

Diseño de un sistema de reconocimiento de matrículas automotrices usando servidor Raspberry Pi e IA en Amazon Web Services

Ilse K. Leyva-Villanueva¹, Iván U. Aguilar-Pillado¹,
Héctor R. Martínez-Anselmo¹, José J. Rodríguez-Senday²

¹ Universidad Tecnológica de Nogales,
Sonora,
México

² Instituto Tecnológico de Nogales,
Sonora,
México

karim.iklv@gmail.com

Resumen. El presente trabajo muestra cómo es posible que un sistema embebido constituido como un servidor apache, con pocos o casi nulos recursos de procesamiento para aplicaciones de inteligencia artificial pueda ser un medio para comunicarse y enviar información a los servicios en la nube de Amazon Web Services, AWS, y pueda utilizar esas aplicaciones para reconocimiento de matrículas automotrices. Los servicios en la nube que son proporcionados por AWS y Google son similares en cuanto a lo que ofrecen para el área de reconocimiento por visión artificial, pero se desconocen ciertos resultados cuando estos interactúan con sistemas embebidos como Raspberry Pi o cualquier otro. La finalidad es conocer la eficiencia de esos recursos en la nube para desarrollar aplicaciones en hardware que nos ayuden a simplificar algunas tareas de la vida diaria y posteriormente podamos iniciar con el desarrollo de aplicaciones diversas para solucionar algunas problemáticas de nuestro entorno social.

Palabras clave: AWS, Google, inteligencia artificial, sistemas embebidos.

Design of an Automotive License Plate Recognition System Using Raspberry Pi Server and AI in Amazon Web Services

Abstract. This paper shows how it is possible that an embedded system constituted as an Apache server, with little or no processing resources for artificial intelligence applications can be a means to communicate and send information to the Amazon Web Services cloud services, AWS, and can use those applications for license plate recognition. The cloud services provided by AWS and Google are similar in terms of what they offer in the area of machine vision recognition, but certain results are unknown when they interact with embedded systems such as raspberry pi or any other. The purpose is to know the efficiency of these resources in the cloud to develop applications in hardware that help us

to simplify some tasks of daily life and then we can start with the development of diverse applications to solve some of the problems of our social environment.

Keywords: AWS, Google, artificial intelligence, embedded system.

1. Introducción

Con los avances en electrónica y las comunicaciones, por internet han surgido los sistemas embebidos los cuales tienen un desempeño bastante aceptable para procesar información simple y son bastante buenos para comunicarse al internet y poder interactuar con páginas WEB y aplicaciones móviles.

Amazon Web Services, AWS, es un servicio en la nube que nos ofrece distintas características para el tratamiento de datos, siempre y cuando exista un medio que ayude a subirlos a la nube. Uno de los servicios que ofrece es el reconocimiento por visión artificial con algoritmos conceptuales de Machine y Deep learning.

El medio por el cual podemos subir esta información a la nube son los llamados sistemas embebidos, los cuales pueden ser constituidos como servidores apache mediante la instalación de ciertas librerías específicas. La desventaja de este tipo de hardware programable es su baja capacidad de procesamiento, aún y cuando se considera como una computadora personal, no tiene mucha memoria RAM, no tiene procesadores capaces de trabajar con algoritmos pesados en cuanto al cálculo necesario para procesar algoritmos de inteligencia artificial.

El presente trabajo muestra una manera de trabajar estos algoritmos de inteligencia artificial en la nube y los sistemas embebidos mediante el desarrollo de un proceso de investigación para reconocer matrículas automotrices.

2. Trabajos relacionados

Las organizaciones emplean cada vez más la IA para llevar a cabo complejas tareas de Inteligencia Artificial que antes se creía que solo los humanos eran capaces de realizar. En algunos ámbitos limitados del aprendizaje como servicio, la IA supera incluso el rendimiento de los humanos.

Para fomentar la difusión y aplicación de la IA, los proveedores de la nube, como Amazon, Google, IBM, Microsoft o Salesforce, han empezado a ofrecer aprendizaje automático, aprendizaje profundo, analítica e inferencia como servicio, llevando a la práctica el debate sobre el aprovisionamiento de capacidades de IA desde la nube [1]. [2] Explora el desarrollo de la Inteligencia Artificial y su impacto en los modelos de negocio, la organización y el trabajo.

En primer lugar, se ofrece una historia estilizada de la IA en la que se destacan los factores tecnológicos, organizativos y de mercado que fomentan su difusión y su potencial transformador. En segundo lugar, se analizan las consecuencias de la adopción de la IA para los modelos empresariales, la organización y el trabajo.

Este debate contribuye a mostrar cómo el desarrollo y la difusión de este ámbito tecnológico dan nueva fuerza al paradigma de lean production, tanto en el sector manufacturero como en el de servicios.

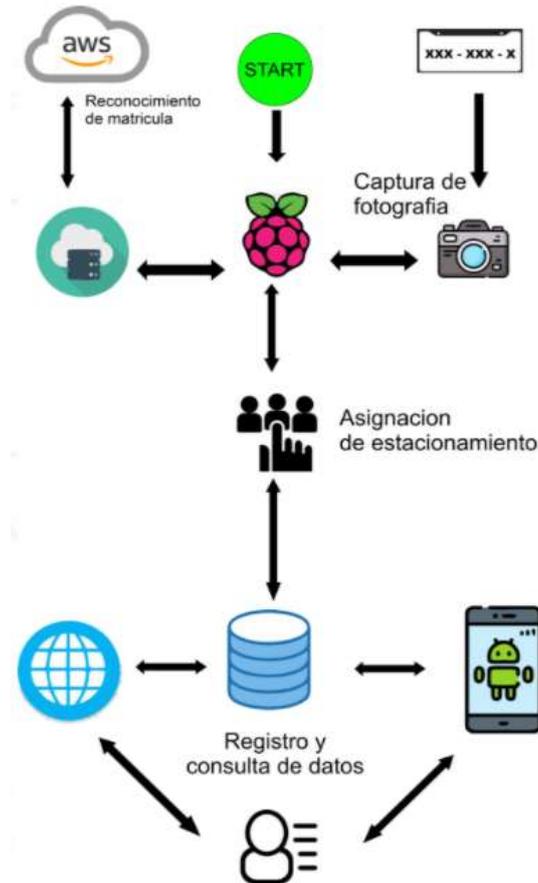


Fig. 1. Diagrama de bloques del proyecto propuesto.

Actualmente, existe un auge creciente sobre tecnologías basadas en microservicios y en el cloud computing, esto debido a su alta escalabilidad, mantenibilidad y facilidad para crear infraestructuras de forma segura. Amazon Web Services (AWS) ofrece diversos servicios que permiten convertir plataformas sencillas en aplicaciones robustas, usando diferentes tecnologías y bases de datos, así como también permite agregar seguridad tanto a las aplicaciones como a los datos que son la fuente principal de todo sistema [3].

En [4] se usó una unidad construida con Raspberry Pi, que se presenta como nodo central del sistema. Para recopilar datos de múltiples nodos sensores de forma rápida y enviar/recibir mensajes desde/hacia la aplicación que se ejecuta en la nube, se utiliza el protocolo ligero de transporte de mensajería publisher/subscriber, MQTT. [5] Presenta una solución que integra un asistente de voz en un escenario de oficina inteligente.

El interés particular era desarrollar el entorno para interactuar con dispositivos específicos de oficinas inteligentes, así como con herramientas de gestión de proyectos, ofreciendo nuevas perspectivas científicas que puedan guiar a los investigadores en futuros trabajos similares.

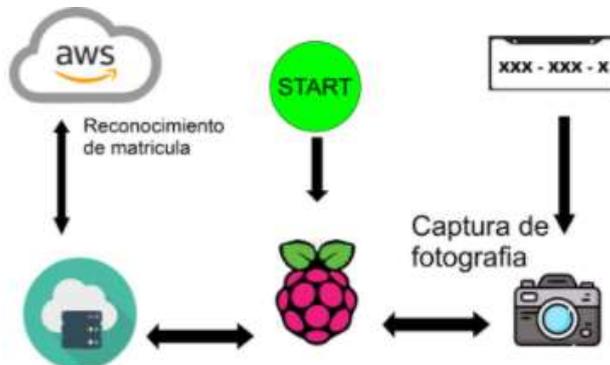


Fig. 2. Estructura de procesamiento con IA propuesta.

3. Sistema propuesto

Se propone un sistema de reconocimiento de matrículas automotrices que usa una Webcam para tomar estas fotografías, un sistema embebido raspberry pi configurado para trabajar como un servidor apache, el uso de los servicios de inteligencia artificial de AWS y todo esto trabajando en el concepto de internet de las cosas para probar la eficiencia que tienen estos algoritmos de inteligencia artificial. La figura 1 muestra el diagrama conceptual del proyecto.

La metodología propuesta para el término de este trabajo de investigación es dividir el trabajo en 2 casos.

La primera parte propuesta es constituir el servidor apache con la raspberri pi, configurada para la toma de fotografías de la matrícula del automóvil, esta puede ser guardada en cualquiera de los formatos conocidos, png, jpeg, etc.

Lo siguiente es a través del desarrollo de ciertos scripts de Python, que trabajan bien en el sistema embebido, subirlas a la plataforma de AWS, una vez estando estos datos en la nube de AWS, poder aplicarle los algoritmos de reconocimiento para determinar la cadena de caracteres de la matrícula, tal y como lo muestra la figura 2.

La segunda parte del proyecto, para proporcionarle un sentido de uso práctico es utilizar la información generada en la parte inteligente del mismo e interactuar mediante consultas a una base de datos construida en la nube de Google, en Firebase. Cuando el sistema inteligente terminó de procesar la información, se envía la cadena de caracteres de nueva cuenta al servidor con raspberry pi, no se sabe aún si es una cadena que corresponde a una matrícula.

Entonces, la raspberri establece una consulta a la base de datos para verificar que el usuario efectivamente se encuentre registrado y si eso resulta correcto, se enviará un aviso al dispositivo móvil del usuario avisándole que está activo en el sistema.

El usuario podrá interactuar también con una APP WEB para verificar su estado en el sistema. La figura 3 nos muestra cómo es esto posible.

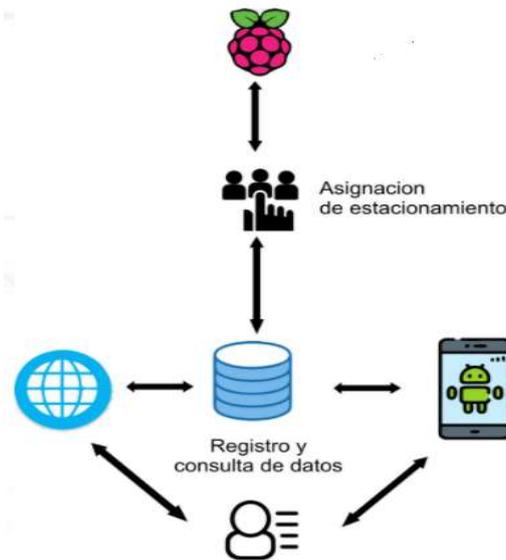


Fig. 3. Estructura del bloque de consulta para validar al usuario.



Fig. 4. Vista frontal de una matrícula.

4. Resultados

Se tomaron diversas fotografías de matrículas automotrices para probar el sistema de reconocimiento, en las cuales se cuidó que tuvieran ciertas características para hacer que el sistema tuviera retos en la identificación y que probablemente pudiera clasificar mal. La figura 4 muestra cómo sería una fotografía perfecta para el sistema, con una excelente nitidez y totalmente de frente.

La figura 5, muestra una fotografía con desviación hacia la izquierda y con menor grado de nitidez con respecto a la figura 4. Esta característica es con la finalidad de que el sistema de reconocimiento tenga un reto en la clasificación de la imagen.

La figura 6, muestra una matrícula escrita totalmente a mano y con un color distinto al que muestran de manera normal las matrículas automotrices, con el fin de agregarle otro reto más al sistema con el color y con los caracteres escritos de manera manual.



Fig. 5. Vista de lado de una matrícula.

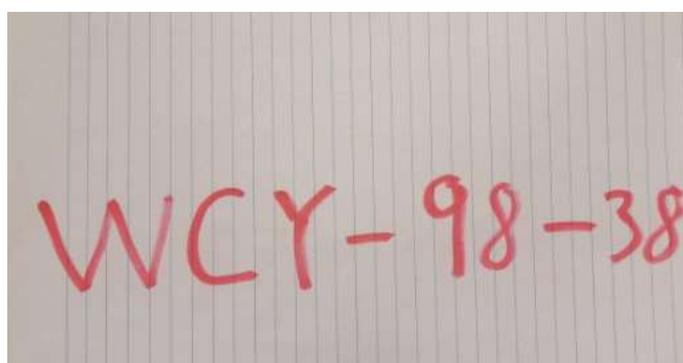


Fig. 6. Matrícula escrita manualmente.

Se tomó un número de 30 fotografías que corresponden a la capacidad que tiene un estacionamiento, tratando de que cada una de las fotografías tuviera alguna de las características mostradas en las figuras 4, 5 y 6. Los resultados se muestran en las matrices de confusión indicadas en la tabla 2. Esta matriz de confusión propuesta es con la finalidad de mostrar las métricas de precisión y exhaustividad que nos servirán para determinar qué tan buena es la propuesta.

Los resultados se muestran la matriz de confusión de la tabla 2, tal como el ejemplo mostrado en la tabla 1.

- **TP:** Foto válida y el sistema la reconoció.
- **TN:** Foto no válida y el sistema no lo reconoció.
- **FP:** Foto no válida y el sistema la reconoce.
- **FN:** Foto válida y el sistema no la reconoce.

La matriz de confusión mostrada en la tabla 2, muestra como las 30 fotografías fueron clasificadas como TP, verdaderos positivos. Esto es, el renglón correspondiente tiene un valor booleano de 1 que corresponde a los que realmente es y la columna también tiene el mismo valor booleano de 1 que corresponde a lo que el sistema predice. Este tipo de matriz se utilizó para medir el rendimiento del sistema propuesto.

Tabla 1. Matriz de confusión propuesta.

		Predicción	
		0	1
Realidad	0	TN	FP
	1	FN	TP

Tabla 2. Matriz de confusión de resultados.

		Predicción	
		0	1
Realidad	0	0	0
	1	0	30

De los datos proporcionados por la matriz de confusión se calculó el parámetro de precisión, para determinar qué tan cerca estaban los resultados predichos de lo que realmente debería de ser:

$$Precisión = \frac{TP}{TP + TN} = 1. \quad (1)$$

Como podemos observar en el resultado, que fue de 1, lo podemos interpretar como excelente, ya que indica que cada una de las fotografías tomadas las clasificó de forma correcta:

$$Precisión = \frac{30}{30 + 0} = 1.$$

El siguiente parámetro calculado fue la exhaustividad para determinar que tan bien clasificaba el sistema:

$$Exhaustividad = \frac{TP}{TP + FN} = 1. \quad (2)$$

Como resultado de esto, el sistema indica que el hecho de identificar de manera correcta las 30 fotografías era lo indicado:

$$Exhaustividad = \frac{30}{30 + 0} = 1.$$

Adicionalmente, se obtuvieron Apps Web y Móviles que ayudan al usuario a utilizar de una manera más simple el sistema como es en el registro, en el acceso al sistema y sobre todo la App Web que sirve para que el administrador del sistema tenga datos confiables para la toma de decisiones, en las figuras 7, 8 y 9 se muestran estas interfaces.

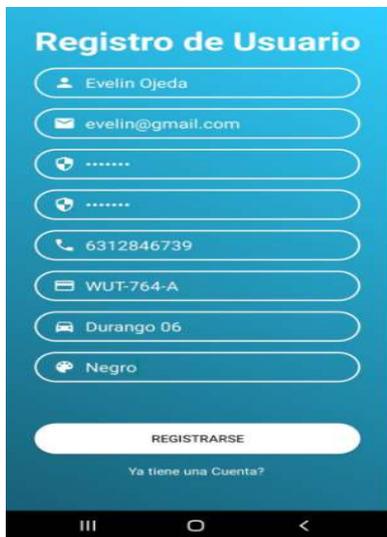


Fig. 7. App Móvil.

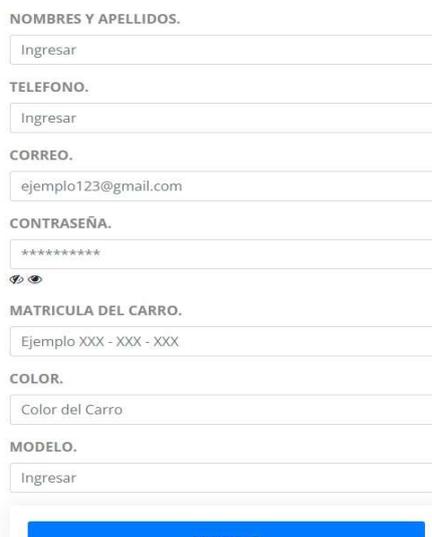


Fig. 8. Alta de usuarios en la App Web.

ID_Estacionamiento	Nombre del Usuario	Matrícula del Carro	Fecha y Hora (Entrada)	Fecha y Hora (Salida)
2	Dayan Leiliany Lagarda Roldan	DAY-16-17	27/Marzo/2023 7:58 a.m.	27/Marzo/2023 12:22 p.m.
1	Hector Raúl Martínez Anselmo	HEC-19-20	27/Marzo/2023 7:44 a.m.	27/Marzo/2023 1:20 p.m.
0	Ivan Ulises Aguilar Pillado	WCY-98-38	27/Marzo/2023 8:02 a.m.	27/Marzo/2023 1:25 p.m.

Fig. 9. Información para el administrado.

5. Conclusiones y trabajo a futuro

De acuerdo a lo que observamos en la matriz de confusión, las métricas indican que las herramientas de AWS en la nube con IA son excelentes inclusive, aún y cuando las fotografías de las matrículas tengan características que en su momento pueden hacer que el sistema identifique de manera errónea.

También probamos que es posible construir un servidor con el sistema embebido usado, raspberry pi, para preprocesar la fotografía de acuerdo a las instancias que solicita la plataforma de IA de AWS y llevar a cabo el trabajo de identificación.

El trabajo muestra también, como es posible que con las herramientas del internet de las cosas, IoT, se puede interconectar una App Web, App móvil, una base de datos de proveedores distintos a AWS, sistemas embebidos de tal manera que se construya un proyecto completo de gestión de información para el usuario.

Trabajos futuros implican darle aplicaciones al sistema desarrollado para ofertar diferentes servicios a los usuarios de las alternativas que ofrecen las empresas privadas como las instituciones públicas.

Referencias

1. Lins, S., Pandl, K. D., Teigeler, H., Thiebes, S., Bayer, C., Sunyaev, A.: Artificial intelligence as a service. *Business and Information Systems Engineering*, vol. 63, pp. 441–456 (2021) doi: 10.1007/s12599-021-00708-w
2. Fantl, L., Guarascio, D., Moggi, M.: From Heron of Alexandria to Amazon’s Alexa: A stylized history of AI and its impact on business models, organization and work. *Journal of Industrial and Business Economics*, vol. 49, pp. 409–440 (2022) doi: 10.1007/s40812-022-00222-4
3. Cárdenas-Sánchez, B. C., Olarte-Rojas, C. A.: Análisis de seguridad entre microservicios con Amazon web service. *Revista Logos Ciencia and Tecnología*, vol. 14, no. 2, pp. 42–52 (2022) doi: 10.22335/rlct.v14i2.1546
4. Khriji, S., Benbelgacem, Y., Chéour, R., Houssainil, D., Kanoun, O.: Design and implementation of a cloud-based event-driven architecture for real-time data processing in wireless sensor networks. *The Journal of Supercomputing*, vol. 78, pp. 3374–3401 (2022) doi: 10.1007/s11227-021-03955-6
5. Bogdan, R., Tatu, A., Crisan-Vida, M. M., Popa, M., Stoicu-Tivadar, L.: A practical experience on the Amazon Alexa integration in smart offices. *Sensors*, vol. 21, no. 734 (2021) doi: 10.3390/s21030734
6. Swaroop, T.: Create your own face recognition service with AWS Rekognition! (2022) <https://youtu.be/oHSesteFK5c>
7. Zaamout, E.: Laravel tutorial - Deploy any laravel app in AWS. AHT Cloud (2021) <https://youtu.be/W2fQFbkEQo0>
8. Aprendible: Aprende Laravel en 3 horas (2022) <https://youtu.be/rQZmhqah0PQ>
9. AWS: AWS SDK for PHP 3.x (2023) <https://docs.aws.amazon.com/aws-sdk-php/v3/api/namespace-Aws.html>
10. Microsoft: Overview of ASP.NET Core MVC (2022) <https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-7.0>