Control de Ingresos y Egresos de Vehículos en Patios de Distribución mediante Redes Neuronales Convolucionales y RFID en Raspberry Pi con Unidad Hailo-8L

Profesora Titular: Ing. Grettel Barceló Alonso, PhD

Equipo #20

Avance 6. Conclusiones clave

- A01794188 - Francisco Xavier Bastidas Moreno
- A01794653 - Raúl Jesús Coronado Aguirre
- A01794327 - Juan Sebastián Ortega Briones

Análisis de los resultados

En este avance, se busca fundamentar si es viable implementar el modelo, evaluando su adecuación a los criterios de éxito previamente establecidos y considerando los resultados obtenidos durante la etapa de modelado. En cualquier caso, será importante también incluir los siguientes cuestionamientos:

¿El rendimiento del modelo es lo suficientemente bueno para su implementación en producción?

El rendimiento del modelo es excelente para desplegarlo en nuestro caso particular, ya que hicimos ajustes específicos para los ángulos de cámara, iluminación y objetos a omitir que son comunes en el entorno a solucionar.

Para que el modelo fuera escalable a muchas más aplicaciones, se debería adaptar a todo tipo de ángulo de cámara, iluminación, entorno y otras variables que dependen mucho del área a analizar. En nuestro caso, es un ángulo de cámara en que los autos se ven en su mayor parte en su cara superior, un ángulo en el que no todos los modelos comunes y de fuente abierta son entrenados, ya que se acostumbran a usar para detectar frentes de carros o ángulos donde existen más características principales. Puede ser el caso que un proyecto, su infraestructura sea con cámaras que apunten del suelo hacia arriba, lo que causaría otro problema al no tener modelos entrenados desde ese ángulo, por lo que el fine-tuning para cada caso es esencial para poder implementarlo en producción con una escalabilidad alta.

¿Existe margen para mejorar aún más el rendimiento?

Siempre existe margen para mejorar, especialmente en un campo tan dinámico como el de la inteligencia artificial.

Algunas propuestas de mejora son:

Mejora del Algoritmo de Seguimiento

- **Implementación de Filtros de Kalman**: Los filtros de Kalman pueden predecir la posición futura de un objeto basándose en su estado actual y modelos de movimiento, lo que mejora la estabilidad del seguimiento y reduce la pérdida temporal de objetos. Esto es especialmente útil en escenarios donde la detección puede fallar intermitentemente.
- **Uso de SORT (Simple Online and Realtime Tracking)**: SORT es un algoritmo de seguimiento que combina detección y predicción de movimiento para facilitar un seguimiento más robusto y eficiente en tiempo real. Es sencillo de implementar y puede mejorar significativamente el rendimiento del seguimiento sin requerir recursos computacionales intensivos.
- **Integración de Deep SORT**: Como una extensión de SORT, Deep SORT incorpora características de apariencia utilizando redes neuronales, lo que permite mantener la identidad de los objetos incluso en presencia de múltiples objetos similares. Esto mejora la asociación de detecciones entre frames y reduce la tasa de error en el seguimiento.
- **Exploración de Algoritmos Avanzados como Tracktor o ByteTrack**: Estos algoritmos han demostrado un rendimiento superior en diversos escenarios y podrían ofrecer mejoras adicionales en el seguimiento. Dependiendo de los recursos disponibles y las necesidades específicas del proyecto, podrían ser opciones viables para considerar.
- **Optimización del Algoritmo Existente**: Dado que el algoritmo actual solo mantiene un arreglo de los **bounding boxes** y permite un número determinado de frames con pérdidas, mejorar este enfoque podría ser un paso inicial. Ajustar los parámetros de tolerancia a las pérdidas o incorporar métodos de asociación más sofisticados puede incrementar la eficacia sin requerir una revisión completa del sistema.

Sustitución del hardware por alguno específico para estos flujos de trabajo como Jetson.

Aunque el hardware de Hailo es específico para este flujo de trabajo, es demasiado nuevo y no están implementadas todas las librerías para su soporte, por ejemplo, las librerías de SORT, en otros casos si estén implementadas, pero no están bien documentadas o solo están implementadas para C++ o C#.

Migrar este modelo a Jetson podría permitir el uso de más librerías somo SORT, ByteTrack, etc.

Otras

Podríamos explorar la integración de tecnologías complementarias, como sistemas de visión estereoscópica, para mejorar la detección en condiciones de baja visibilidad o durante la noche. Además, la implementación de algoritmos de aprendizaje automático más avanzados o la actualización a versiones más potentes de YOLO podrían ofrecer mejoras en la precisión y la velocidad de procesamiento. Al igual que estar al tanto de las nuevas tecnologías emergentes que son desarrolladas con tanta frecuencia en estos tiempos.

Sin embargo, es importante reconocer que, si el modelo está funcionando para lo que el cliente necesita, en este caso, un desarrollo para proveer una solución particular, y si el modelo soluciona este problema, no es necesario caer en la búsqueda de la perfección y tratar de optimizar por el hecho de conseguir métricas cercanas a la perfección, ya que podría retrasar el despliegue de la herramienta. Con datos históricos y nuestro modelo ya en funcionamiento, podemos hacer ajustes y mejoras; incluso podemos ofertar mayores funcionalidades como un registro completo en una plataforma de seguimiento con creación de dashboards detallados de las entradas y salidas de sus productos.

Una de las áreas clave para la mejora continua es la adaptación del modelo a diferentes tipos de hardware, que pueden variar desde Raspberry Pi hasta computadoras con potentes GPUs, servidores con capacidades de GPU avanzadas o incluso cámaras que tienen tecnología de procesamiento integrada.

La optimización del rendimiento del modelo en estos dispositivos diferentes puede ofrecer mejoras significativas en términos de velocidad de procesamiento y eficiencia energética, lo cual es crucial para aplicaciones en tiempo real. Por ejemplo, mientras que una Raspberry Pi puede ser adecuada para prototipos y aplicaciones de baja escala, una GPU en una computadora o servidor puede procesar datos mucho más rápido, lo que resulta en tiempos de respuesta más rápidos y la capacidad de manejar un volumen mucho mayor de datos en tiempo real.

Además, la optimización del modelo para diferentes plataformas requerirá una inversión variada en hardware, que dependerá directamente de la cantidad de inversión que los stakeholders estén dispuestos a realizar. Con una inversión adecuada, se pueden explorar mejoras en la implementación de algoritmos de aprendizaje profundo más avanzados o la actualización a versiones más recientes y potentes de modelos como YOLO, lo que podría mejorar significativamente tanto la precisión como la eficiencia del sistema.

Estas mejoras no solo dependen de la capacidad tecnológica, sino también del compromiso financiero de las partes interesadas para apoyar las actualizaciones y el mantenimiento continuo del sistema. Esto permitirá una adaptación más eficiente del sistema a las cambiantes demandas y condiciones del entorno operativo, asegurando que la solución siga siendo competitiva y efectiva a lo largo del tiempo.

¿Cuáles serían las recomendaciones clave para poder implementar la solución?

Existirían 3 situaciones a considerar, si ya se tiene un hardware con el que se tiene que trabajar, si se quiere hacer uso del hardware actual y agregar ciertas cosas o si se tiene el interés de comprar el hardware para asegurar el rendimiento optimo del modelo

Situación 1: Uso de Hardware Existente

Consideraciones:

- **Hardware**: Evaluar las especificaciones del hardware existente para determinar si cumple con los requisitos mínimos necesarios para ejecutar el modelo de manera eficiente. Esto incluye procesador, memoria, capacidad de almacenamiento y capacidades gráficas si es relevante.
- **Uso del Programa**: Ajustar la configuración del modelo para optimizar el uso de recursos del hardware actual sin comprometer demasiado el rendimiento. Esto puede implicar modificar la resolución de entrada, reducir la tasa de frames o simplificar el modelo.
- **Condiciones de las Capturas de las Cámaras**: Analizar las limitaciones que las configuraciones actuales de las cámaras puedan presentar, como ángulos fijos o iluminación insuficiente, y adaptar el modelo para compensar estas limitaciones.
- **Tipos de Autos**: Asegurarse de que el modelo actualizado continúe reconociendo con eficacia la variedad de autos que normalmente ingresan y salen del patio, ajustando la capacidad del modelo para generalizar a partir de los datos visuales disponibles.

Situación 2: Mejora del Hardware Actual

Consideraciones:

- **Hardware**: Identificar componentes del hardware actual que pueden ser mejorados, como aumentar la memoria RAM, mejorar la CPU o agregar GPUs para acelerar el procesamiento.
- **Uso del Programa**: Optimizar el software para aprovechar las mejoras en el hardware, lo cual podría incluir habilitar procesamiento en paralelo o utilizar versiones más avanzadas del modelo que eran previamente no ejecutables.
- **Condiciones de las Capturas de las Cámaras**: Mejorar la infraestructura de cámaras si es necesario, como agregar cámaras con mejores capacidades en condiciones de baja luz o cámaras con mayor resolución, para mejorar la calidad de los datos recogidos. Agregar iluminación al entorno, eliminar obstrucciones o delimitar una zona en la que solo puedan pasar los autos.
- **Tipos de Autos**: Expandir la base de datos de entrenamiento para incluir más variabilidad en los tipos de autos, asegurando que el sistema mejorado pueda identificar correctamente una gama más amplia de vehículos.

Situación 3: Adquisición de Nuevo Hardware

Consideraciones:

- **Hardware**: Investigar y adquirir el mejor hardware disponible que garantice el rendimiento óptimo del modelo, considerando las últimas tecnologías en procesamiento, almacenamiento y rendimiento gráfico. Al igual que si se puede diseñar la entrada de los autos se puede asegurar que solo pasen los autos, así eliminamos cualquier ruido que otro objeto pueda ocasionar.

- **Uso del Programa**: Configurar el sistema desde cero para maximizar las capacidades del nuevo hardware, utilizando las configuraciones de software más avanzadas y eficientes.
- **Condiciones de las Capturas de las Cámaras**: Diseñar un sistema de cámaras que integre lo último en tecnología de visión por computadora, incluyendo cámaras con capacidades de ajuste automático de enfoque y exposición, amplio rango dinámico y alta resolución.

- **Tipos de Autos**: Asegurar que el modelo sea capaz de identificar una amplia variedad de vehículos, incorporando un extenso conjunto de datos de entrenamiento que refleje la diversidad de vehículos que el sistema espera procesar.

Cada una de estas situaciones requiere un enfoque cuidadoso y considerado para asegurar que las decisiones tomadas maximicen la efectividad del sistema y proporcionen una solución sostenible y escalable para el control de ingresos y egresos de vehículos en patios de distribución, al igual que la solución se puede ajustar al nivel de precisión que sea necesario esto en un escenario en que los inversionistas conozcan los beneficios que podría traer y que la calidad delos modelos depende en una parte de las capacidades de cómputo que se cuenten.

Accionables para las partes interesadas

Para implementar la solución en tiempo real, las partes interesadas deben evaluar plataformas en la nube como Azure, AWS, GCP e IBM Watson. El análisis debe considerar la facilidad de uso, escalabilidad, costos y servicios especializados que ofrece cada proveedor. Azure se destaca por su oferta competitiva en GPU, pero se recomienda realizar una evaluación de costos en comparación con el hardware local. Este análisis debe proyectar los gastos operativos, mantenimiento y personal especializado para determinar la mejor opción en términos de retorno de inversión (ROI).

Además, se debe planificar la escalabilidad y monitoreo del sistema para asegurar su rendimiento en tiempo real, estableciendo sistemas de alertas y opciones de administración global en caso de crecimiento. También es fundamental que las partes interesadas desarrollen una justificación técnica y económica para el proveedor elegido, explicando cómo esta elección respalda la sostenibilidad del modelo de negocio y el uso

eficiente de los recursos financieros y tecnológicos. Esto permitirá una decisión informada que equilibre los requisitos del proyecto con la sostenibilidad económica y operativa de la solución de ML.

Adicionalmente se recomienda que los stakeholdres promuevan una cultura de información interna que permita a los usuarios conocer la solución a fin de evitar algunas prácticas que podrían afectar el modelo como modificar las áreas donde fue entrenado el modelo, mover las cámaras, etc.

Análisis de plataformas

Para nuestro flujo de trabajo en las plataformas tradicionales de la nube no encontramos servicios de Software as a Service (SaaS) para este tipo de inferencias, existen para imágenes, pero no para video en tiempo real. Por lo que las opciones son el uso de máquinas virtuales con GPU para implementar nuestro modelo.
Comparando las opciones encontramos que, la máquina virtual con GPU más económica está en Azure.

Máquina virtual con GPU más económica (precio en dólares por mes)			
Azure	AWS	Google	IBM
54.37	165.71	204.12	1,336.74

La capacidad requerida para nuestro modelo no es importante en CPU y memoria, pero si en GPU, así que la opción de la VM de Azure se ajusta perfecto.

Estos son algunas opciones de Azure:

GPU

Specialized virtual machines targeted for heavy graphic rendering and video editing available with single or multiple GPUs.

NC-series

N-series virtual machines are ideal for compute and graphics-intensive workloads, helping customers to fuel innovation through scenarios like high-end remote visualization, deep learning, and predictive analytics. NC-series virtual machines feature the NVIDIA Tesla accelerated platform and these virtual machines do not support the NVIDIA RTX Enterprise technology for graphics and visualization applications. In addition, N-series offers a NC24r configuration that provides a low latency, high-throughput network interface optimized for tightly coupled parallel computing workloads.

Instance	vCPU(s)	RAM	Temporary storage	GPU	Pay as you go	1 year savings plan	3 year savings plan	Spot	Add to estimate
NC6	6	56 GiB	340 GiB	1X K80	$657.0000/month	$481.2525/month ~26% savings	$349.6554/month ~46% savings	$91.9800/month ~86% savings	+
NC12	12	112 GiB	680 GiB	2X K80	$1,314.0000/month	$962.5050/month ~26% savings	$699.3108/month ~46% savings	$183.9600/month ~86% savings	+
NC24r	24	224 GiB	1,440 GiB	4X K80	$2,890.8000/month	$2,117.5110/month ~26% savings	$1,538.4823/month ~46% savings	$404.7120/month ~86% savings	+
NC24	24	224 GiB	1,440 GiB	4X K80	$2,628.0000/month	$1,925.0100/month ~26% savings	$1,398.6216/month ~46% savings	$367.9200/month ~86% savings	+

Reserved Virtual Machine Instances are currently not available for the NC-series.

The Azure NC-series Virtual Machine sizes will be retired on August 31, 2023. Refer to our migration guide for help on next steps.

NCas_T4_v3-series

NCas_T4_v3-series virtual machine is a new addition to the Azure GPU family specifically designed for the AI and machine learning workloads. The VMs feature 4 NVIDIA T4 GPUs with 16 GB of memory each, up to 64 non-multithreaded AMD EPYC 7V12(Rome) processor cores, and 448 GiB of system memory. These virtual machines are ideal to run ML and AI workloads utilizing Cuda, TensorFlow, Pytorch, Caffe, and other Frameworks or the graphics workloads using NVIDIA RTX Enterprise technology.

Instance	vCPU(s)	RAM	Temporary storage	GPU	Pay as you go	1 year savings plan	3 year savings plan	Spot	Add to estimate
NC4as T4 v3	4	28 GiB	180 GiB	1X T4	$383.9800/month	$259.6464/month ~32% savings	$180.6239/month ~52% savings	$54.3719/month ~85% savings	+
NC8as T4 v3	8	56 GiB	360 GiB	1X T4	$548.9600/month	$371.2050/month ~32% savings	$258.2302/month ~52% savings	$77.7326/month ~85% savings	+
NC16as T4 v3	16	110 GiB	360 GiB	1X T4	$878.9200/month	$594.3222/month ~32% savings	$413.4428/month ~52% savings	$124.4548/month ~85% savings	+
NC64as T4 v3	64	440 GiB	2,880 GiB	4X T4	$3,176.9600/month	$2,148.2586/month ~32% savings	$1,494.4414/month ~52% savings	$449.8574/month ~85% savings	+

NVads A10 v5 series

The NVads A10 v5 is based on the Nvidia A10 GPU and optimized for graphics and visualization workloads. The NVads A10 v5 VM-series provides GPU resourcing flexibility and allows you to choose VMs starting with 1/6th of GPU and scale to multi-GPU configurations. The GPU partitioning technology combined with the powerful AMD Milan processors and up to 880GB of memory, provides right sized GPU VMs from entry level VDI workload to high end graphics workload requiring more vCPU, Memory and GPU resources.

Instance	vCPU(s)	RAM	Temporary storage	GPU	Pay as you go	1 year savings plan	3 year savings plan	Spot	Add to estimate
NV6adsv5	6	55 GiB	180 GiB	1/6X A10	$331.4200/month	$275.8743/month ~16% savings	$204.4511/month ~38% savings	$66.2840/month ~80% savings	+
NV12ads A10 v5	12	110 GiB	320 GiB	1/3X A10	$662.8400/month	$551.7486/month ~16% savings	$408.9095/month ~38% savings	$132.5680/month ~80% savings	+
NV18ads A10 v5	18	220 GiB	720 GiB	1/2X A10	$1,168.0000/month	$972.2432/month ~16% savings	$720.5392/month ~38% savings	$233.6000/month ~80% savings	+
NV36ads A10 v5	36	440 GiB	720 GiB	1X A10	$2,336.0000/month	$1,944.4864/month ~16% savings	$1,441.0784/month ~38% savings	$467.2000/month ~80% savings	+
NV36adms A10 v5	36	880 GiB	720 GiB	1X A10	$3,299.6000/month	$2,746.5885/month ~16% savings	$2,035.5247/month ~38% savings	$659.9200/month ~80% savings	+
NV72ads A10 v5	72	880 GiB	1,400 GiB	2X A10	$4,759.6000/month	$3,961.8925/month ~16% savings	$2,936.1987/month ~38% savings	$951.9200/month ~80% savings	+

Estas son las opciones de AWS:

Instancias recomendadas GPU

PDF | RSS

Recomendamos una GPU instancia para la mayoría de los fines del aprendizaje profundo. Entrenar nuevos modelos es más rápido en una GPU instancia que en una CPU instancia. Puede escalar de forma sublineal si tiene varias GPU instancias o si utiliza la formación distribuida en muchas instancias con ellas. GPUs

Los siguientes tipos de instancias son compatibles con. DLAMI Para obtener información sobre GPU las opciones de tipos de instancias y seleccione **Computación acelerada**.

> **nota**
>
> El tamaño del modelo debe ser un factor a la hora de elegir una instancia. Si tu modelo supera el de una instancia disponibleRAM, elige un tipo de instancia diferente con suficiente memoria para tu aplicación.

- Las instancias Amazon EC2 P5e tienen hasta 8 NVIDIA Tesla H200. GPUs
- Las instancias Amazon EC2 P5 tienen hasta 8 NVIDIA Tesla GPUs H100.
- Las instancias Amazon EC2 P4 tienen hasta 8 NVIDIA Tesla GPUs A100.
- Las instancias Amazon EC2 P3 tienen hasta 8 NVIDIA Tesla GPUs V100.
- Las instancias Amazon EC2 G3 tienen hasta 4 NVIDIA Tesla GPUs M60.
- Las instancias Amazon EC2 G4 tienen hasta 4 NVIDIA GPUs T4.
- Las instancias Amazon EC2 G5 tienen hasta 8 NVIDIA GPUs A10G.
- Las instancias Amazon EC2 G6 tienen hasta 8 NVIDIA GPUs L4.
- Las instancias Amazon EC2 G6e tienen hasta 8 núcleos Tensor NVIDIA L40S. GPUs
- Las instancias Amazon EC2 G5g tienen procesadores Graviton2 basados en ARM64 AWS .

DLAMIlas instancias proporcionan herramientas para monitorear y optimizar sus procesos. GPU Para obtener más información sobre la supervisión de sus GPU procesos, consulteGPUSupervisión y optimización.

	Instance name	vCPUs	Memory	Network Performance	Storage	On-Demand Hourly Cost ▲	CurrentGeneration	Potential Effective Hourly Cost (Savings %)
◉	g4dn.xlarge	4	16 GiB	Up to 25 Gigabit	125 GB NVMe SSD	0.526	Yes	0.1978 (62%)
○	g4dn.2xlarge	8	32 GiB	Up to 25 Gigabit	225 GB NVMe SSD	0.752	Yes	0.2828 (62%)
○	g4dn.4xlarge	16	64 GiB	Up to 25 Gigabit	225 GB NVMe SSD	1.204	Yes	0.4527 (62%)
○	g4dn.8xlarge	32	128 GiB	50 Gigabit	900 GB NVMe SSD	2.176	Yes	0.8182 (62%)
○	g4dn.12xlarge	48	192 GiB	50 Gigabit	900 GB NVMe SSD	3.912	Yes	1.4709 (62%)
○	g4dn.16xlarge	64	256 GiB	50 Gigabit	900 GB NVMe SSD	4.352	Yes	1.6363 (62%)
○	g4dn.metal	96	384 GiB	100 Gigabit	2 x 900 GB NVMe SSD	7.824	Yes	2.9418 (62%)

Payment options

Estimated commitment price based on the following selections:
Instance type: **g4dn.xlarge** Operating system: **Linux**

Select the container and options to find your best price

○ Compute Savings Plans
One plan that automatically applies to all usage on EC2, Fargate, and Lambda. Up to 66% discount. Learn more
Reservation term
◉ 1 year
○ 3 year
Payment Options
◉ No upfront
○ Partial upfront
○ All upfront
Upfront: 0.00
Monthly: 278.13/Month

○ EC2 Instance Savings Plans
Get deeper discount when you only need one instance family and region. Up to 72% discount. Learn more
Reservation term
○ 1 year
◉ 3 year
Payment Options
◉ No upfront
○ Partial upfront
○ All upfront
Upfront: 0.00
Monthly: 165.71/Month

◉ On-Demand
Maximize flexibility. Learn more
Expected utilization
Enter the expected usage of Amazon EC2 instances
Usage
100
Usage type
Utilization percent per month
Instance: 0.526/Hour
Monthly: 383.98/Month

○ Spot Instances
Minimize cost by leveraging EC2's spare capacity. Recommended for fault tolerant and interruption tolerant applications. Learn more
The historical average discount for g4dn.xlarge is 66%
Assume percentage discount for my estimate
-1
Actual spot instance pricing varies
With spot instances, you pay the spot price that's in effect for the time period your instance is running
Instance: 0.526/Hour
Monthly: 387.82/Month

▶ Other purchasing options

Estas son las opciones de Google:

Modelos generales

Tipo de carga de trabajo	Uso adecuado	Buena alternativa (en orden recomendado)
Entrenamiento de servidores múltiples (distribuidos)	A3	• A2 • G2 • N1+V100
Entrenamiento de servidor único	A3, A2	• G2 • N1+V100
Inferencia	G2	• N1+T4 • N1+V100

Inference and visualization with GPUs Best performance for inference and visualization tasks requiring GPUs.	• CUDA-enabled ML training and inference • Video transcoding	Specialized **G-series** G2
All other GPU tasks Highest performing GPUs.	• Massively parallelized computation • BERT natural language processing • Deep learning recommendation model (DLRM)	Specialized **A-series** A3

Estimación mensual

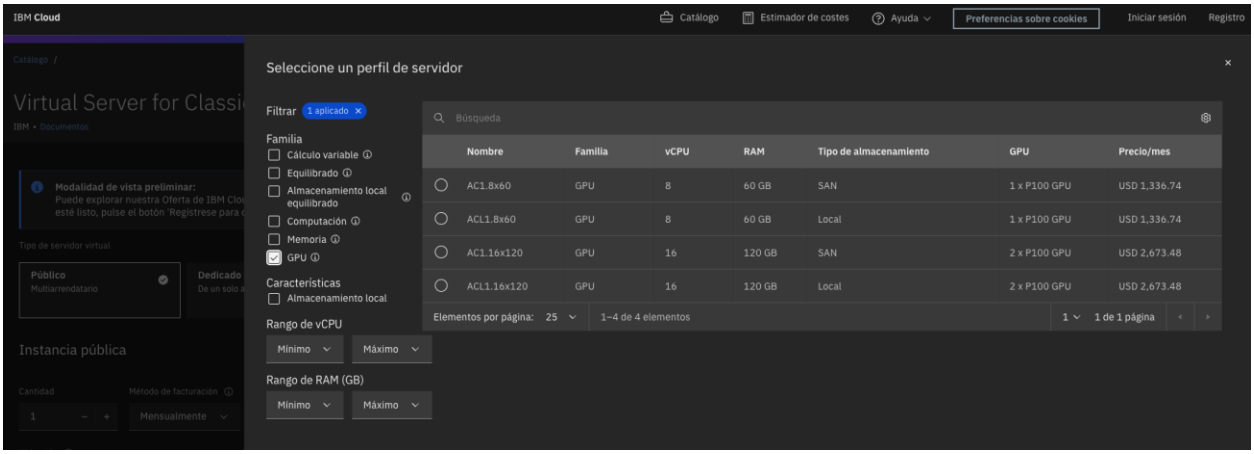
USD204.12

Equivale a alrededor de USD0.28 por hora

Paga por lo que usas, con facturación por segundo y sin pagos por adelantado

Elemento	Estimación mensual
1 vCPU + 3.75 GB memory	USD34.67
1 NVIDIA T4	USD255.50
Disco persistente balanceado de 10 GB	USD1.00
Descuento por uso	-USD87.05
Total	USD204.12

Estas son las de IBM:



Escalabilidad

En todos los casos las máquinas virtuales son escalables tanto vertical como horizontalmente, lo que nos permite replicar la solución de forma importante. Sin embargo, en caso de qué escale de forma importante, sería necesario implementar algún método de monitoreo de la solución completa para tener visibilidad del Estado, que incluso implementar algún método de administración global.

Evaluación y Justificación del Proveedor de Servicios en la Nube

Para el caso de implementar esta solución en la nube, se sugeriría hacerlo en Azure debido a los costos y la escalabilidad. La implementación en la nube nos ahorra la adquisición del hardware local, así como de su mantenimiento e instalación, sin embargo, su costo de $652 al año, es cuatro veces el costo del hardware, y a esto hay que sumarle el costo de la persona especializada en Azure para que realice la implementación. Se tendría que evaluar la cantidad de instancia simultáneas que podría soportar esta máquina virtual, para determinar si es económicamente viable o si tiene un retorno de inversión.

Referencias

Miller, G. (2022, diciembre). Stakeholder roles in artificial intelligence projects. Project Leadership and Society, Volume 3. https://doi.org/10.1016/j.plas.2022.100068

Korolov, M. (2022, septiembre 7). Measuring the business impact of AI. CIO. https://www.cio.com/article/405620/measuring-the-business-impact-of-ai.html

Ultralytics. (2024, 11 septiembre). kalman_filter. Ultralytics YOLO Docs. https://docs.ultralytics.com/reference/trackers/utils/kalman_filter/

Mosesdaudu. (2024, 1 febrero). Object Detection & Tracking With Yolov8 and Sort Algorithm. Medium. https://medium.com/@mosesdaudu001/object-detection-tracking-with-yolov8-and-sort-algorithm-363be8bc0806

Review Estimate - IBM Cloud. (s. f.). https://cloud.ibm.com/estimator

Precios de instancias de VM | Compute Engine: Virtual Machines (VMs) | Google Cloud. (s. f.). Google Cloud. https://cloud.google.com/compute/vm-instance-pricing?hl=es-419

Nuevas instancias G4ad de Amazon EC2 (1:59). (s. f.). [Vídeo]. Amazon Web Services, Inc. https://aws.amazon.com/es/ec2/instance-types/g4/

Azure Linux virtual machines pricing. (s. f.). https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/#n-series