a few months? Here is a list of key takeaways:

- Fully Serverless avoid the undifferentiated heavy lifting of managing underlying infrastructure complexity and use pre-built software and pretrained models
- Performance Rekognition is quick, around 600ms in most cases
- Easy to use The API is simple and easy to integrate with
- Continuous improvement AWS are constantly maintaining and iterating on Rekognition meaning it will become increasingly better over time
- Accuracy Rekognition takes advantage of Amazon's broad set of customers; resulting in highly accurate image label detection
- Has a generous free tier so you can trial before deciding
- Cost We pay only for what we use. Rekognition allows us to scale up and down based on business needs.

Summary

TLDR: Rekognition enabled us to rapidly build an image object detection feature that's accurate, fast and scalable.



Serverless-Transformation is an <u>aleios</u> initiative. <u>aleios</u> helps startups disrupt and large organisations to remain competitive using the best of Cloud-Native, Serverless.

Serverless Image Classification Rekognition AWS Object Detection



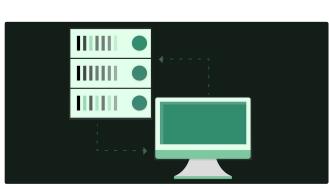
Written by Aiden Walton

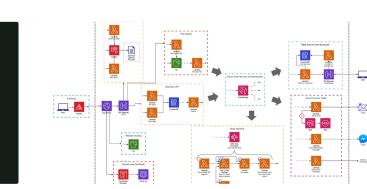
9 Followers · Writer for Serverless Transformation



Serverless Enthusiast | Software Engineer at aleios (https://www.aleios.com/)

More from Aiden Walton and Serverless Transformation





Xavier Lefèvre in Serverless Transformation

Xavier Lefèvre in Serverless Transformation

Asynchronous client interaction in AWS Serverless: Polling,... Event-driven Serverless often requires async update to the client. There are several ways ...

What a typical 100% Serverless **Architecture looks like in AWS!** If you are new to serverless and looking for a high level web architecture guide, you've...

11 min read · May 19, 2020

6 min read · Apr 18, 2020