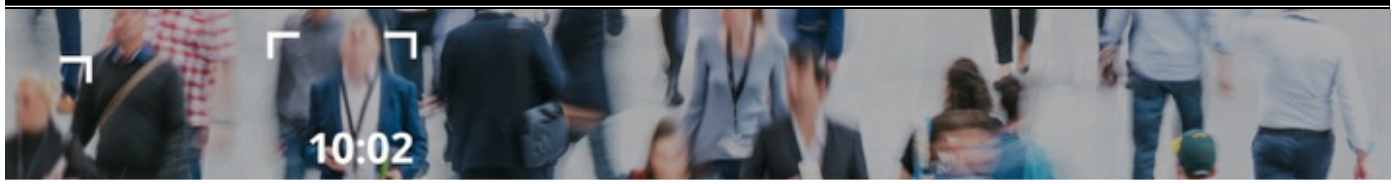
BriefCam

TRANSFORMING VIDEO SURVEILLANCE INTO ACTIONABLE INTELLIGENCE



BriefCam® White Paper

Maximizing Your Video Processing Performance

September 2020

Contents

Introduction

BriefCam is a powerful suite of solutions intended to unlock the data buried in video footage. This is done by processing the unstructured video and transforming it to structured data, which can then be used for the various BriefCam solutions.

BriefCam converts a raw video stream, either a file or a camera, from its proprietary format into a format that BriefCam can process. BriefCam then takes the video and processes it with computer vision and deep learning technologies.

As BriefCam processes video, it detects and recognizes objects, along with information about their type and attributes. Objects are then classified according to different classes, attributes, color, dwell time, size and speed.

BriefCam processes the video stream once and then stores all metadata in the database.

This document discusses the video processing pipeline and explains how users and administrators can optimize its throughput and maximize performance.

Processing Warmup

When the processing service is started, it takes time to load the different DNN networks and to learn the video characteristics before any valuable information can be extracted. During the warmup period, BriefCam is attempting to learn the first frames by determining the frame rate, figuring out what parts of the frame are the background. This is why, for example, when you run a RESPOND rule, alerts originating from the rule might not begin immediately.

Processing Queue

The processing queue in BriefCam is optimized to assure the fastest possible generation of synopses. If you select a date/time range that coincides – even partially – with that of a previously processed video source, BriefCam will make optimal use of the respective video segments that have already been processed to reduce processing time. If a new source's date/time range overlaps a previous source's range completely, no processing will take place, and the source will instantaneously be available for consumption.

Processing Concepts

There are three main concepts to cover when explaining video processing performance and accuracy with BriefCam: throughput, task length, and parallel workers.

Throughput

Throughput is used to measure the performance of the processing. Throughput is defined as how many hours of video can be processed in one hour of continuous processing by BriefCam's processing server.

For example, a throughput of 10 Hs/H means that it will take BriefCam's processing server one hour to completely process a 10-hour video (assuming it is the only video currently being processed).

Throughput is a measured number that is influenced by the resolution, the GPU characteristics and the complexity of the scene, including the number of objects, frame rate, the noise in the scene, such as flags that are flapping or trees that are moving, as well as a changing background, such as the moving shadows as the sun changes angle throughout the day.

Task Length

A task represents a chunk of video submitted to the BriefCam engine for processing. A task is handled by a GPU worker.

When BriefCam is processing a task, it learns throughout the length of the task. BriefCam learns about distance, size of near and far objects, noise, background and so on. This learning process significantly improves accuracy.

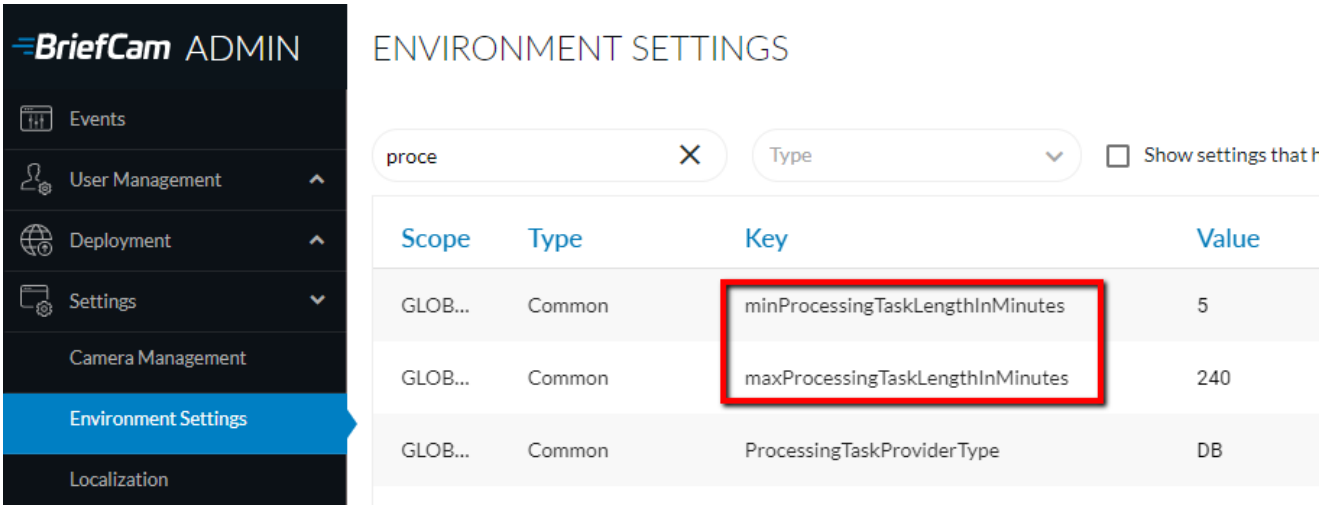
For this learning process, the longer the task, the better the accuracy.

On the other hand, splitting tasks into smaller chunks increases the throughput of the system since more tasks can be run in parallel and can take advantage of the GPU parallel architecture.

Based on extensive testing, we have identified the 4 hours' mark as the sweet spot for balancing between best accuracy and throughput optimization.

This does not mean that shorter videos will not work. It just means that better accuracy and throughput are achieved with longer videos.

When a long video (longer than 4 hours) is submitted to BriefCam, it is split into 4-hour chunks and each chunk is processed by a task in parallel. The length of the chunks is set in the **maxProcessingTaskLengthInMinutes** admin setting (as seen in the image below).



Parallel Workers

Now let's explain the workers and their influence on performance.

A GPU can handle several parallel processing tasks and those tasks are processed by workers. Each GPU can handle a certain number of parallel workers based on its throughput and RAM.

The number of parallel tasks does not change the overall throughput of the GPU (with one exception, which we'll discuss later).

To demonstrate this point, let's assume that a certain GPU has a throughput of 10 hours of video in 1 hour of processing and there is no limit to the length of a task.

In theory, if a single video of 10 hours is submitted, it will be processed by a single worker and will be finished in 1 hour.

Assuming a single GPU, if two 10-hour videos are submitted, each will be processed in parallel by a worker (if available). Since the overall GPU throughput is distributed among the workers, it will take 2 hours to process the two 10-hour videos. In the above example, adding another GPU will double the throughput. As a result, if the two workers processing the two 10-hour videos will be distributed between the two GPUs – this will reduce the time it takes to complete the processing from 2 hours to 1 hour.

This resource distribution is approximate, since it also relies on resolution, video complexity and specific objects appearing in the videos.

In actuality, when it comes to performance optimization, the best performance is obtained when at least 4 workers are processing in parallel. This can be achieved by submitting a long video (12+ hours), by submitting at least 4 videos or any combination of the two.

If we have a video of 13 hours, it's divided into four tasks of three hours each, and then one additional task for the additional hour.

When running a single task, you should expect a 30-35% decrease in performance. When running two parallel tasks you should expect a 20% decrease in performance. Three tasks will give you a 10% decrease and with four workers you will get the full throughput.

Number of workers	Throughput
1	30-35% decrease in performance
2	20% decrease in performance
3	10% decrease in performance
4	Full throughput

BriefCam Processing Server – RAM Considerations

BriefCam's processing server requires both CPU and the GPU memory resources to operate.
The below table lists the GPU and CPU requirements.

Entity	GPU RAM	Machine RAM
Fixed memory requirements		
Operating system	Not connected to screen	5.3 GB
Processing engine	2.44 GB	2.5 GB
Requirements per worker		
Single worker – 4K	0.98 GB	8 GB
Single worker – 1080P	0.35 GB	4 GB
Single worker – 720P	0.16 GB	2.1 GB
Single worker – 4CIF	0.07 GB	1 GB

For example, when configuring the system to handle 10 parallel tasks and the expected video quality is 1080P, you should expect:

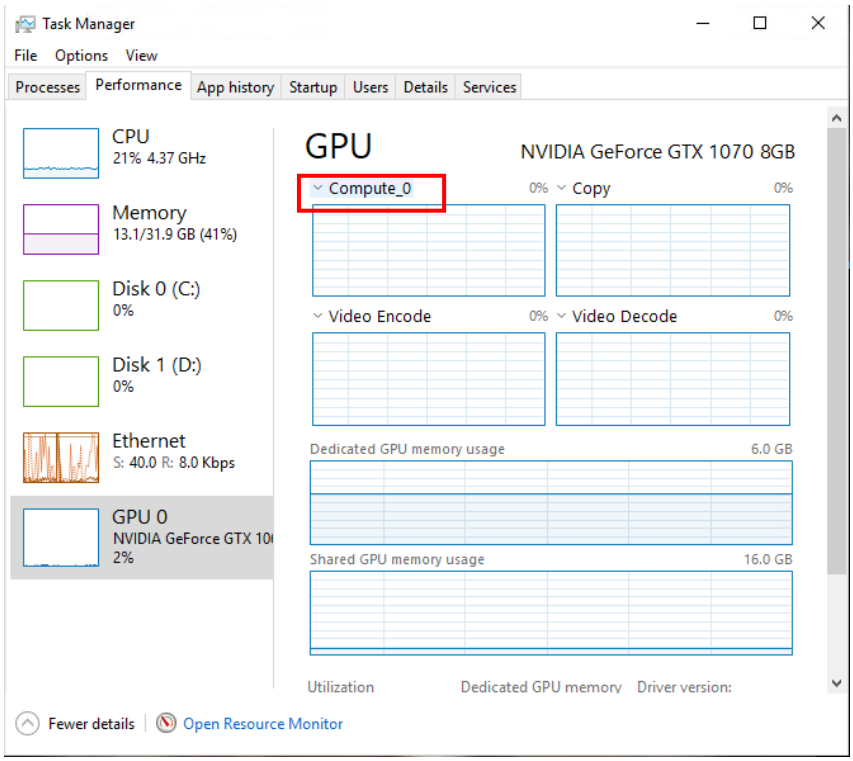
GPU RAM: 0 (OS) + 2.44 (PS) + 10*0.35 (Worker) = GPU with at least 5.94GB RAM

Machine RAM: 5.3 (OS) + 2.5 (PS) + 10*4 (Worker) = Machine with at least 47.8 GB RAM dedicated to BriefCam

GPU Processing Explanation

To understand how the GPU is being used, take into account the following three points.

1. To check the actual utilization, look at **Compute_0** in the Task Manager (as shown in the image below) and not in other locations.



2. Even if the GPU is used, the CPU is still involved. That is why you will see that it's not completely idle and this is the expected behavior.

If your video does not allow hardware coding and cannot be decoded on the GPU resource, the CPU will be involved and not the GPU.

BriefCam

www.BriefCam.com

Visit us on social media