

Cloud Foundation

Project Report

ING 4 Cloud computing & virtualisation

Made by:

Ahmed Reda SIKILLI
Mohamed Ali OULGHAZI
Ghita EL KADIRI

College year: 2023/2024

Introduction

The **Car Brand Detection Web Application** is a project designed to leverage cutting-edge technology to detect and identify car brands from uploaded images. This application seamlessly integrates Azure cloud services with frontend and backend technologies, creating a user-friendly and efficient platform for car brand recognition.

Project Overview

The primary objective of this project is to empower users to upload car images through a sleek and intuitive user interface. Leveraging the power of Azure cloud services specifically **Azure Blob Storage** for image storage, **Azure Custom Vision** for image recognition, **Azure Functions** for asynchronous processing, and **Azure App Services** for hosting the application swiftly processes these uploaded images to identify the respective car brands.

Core Components

- **Frontend Interface (HTML/JavaScript):** The frontend interface allows users to upload images effortlessly and displays the detected car brands.
- **Backend Processing (Node.js/Express):** The backend, powered by Node.js and Express, manages the server setup, routes, and image processing logic. It integrates seamlessly with Azure Blob Storage for image storage and retrieval.

Azure Services Integration

The utilization of Azure Blob Storage facilitates secure and scalable storage for uploaded images and detected brand results. Azure Custom Vision, an AI-powered service, plays a pivotal role in recognizing car brands from the uploaded images. The Azure Function orchestrates asynchronous image processing tasks, ensuring efficient brand detection upon image uploads. Azure App Services, as the hosting platform, delivers the frontend interface and backend functionality seamlessly to users.

Azure Resources Utilized

1. Azure Blob Storage



Azure Storage Accounts provide a flexible and scalable cloud-based storage solution for various data types, including blobs, files, queues, and tables. In our project, Azure Blob Storage specifically plays a crucial role in storing both the uploaded car images and the results of the car brand detection process.

By utilizing Azure Blob Storage's containers and blobs, the project ensures a structured and organized approach to store both the uploaded car images and the detected brand responses. The segregation of data within containers aids in efficient management, enabling clear differentiation between the types of stored information and facilitating streamlined retrieval for further processing and display within the web application.

Storage Purpose

Azure Blob Storage is utilized for:

- **Storing Uploaded Car Images:** Images uploaded by users through the web interface are stored as blobs within Azure Blob Storage.
- **Storing Detected Brand Responses:** The results of the car brand detection process are stored as a specific file, such as `result.txt`, within the storage account.

Structured Data Organization

The project structures data within Azure Blob Storage using containers, ensuring an organized approach:

- **Container Creation:** A dedicated container named `car4container` is created to segregate and manage uploaded car images.
- **Blob Storage for Results:** Apart from image storage, a specific blob, such as `result.txt`, stores the detected brand responses, maintaining data separation and clarity.

Efficient Data Management

Utilizing Azure Blob Storage facilitates:

- **Clear Differentiation:** The segregation of data within containers enables a clear distinction between uploaded images and detected brand responses.
- **Efficient Retrieval:** This structured approach streamlines data retrieval, facilitating further processing and display within the web application.

Home > car4all | Containers >

car4container

Container

Search

Upload Change access level Refresh Delete Change tier Acquire lease Break lease View snapshots Create snapshot Give feedback

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Shared access tokens

Access policy

Properties

Metadata

Authentication method: Access key (Switch to Microsoft Entra user account)

Location: car4container

Search blobs by prefix (case-sensitive)

Show deleted blobs

Add filter

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state	
<input type="checkbox"/> 092aee6d1b8048e8204b6b706f0c13b7.jpg	19/12/2023, 17:18:15	Hot (Inferred)		Block blob	12.85 KiB	Available	***
<input type="checkbox"/> 09a2c07824b58b2bbb77f70b74695c5d.jpg	15/12/2023, 20:32:48	Hot (Inferred)		Block blob	12.99 KiB	Available	***
<input type="checkbox"/> 16c99ae15f78222840a78b0410e1816b.jpg	15/12/2023, 20:33:13	Hot (Inferred)		Block blob	120.53 KiB	Available	***
<input type="checkbox"/> 20230503053437_Website images_4_.jpg	15/12/2023, 14:45:32	Hot (Inferred)		Block blob	205.07 KiB	Available	***
<input type="checkbox"/> 5c5092028b84cc6088ef3d5e2ed2db19.jpg	15/12/2023, 20:31:23	Hot (Inferred)		Block blob	22.97 KiB	Available	***
<input type="checkbox"/> 68105fa72a8b8e2f0089d0b6f6e9134f8.jpg	15/12/2023, 20:34:40	Hot (Inferred)		Block blob	8.49 KiB	Available	***
<input type="checkbox"/> 73a900a3dc9a695205b0d546d9148cb1.jpg	19/12/2023, 16:29:03	Hot (Inferred)		Block blob	16.23 KiB	Available	***
<input type="checkbox"/> 800f62beeeae091e66118eacff534af.jpg	19/12/2023, 16:30:03	Hot (Inferred)		Block blob	13.67 KiB	Available	***
<input type="checkbox"/> 8af18e698da0d2424de2b4a3635ff277.jpg	15/12/2023, 13:48:44	Hot (Inferred)		Block blob	61.82 KiB	Available	***

2. Azure Custom Vision



Azure Custom Vision is a machine learning service provided by Azure Cognitive Services, designed for building and deploying custom image recognition models. In our project, Azure Custom Vision serves a critical role in the process of identifying and recognizing car brands from the uploaded images.

Service Purpose

Azure Custom Vision is utilized for:

- **Car Brand Identification:** The service is specifically employed to identify and recognize various car brands depicted in the uploaded images.

Machine Learning Model Training

The workflow involves:

- **Model Creation:** Developers can create custom image recognition models tailored to identify specific car brands.
- **Training with Data:** The model is trained using labeled datasets containing images of different car brands.

Brand Recognition Integration

Azure Custom Vision is integrated into the project's backend for:

- **Image Processing:** Upon image uploads, an Azure Function likely triggers Custom Vision to process and recognize the car brand within the image.

Image Detail Undo Changes Regions Shown Suggested objects off



My Objects

Audi X Volkswagen X

Only show suggested objects if the probability is above the selected threshold.

Threshold Value: 58%

Predictions

Predictions are shown in red

Tag	Probability
Audi	81.8%

3. Azure Functions



Azure Functions provide a serverless execution environment that can be triggered by various events. In our project, Azure Functions are specifically configured with a Blob Trigger, allowing them to respond automatically to new or updated blobs within the Azure Blob Storage.

Service Purpose

Azure Functions with Blob Trigger are utilized for:

- **Event-Driven Image Processing:** Automatically triggering image processing tasks in response to new image uploads within the specified Blob Storage container.

Blob Trigger Functionality

Azure Functions with Blob Trigger enable:

- **Automatic Invocation:** Functions are triggered upon specific blob-related events, such as the creation or modification of blobs within a defined container.
- **Real-time Processing:** Immediate execution of processing logic as new blobs are added to the container.

Integration with Blob Storage

The Blob Trigger Function is integrated for:

- **Dynamic Processing:** Reacts dynamically to new images uploaded to the designated Blob Storage container.
- **Blob Metadata Access:** Accesses blob metadata and content for processing, likely fetching images for brand recognition.

Image Processing Logic

Within the Blob Trigger Function:

- **Blob Fetching Logic:** Retrieves newly uploaded images from Blob Storage.
- **Custom Vision Integration:** Initiates Azure Custom Vision service calls to analyse and identify car brands from the fetched images.

```

BlobTrigger1 > _init_.py > get_blob_name_only
1 import azure.functions as func
2 from azure.storage.blob import BlobServiceClient
3 from azure.cognitiveservices.vision.customvision.prediction import CustomVisionPredictionClient
4 from msrest.authentication import ApiKeyCredentials
5 import io
6 import azure.core.exceptions
7 import json
8 import os
9 storage_connection_string = os.environ["AzureWebJobsStorage"]
10 def get_blob_name_only(full_blob_path):
11     # Splitting the full blob path by the '/' separator
12     blob_parts = full_blob_path.split('/')
13
14     # The last part of the split path is the blob name
15     blob_name = blob_parts[-1]
16
17     return blob_name
18
19 def process_custom_vision_results(results):
20     if results.predictions:
21         prediction = results.predictions[0] # Assuming the first prediction contains the brand info
22         detected_brand = prediction.tag_name # Assuming the brand name is stored in 'tag_name'
23         return detected_brand
24     return "Brand not found in results"
25
26 def save_result_to_blob(detected_brand):

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS AZURE

```

'Etag': '0x8DC00D23BF3EA34'
'Server': 'Windows-Azure-Blob/1.0 Microsoft-HTTPAPI/2.0'
'x-ms-request-id': 'ea6572e9-901e-0024-26bb-3285a6000000'
'x-ms-client-request-id': '58e10a53-9eae-11ee-af94-747827407244'
'x-ms-version': 'REDACTED'
'x-ms-content-crc64': 'REDACTED'
'x-ms-request-server-encrypted': 'REDACTED'
'Date': 'Tue, 19 Dec 2023 20:36:52 GMT'
[2023-12-19T20:36:55.386Z] Blob name: IMG_8069.jpeg
[2023-12-19T20:36:55.387Z] Detected Car Brand: Renault
[2023-12-19T20:36:55.411Z] Executed 'Functions.BlobTrigger1' (Succeeded, Id=b3d11235-1605-4c35-b30d-18f1b9da86ed, Duration=12008ms)
[2023-12-19T20:37:05.896Z] Executing 'Functions.BlobTrigger1' (Reason='New blob detected(LogsAndContainerScan): car4container/result.txt', Id=92369d19-9f3a-4d67-8ea9-5aca14debb4e)

```

4. Azure App Services



Azure App Service offer a fully managed platform for building, deploying, and scaling web applications. In our project, Azure App Services host the frontend HTML interface and the backend Node.js application, providing a seamless user experience and facilitating the backend functionality for image uploads and processing.

Service Purpose

Azure App Services are utilized for:

- **Hosting the Web Application:** Serving the frontend interface (`index.html`) to users for uploading images and displaying detected car brands.
- **Hosting the Backend Application:** Deploying and managing the backend Node.js application (`server.js`) responsible for handling image uploads, processing, and interaction with Azure services.

Web Hosting Capabilities

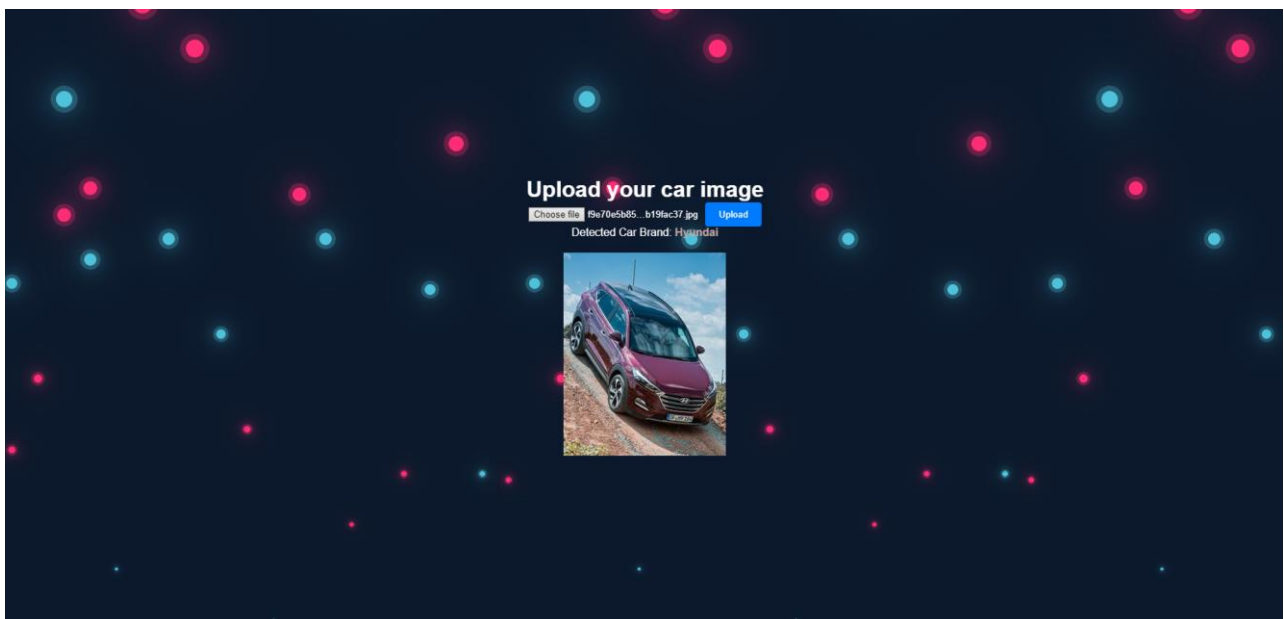
Azure App Services facilitate:

- **Scalable Web Hosting:** Offering a scalable environment for hosting web applications with dynamic scaling options.
- **Integrated Deployment:** Seamless deployment of web applications using various deployment methods like Git, Azure DevOps, or Visual Studio. In our case we used GitHub.

Backend Application Hosting

The Node.js Backend Application is hosted for:

- **Server Setup and Execution:** Managing the server setup and executing the backend logic for image processing upon image uploads.
- **Interfacing with Azure Services:** Establishing connections with Azure Blob Storage, Azure Custom Vision, and other services for data storage and image recognition.



You can access to our web app through this link: <https://cardet.azurewebsites.net>
Front-end & Back-end code: <https://github.com/Ahmedreda2002/webtest>

Conclusion

The Car Brand Detection Project harnesses the power of Azure's diverse suite of services to deliver a sophisticated yet user-friendly application. By seamlessly integrating Azure Blob Storage, Azure Custom Vision, Azure Functions with Blob Trigger, and Azure App Services, the project achieves a robust platform for identifying car brands from uploaded images.

This project stands as a testament to the potential of cloud-based solutions, showcasing the efficiency of Azure services in handling image recognition tasks. Its success lies in the cohesive integration of frontend and backend technologies, offering users an intuitive interface and accurate brand detection.

Moving forward, this project sets the stage for further advancements, inviting opportunities for scalability improvements and the integration of enhanced recognition models. Ultimately, the Car Brand Detection Project demonstrates the immense capabilities of Azure services in driving innovation and delivering impactful solutions in the domain of image recognition applications.