

Reconocimiento facial usando herramientas de IA de Amazon Web Services y sistemas embebidos

Eduardo Saavedra Quijada², Luis A Medina Muñoz¹,
Felipe Morales Solís¹, Gabriel López Valencia²

¹ Instituto Tecnológico de Nogales,
Sonora,
México

² Universidad Tecnológica de Nogales,
Sonora,
México

eduardosaavedraq687@gmail.com

Resumen. Los diversos sistemas embebidos que actualmente están disponibles para los usuarios, aun y cuando tienen las características de una computadora personal, carecen de la capacidad de procesamiento necesaria para ejecutar trabajos de reconocimiento facial, a menos que, se le agreguen dispositivos coprocesadores como los neural computer stick de Intel o Coral USB accelerator de google. Amazon Web Services y recientemente google, tienen herramientas de inteligencia artificial que ayudan a acelerar la implementación de un proyecto de reconocimiento facial incorporadas en sus espacios virtuales llamados nube. El presente trabajo, muestra cómo es posible implementar acciones de reconocimiento facial en un sistema embebido, el cual es raspberry pi, cuya función es servir como un hardware auxiliar para subir fotografías grupales a la nube de Amazon Web Services y que con las herramientas de IA alojadas en ese medio se lleve a cabo el procesamiento pesado de reconocimiento facial y devuelva hacia al sistema embebido la información de cada uno de las personas presentes en la fotografía.

Palabras clave: Amazon web, google, reconocimiento facial, sistemas embebidos.

Facial Recognition Using AI Tools from Amazon Web Services and Embedded Systems

Abstract. The various embedded systems that are currently available to users, even though they have the features of a personal computer, lack the processing power needed to run facial recognition jobs, unless co-processor devices such as Intel's neural compute stick or Google's Coral USB accelerator are added. Amazon Web Services and recently Google have artificial intelligence tools that help accelerate the implementation of a facial recognition project embedded in their virtual spaces called cloud. This paper shows how it is possible to implement facial recognition actions in an embedded system, which is raspberry pi, whose

function is to serve as an auxiliary hardware to upload group photographs to the Amazon Web Services cloud and that with the AI tools hosted in that environment the heavy processing of facial recognition is carried out and returns to the embedded system the information of each of the people present in the photograph.

Keywords: Amazon web, google, facial recognition, embedded system.

1. Introducción

Los sistemas embebidos son herramientas de cómputo donde todos sus componentes están integrados en una sola placa, reduciendo con esto el tamaño, pero con características similares a una computadora personal, con una menor capacidad de procesamiento, sobre todo para ejecutar aplicaciones donde se utilicen herramientas de inteligencia artificial.

Estas debilidades del sistema embebido pueden ser cubiertas con el uso de herramientas de inteligencia artificial quizás, de los 2 proveedores más importantes como son Amazon Web Services y Google. Amazon Web Services provee de servicios en la nube a usuarios registrados para acelerar sus procesos de diseño basados en inteligencia artificial.

Estos procesos ayudan a los sistemas embebidos a formar parte de un proyecto de reconocimiento facial sin utilizar sus características de cómputo, únicamente convirtiéndose en un servidor de información hacia la nube, lo cual con sus propiedades de conectividad es relativamente sencillo.

Este trabajo propone utilizar Raspberri Pi como sistema embebido, el cual trabajará como medio para subir fotografías individuales o grupales a la nube de Amazon Web Services en la cual se llevará a cabo el procesamiento de la imagen con servicios de inteligencia artificial y regresará al servidor implementado con Raspberri Pi la información del o los individuos que están en esa fotografía.

2. Trabajos relacionados

En [1] se implementó en Python un prototipo de un sistema de reconocimiento facial usando machine learning, para lograrlo se usaron algunos temas relacionados con la inteligencia artificial y el impacto que ha tenido el desarrollo de estas tecnologías a nivel mundial, siendo implementadas en muchos campos como la medicina, la seguridad en los aeropuertos entre otras aplicaciones que han mejorado la calidad de vida de las personas.

La investigación “Aplicación de inteligencia artificial para monitorear el uso de mascarillas de protección” creo una aplicación web que permite monitorear el uso de mascarillas protectoras en ambientes públicos. Utilizando el framework Flask, en el lenguaje de Python, la aplicación cuenta con un panel de control que ayuda a visualizar los datos obtenidos. El proceso de detección utiliza el algoritmo Haar Cascade para clasificar rostros con y sin mascarillas protectoras.

Como resultado, la aplicación web es liviana y permite detectar y almacenar en la nube las imágenes capturadas y la posibilidad de un mayor análisis de datos.

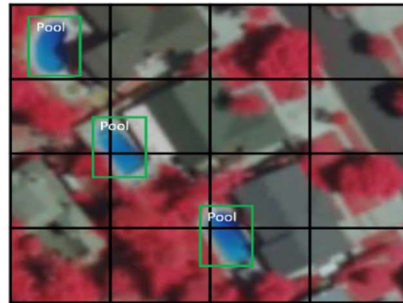


Fig. 1. Single shoot detector.

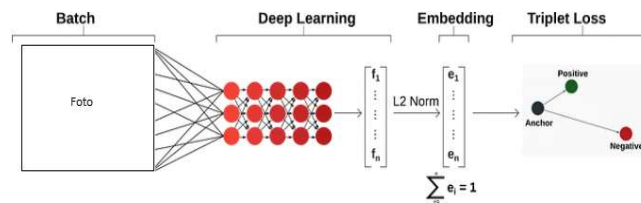


Fig. 2. Arquitectura de FaceNet.

El clasificador presenta precisión, revocación y f-score de 63%, 93% y 75%, respectivamente. Aunque la precisión fue satisfactoria, se realizarán nuevos experimentos para explorar nuevas técnicas de visión por computadora, como el uso de aprendizaje profundo [2].

En [4] muestran cómo implementar un servicio basado en la nube que proporciona los algoritmos faciales más avanzados de reconocimiento y detección de rostros con atributos bajo la plataforma Microsoft Azure.

Con lo que respecta a la implementación de estos servicios se llevaron a cabo experimentos diferenciados de cada una de las fases del desarrollo del proyecto, de modo que se puedan evaluar las fortalezas y debilidades del servicio en la nube. Los análisis de las imágenes procesadas se han centrado en observar el potencial de exactitud, la eficiencia y rapidez del servicio.

En [5] se presenta un método de reconocimiento de imágenes mediante el aprendizaje de diccionarios específicos de clase para separar las salidas de la red neuronal en características dependientes de la clase, potenciando así su capacidad de discriminación. Específicamente, se desarrolló una red de atención de clase (CANet) mediante la integración de un módulo de codificación de atención de clase específica (CAE) simple pero eficaz en la parte superior de las capas convolucionales.

En [6], se presenta una revisión de los avances recientes con sugerencias sobre las nuevas direcciones posibles para mejorar la eficiencia de los enfoques de reconocimiento visual relacionados con las DNN e inspirados en el cerebro, incluida la compresión eficiente de redes y las redes dinámicas inspiradas en el cerebro. Se investiga no sólo desde el punto de vista del modelo, sino también de los datos (lo que no ocurre en los estudios existentes) y se centra en cuatro tipos de datos típicos (imágenes, vídeo, puntos y eventos).



Fig. 3.: Modo de prueba del sistema propuesto.

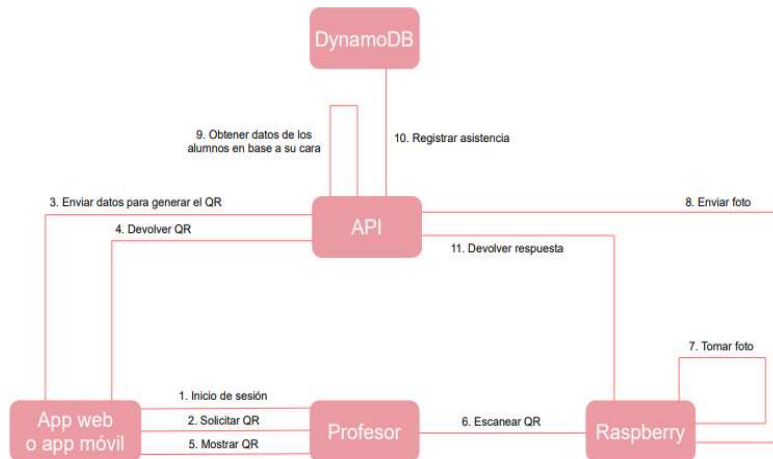


Fig. 4. Modelado gráfico de la aplicación.

Este estudio pretende ofrecer un resumen sistemático exhaustivo que pueda servir de valiosa referencia e inspirar tanto a investigadores como a profesionales que trabajen en problemas de reconocimiento visual.

3. Desarrollo

3.1. Single Shot Detector

El SSD tiene dos componentes: un modelo troncal y un principal. El modelo troncal suele ser una red de clasificación de imágenes preentrenadas como extractor de características. Normalmente se trata de una red como ResNet entrenada en ImageNet de la que se ha eliminado la última capa de clasificación totalmente conectada.

De este modo, obtenemos una red neuronal profunda capaz de extraer el significado semántico de la imagen de entrada, conservando al mismo tiempo la estructura espacial de la imagen, aunque a una resolución inferior.

Tabla 1. Porcentajes de similitud de Amazon Recognition.

| No. de foto | % de similitud |
|-------------|----------------|
| Foto 1 | 99.999 |
| Foto 2 | 99.997 |
| Foto 3 | 99.999 |
| Foto 4 | 99.983 |
| Foto 5 | 99.981 |

Para ResNet34, la columna vertebral da como resultado 256 mapas de características de 7×7 para una imagen de entrada. El SSD principal no es más que una o varias capas convolucionales añadidas a esta columna vertebral y los resultados se interpretan como los cuadros delimitadores y las clases de objetos en la ubicación espacial de las activaciones de las capas finales.

En lugar de utilizar ventanas deslizantes, SSD divide la imagen utilizando una cuadrícula y hace que cada celda de la cuadrícula sea responsable de detectar objetos en esa región de la imagen. Detectar objetos significa simplemente predecir la clase y la ubicación de un objeto dentro de esa región.

Si no hay ningún objeto presente, lo consideramos como la clase de fondo y se ignora la ubicación. Por ejemplo, podríamos utilizar una cuadrícula de 4×4 como muestra la figura 1. Cada celda de la cuadrícula es capaz de emitir la posición y la forma del objeto que contiene [11]

3.2. FaceNet

FaceNet es un sistema de reconocimiento facial que aprende a asignar caras a una posición en un espacio multidimensional en el que la distancia entre puntos corresponde directamente a una medida de similitud facial.

1. Preprocesamiento: Un método utilizado para tomar un conjunto de imágenes y convertirlas todas a un formato uniforme, en este caso, una imagen cuadrada que contiene sólo la cara de una persona. Un conjunto de datos uniforme es útil para reducir la varianza durante el entrenamiento, ya que se disponen de recursos informáticos limitados al utilizar la TPU Edge.
2. Embedding: Un proceso fundamental para el funcionamiento de FaceNet, que aprende representaciones de caras en un espacio multidimensional donde la distancia corresponde a una medida de similitud de caras.
3. Clasificación: Paso final en el que se utiliza la información proporcionada por el proceso de incrustación para separar los distintos rostros [10].

3.3. Sistema Propuesto

El sistema propuesto se probó en un grupo de una institución educativa, identificando a los usuarios por clase asignada en diferentes horas y llevando un control



Fig. 5. Encuadre de caras para reconocimiento.

de lista de manera automatizada, en este caso solo se adjuntarán resultados de lo eficiente que es el sistema de reconocimiento facial.

El diagrama de bloque muestra que se puede usar una APP móvil o una APP web, el usuario es libre de elegir, de acuerdo a los recursos con los que cuenta, cuál de las 2 APPs usará. Inicialmente se solicitará generar un código QR que contenga la información que identifica a la materia que imparte, cuando este código es generado se le regresará al usuario a la APP para posteriormente ser usada en la siguiente etapa.

Un sistema embebido, raspberry pi, mediante el uso de una cámara WEB leerá el código QR y en un intervalo de tiempo de 5 segundos tomará una foto de manera grupal y subirá esta información a los servicios de Amazon para que inicie con el reconocimiento facial de las personas que se encuentren en dicha foto.

Las coincidencias de los rostros se basan en su geometría visual, incluida la relación entre los ojos, la nariz, las cejas, la boca y otras características faciales. Cuando Amazon Rekognition analiza imágenes, existe una línea alrededor del rostro, un recuadro, que determina la única parte de la imagen que Rekognition considera en su análisis.

A continuación, el análisis produce números de notación del objeto para la imagen que indican la “ubicación” de los elementos principales del rostro. Cuando los clientes ejecutan la búsqueda de un rostro, la tecnología compara los datos de la imagen fuente con cada una de las imágenes en las que busca. A partir de ahí, el servicio asigna a cada rostro de la imagen una puntuación de similitud. Este enfoque garantiza que Amazon Rekognition no tenga información sobre la identidad de una persona, sino solo de la probabilidad de que un rostro coincida con otro [9].

La tabla 1, indica la información del porcentaje de similitud de 5 fotos, tomadas aleatoriamente del conjunto de datos, para demostrar la forma como AWS indica el

Tabla 2. Matriz de confusión propuesta.

| | | Predicción | |
|----------|---|------------|----|
| | | 0 | 1 |
| Realidad | 0 | TN | FP |
| | 1 | FN | TP |

Tabla 3. Datos estadísticos tomando la fotografía de frente 126 fotos.

| | | Predicción | |
|----------|---|------------|-----|
| | | 0 | 1 |
| Realidad | 0 | 0 | 0 |
| | 1 | 2 | 124 |

reconocimiento y como en esta información nos podemos dar cuenta que la identidad de la persona está reservada. Una vez finalizado el proceso de reconocimiento se hacen accesos a la base de datos para ver si es una persona existente o no y determinar la identificación de la misma con la información personal.

4. Resultados

Se construyó una matriz de confusión como la que se muestra en la tabla 3, la cual se llenó con la información obtenida del reconocimiento para calcular las métricas del sistema, que fueron 14 fotos grupales con 9 integrantes cada una, donde:

- **TP:** Está en la foto y el sistema lo reconoció.
- **TN:** No está en la foto y el sistema no lo reconoció.
- **FP:** No está en la foto y el sistema lo reconoce.
- **FN:** Está en la foto y el sistema no lo reconoce.

$$\text{Precisión} = \frac{124}{124+0} = 1, \quad (1)$$

$$\text{Recall} = \frac{124}{124+2} = 0.98, \quad (2)$$

$$F_1 = 2 \left(\frac{(0.98)(1)}{0.98+1} \right) = 0.98, \quad (3)$$

$$\text{Precisión} = \frac{101}{101+0} = 1, \quad (4)$$

$$\text{Recall} = \frac{101}{101+25} = 0.80, \quad (5)$$

$$F_1 = 2 \left(\frac{(0.80)(1)}{0.80+1} \right) = 0.88. \quad (6)$$

Tabla 4. Datos estadísticos tomando la fotografía agregando un cubrebocas a 2 personas.

| | | Predicción | |
|----------|---|------------|-----|
| | | 0 | 1 |
| Realidad | 0 | 0 | 0 |
| | 1 | 25 | 101 |

5. Conclusiones y trabajo a futuro

De acuerdo a los resultados obtenidos, vemos primeramente que el posible implementar diversas aplicaciones con el concepto de internet de las cosas, es decir, comunicar APPs de servicios WEB y dispositivos móviles y los diferentes servicios en la nube como pueden ser de Google o AWS, en el caso de este trabajo fue con AWS. Todo esto no sería posible en el concepto en el cual se realizó el trabajo sin un sistema embebido como Raspberry Pi.

También, este trabajo muestra que es posible que los recursos de procesamiento para aplicaciones que ocupen inteligencia artificial, como lo es el reconocimiento de rostros, y de los cuales los sistemas embebidos no cuentan con la suficiente capacidad para ello, los recursos de procesamiento que se encuentran en el caso nuestro en AWS son suficientes y pueden ser utilizados en aplicaciones específicas.

Las métricas porcentuales indicadas por AWS en promedio, si las analizamos, nos dicen el sistema de reconocimiento es bastante bueno con alrededor de menos del 1% de error, datos que son comprobados al implementar las matrices de confusión y que arrojan una precisión en el reconocimiento facial correcto del 100%. Trabajos futuros implica darle aplicaciones educativas como toma de lista, tutorías y otros servicios relacionados con el alumno.

Referencias

1. Baresi, L., Colazzo, S., Mainetti, L., Morasca, S.: W2000: A modelling notation for complex web applications. In: Web Engineering, pp. 335–364 doi: 10.1007/3-540-28218-1_11
2. Koch, N., Knapp, A., Zhang, G., Baumeister, H.: UML-based web engineering. In: Web Engineering: Modelling and Implementing Web Applications, pp. 157–191 (2008) doi: 10.1007/978-1-84628-923-1_7
3. Sánchez-Santamaria, M., García-García, L. A.: La ingeniería web: Desarrollo de aplicaciones web de alta calidad. Revista de Divulgación Científico-Tecnológica del Gobierno del Estado de Morelos (2011)
4. Martín-Vera, P., Pons, C., Gonzáles, C., Giulianelli, D. A., Rodríguez, R. A.: Metodología de modelado de aplicaciones web móviles basada en componentes de interfaz de usuario. In: Argentine Symposium on Software Engineering (2013)
5. Cheng, G., Lai, P., Gao, D., Han, J.: Class attention network for image recognition. Science China Information Sciences, vol. 66, no. 3 (2023) doi: 10.1007/s11432-021-3493-7
6. Wu, Y., Wang, D., Lu, X., Yang, F., Yao, M., Dong, W., Shi, J., Li, G.: Efficient visual recognition: A survey on recent advances and brain-inspired methodologies. Machine Intelligence Research, vol. 19, no. 5, pp. 366–411 (2022) doi: 10.1007/s11633-022-1340-5

7. Moroni, N., Señas, P.: Uso de grafos para el modelado de experiencias educativas colaborativas basadas en la web. In: VI Workshop de Tecnología Informática Aplicada en Educación, pp. 1134–1146 (2007)
8. Bouchrika, I., Ait-Oubelli, L., Rabir, A., Harrathi, N.: Mockup-based navigational diagram for the development of interactive web applications. In: Proceedings of the 2013 International Conference on Information Systems and Design of Communication, pp. 27–32 (2013) doi: 10.1145/2503859.2503864
9. AWS: Datos sobre el reconocimiento facial mediante inteligencia artificial. Amazon Web Services (2023) aws.amazon.com/es/rekognition/the-facts-on-facial-recognition-with-artificial-intelligence/
10. ArcGIS developers: How single-shot detector (SSD) works? ArcGIS API for Python (2023) developers.arcgis.com/python/guide/how-ssd-works/
11. Madio, P.: A facenet-style approach to facial recognition on the google coral development board. Towards Data Science (2019)
12. Lipschutz, S.: Theory and problems of set theory and related topics. McGraw-Hill, pp. 5–6 (1998)
13. Lipschutz, S., Lipson, M.: Discrete Mathematics. McGraw-Hill, pp. 1–22 (2007)
14. Swaroop, T.: Create your own face recognition service with AWS Rekognition! (2022)
15. AHT cloud: Laravel tutorial - Deploy any Laravel app in AWS (2021) youtu.be/W2fQFbkEQo0
16. Aprendible: Aprende Laravel en 3 horas (2022) youtu.be/rQZmhqah0PQ
17. AWS: AWS SDK for PHP 3.x (2023) docs.aws.amazon.com/aws-sdk-php/v3/api/name-space-Aws.html
18. Microsoft learn: Overview of ASP.NET Core MVC, ASP.NET (2023) learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-7.0