Video Analysis via Video Intelligence API

CPSC – 5207 Intro to Cloud Technologies

Lecturer: Jaspreet Bhatia

Group 9

Chathura Lakmal H P (0422854), Harshkumar Vyas (0424484), Jiazhou Ye (0426609),

Rupa Chakrabarti (0430308), Devanshi Patel (0444549)

Table of Contents

Abstract

This paper dives into the step-by-step implementation of a surveillance system using Google Cloud Platform. With the rapid growth in surveillance needs and the evolution of HD technology, the system integrated computer vision, machine learning (ML) and image processing to automatically interpret video scenes. The report guides towards creation of Google Cloud projects, enabling the Cloud Video Intelligence API, setting up storage buckets, implementing Cloud Functions for video analysis, and utilizing BigQuery for data storage.

During implementation the challenges encountered were like access permissions, function execution errors, and handling large video files. The paper includes a detailed block diagram illustrating the integration of Google Cloud services.

The proposed surveillance system has potential applications in detecting various elements in videos, from objects and people to logos and text. Future recommendations focus on addressing challenges like shadow detection, weather changes, and tracking in crowded environments. Real-world applications range from identifying vehicle thieves to creating a CCTV alarm system. The paper concludes by emphasizing ongoing research in video surveillance and suggesting future enhancements such as supporting large video files, generating results in multiple formats, and incorporating audio analysis alongside video processing.

Introduction and background

The practical needs of the surveillance system have been growing rapidly in the past decade. This growth rate is also supported by the development of high definition technology. Surveillance systems have earned a huge attention as application-oriented research which are integrating the ability of computer vision, machine learning, and image processing. The major purpose of this approach is to automatically interpret the scene in a video, to observe, and to predict the interaction of an object in the scene based on information that has been gathered from camera sensors. The initial utilization of the surveillance system was constructed from tube cameras which were distributed to monitor or to relay the factory activities in the 1940s. The conventional video surveillance system was often called Close-Circuit Television (CCTV), which was fragile and expensive, since the cameras were provided and installed by a security team to inspect the occurrence in the scene from a video display. The appearance of digitalized CCTV and advanced computation systems has increased the expansion of a (semi-) self-loading system which is known as the 2nd generation of surveillance systems. This new system has received advantages from the previous advancement in digital video communication, such as effective data compression, steady transmission, and bandwidth decrement [1].

In their research, Kyungroul Lee et al. has introduced a framework that manages to adopt heterogeneous video networks and protocols in supervisory management. Meanwhile, Mei Kuan Lim et al. have proposed iSurveillance that is able to detect several events, in various areas of interest (ROI) of a scene, at certain times by adapting and applying knowledge-based

---

[1] B. Cheng, J. Yang, S. Wang, and J. Chen, "Adaptive video transmission control system based on reinforcement learning approach over heterogeneous networks," IEEE Trans. Autom. Sci. Eng., vol. 12, no. 3, pp. 1104–1113, Jul. 2015

architecture to the video surveillance domain for a broader and integrated analysis of real-world surveillance scenarios. Antonio C. Nazare Jr and William Robson Schwartz have proposed SSF as a tool that provides an environment and a set of functions that enable researchers to implement and evaluate their algorithms in relation to an integrated monitoring system. Bita Darvish Rouhani et al. have also proposed RISE, which is designed by applying a background subtraction algorithm, a new adaptive methodology, based on streaming that studies/updates the appropriate dictionary matrix of background pixels for real-time monitoring of persistent and transient objects in an uncontrolled environment. Loredana Caruccio et al. have proposed EDCAR which implements video-based event recognition to interpret activities or behaviors in video sequences to detect, recognize, isolate certain events, and then warn of any unplanned activities. There are many limitations and open challenges for researchers in the field of video surveillance. The problem with motion detection in dynamic scenes is a difficult task to deal with illumination and weather changes, and shadow detection. A fast and accurate method is still needed to apply segmentation techniques to improve process performance. Tracking many people or groups of people is difficult due to crowded environments, poor lighting, noisy images, and the presence of camera movements.[2]

Our intention is to develop a system that analyzes CCTV videos in a distributed manner and extract the important incidents and inform users in a distributed computing environment. As the first part of the project we will implement a video data analysis and generate text labels via google intelligence API and store data in BigQuery for further analysis in the next step.

[2] Shidik, G. F., Noersasongko, E., Nugraha, A., Andono, P. N., Jumanto, J., & Kusuma, E. J. (2019). A systematic review of intelligence video surveillance: Trends, techniques, frameworks, and datasets. *IEEE Access*, *7*, 170457-170473. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8911368

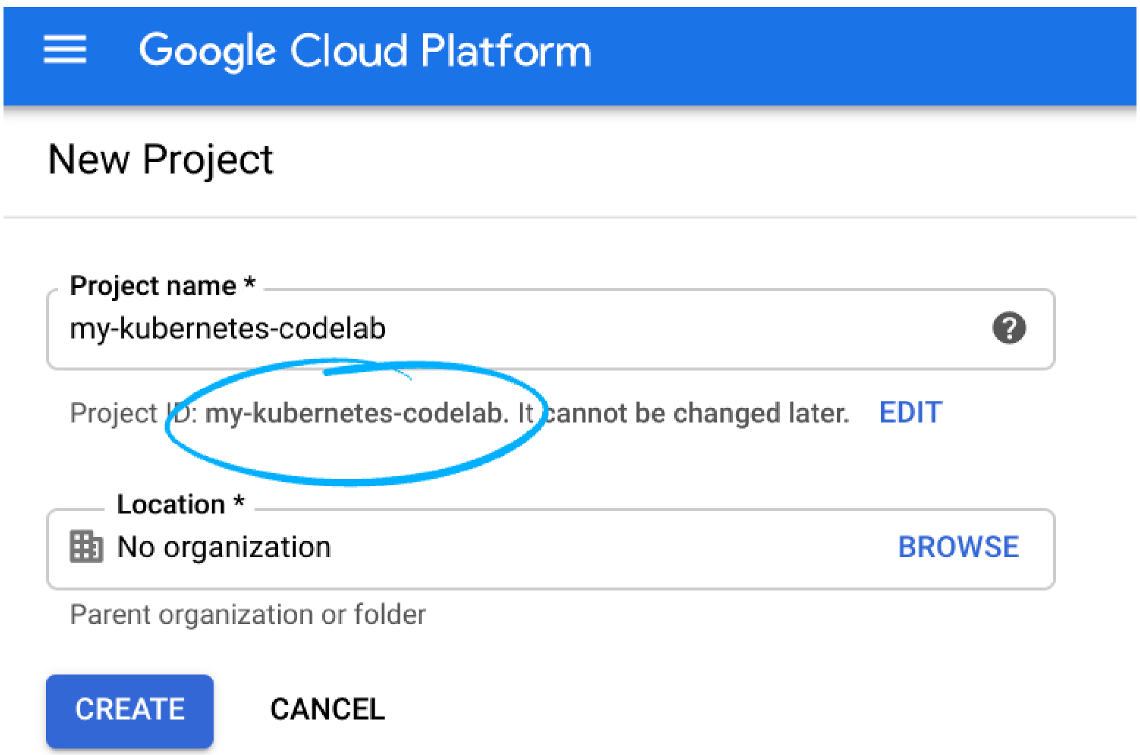
Implementation

For more detailed steps to implement yourself refer to the Project setup guide.pdf in the Git Hub.

Github Link: https://github.com/ChathuraP/Video-Analysis-via-GCP/tree/main

Step 1: Google Cloud Project Creation

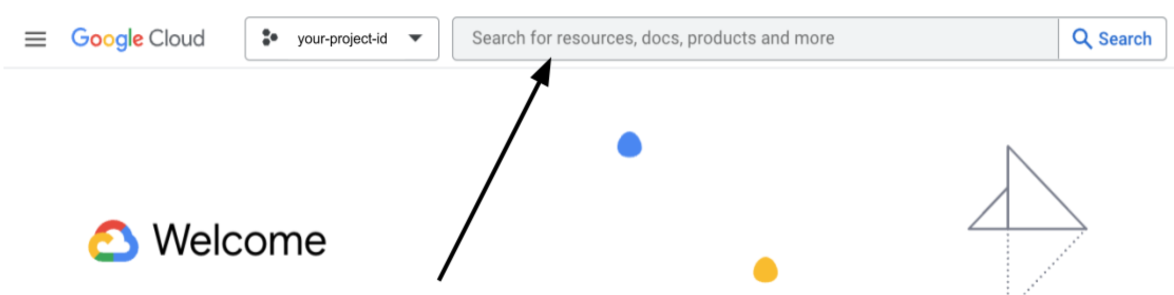
- Sign-in to the Google Cloud Console and create a new project or reuse an existing one. If we don't already have a Gmail or Google Workspace account, we must create one.



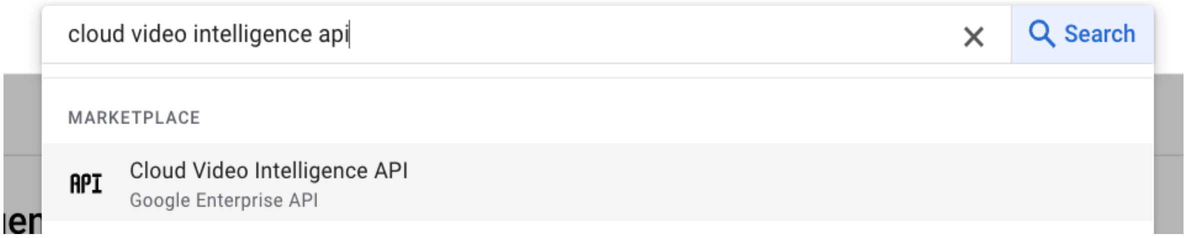
- The Project name is the display name for this project's participants. It is a character string not used by Google APIs. We can update it at any time.
- The Project ID is unique across all Google Cloud projects and is immutable (cannot be changed after it has been set). The Cloud Console auto-generates a unique string; usually we don't care what it is. In most codelabs, we'll need to reference the Project ID (it is typically identified as PROJECT_ID). If you don't like the generated ID, you may generate another random one. Alternatively, you can try your own and see if it's available. It cannot be changed after this step and will remain for the duration of the project.
- For your information, there is a third value, a Project Number which some APIs use. Learn more about all three of these values in the documentation.

Step 2: Enable Video Intelligence API

- In the Cloud Console, click on the search bar at the top of the page.

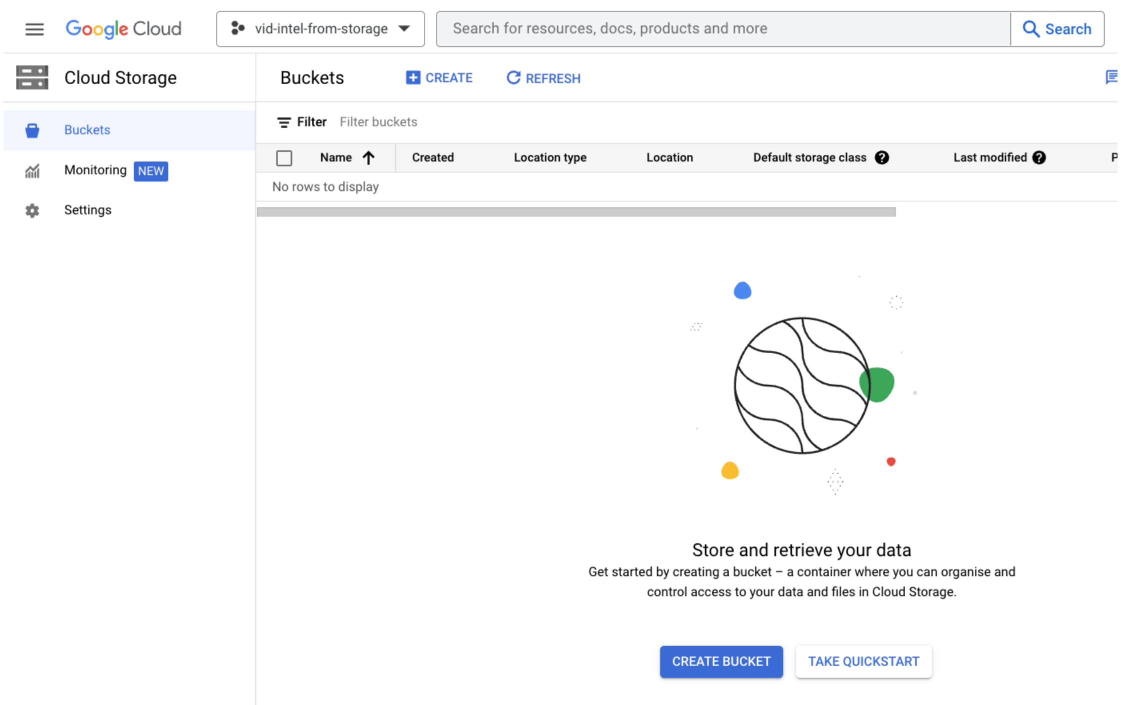


- Enable Video Intelligence API
- Search for "Cloud Video Intelligence API" and click on the "API" result
- On the Cloud Video Intelligence API product page, click "ENABLE".



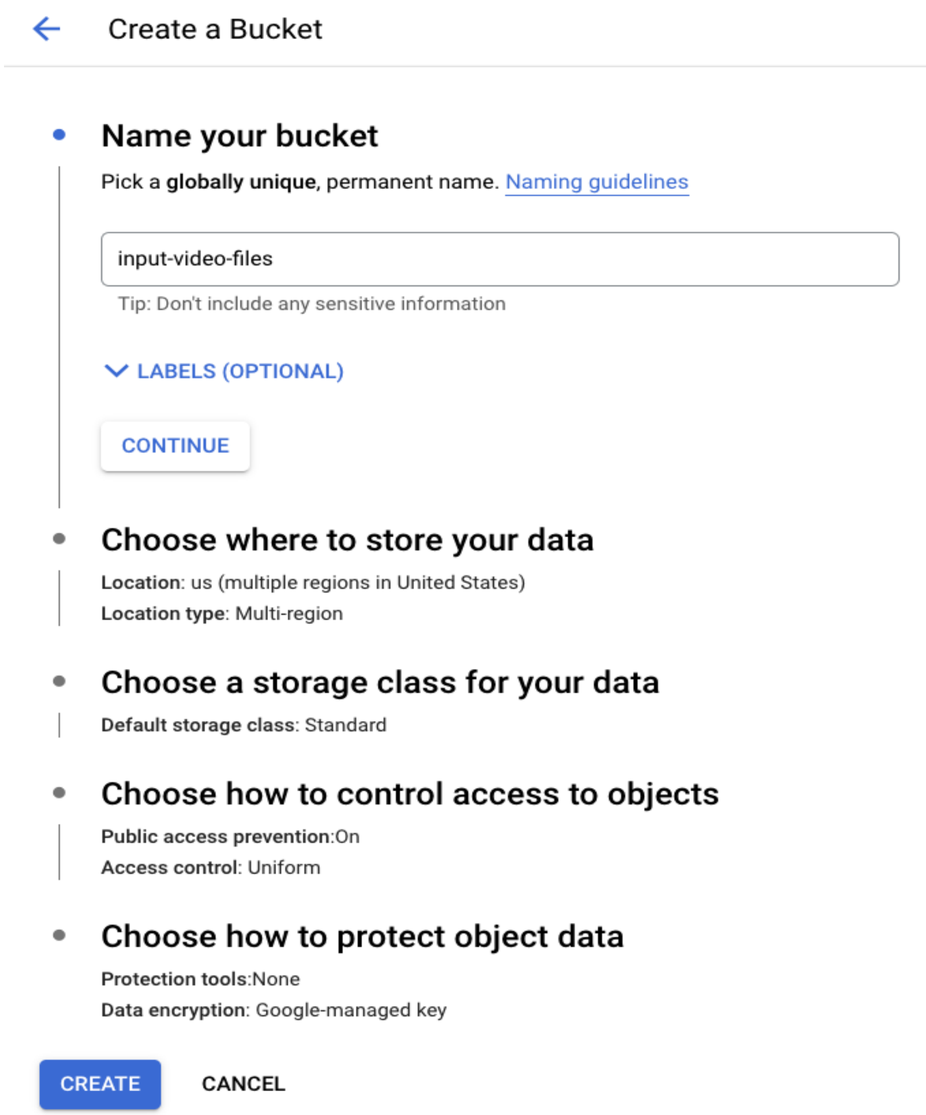
Step 3: Create Storage Buckets

- Navigate to Cloud Storage either by using the navigation menu on the left of the screen or the console search bar.

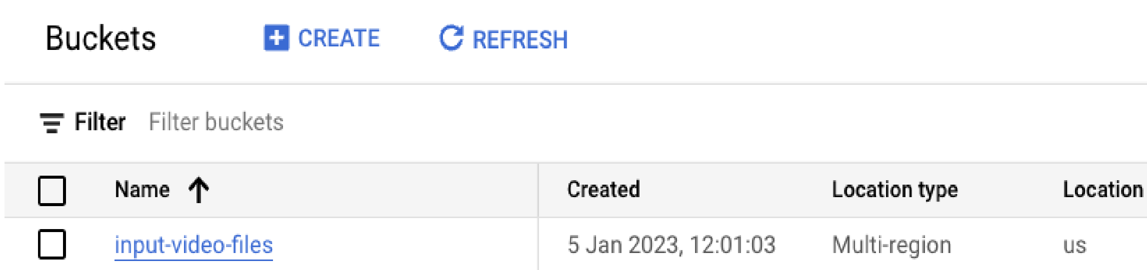


- Click "CREATE" or "CREATE BUCKET".
- We are going to create two buckets, an input bucket to upload our video files to, and an output bucket where the Cloud Video Intelligence API will store its output.

- In the "Create a Bucket" menu give our input bucket a unique name (ideally including the word "input" so that we know which bucket is which), then click "CREATE". (for indepth detail configurations and selection please refer to the [Project setup guide.pdf] in the Git Hub.)



Navigate back to the main Cloud Storage page. We would see the new input bucket listed:

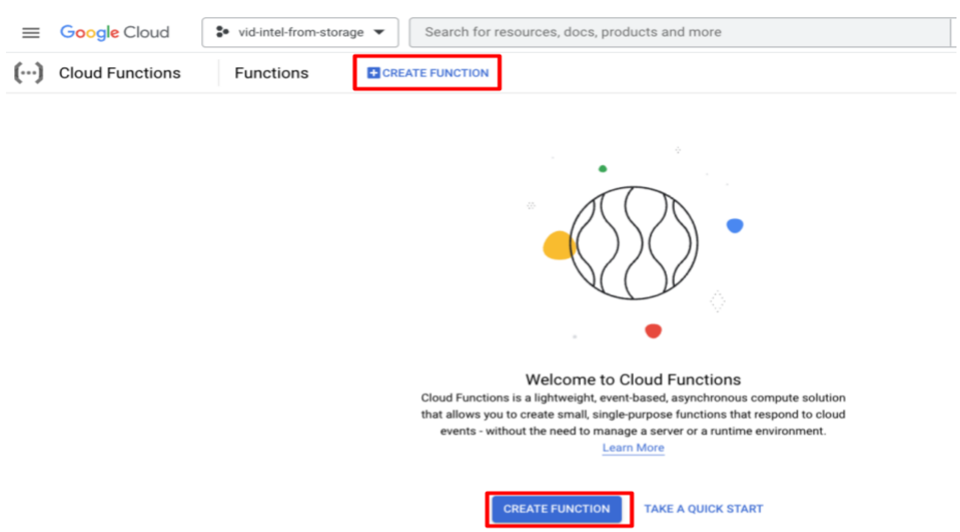


- Create an output bucket. On the main Cloud Storage page, click "CREATE" and create the output bucket. Same as the previous step but ideally including the word "output" in its name so that we know which bucket is which, then click "CREATE".

Step 4: Create Cloud Function

- Navigate to the Cloud Function page either by using the navigate menu on the left or the console search bar at the top. Then click "CREATE FUNCTION".

Configure Cloud Function:

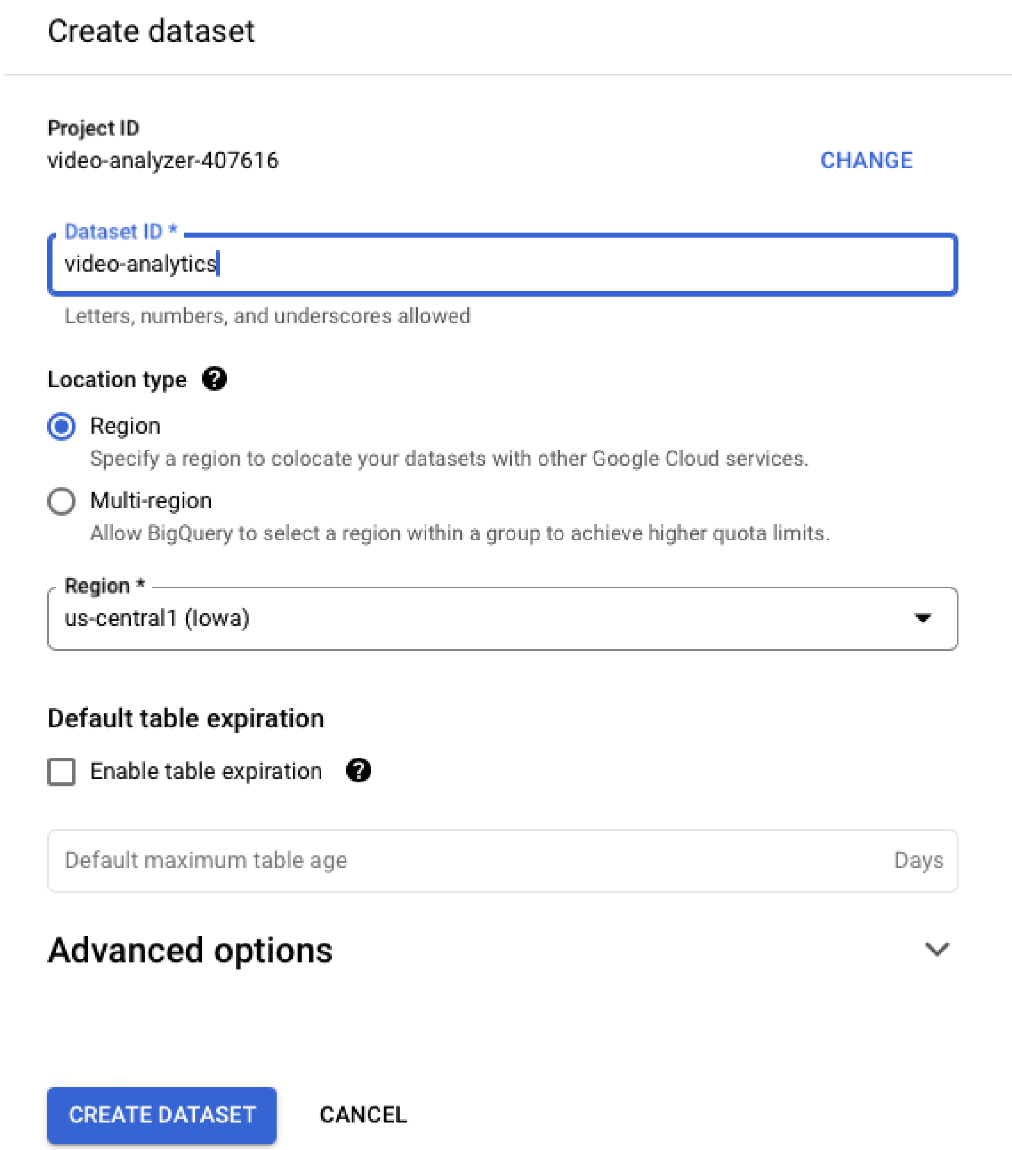


- In the configuration menu, select "1st gen" as the environment, then give the function a descriptive name like "analyze-video-trigger-function".
- Next for the trigger type, select "Cloud Storage", with event type "On (finalising/creating)", and for the bucket click "BROWSE" and select the input bucket we created earlier. Then click "SAVE".
- Click "NEXT" to proceed to the code section.
- Code the Cloud Function
- Within the code section select "Python 3.10" as the runtime, and for the entry point type in "analyze_video" (this is the specific function in our code that is going to be called when it is triggered).
- **Caution**: Don't forget to specify the "Entry point", in this case it should be set to "analyze_video" for it to run the start function in our code.
- Within the code editor for the "main.py" file, delete all of the code that is already there and then paste in the following code. We will need to replace the "gs://YOUR OUTPUT BUCKET NAME" string with the name of the output bucket you created earlier.

Step 5: Create BigQuery Data Set

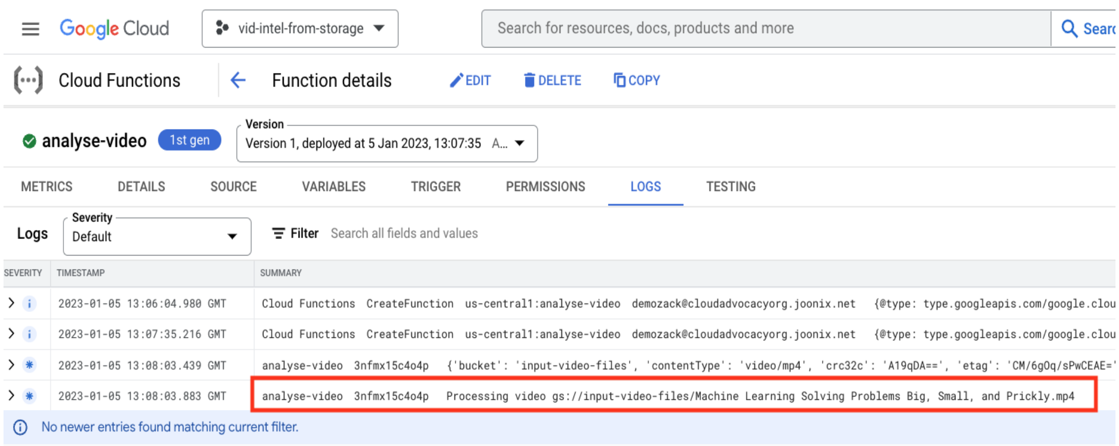
- Go to the BigQuery dashboard via the project dashboard.
- Select our project and click three dots near it to open the menu and create a dataset providing a proper name that you will later need to access the dataset. This name needs to be inserted to the cloud function source code so the cloud function is able to access it. Furthermore, Cloud function service account “App Engine service account” must be

given new role permission as “BigQuery Data Editor” from IAM console. please refer to the Project setup guide.pdf in the Git Hub

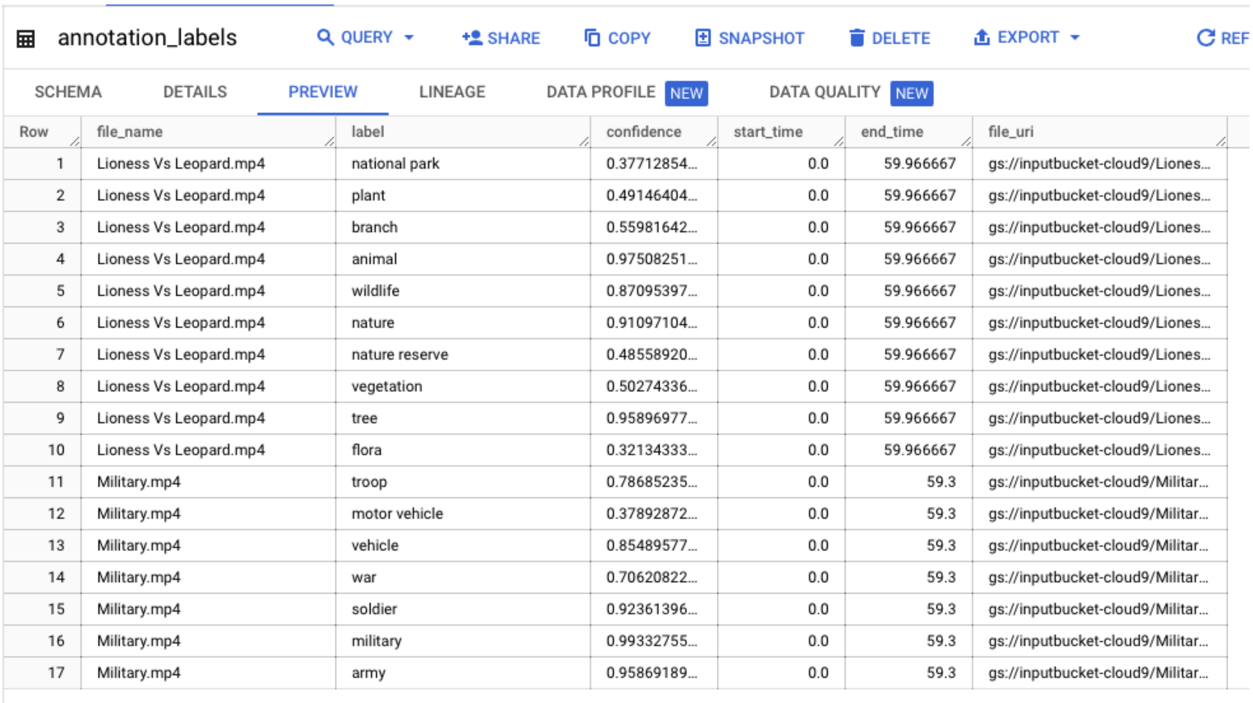


Step 6: Run the Project

- Navigate to the input storage Bucket and upload the video file. We can drag and drop it in, or click "UPLOAD FILES".
- This should automatically trigger the Cloud Function and call the Cloud Video Intelligence API to analyze the video.

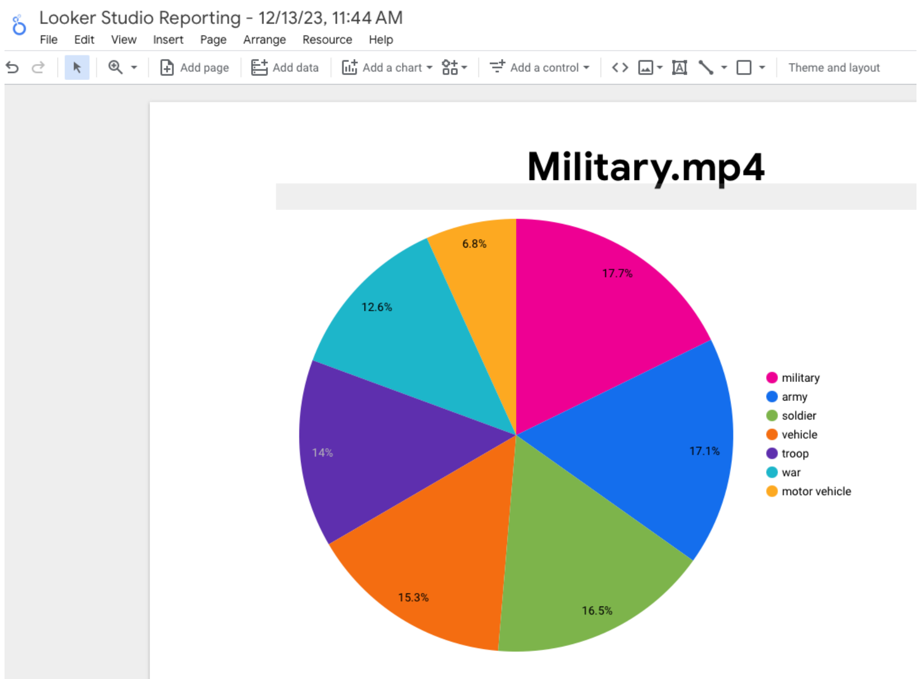


- Look for the logs in the cloud function and see each step completion or if any error occurs.
- After a couple of minutes, we can see the result printed to the logs and should see a new .json file in the output bucket. If it's not there, debug the Cloud Function is running properly using the guide above, redeploy it and try again.
- Furthermore, in the bigquery window we can see new tables were created under the dataset and label data were populated into those tables.
- See the demo video in [YouTube]



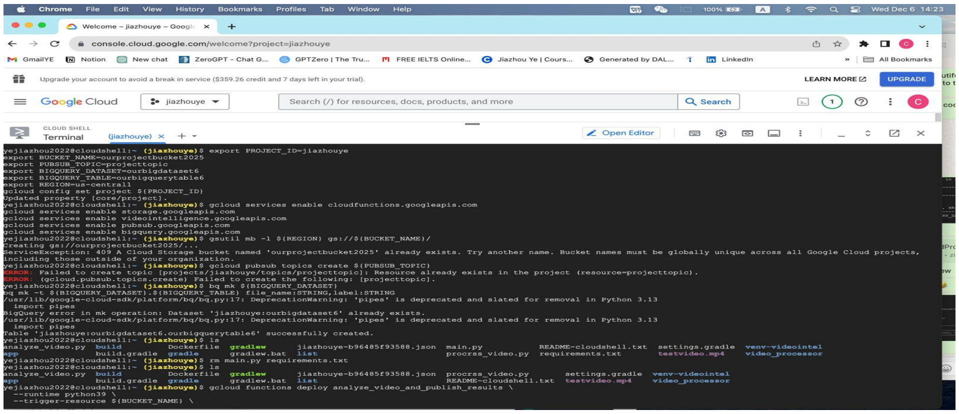
Row	file_name	label	confidence	start_time	end_time	file_uri
1	Lioness Vs Leopard.mp4	national park	0.37712854...	0.0	59.966667	gs://inputbucket-cloud9/Liones...
2	Lioness Vs Leopard.mp4	plant	0.49146404...	0.0	59.966667	gs://inputbucket-cloud9/Liones...
3	Lioness Vs Leopard.mp4	branch	0.55981642...	0.0	59.966667	gs://inputbucket-cloud9/Liones...
4	Lioness Vs Leopard.mp4	animal	0.97508251...	0.0	59.966667	gs://inputbucket-cloud9/Liones...
5	Lioness Vs Leopard.mp4	wildlife	0.87095397...	0.0	59.966667	gs://inputbucket-cloud9/Liones...
6	Lioness Vs Leopard.mp4	nature	0.91097104...	0.0	59.966667	gs://inputbucket-cloud9/Liones...
7	Lioness Vs Leopard.mp4	nature reserve	0.48558920...	0.0	59.966667	gs://inputbucket-cloud9/Liones...
8	Lioness Vs Leopard.mp4	vegetation	0.50274336...	0.0	59.966667	gs://inputbucket-cloud9/Liones...
9	Lioness Vs Leopard.mp4	tree	0.95896977...	0.0	59.966667	gs://inputbucket-cloud9/Liones...
10	Lioness Vs Leopard.mp4	flora	0.32134333...	0.0	59.966667	gs://inputbucket-cloud9/Liones...
11	Military.mp4	troop	0.78685235...	0.0	59.3	gs://inputbucket-cloud9/Militar...
12	Military.mp4	motor vehicle	0.37892872...	0.0	59.3	gs://inputbucket-cloud9/Militar...
13	Military.mp4	vehicle	0.85489577...	0.0	59.3	gs://inputbucket-cloud9/Militar...
14	Military.mp4	war	0.70620822...	0.0	59.3	gs://inputbucket-cloud9/Militar...
15	Military.mp4	soldier	0.92361396...	0.0	59.3	gs://inputbucket-cloud9/Militar...
16	Military.mp4	military	0.99332755...	0.0	59.3	gs://inputbucket-cloud9/Militar...
17	Military.mp4	army	0.95869189...	0.0	59.3	gs://inputbucket-cloud9/Militar...

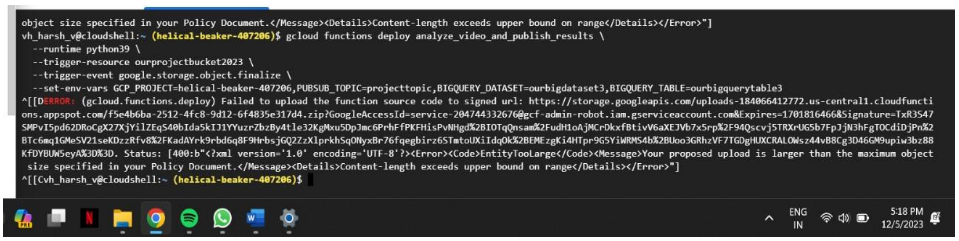
Then we can use this data from BigQuery to export in to Looker Studio Reporting and represent them in graphical format.

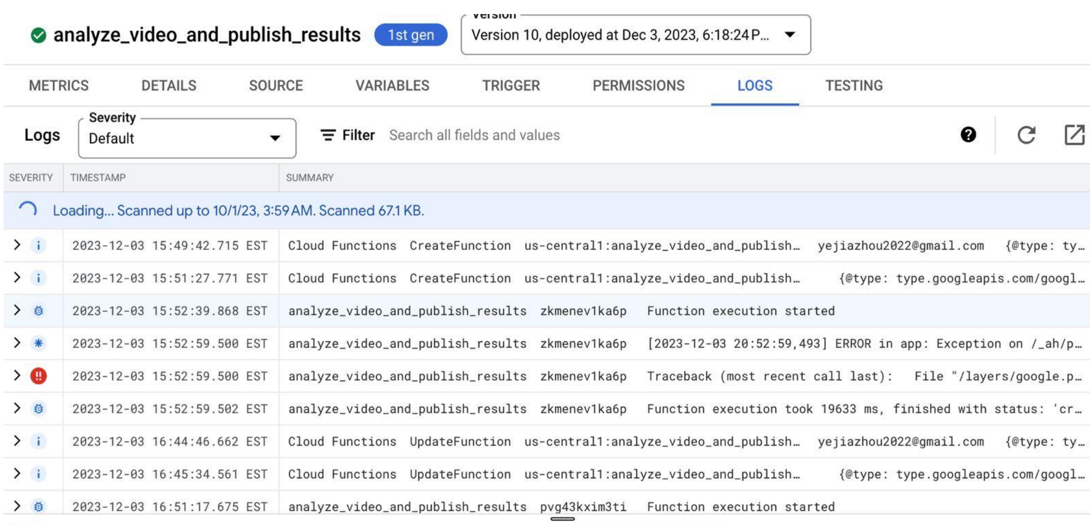


Issue faced while Implementation

- We have faced many issues when we write the cloud function which was throwing errors many times. Access issues, permission issues, Issues with connecting to big data.
- Large size videos are not supported and tend to timeout.
- When giving the proper rights to the functions from the IAM console.
- Executing the python code without errors was the biggest challenge we faced and we were able to successfully overcome it with the help of google cloud logs.


console.cloud.google.com/welcome?project=jiazhouye
Upgrade your account to avoid a break in service ($359.26 credit and 7 days left in your trial).
LEARN MORE
UPGRADE
Google Cloud
jiazhouye
Search (/) for resources, docs, products, and more
Search
CLOUD SHELL
Terminal
(jiazhouye)
Open Editor
yejiazhou2022@cloudshell:~ (jiazhouye)$ export PROJECT_ID=jiazhouye
export BUCKET_NAME=ourprojectbucket2025
export PUBSUB_TOPIC=projecttopic
export BIGQUERY_DATASET=ourbigdataset6
export BIGQUERY_TABLE=ourbigquerytable6
export REGION=us-central1
gcloud config set project ${PROJECT_ID}
Updated property [core/project].
yejiazhou2022@cloudshell:~ (jiazhouye)$ gcloud services enable cloudfunctions.googleapis.com
gcloud services enable storage.googleapis.com
gcloud services enable videointelligence.googleapis.com
gcloud services enable pubsub.googleapis.com
gcloud services enable bigquery.googleapis.com
yejiazhou2022@cloudshell:~ (jiazhouye)$ gsutil mb -l ${REGION} gs://${BUCKET_NAME}/
Creating gs://ourprojectbucket2025/...
ServiceException: 409 A Cloud Storage bucket named 'ourprojectbucket2025' already exists. Try another name. Bucket names must be globally unique across all Google Cloud projects, including those outside of your organization.
yejiazhou2022@cloudshell:~ (jiazhouye)$ gcloud pubsub topics create ${PUBSUB_TOPIC}
ERROR: Failed to create topic [projects/jiazhouye/topics/projecttopic]: Resource already exists in the project (resource=projecttopic).
ERROR: (gcloud.pubsub.topics.create) Failed to create the following: [projecttopic].
yejiazhou2022@cloudshell:~ (jiazhouye)$ bq mk ${BIGQUERY_DATASET}
bq mk -t ${BIGQUERY_DATASET}.${BIGQUERY_TABLE} file_name:STRING,label:STRING
/usr/lib/google-cloud-sdk/platform/bq/bq.py:17: DeprecationWarning: 'pipes' is deprecated and slated for removal in Python 3.13
import pipes
BigQuery error in mk operation: Dataset 'jiazhouye:ourbigdataset6' already exists.
/usr/lib/google-cloud-sdk/platform/bq/bq.py:17: DeprecationWarning: 'pipes' is deprecated and slated for removal in Python 3.13
import pipes
Table 'jiazhouye:ourbigdataset6.ourbigquerytable6' successfully created.
yejiazhou2022@cloudshell:~ (jiazhouye)$ ls
analyze_video.py build Dockerfile gradlew jiazhouye-b96485f93588.json main.py README-cloudshell.txt settings.gradle venv-videointel
app build.gradle gradle gradlew.bat list procrss_video.py requirements.txt testvideo.mp4 video_processor
yejiazhou2022@cloudshell:~ (jiazhouye)$ rm main.py requirements.txt
yejiazhou2022@cloudshell:~ (jiazhouye)$ ls
analyze_video.py build Dockerfile gradlew jiazhouye-b96485f93588.json procrss_video.py settings.gradle venv-videointel
app build.gradle gradle gradlew.bat list README-cloudshell.txt testvideo.mp4 video_processor
yejiazhou2022@cloudshell:~ (jiazhouye)$ gcloud functions deploy analyze_video_and_publish_results \
--runtime python39 \
--trigger-resource ${BUCKET_NAME} \

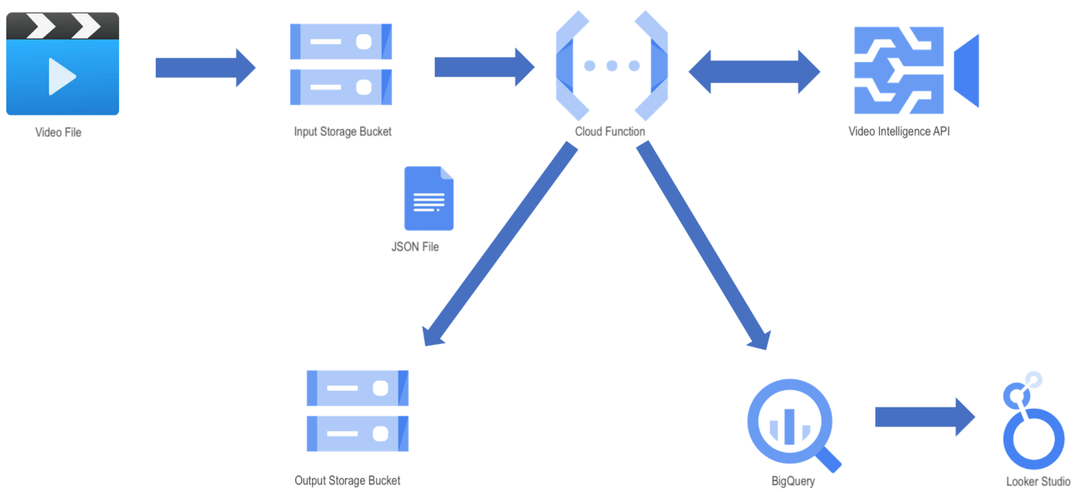

object size specified in your Policy Document.</Message><Details>Content-length exceeds upper bound on range</Details></Error>"]
vh_harsh_v@cloudshell:~ (helical-beaker-407206)$ gcloud functions deploy analyze_video_and_publish_results \
--runtime python39 \
--trigger-resource ourprojectbucket2023 \
--trigger-event google.storage.object.finalize \
--set-env-vars GCP_PROJECT=helical-beaker-407206,PUBSUB_TOPIC=projecttopic,BIGQUERY_DATASET=ourbigdataset3,BIGQUERY_TABLE=ourbigquerytable3
ERROR: (gcloud.functions.deploy) Failed to upload the function source code to signed url: https://storage.googleapis.com/uploads-184066412772.us-central1.cloudfunctions.appspot.com/f5e4b6ba-2512-4fc8-9d12-6f4835e317d4.zip?GoogleAccessId=service-204744332676@gcf-admin-robot.iam.gserviceaccount.com&Expires=1701816466&Signature=
Status: [400:b"<?xml version='1.0' encoding='UTF-8'?><Error><Code>EntityTooLarge</Code><Message>Your proposed upload is larger than the maximum object size specified in your Policy Document.</Message><Details>Content-length exceeds upper bound on range</Details></Error>"]
vh_harsh_v@cloudshell:~ (helical-beaker-407206)$
ENG
IN
5:18 PM
12/5/2023


analyze_video_and_publish_results
1st gen
Version
Version 10, deployed at Dec 3, 2023, 6:18:24 P...
METRICS
DETAILS
SOURCE
VARIABLES
TRIGGER
PERMISSIONS
LOGS
TESTING
Logs
Severity
Default
Filter
Search all fields and values
SEVERITY
TIMESTAMP
SUMMARY
Loading... Scanned up to 10/1/23, 3:59 AM. Scanned 67.1 KB.
2023-12-03 15:49:42.715 EST Cloud Functions CreateFunction us-central1:analyze_video_and_publish… yejiazhou2022@gmail.com {@type: ty…
2023-12-03 15:51:27.771 EST Cloud Functions CreateFunction us-central1:analyze_video_and_publish… {@type: type.googleapis.com/googl…
2023-12-03 15:52:39.868 EST analyze_video_and_publish_results zkmenev1ka6p Function execution started
2023-12-03 15:52:59.500 EST analyze_video_and_publish_results zkmenev1ka6p [2023-12-03 20:52:59,493] ERROR in app: Exception on /_ah/p…
2023-12-03 15:52:59.500 EST analyze_video_and_publish_results zkmenev1ka6p Traceback (most recent call last): File "/layers/google.p…
2023-12-03 15:52:59.502 EST analyze_video_and_publish_results zkmenev1ka6p Function execution took 19633 ms, finished with status: 'cr…
2023-12-03 16:44:46.662 EST Cloud Functions UpdateFunction us-central1:analyze_video_and_publish… yejiazhou2022@gmail.com {@type: ty…
2023-12-03 16:45:34.561 EST Cloud Functions UpdateFunction us-central1:analyze_video_and_publish… {@type: type.googleapis.com/googl…
2023-12-03 16:51:17.675 EST analyze_video_and_publish_results pvg43kxim3ti Function execution started

System architecture

If we want to use AI to detect things in videos (e.g objects, people, logos, speech, text etc), then the Cloud Video Intelligence API might be exactly what we need.[3] The following diagram represents the pipeline of our project that will automatically analyze any video you upload results to a specific Cloud Storage bucket as well as BigQuery.

Block Diagram and pipeline



This data pipeline starts with video inserted into the cloud basket. On upload, a finalize event could be triggered, and video will be sent to the Video Intelligence API. Once it is processed, the result will be stored in another cloud bucket. At the same time, the result data will be sent to BigQuery to store and used in Looker Studio to create a graphical representation.

[3] https://cloud.google.com/video-intelligence-storage-trigger#0
Video Analysis via Video Intelligence API

Google Cloud Video Intelligence API

It is capable of recognizing over 20,000 objects, places, and actions in stored and streaming video. Extract rich metadata at the video, shot, or frame level. Create custom entity labels with AutoML Video Intelligence. We use this API to detect video labels and transcripts from the video clips. Since this is easily accessible and practically free of charge to test a demo project, It is already a trained ML model with a vast amount of data, which takes significant effort and resources. Also, it does not require infrastructure costs.

Google Cloud Storage (GCS)

Cloud Storage is a service for storing objects in Google Cloud easily. An object is an immutable piece of data consisting of a file of any format. They store objects in containers called buckets. Buckets can also contain managed folders, which we use to provide expanded access to groups of objects with a shared name prefix. All buckets should be associated with a project, and we can group projects under an organization. Each project, bucket, managed folder, and object in Google Cloud is a resource that can be easily accessed via compute engine instances.[4]

We use two buckets in our project, one for storing input videos and the other for storing video analysis results as JSON files. Since videos are large, unstructured object files, this is the perfect storage for our scenario. Also, we use this service because it can handle large amounts of data and is cost-effective. As a video surveillance system, the data stored will be in large video files. Furthermore, we use a single region for cost-effective storage as we do not require high availability. Also, we use the standard storage class since we need to access videos immediately

[4] https://cloud.google.com/storage?hl=en

and delete it after one day (or move to Achieve class depending on future requirements of the project). We have set uniform access as we only need to upload the videos, further processing is done via the service account of the cloud function.

Google Cloud Functions

Cloud Functions has a simple and intuitive developer experience. Just write the code and let Google Cloud handle the operational infrastructure. Develop faster by writing and running small code snippets that respond to events. Streamline challenges orchestration problems by connecting Google Cloud products to one another or third party services using events.

We are only billed for the function's execution time, metered to the nearest 100 milliseconds. We pay nothing when your function is idle. Cloud Functions automatically spins up and backs down in response to events.

Use open source FaaS (function as a service) framework to run functions across multiple environments and prevent lock-in. Supported environments include Cloud Functions, local development environment, on-premises, Cloud Run, and other Knative-based serverless environments.

Use Cloud Functions to surface microservices via HTTP APIs or integrate with third-party services that offer webhook integrations to quickly extend applications with powerful capabilities such as sending a confirmation email after a successful Stripe payment or responding to Twilio text message events.[5] We use cloud functions because they only charge us for the function's execution time, which benefits us financially. Also, the cloud function has the benefit of triggering the file upload completion event of the cloud storage bucket. Also, it provides a variety of programming languages to choose from to write our functions. Furthermore, it can easily access other cloud resources, such as BigQuery.

Google BigQuery

BigQuery Studio provides a single, unified interface for all data practitioners of various coding skills to simplify analytics workflows from data ingestion and preparation to data exploration and visualization to ML model creation and use. It also allows the use simple SQL to access Vertex AI foundational models directly inside BigQuery for text processing tasks, such as sentiment analysis, entity extraction, and many more without having to deal with specialized models.

BigQuery editions allow us to pick the right feature set for individual workload requirements with the ability to mix and match for the right price-performance. Compute capacity auto scaling adds fine-grained compute resources in real time to match the needs of workload demands, and ensure only pay for the compute capacity we use. With compressed storage pricing, we can reduce the storage costs while increasing your data footprint at the same time.

BigQuery ML enables data scientists and data analysts to build and operationalize ML models on planet-scale structured, semi-structured, and now unstructured data directly inside BigQuery, using simple SQL—in a fraction of the time. Export BigQuery ML models for online prediction

[5] https://cloud.google.com/functions?hl=en

into Vertex AI or serving layer. Learn more about the models we currently support.[6] We use this because it allows us to efficiently store large amounts of data and provide fast execution of SQL query results. It also provides easy handling of sore JSON files for future steps of this project. Moreover, it reduces the overall cost of storage.

Google Looker Studio

This google product can transform data to impactful business metrics and dimensions with intuitive smart reports and graphs. We can share compelling reports with our team or with the world, collaborating in real time. We use this platform to display the result from video analysis from the Video intelligence API since it provides easy integration for BigQuery and free of charge to use.

[6] https://cloud.google.com/bigquery?hl=en

Conclusion and Future plans

In conclusion this project discusses step by step implementation and integration of several google cloud products. future path of this project. It benefits in real life and describes all the functions used while implementing. There are still many challenges to overcome like shadow detection, weather changes, dynamic changes, and accuracy in a crowded environment. How to deal with the large files and high number of file processing at the same time. We will need to modularise the cloud function and use cloud Pub/Sub stream to handle a high load of incoming videos.

As a final product to market, this project can be improved in multiple ways like

- Video processing and AI integration to identify vehicle thieves
    - A system that integrated with vehicle cameras which detect people moving around it while parked and use realtime camera footage(its own and nearby other vehicles which have the same system installed). Video intelligence processing and AI to identify vehicle thieves and take preventive actions against vandalism.
- CCTV alarm processing system streamline process like
    - once run the set up, it create all the resources and pipeline to access realtime video footage from 16 CCTV cameras in a house, process it parallelly use image enhancement tech to improve process speed, detect humans in the video frames and process those human images against familiar face database which given by the user to filter out strangers. process those video clips with ML/AI to decide intruders or not then proceed to alarm Owner over his mobile or given contact to Police providing evidence as well.

Future project recommendations

- Improve Cloud function to modularise and support analyse real time video files. Then publish into pub/sub stream for further analyse the result using ML.
- Create a distributed system that interacts with relevant feeds to generate accurate context of the video field to predict what is actually going on.
- Create infrastructure for report users or inform authorities via Pub/Sub alert system and integrate with main system
- Create infrastructure to use IoT devices to take further custom actions predefined by the user.

Bibliography

1. Cheng, B., Yang, J., Wang, S., & Chen, J. (2015). Adaptive Video Transmission Control System based on reinforcement learning approach over heterogeneous networks. *IEEE Transactions on Automation Science and Engineering*, *12*(3), 1104–1113. https://doi.org/10.1109/tase.2014.2387212
2. Ambawane, P., Bharatia, D., & Rane, P. (2019). Smart e-stick for Visually Impaired using Video Intelligence API. *Conference: 2019 IEEE Bombay Section Signature Conference (IBSSC)*. https://doi.org/10.1109/ibssc47189.2019.8973060
3. *Google Cloud documentation | Documentation*. (n.d.). Google Cloud. https://cloud.google.com/docs/
4. Islam, R., Iqbal, F., Akhter, S., Rahman, M., & Khan, R. (2023). Deep learning based object detection and surrounding environment description for visually impaired people. *Heliyon*, *9*(6), e16924. https://doi.org/10.1016/j.heliyon.2023.e16924
5. Gharat, M., Patanwala, R., & Ganaparthi, A. (2017). Audio guidance system for blind. In *International Conference on Electronics Communication and Aerospace Technology*
6. Thakare, P. U., Shubham, K., Pawale, A., Ajinkya, R., & Shelke, O. (2017). Smart Assistance System for the Visually Impaired. *International Journal of Scientific and Research Publications, 7*(12), December 2017.
7. Saquib, Z., Murari, V., & Bhargav, S. N. (2017). BlinDar: An Invisible Eye for the Blind People. In *IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 2017.*

8. Nada, A., Mashelly, S., Fakhr, M. A., & Seddik, A. F. (2015). Effective Fast Response Smart Stick for Blind People. *Research Gate Conference Paper, April 2015.*
9. Sekar, D., Sivakumar, S., Thiyagarajan, P., Premkumar, R., & Vivek Kumar, M. (2016). Ultrasonic and Voice Based Walking Stick for Blind People. *International Journal of Innovative Research in Electrical Electronics Instrumentation and Control Engineering, 4*(3), March 2016.
10. Mashat, M. D., & Albani, A. A. (2017). Intelligent Blind Cane System. In *UKSim-AMSS 19th International Conference on Modelling Simulation.*
11. Kanzariya, S., & Vora, P. V. (2015). Real Time Video Monitoring System Using Raspberry Pi. In *National Conference on Emerging Trends in Computer Electrical Electronics.*
12. Zhang, L., Li, S. Z., Yuan, X., & Xiang, S. (2007). Real-time Object Classification in Video Surveillance Based on Appearance Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*.
13. Selvi, M. S., & Chellam, J. A. I. (2018). Smart Video Surveillance: Object Detection Tracking and Classification. *International Journal of Innovations Advancement in Computer Science, March 2018.*
14. Valentín, L., Serrano, S. A., García, R. O., Andrade, A., Palacios-Alonso, M. A., & Sucar, L. E. (2017). A Cloud-Based Architecture for Smart Video Surveillance. In *2nd International Conference on Smart Data and Smart Cities, October 2017.*
15. Google Cloud. (2023). Vodafone picks Google Cloud to safely share mobile phone data. Retrieved from

https://cloud.google.com/blog/products/data-analytics/vodafone-picks-google-cloud-to-safely-share-mobile-phone-data

16. de Diego, I. M., Román, I. S., Montero, J. C., Conde, C., & Cabello, E. (2019). Scalable and flexible wireless distributed architecture for intelligent video surveillance systems. *Multimedia Tools Appl., 78*(13), 17437–17459. https://doi.org/10.1007/s11042-018-7035-5
17. Kumar, N., Lee, J.-H., & Rodrigues, J. J. P. C. (2015). Intelligent mobile video surveillance system as a Bayesian coalition game in vehicular sensor networks: Learning automata approach. *IEEE Trans. Intell. Transp. Syst., 16*(3), 1148–1161. https://doi.org/10.1109/TITS.2014.2368262
18. Hu, P., Tan, W. L., Wishart, R., Portmann, M., & Indulska, J. (2010). MeshVision: An adaptive wireless mesh network video surveillance system. *Multimedia Syst., 16*(4–5), 243–254. https://doi.org/10.1007/s00530-010-0181-8
19. Lubobya, S. C., Dlodlo, M. E.-H., de Jager, G., & Zulu, A. (2018). Mesh IP video surveillance systems model design and performance evaluation. *Wireless Pers. Commun., 100*(2), 227–240. https://doi.org/10.1007/s11277-018-5389-5
20. Lim, K.-W., Seo, Y., Jung, W.-S., Ko, Y.-B., & Park, S. (2013). Design and implementation of adaptive WLAN mesh networks for video surveillance. *Wireless Netw., 19*(7), 1511–1524. https://doi.org/10.1007/s11276-013-0628-8
21. Zhizhuo, S., Yu-An, T., & Yuanzhang, L. (2014). An energy-efficient storage for video surveillance. *Multimedia Tools Appl., 73*(1), 151–167. https://doi.org/10.1007/s11042-014-2082-7

22. Xiao, Y., Changyou, Z., Yuan, X., Hongfei, Z., Yuanzhang, L., & Yu-An, T. (2018). An extra-parity energy saving data layout for video surveillance. *Multimedia Tools Appl., 77*(4), 4563–4583. https://doi.org/10.1007/s11042-017-4803-5
23. Subudhi, B. N., Rout, D. K., & Ghosh, A. (2019). Big data analytics for video surveillance. *Multimedia Tools Appl., 78*(18), 26129–26162. https://doi.org/10.1007/s11042-019-08903-2
24. Alamri, A., Hossain, M. S., Almogren, A., Hassan, M. M., Alnafjan, K., Zakariah, M., Seyam, L., & Alghamdi, A. (2016). QoS-adaptive service configuration framework for cloud-assisted video surveillance systems. *Multimed. Tools Appl., 75*(21), 13333–13348. https://doi.org/10.1007/s11042-015-2888-7
25. Hossain, M. S. (2014). QoS-aware service composition for distributed video surveillance. *Multimedia Tools Appl., 73*(1), 169–188. https://doi.org/10.1007/s11042-014-1977-8
26. Zhang, W., Duan, P., Xie, X., Xia, F., Lu, Q., Liu, X., & Zhou, J. (2016). QoS4IVSaaS: A QoS management framework for intelligent video surveillance as a service. *Pers. Ubiquitous Comput., 20*(5), 795–808. https://doi.org/10.1007/s00779-015-0885-8
27. Cloud Video Intelligence API triggered by Cloud Storage by Zack Akil.
28. Nazare, A. C., Jr., & Schwartz, W. R. (2016). A scalable and flexible framework for smart video surveillance. *Comput. Vis. Image Understand., 144*, 258–275. https://doi.org/10.1016/j.cviu.2015.11.010
29. Rouhani, B. D., Mirhoseini, A., & Koushanfar, F. (2017). RISE: An automated framework for real-time intelligent video surveillance on FPGA. *ACM Trans. Embedded Comput. Syst., 16*(5s), Art. no. 158. https://doi.org/10.1145/2998453

30. Cherukuri, P. A. A., Linh, N. T. D., Vududala, S. K., & Cherukuri, R. (2021). Keras convolutional neural network model in sequential build of 4 and 6 layer architectures. In *Intelligent Systems and Networks.* Springer Singapore; Singapore: pp. 465–472.