

SISTEMA INTELIGENTE DE AUTOMATIZACIÓN DE PARQUEOS CON RECONOCIMIENTO DE PLACAS

CASO DE ESTUDIO: PARQUEOS ADMINISTRADOS POR EL GAMLP



Ing. Esther Marisol Alejo Gómez
Carrera de Ingeniería de Sistemas, Escuela Militar de Ingeniera
La Paz, Bolivia
ealejog@est.emi.edu.bo



INTELLIGENT PARKING AUTOMATION SYSTEM WITH PLATE RECOGNITION CASE STUDY: PARKING LOTS MANAGED BY GAMLP

Resumen - El presente proyecto trabaja con las redes neuronales convolucionales que es un campo que está siendo ampliamente explotado dentro de la inteligencia artificial. El sistema desarrollado es un sistema de transaccional de información que aloja una red neuronal convolucional, el cual fue entrenado para poder realizar la detección de las placas de los vehículos. Este está desarrollado para la gestión de parqueos, siendo de ayuda para los operadores que trabajan directamente en el parqueo y también para los administradores, dando la información de manera sistematizada.

Palabras Claves— Inteligencia artificial, gestión de parqueos, redes neuronales convolucionales.

Abstract - This project works with convolutional neural networks, which is a field that is being widely exploited within artificial intelligence. The developed system is an information transactional system that hosts a convolutional neural network, which was trained to be able to detect vehicle license plates. This is developed for parking management, being helpful for operators who work directly in the parking lot and also for administrators, providing information in a systematized way.

Keywords— Artificial intelligence, convolutional neural networks, parking management.

I. INTRODUCCIÓN

El crecimiento de la tecnología ha ido a pasos agigantados a nivel mundial, afectando drásticamente en la forma de transportarse, aprender, entre otras actividades. Uno de los grandes avances de la tecnología en los últimos años ha sido la Inteligencia Artificial que intenta emular el pensamiento lógico racional de los humanos.

Actualmente, la inteligencia artificial es usada en un amplio número de campos, está presente en las aplicaciones para teléfonos inteligentes, redes sociales, automóviles inteligentes, publicidad y es parte de la vida cotidiana de los seres humanos. La aplicación de la inteligencia artificial para crear las ciudades inteligentes ha supuesto un antes y después, se ha visto mejoras en la gestión automática y eficiente de las infraestructuras urbanas, entre otros.

Se ha visto un incremento del parque automotor a nivel mundial y Bolivia no es la excepción, donde solo en la ciudad de La Paz se tiene grandes congestionamientos vehiculares, una de las razones es porque se tiene vehículos estacionados en ambos lados de las vías y también por vehículos que hacen fila para ingresar a un parqueo.

En el presente proyecto de grado se busca utilizar los avances tecnológicos que ofrece la inteligencia artificial, para desarrollar un sistema que coadyuve a los parqueos del Gobierno Autónomo Municipal de La Paz para brindar un mejor servicio a la ciudadanía.

II. MATERIALES Y MÉTODOS

La metodología usada para el desarrollo del proyecto es: SCRUM y Desarrollo de la Red Neuronal, en donde cada una tienen fases, tareas y entregables.

III. ANÁLISIS DE LA SITUACIÓN ACTUAL

Se realizó la observación participante, no participante y entrevista al personal a cargo, identificando los procesos actuales que se realizan en la atención del parqueo y así también los involucrados que participan en el mismo. En la siguiente figura se puede ver los procesos dentro del parqueo.



Fig 1. Procesos Elaboración propia

El parqueo Mercado Lanza cuenta con 65 espacios para vehículos, El tamaño de los espacios para vehículos es de 6x3 metros, no teniendo variación de tamaños entre estos.

IV. DISEÑO DEL SISTEMA

Para iniciar el diseño del sistema se realizó la determinación de los casos de uso para el entendimiento del sistema teniendo su descripción correspondiente cada uno.

El sistema de detección de placas comprende de 4 módulos, que son la gestión de usuarios, gestión de ingresos y salidas (módulo de la red neuronal), gestión de clientes y gestión de reportes, como se puede ver en la siguiente figura.

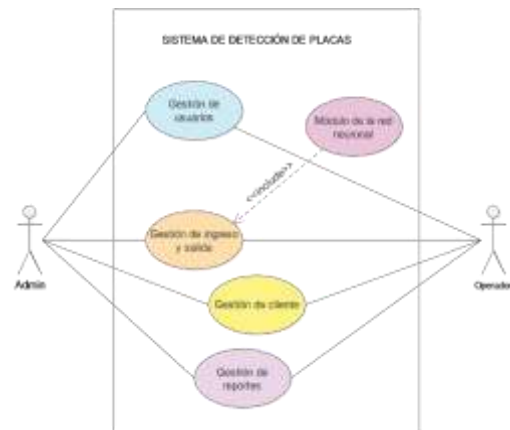


Fig. 2. Caso de uso de alto nivel. Elaboración propia

La arquitectura escogida para el sistema es de 3 capas: Capa de presentación: Es lo que verá el usuario, a esta capa accederá el administrador y el encargado. Capa de negocio: Esta capa recibirá las peticiones de los distintos usuarios, en el caso que el administrador ingrese para ver los reportes del mes, el sistema recibirá la petición y le enviará los datos tras el proceso de conexión con la capa de datos. Capa de datos: Se tiene en esta capa almacenado los datos, es decir la base de datos del sistema. Se alberga los datos sobre los usuarios, clientes, ingresos y salidas de los vehículos. En la siguiente figura se puede comprender mejor la arquitectura



Fig. 3 Arquitectura del sistema. Elaboración propia

El diagrama de clases presenta los datos relacionados según a las tablas con sus respectivos atributos, como se puede en la siguiente figura.

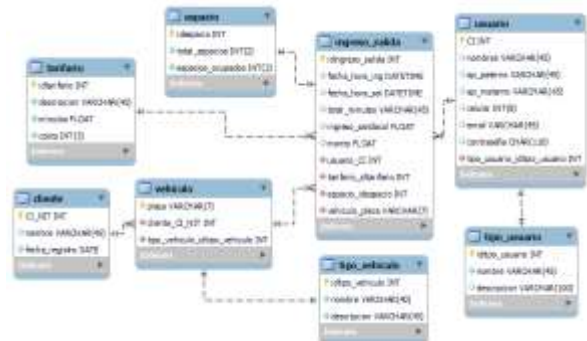


Fig. 4 Diagrama de clases. Elaboración propia

La arquitectura de la red neuronal convolucional que se presenta está en base al modelo de red convolucional Darknet para la clasificación y para la parte de detección se utiliza el modelo de la red convolucional YOLO que es el acrónimo de “You Only Look Once”, esta detección devuelve las posiciones de x e y, alto y ancho de la placa encontrada. En la arquitectura presentada se tiene 24 capas convolucionales seguida de dos capas completamente conectadas. La alternancia de capas convolucionales 1x1 reduce el espacio de características de las capas anteriores (Ver Figura 4).

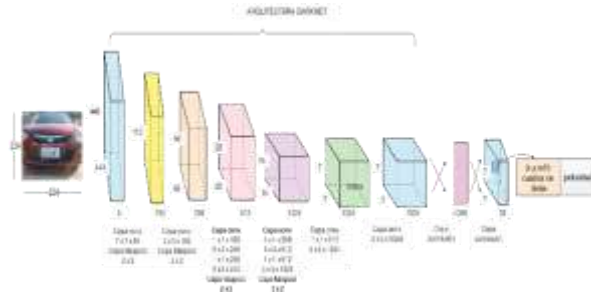


Fig. 5 Arquitectura Red Neuronal. Elaboración propia

Los parámetros que se manejan para esta red neuronal son los siguientes: Tamaño de imagen que procesa la red: El tamaño será fijo (224x224), todas las imágenes usadas son redimensionadas antes de entrar en la red. Cantidad de cajas por imagen: Estas son la cantidad de objetos máximos que se quiere detectar. Etiquetas: Estas son de los objetos que se quiere detectar. En este caso se detectará solo uno tipo de objeto (placa).

Epochs: Es la cantidad de iteraciones que realiza la red neuronal para entrenar.

V. DESARROLLO DEL SISTEMA

Para el proceso de entrenamiento se ajusta los ciclos llamados epoch (épocas), estas son las iteraciones que realiza la red neuronal convolucional, se estableció para la primera etapa 100 (Ver Figura 5).

```

In [11]: history = model.fit_generator(train_generator, validation_data=(val_gen, val_gen),
Epoch 1/100
10/100 [=====] - 10s 10s/step - loss: 0.1867 - val_loss: 0.0908
Epoch 2/100
10/100 [=====] - 10s 10s/step - loss: 0.0911 - val_loss: 0.0718
Epoch 3/100
10/100 [=====] - 10s 10s/step - loss: 0.0901 - val_loss: 0.0542
Epoch 4/100
10/100 [=====] - 10s 10s/step - loss: 0.0612 - val_loss: 0.0518
Epoch 5/100
10/100 [=====] - 10s 10s/step - loss: 0.0503 - val_loss: 0.0446
Epoch 6/100
10/100 [=====] - 10s 10s/step - loss: 0.0408 - val_loss: 0.0374
Epoch 7/100
10/100 [=====] - 10s 10s/step - loss: 0.0447 - val_loss: 0.0309
Epoch 8/100

```

Fig. 6. Entrenamiento. Elaboración propia

En el proceso del entrenamiento la red neuronal al principio llega a tener un mayor porcentaje de errores, a medida que pasa las épocas el modelo tiende a presentar menos errores.

A continuación, se muestra los errores obtenidos de la fase de entrenamiento y la fase de validación, dentro del proceso de aprendizaje con 200 épocas en total para que la red neuronal detecte los patrones característicos de las placas (Ver Figura 6).

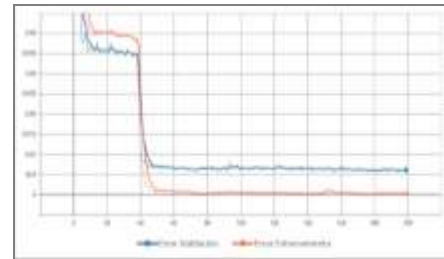


Fig. 7 Errores de entrenamiento y validación. Elaboración propia

Se desarrolló los módulos de control, usuarios, clientes y reportes siguiendo los sprints planificados. En la siguiente figura se puede ver la interfaz de inicio de sesión.



Fig. 8 Iniciar sesión. Elaboración propia

En la siguiente imagen se muestra los ingresos y salidas registrados a través de la red neuronal convolucional.

#	Placa	Fecha y Hora Ingreso	Fecha y Hora Salida	Total Cobro	Estado	Tipo	Observación
1	12345678	2020-01-01 10:00:00	2020-01-01 11:00:00	10.0	OK	Normal	
2	87654321	2020-01-01 12:00:00	2020-01-01 13:00:00	10.0	OK	Normal	
3	23456789	2020-01-01 14:00:00	2020-01-01 15:00:00	10.0	OK	Normal	
4	98765432	2020-01-01 16:00:00	2020-01-01 17:00:00	10.0	OK	Normal	
5	56789012	2020-01-01 18:00:00	2020-01-01 19:00:00	10.0	OK	Normal	
6	34567890	2020-01-01 20:00:00	2020-01-01 21:00:00	10.0	OK	Normal	
7	78901234	2020-01-01 22:00:00	2020-01-01 23:00:00	10.0	OK	Normal	
8	65432109	2020-01-02 00:00:00	2020-01-02 01:00:00	10.0	OK	Normal	
9	01234567	2020-01-02 02:00:00	2020-01-02 03:00:00	10.0	OK	Normal	
10	45678901	2020-01-02 04:00:00	2020-01-02 05:00:00	10.0	OK	Normal	

Fig. 9 Ingresos y salidas. Elaboración propia

Dentro del sistema se registra a los clientes, solo la primera vez que acuden al parqueo (Ver Figura 9).

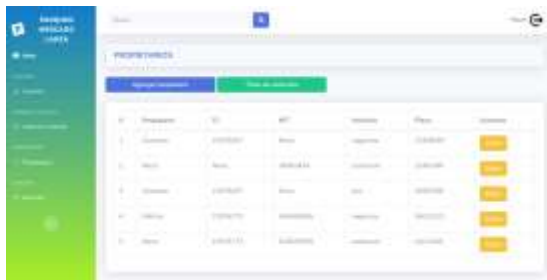


Fig. 10 Registro clientes. Elaboración propia

El sistema cuenta con un módulo de reportes de donde se filtra los datos requeridos y se pueden imprimir (Ver Figura 10).

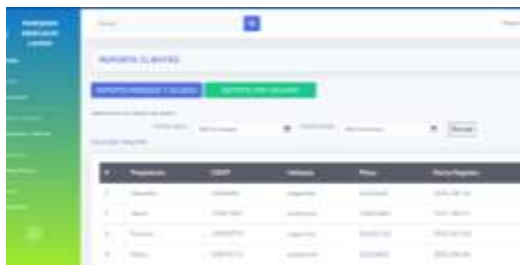


Fig. 11 Reportes. Elaboración propia

VI. RESULTADOS

Se logró detectar satisfactoriamente las placas de los vehículos, como se puede ver en la siguiente figura, verificando que la red neuronal convolucional es capaz de detectar las placas.



Fig. 12 Detección de placa. Elaboración propia

El sistema así también registra a los usuarios, donde el administrador es quien registra a los operadores. Se realizó las pruebas unitarias en cada módulo por lo que, los resultados obtenidos de cada módulo van de acuerdo con lo planificado de acuerdo a los requerimientos.

VII. CONCLUSIONES

- Conforme al desarrollo del sistema planteado para el presente proyecto con base en las entrevistas y en la observación del trabajo de campo realizado, se pudo analizar los procesos y procedimientos en la gestión del parqueo habiéndose identificado que los factores que generan demoras son: (1) los procedimientos manuales al abrir la valla para entrar y salir, (2) verificar los espacios libres y (3) el cálculo de tiempo de permanencia a partir de una hoja de cálculo, la solución a lo mencionado, ha sido incorporado en el presente proyecto por lo que se logró coadyuvar en la gestión de los parques administrados por el Gobierno Autónomo Municipal de La Paz conforme lo planteado inicialmente
- Como aporte de ingeniería para el desarrollo del presente proyecto se aplicó el modelo de red neuronal convolucional Darknet y Yolo que fue utilizada en la detección e identificación de las placas, procedimiento mediante el cual se facilitó y agilizó los procesos de registro al ingreso de vehículos; y al momento de la salida, el sistema brinda información exacta de los tiempos de permanencia y uso del servicio, que es la base para la aplicación de tarifas de cobro.
- Con el desarrollo del módulo de control de ingresos y salidas de vehículos se puede determinar la cantidad de espacios disponibles, evitándose de esta manera tener que revisar con el apoyo de un funcionario la opción de determinar la disponibilidad de espacios, dicha situación facilita y agiliza la atención de los clientes.

REFERENCES

- [1] AddPPTo. (21 de agosto de 2015). ¿Qué es un sistema web? AddPPTo. <http://www.addappto.com/que-es-un-sistema-web/>
- [2] Andrade, M. (marzo de 2016). Metodologías de desarrollo de software. definicionabc. Obtenido de <https://www.definicionabc.com>
- [3] Armando Duany Dangel (24 de Feb de 2010). "Sistemas de Información". <https://www.econlink.com.ar/sistemas-informacion/definicion>
- [4] Banda, H. (2014). Inteligencia Artificial: Principios y Aplicaciones. (1era, Ed.) Quito
- [5] Bergenti, F., Gleizes, M.-P., & Zambonelli, F. (Eds.). (2014). Methodologies and Software Engineering for Agent Systems. Springer.
- [6] Boesh G. (s.f.). What is Computer Vision? The Complete Technology Guide for 2022. Viso.ai. Recuperado el 2022: <https://viso.ai/computer-vision/what-is-computer-vision/>
- [7] Campo, L. (s/f). Sistema de Información. Incap.int. Recuperado el 16 de marzo de 2022, de <http://www.incap.int/sisvan/index.php/es/acerca-de-san/conceptos/797-sin-categoria/501-sistema-de-informacion>
- [8] CEPAL. (s.f.). Gobiernos autónomos municipales de Bolivia. Recuperado el 2022, de <https://observatorioplanificacion.cepal.org/es/instituciones/gobiernos-autonomos-municipales-de-bolivia>

- [9] Chavez, J. (s.f.). Aprendizaje supervisado: Qué es, tipos y ejemplo. Ceupe. Recuperado el 2022, de <https://www.ceupe.com/blog/aprendizaje-supervisado.html>
- [10] Determinación de costos. (s.f.). Blogspot.Com. Recuperado el 17 de abril: <https://determinaciondecostospedrop.blogspot.com/2015/12/determinacion-de-costos.html>
- [11] Diagrama de actividades. (s.f.). DiagramasUML. <https://diagramasuml.com/actividades/>
- [12] Diagrama de componentes. (s.f.). DiagramasUML. <https://diagramasuml.com/componentes/>
- [13] digite. (20 de mayo de 2021). ¿Qué Es La Metodología Scrum? Y Gestión De Proyectos Scrum. Digite. <https://www.digite.com/es/agile/que-es-scrum/>
- [14] Dölen, M., & Lorenz, R. D. (2002). General methodologies for neural network programming. International journal of smart engineering system design, 4(1), 63–73. <https://doi.org/10.1080/10255810210629>
- [15] Equipo Coremain. (s.f.). Recuperado el 2022, de <https://www.coremain.com/desarrollo-software-agilscrum/#:~:text=Scrum%20es%20un%20conjunto%20de,aplicar%20una%20u%20otra%20metodología>
- [16] Fowler, M. (1999). UML gota a gota. México: ADDISON WESLEY LONGMAN.
- [17] GAMLP. (s.f.). Misión y visión. Recuperado el 11 de abril de 2022, de <https://www.lapaz.bo/gobierno/mision-y-vision/#:~:text=Somos%20una%20entidad%20p%C3%BAblica%20municipal,su%20territorio%20y%20prestando%20servicios>
- [18] Garzas J. (19 de mayo de 2012). El ciclo de vida iterativo e incremental. JavierGarzas. <https://www.javiargarzas.com/2012/10/iterativo-e-incremental.html>
- [19] Herazo L. (4 de septiembre de 2020). ¿Qué es una aplicación móvil? Anincubator Website. <https://anincubator.com/que-es-una-aplicacion-movil/>
- [20] Historia de usuario. (26 de abril de 2021). Scrum Manager BoK. https://www.scrummanager.net/bok/index.php/Historia_de_usuario
- [21] IBM Cloud Education (junio de 2020). Inteligencia Artificial. IBM. Obtenido de <https://www.ibm.com>
- [22] IEEE Std (1993). IEEE Software Engineering Standard: glosary of Software Engineering Terminology. IEEE Computer Society Press.
- [23] Javier, A. (abril de 2019). Aprendizaje Automático. sitiointeligencia. Obtenido de <https://sitiointeligenciaa.wordpress.com>
- [24] Ley N° 2028. Ministerio de Planificación del Desarrollo, 28 de octubre de 1999. <http://www.planificacion.gob.bo/uploads/marco-legal/Ley%20N%C2%B0%202028%20DE%20MUNICIPALIDADES.pdf>
- [25] Lopez Briega, R. (29 de noviembre de 2018). Deep Learning. Github.Io. <https://iaarbook.github.io/deeplearning/>
- [26] Luca. (s.f.). Luca AI Powered Decisions. Recuperado el 2022, de <https://luca-d3.com/es/dataspeaks/diccionario-tecnologico/python-lenguaje>
- [27] Luna Gonzales, J. (8 de febrero de 2018). Tipos de aprendizaje automático. SoldAI. <https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-6413e3c615e2>
- [28] Mazón B. (20 de diciembre de 2015). SCRUM Vida. ResearchGate. https://www.researchgate.net/figure/Figura-27-Ciclo-de-Vida-Scrum_fig3_318280962
- [29] Meel V. (s.f.). Image Recognition: The Basics and Use Cases (2022 Guide). Viso.ai. Recuperado el 2022: <https://viso.ai/computer-vision/image-recognition/>
- [30] Microsoft 365 Team. (24 de septiembre de 2019). La guía sencilla para la diagramación de UML. Microsoft. <https://www.microsoft.com/es-ww/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling>
- [31] Migani, S. (20 de diciembre de 2013). Un modelo de estimación de proyectos de software. https://www.academia.edu/4853589/UN_MODELO_DE_ESTIMACION_DE_PROYECTOS_DE_SOFTWARE
- [32] Mobile-D Homepage (12 de noviembre de 2015). Mobile-D VTT's methodology for agile software development. <http://agile.vtt.fi/mobile.html>
- [33] Modelo de desarrollo de software. (5 de mayo de 2013). Ecured. https://www.ecured.cu/Modelo_de_desarrollo_de_software
- [34] Moreno, P. A. (2012). Capítulo 3. Sistemas. INTRODUCCIÓN A LA INGENIERÍA DE SISTEMAS (pp. 42-43). Bogotá: Universidad Nacional Abierta y a Distancia.
- [35] Padgham, L., & Winikoff, M. (2004). Developing Intelligent Agent. England: John Wiley & Sons Ltd.
- [36] Ponce Cruz, P. (2010). Inteligencia Artificial con Aplicaciones a la Ingeniería. México: Alfaomega.
- [37] Project Management Institute. (2018). Um guia do Conhecimento em Gerenciamento de Projetos (guia PMBOK): (Brazilian Portuguese version of: A guide to the Project Management Body of Knowledge :PMBOK Guide) (6th ed.). Project Management Institute.
- [38] Ramirez, R. (2012). Métodos para el desarrollo de aplicaciones móviles. España: Universidad Oberta de Catalunya.

Biografía Autor(es)



Esther Marisol Alejo Gomez, es estudiante egresada en grado de licenciatura de la Facultad de Ingeniería en la carrera de Ingeniería de Sistemas en la Escuela Militar de Ingeniería (EMI), en la ciudad de La

Paz, Bolivia. Ella realizó sus pasantías en el Ministerio de Defensa, 2021, durante sus estudios en la universidad, participó de la Feria Regional de La Paz con el proyecto “Sistema De Reconocimiento De La Lengua De Señas Para Coadyuvar En La Comunicación Entre Sordomudos Y La Sociedad”, 2020; también fue parte de la Feria Regional con el proyecto “Interfaz Holográfica para la Manipulación del Ordenador Mediante el Reconocimiento de la Mano”, 2022.

Ing. Alejo, premio de segundo lugar al proyecto “Sistema De Reconocimiento De La Lengua De Señas Para Coadyuvar En La Comunicación Entre Sordomudos Y La Sociedad” el año 2020; premio con primer lugar al proyecto “Sistema De Reconocimiento De La Lengua De Señas Para Coadyuvar En La Comunicación Entre Sordomudos Y La Sociedad” el año 2022.

Fecha de Envío del Artículo: La Paz, 11 de noviembre de 2022

Fecha de Recepción de artículo: La Paz, 6 de diciembre de 2022