AWS Rekognition Image Labels Generator

Automatically detect and label objects in images using Amazon Rekognition.

Project Created by ***Suyash Sable***

Email: sablesuyashsopan@gmail.com
Mobile: +91 8369086647

Overview

Understanding the System

Automated Object Detection

Detects and labels objects in images stored in Amazon S3.

Detailed Output

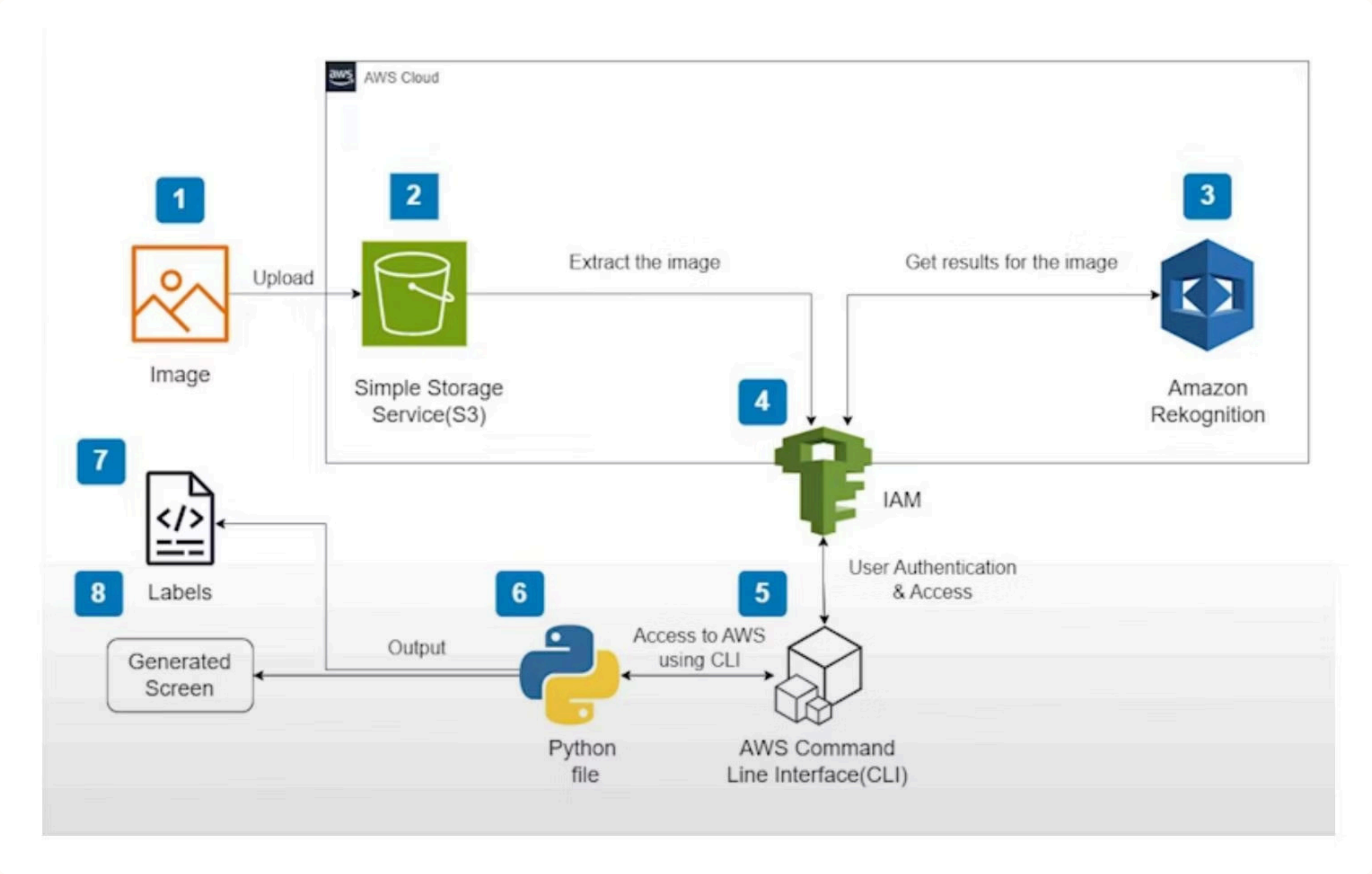
Provides labels with confidence scores and visual bounding boxes.

Python-Based Solution

Easy integration with a Python-based solution.

Architecture

System Components



Amazon S3	Stores images for analysis.
AWS Rekognition	AI service for image label and object detection.
IAM	Manages secure access to AWS services.
Python (Boto3)	SDK to call AWS services and process results.

Setup Guide

Prerequisites

1 AWS Account

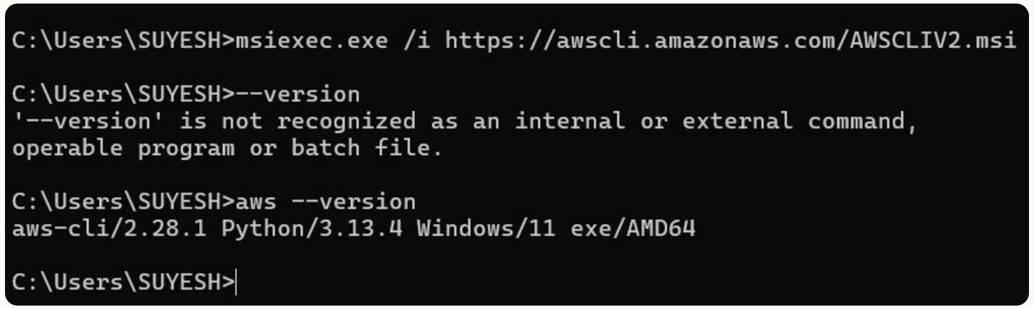
Ensure you have an active AWS Account.

2 AWS CLI Installed

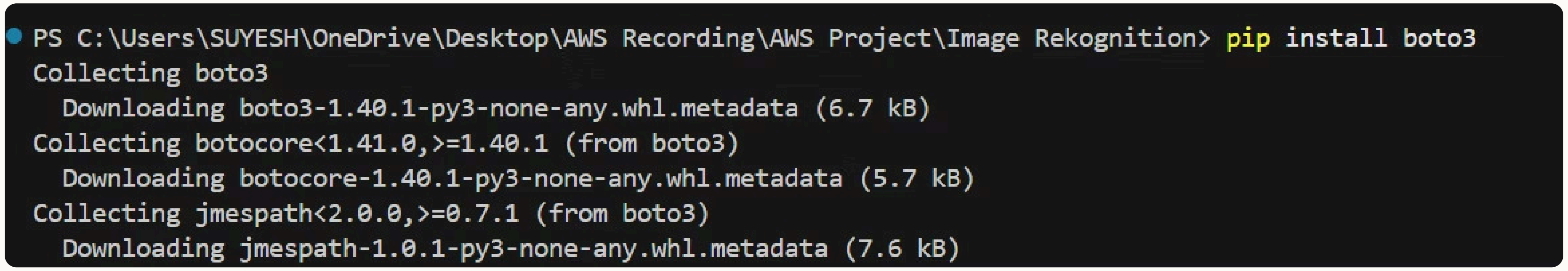
Configure your AWS Command Line Interface.

3 Python + Boto3

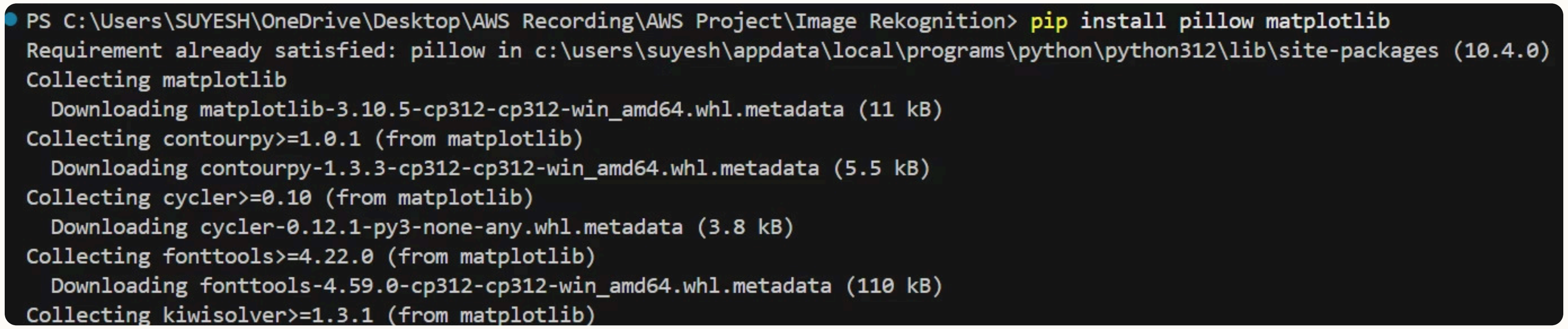
Install Python and the **Boto3 SDK** along with **Pillow** and **Matplotlib**.



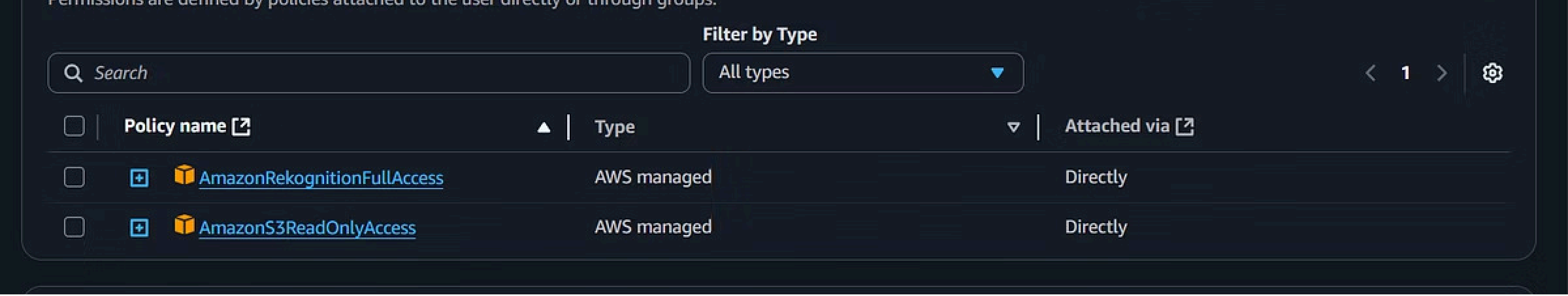
```
C:\Users\SUYESH>msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi

C:\Users\SUYESH>--version
'--version' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\SUYESH>aws --version
aws-cli/2.28.1 Python/3.13.4 Windows/11 exe/AMD64

C:\Users\SUYESH>
```



```
PS C:\Users\SUYESH\OneDrive\Desktop\AWS Recording\AWS Project\Image Rekognition> pip install boto3
Collecting boto3
  Downloading boto3-1.40.1-py3-none-any.whl.metadata (6.7 kB)
Collecting botocore<1.41.0,>=1.40.1 (from boto3)
  Downloading botocore-1.40.1-py3-none-any.whl.metadata (5.7 kB)
Collecting jmespath<2.0.0,>=0.7.1 (from boto3)
  Downloading jmespath-1.0.1-py3-none-any.whl.metadata (7.6 kB)
```



```
PS C:\Users\SUYESH\OneDrive\Desktop\AWS Recording\AWS Project\Image Rekognition> pip install pillow matplotlib
Requirement already satisfied: pillow in c:\users\suyesh\appdata\local\programs\python\python312\lib\site-packages (10.4.0)
Collecting matplotlib
  Downloading matplotlib-3.10.5-cp312-cp312-win_amd64.whl.metadata (11 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.3-cp312-cp312-win_amd64.whl.metadata (5.5 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.59.0-cp312-cp312-win_amd64.whl.metadata (110 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
```



Permissions are defined by policies attached to the user directly or through groups.

Search

Filter by Type: All types

□	Policy name	Type	Attached via
□	AmazonRekognitionFullAccess	AWS managed	Directly
□	AmazonS3ReadOnlyAccess	AWS managed	Directly

Setup Guide

IAM Policy Configuration

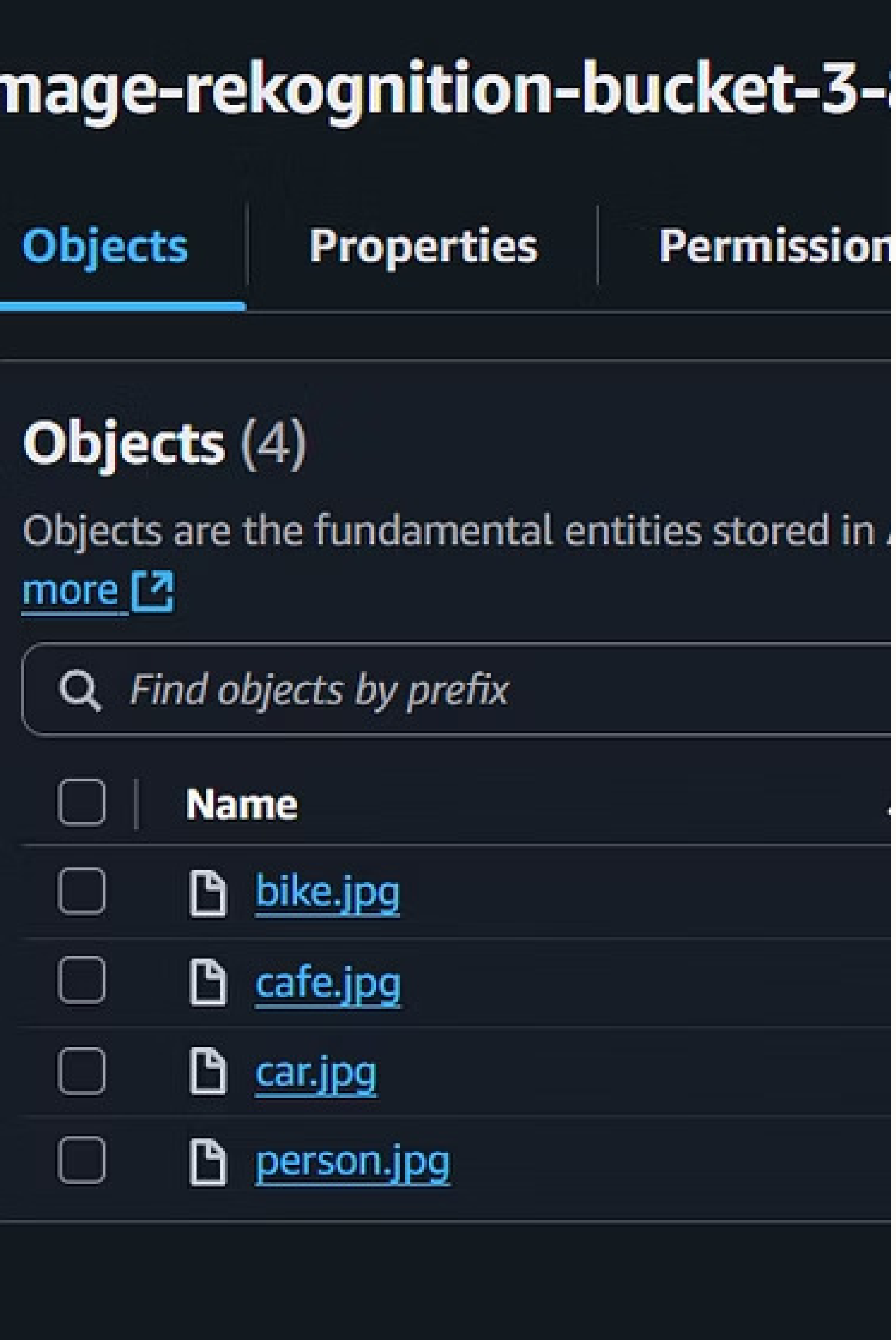
Attach the following policies to your **IAM** user for secure access:

AmazonS3ReadOnlyAccess

Allows read-only access to Amazon S3 buckets.

AmazonRekognitionFullAccess

Grants full access to AWS Rekognition services.



Objects (4)

Objects are the fundamental entities stored in
more

□	Name
□	bike.jpg
□	cafe.jpg
□	car.jpg
□	person.jpg

Setup Guide

S3 Bucket Creation

Create Bucket

Go to Amazon S3 Console and select "Create Bucket".

Name Your Bucket

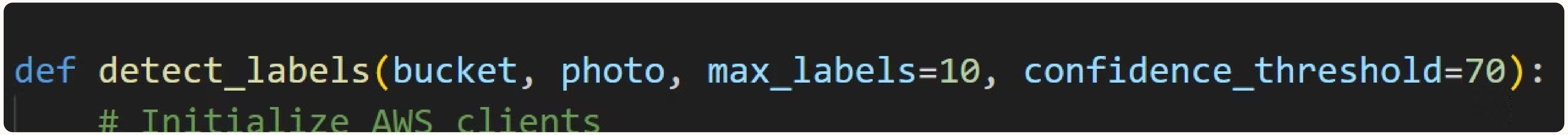
Name it **image-rekognition-bucket-3-8-2025** (or similar).

Upload Images

Upload your JPG images to the newly created bucket.

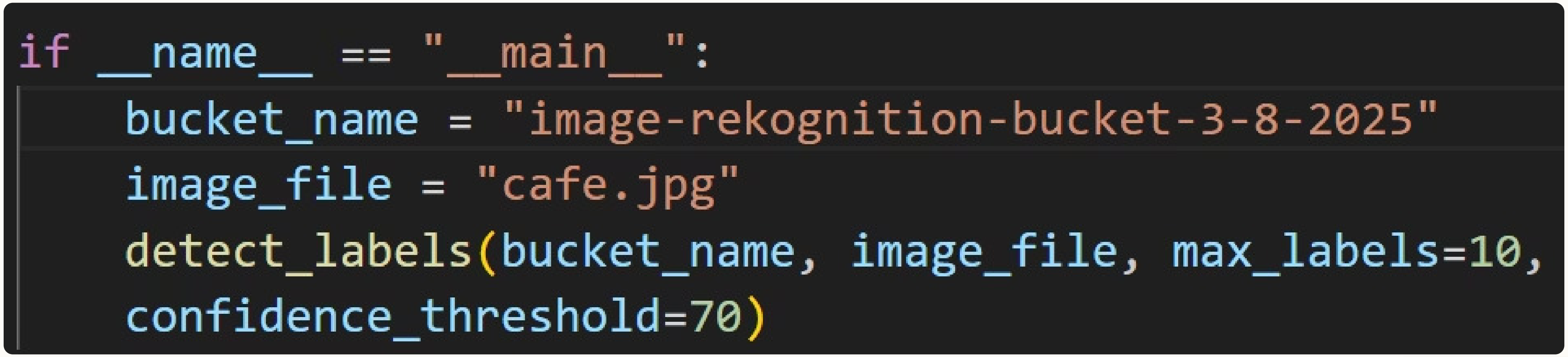
Python Script

detect_labels() Function

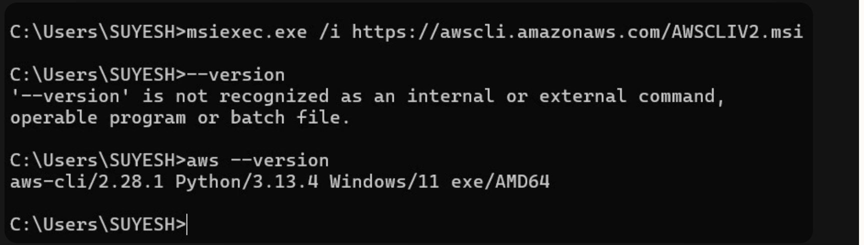


```python
def detect_labels(bucket, photo, max_labels=10, confidence_threshold=70):
    # Initialize AWS clients
```

bucket	S3 bucket name	**image-rekognition-bucket-3-8-2025**
photo	Image filename in S3	**cafe.jpg**
max_labels	Max number of labels to return	**10**
confidence_threshold	Min confidence % for a label	**70**



```python
if __name__ == "__main__":
    bucket_name = "image-rekognition-bucket-3-8-2025"
    image_file = "cafe.jpg"
    detect_labels(bucket_name, image_file, max_labels=10,
    confidence_threshold=70)
```



```
C:\Users\SUYESH>msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi

C:\Users\SUYESH>--version
'--version' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\SUYESH>aws --version
aws-cli/2.28.1 Python/3.13.4 Windows/11 exe/AMD64

C:\Users\SUYESH>
```

Deployment

Running the Script

Save Script

Save the script as `imageRekognition.py`.

Update Variables

Modify variables within the script as needed.

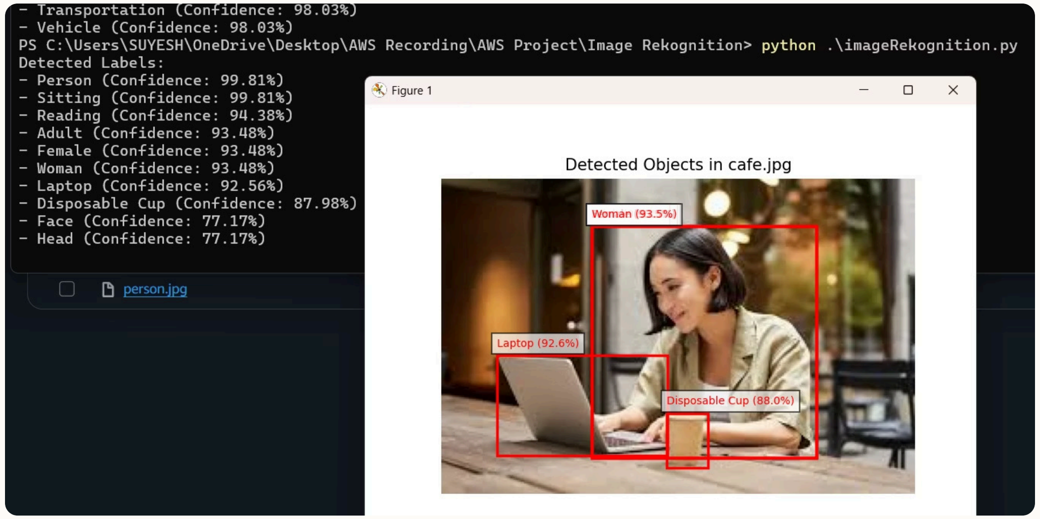
Run Script

Execute with `python ./imageRekognition.py`.

Visualizing the Output

The Python script leverages the results from **AWS Rekognition** to display detailed visual outputs, showcasing **detected labels** and their **precise bounding** boxes directly on the images.

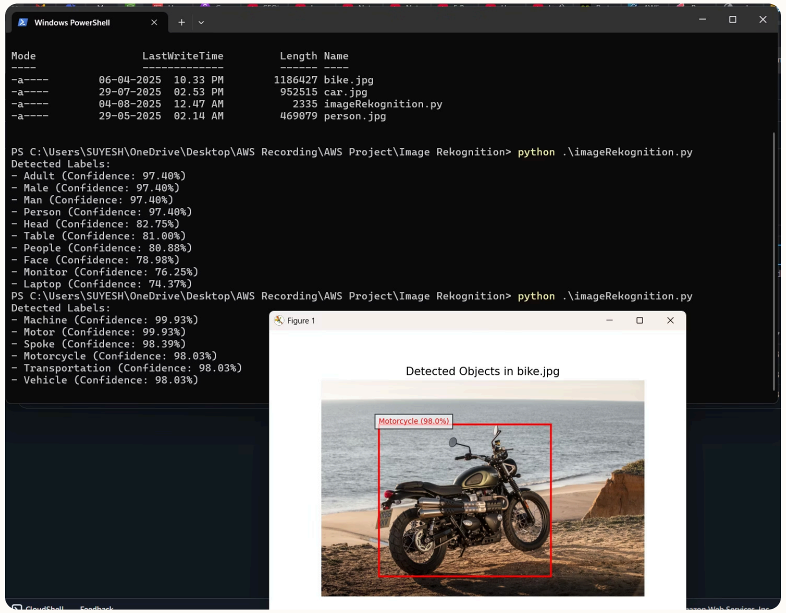
Output 1 with cafe.jpg



```
- Transportation (Confidence: 98.03%)
- Vehicle (Confidence: 98.03%)
PS C:\Users\SUYESH\OneDrive\Desktop\AWS Recording\AWS Project\Image Rekognition> python .\imageRekognition.py
Detected Labels:
- Person (Confidence: 99.81%)
- Sitting (Confidence: 99.81%)
- Reading (Confidence: 94.38%)
- Adult (Confidence: 93.48%)
- Female (Confidence: 93.48%)
- Woman (Confidence: 93.48%)
- Laptop (Confidence: 92.56%)
- Disposable Cup (Confidence: 87.98%)
- Face (Confidence: 77.17%)
- Head (Confidence: 77.17%)
```

This example demonstrates how the script identifies multiple objects and label them, such as "**Person**", "**Sitting**", "**Reading**", and "**Adult**", along with their respective **confidence scores**.

Output 2 with bike.jpg



```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        06-04-2025  10.33 PM        1186427 bike.jpg
-a----        29-07-2025  02.53 PM         952515 car.jpg
-a----        04-08-2025  12.47 AM           2335 imageRekognition.py
-a----        29-05-2025  02.14 AM         469079 person.jpg


PS C:\Users\SUYESH\OneDrive\Desktop\AWS Recording\AWS Project\Image Rekognition> python .\imageRekognition.py
Detected Labels:
- Adult (Confidence: 97.40%)
- Male (Confidence: 97.40%)
- Man (Confidence: 97.40%)
- Person (Confidence: 97.40%)
- Head (Confidence: 82.75%)
- Table (Confidence: 81.00%)
- People (Confidence: 80.88%)
- Face (Confidence: 78.98%)
- Monitor (Confidence: 76.25%)
- Laptop (Confidence: 74.37%)
PS C:\Users\SUYESH\OneDrive\Desktop\AWS Recording\AWS Project\Image Rekognition> python .\imageRekognition.py
Detected Labels:
- Machine (Confidence: 99.93%)
- Motor (Confidence: 99.93%)
- Spoke (Confidence: 98.39%)
- Motorcycle (Confidence: 98.03%)
- Transportation (Confidence: 98.03%)
- Vehicle (Confidence: 98.03%)
```

In this bike scenario, the output highlights various labels like "**Machine**", "**Motorcycle**", "**Spoke**", and "**Vehicle**".

These visual outputs are invaluable for quickly verifying the accuracy of the detection and understanding the data extracted by the AWS Rekognition service.

Applications

Example Use Cases

Automated Image Tagging

Catalog e-commerce product images efficiently.

Content Moderation

Detect inappropriate content in user uploads.

Accessibility Tools

Generate alt-text for visually impaired users.

Support

Troubleshooting & References

Common Issues

- **AccessDeniedException:** Check IAM permissions.
- **NoSuchBucket error:** Verify S3 bucket name.
- **Low confidence labels:** Increase `confidence_threshold`.

References

- **AWS Rekognition Docs**
- **Boto3 Documentation**