

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

División Multidisciplinaria en Ciudad Universitaria

Instituto de Ingeniería y Tecnología

Departamento de Ingeniería Eléctrica y Computación



Clasificación de imágenes de habitaciones de hotel en base a un
algoritmo híbrido, para ayudar a combatir la trata de personas

Anteproyecto de investigación presentado por:

Diego Almanza Beyless 169838

Requisito para la obtención del título de

INGENIERO EN SOFTWARE

Asesores

Dr. Rogelio Florencia Juárez

Dr. Gilberto Rivera Zárate

Ciudad Juárez, Chihuahua a 21 de noviembre de 2022

Asunto: Liberación de Asesoría

Mtro. Ismael Canales Valdiviezo
Jefe del Departamento de Ingeniería
Eléctrica y Computación
Presente.-

Por medio de la presente me permito comunicarle que después de haber realizado las asesorías correspondientes al reporte técnico Clasificación de imágenes de habitaciones de hotel en base a un algoritmo híbrido, para ayudar a combatir la trata de personas, del alumno Diego Almanza Beyless de la Licenciatura en Ingeniería de Software, considero que lo ha concluido satisfactoriamente, por lo que puede continuar con los trámites de titulación intracurricular.

Sin más por el momento, reciba un cordial saludo.

Atentamente

Asesor 1	Asesor 2
Dr. Rogelio Florencia Juárez	Dr. Gilberto Rivera Zárate

Ccp. Mtro. David Absalón Uruchurtu Moreno
Diego Almanza Beyless
Archivo

Ciudad Juárez, Chihuahua a 21 de noviembre de 2022

Asunto: Autorización de impresión

C. Diego Almanza Beyless

Presente.-

En virtud de que cumple satisfactoriamente los requisitos solicitados, informo a usted que se autoriza la impresión del proyecto de Clasificación de imágenes de habitaciones de hotel en base a un algoritmo híbrido, para ayudar a combatir la trata de personas, para presentar los resultados del proyecto de titulación con el propósito de obtener el título de Licenciado en Ingeniería de Software.

Sin otro particular, reciba un cordial saludo.

Dra. Julia Patricia Sánchez Solís
Profesora de Seminario de Titulación II

Declaración de Originalidad

Yo Diego Almanza Beyless declaro que el material contenido en esta publicación fue generado con la revisión de los documentos que se mencionan en la sección de Referencias y que no ha sido utilizado para obtener otro título o reconocimiento en otra Institución de Educación Superior.



Diego Almanza Beyless

Agradecimientos

En este espacio me gustaría dedicar algunos párrafos en agradecimiento a las personas que de alguna manera me ayudaron durante mi proceso universitario hasta llegar a este punto de mi formación académica.

Antes que todo me gustaría agradecer a mi familia, principalmente a mis padres, ya que gracias a su sacrificio he llegado a este punto en mi educación y formación, gracias al apoyo incondicional y la educación que me han brindado durante el tiempo que llevo de vida es que estoy en esta parte final de mi etapa como universitario. Siempre estaré en deuda con mis padres ya que sin ellos nada de esto habría sido posible, al igual que mi hermano ya que siempre ha mostrado disposición para ayudarme en cualquier momento sin importar la circunstancia.

Agradezco a todos los profesores que tuve en cada materia dentro de la universidad, cada uno de ellos fue de gran importancia debido a que nos impartieron parte de su conocimiento hacia mis compañeros y a mí de una manera profesional. Me gustaría agradecer de manera especial a los asesores de este proyecto, el Dr. Rogelio Florencia Juárez y el Dr. Gilberto Rivera Zárate, por ser una excelente guía para mí en el desarrollo de este trabajo tan importante y mostrarse siempre con disponibilidad, amabilidad y paciencia al trabajar conmigo.

Agradezco a mi pareja sentimental ya que fue mi fiel acompañante durante todo este proceso, por darme ánimos cuando las cosas se complicaban y nunca dejarme solo sin importar el escenario. También agradecer a mis amigos más cercanos al ayudarme a crecer como persona y siempre estar para mí en los malos y buenos momentos.

Por último, me gustaría agradecer a la Universidad Autónoma de Ciudad Juárez, en especial al campus de Ciudad Universitaria por ser un excelente ambiente para prepararse profesionalmente, por la buena vibra que se transmite a través de todos los estudiantes, profesores y personal del campus es que siempre voy a considerar a la UACJ-CU como mi segunda casa.

A todos ellos, gracias totales.

Dedicatoria

Me gustaría dedicar este proyecto de titulación a mis abuelos maternos, durante el desarrollo de este proyecto perdí a mis abuelos José y Margarita, siempre soñé con el momento de ir a visitarlos con la intención de darles la noticia que había concluido mi licenciatura de manera exitosa, pero lamentablemente la naturaleza decidió llevárselos antes de poder cumplir ese sueño, es por esto que decido dejar plasmado el recuerdo de mis abuelitos en este proyecto dedicándoles este trabajo tan importante para mí con todo mi amor y respeto hacia ellos.

Índice de contenido

Contenido

Declaración de Originalidad	1
Agradecimientos	2
Dedicatoria.....	3
Índice de contenido	4
Índice de figuras.....	7
Índice de tablas	8
Resumen.....	9
Introducción	10
I. Planteamiento del problema	12
1.1 Antecedentes	12
1.1.1 Contexto.....	12
1.1.2 Antecedentes	12
1.2 Definición del problema.....	15
1.3 Objetivos.....	15
1.3.1 Objetivo general.....	15
1.3.2 Objetivos específicos	16
1.4 Pregunta de investigación.....	16
1.5 Justificación	16
II. Marco Referencial	17
2.1 Marco teórico	17
2.1.1 Clasificación de imágenes.....	17
2.1.2 Análisis y procesamiento de imágenes	18
2.1.3 Inteligencia artificial	19

2.1.4 Machine learning	19
2.1.5 Deep learning	20
2.1.5.1 Redes neuronales	20
2.1.5.2 Redes neuronales convolucionales	21
2.2 Marco tecnológico	24
2.2.1 Python	24
2.2.2 Keras	24
2.2.3 Tensorflow	25
2.2.4 Anaconda	25
2.2.5 Jupyter Notebook	25
2.2.6 Pandas	25
2.2.7 Matplotlib.....	26
III. Desarrollo del proyecto.....	27
3.1 Producto propuesto	27
3.1.1 Delimitaciones y limitaciones.....	28
3.1.1.1 Delimitaciones	28
3.1.1.2 Limitantes	28
3.1.2 Metodología de desarrollo	29
3.2 Comprensión del negocio	30
Determinar objetivos del proyecto.....	30
3.3 Entendimiento de los datos	30
3.3.1 Recolectar los datos	30
3.3.2 Descripción de los datos	31
3.3.3 Exploración de los datos	31
3.3.4 Verificar calidad de los datos.....	34

3.4 Preparación de los datos.....	34
3.5 Modelado	35
3.5.1 Técnica de modelado	35
3.5.3 Construcción del modelo	35
3.5.3.1 CNN entrenada desde cero.....	35
3.5.3.2 SVM.....	36
3.5.3.3 Random Forest	36
3.5.3.3 VGG16.....	37
IV. Resultados y Discusiones	38
4.1 Resultados	38
4.2 Discusiones	40
V. Conclusiones	42
Referencias.....	44
Apéndices.....	47

Índice de figuras

Ilustración 1 Resultados obtenidos en el estudio de Stylianou Abby [3].	14
Ilustración 2 Niveles de procesamiento [10].	18
Ilustración 3 Proceso que sigue la CNN [11].	21
Ilustración 4 Ejemplo de convolución [11].	22
Ilustración 5 Ejemplo de max-pooling y average-pooling [11].	22
Ilustración 6 Resultados utilizando cinco capas de convolución, reducción y ReLU [11].	23
Ilustración 7 Ejemplo de capas totalmente conectadas [11].	23
Ilustración 8 Diagrama de la arquitectura propuesta.	28
Ilustración 9 Fases del modelo CRISP-DM [23].	29
Ilustración 10 Mapa de árbol.	32
Ilustración 11 Cantidad de hoteles por cadena.	32
Ilustración 12 Dimensiones de imágenes en una clase.	33
Ilustración 13 Características de la CNN entrenada desde cero.	36
Ilustración 14 Características de VGG16	37

Índice de tablas

Tabla 1 Resultados obtenidos de las métricas utilizadas para la evaluación.....	39
Tabla 2 Ejemplos de resultados al clasificar	39
Tabla 3 Matriz de confusión del modelo CNN + RF.	40

Resumen

La trata de personas es un problema que radica en todo el mundo, en algunos casos los traficantes suelen tener aprisionadas a sus víctimas en habitaciones de hotel y publicar fotos de ellas en el mercado negro. Estas fotos suelen llegar como evidencia en las investigaciones, por lo que, una de las principales tareas de las autoridades es ubicar el hotel donde fue tomada la fotografía de la víctima.

La clasificación de imágenes mediante Inteligencia Artificial es una tarea que trata de comprender una imagen con el objetivo de asignarla a una categoría en específico. En este proyecto de investigación se desarrollaron múltiples algoritmos de clasificación híbrida con el objetivo de identificar que algoritmo funciona mejor al clasificar imágenes de habitaciones de hotel y de esta manera facilitar la identificación de en qué cadena hotelera fue tomada la fotografía. Los algoritmos que se desarrollaron utilizaron una red neuronal convolucional entrenada desde cero y la red pre entrenada *VGG16*, y se hizo uso de los algoritmos *Random Forest (RF)* y *Support Vector Machine (SVM)* para la clasificación híbrida.

En los resultados obtenidos se logró observar que para la clasificación de imágenes sobre habitaciones de hotel resulta efectivo utilizar una clasificación híbrida ya que, al aplicar RF o SVM sobre la red neuronal se obtuvieron mejores resultados de clasificación que cuando se realizó una clasificación no híbrida.

Palabras clave: Clasificación híbrida, red neuronal convolucional, *Random Forest*, *Support Vector Machine*, *VGG16*, trata de personas, habitaciones de hotel.

Introducción

Un área que está cada vez más involucrada en el desarrollo de aplicaciones con Inteligencia Artificial (IA) es el reconocimiento de imágenes, comúnmente conocido como clasificación de imágenes. El aprendizaje profundo es una de las áreas más populares de la IA y, a menudo, se usa para identificar y clasificar imágenes. Gracias a estas nuevas tecnologías podemos disponer de sistemas más preciso que el ojo humano que puede clasificar e interpretar automáticamente imágenes.

En los casos de trata de personas, los traficantes a menudo colocan fotos de las víctimas tomadas en alguna habitación de hotel. Estas imágenes son pruebas que pueden ayudar a investigar y enjuiciar a los autores de estos delitos. La identificación de hoteles en fotografías de habitaciones de hotel es importante para las investigaciones de trata de personas debido a que estas imágenes conectan a las víctimas directamente con su ubicación y es de ayuda para determinar donde fue traficada la víctima o prevenir futuros intentos de trata de personas.

El objetivo principal de este proyecto fue desarrollar múltiples modelos de clasificación de imágenes con el propósito de identificar qué modelo obtiene mejores resultados al trabajar con imágenes de hotel, de esta manera al obtener modelos con buenos resultados al clasificar imágenes de habitaciones de hotel podrían ser de ayuda en las investigaciones sobre trata de personas al identificar la cadena hotelera donde fue tomada la fotografía que se tenga como evidencia.

Para realizar estos modelos se decidió algo diferente a lo encontrado en la literatura donde por lo general se utilizaba una red neuronal convolucional pre entrenada en sus modelos, es por esto que en este proyecto se decidió realizar un algoritmo híbrido, utilizando una red neuronal convolucional entrenada desde cero y una clasificación *Support Vector Machine*, *Random forest* o ambas.

Para realizar el modelo se cuenta con una cantidad de imágenes superior a las 50 mil, lo cual requiere un equipo de cómputo con altas capacidades de procesamiento para lograr entrenar un modelo con todas las imágenes disponibles, lamentablemente no se contó con un equipo de cómputo que fuera capaz de procesar una gran cantidad de imágenes, además a esta limitante tampoco es posible identificar la ubicación exacta del hotel debido a que el

conjunto de datos a utilizar solo brinda como información la cadena hotelera a la cual pertenece el hotel donde fue tomada la fotografía.

I. Planteamiento del problema

En este capítulo se describe un contexto sobre el tema a tratar, los antecedentes que existen sobre el este, la definición del problema encontrado, los objetivos para el proyecto, la pregunta de investigación y la justificación, esto para entender de una buena manera la problemática y como se tratará.

1.1 Antecedentes

En esta sección se hablará del contexto sobre la trata de personas y como se involucra con el sector hotelero, después se comentarán los distintos trabajos previos encontrados en la literatura que están relacionados con esta investigación.

1.1.1 Contexto

La trata de personas es un fenómeno mundial del cual se sabe que se requiere una respuesta urgente y eficaz ante dicho problema. A pesar de los avances legislativos y la introducción de intervenciones nacionales e internacionales, las respuestas a las víctimas y los perpetradores han sido limitadas [1].

Un alto perfil de casos sobre la trata de personas en cadenas hoteleras del Reino Unido, Estados Unidos y China, demuestran que el sector hotelero es uno de los más vulnerables de este delito [2]. Estadísticas publicadas por la *US National Human Trafficking Hotline* (NHTH) revela que el 10.5% de casos reportados sobre la trata de personas involucran hoteles, haciendo que el sector hotelero sea el segundo lugar más popular donde se realiza este delito, solo por debajo de los burdeles comerciales, los cuales ocupan el primer lugar [2].

1.1.2 Antecedentes

La trata de personas implica el reclutamiento, exportación y explotación sexual o laboral de personas vulnerables por la fuerza. Las estimaciones del alcance de la trata de personas varían enormemente debido a que, la *Organización Internacional del Trabajo* identifica 21 millones de víctimas anuales, mientras que *Walk Free Foundation* reporta hasta 45.8 millones de víctimas anuales [2]. Cualquiera que sea la cifra real, esta evidente violación de los derechos humanos afecta a una gran cantidad de los países del mundo.

La investigación de Paraskevas Alexandros y Maureen Brookes [2] tiene como objetivo identificar y analizar las vulnerabilidades del sector hotelero que explotan los traficantes de personas para utilizar los hoteles como conductos para la trata de seres humanos. El estudio identifica los tipos de trata de personas que ocurren dentro de la industria hotelera y los factores específicos a nivel macro, meso y micro que aumentan la vulnerabilidad de los hoteles a la trata de personas [2].

El trabajo de Stylianou Abby [3] tiene tres contribuciones principales. Primero, proponer y formular el problema del reconocimiento de instancias hoteleras. Segundo, seleccionar y compartir un *dataset* sobre imágenes de habitaciones y un protocolo de evaluación para este problema a una escala que es relevante para un esfuerzo internacional para abordar la trata de personas. Tercero, describir y probar algoritmos que incluyen los pasos necesarios para abordar el problema con una base razonable de comparaciones sobre la efectividad de cada algoritmo [3].

El *dataset* utilizado en este trabajo consiste en 1,027,871 imágenes de alrededor de 50,000 hoteles de todo el mundo. Cada imagen cuenta con los siguientes metadatos: nombre de hotel, localización y la cadena hotelera. Las fuentes de estos datos provienen de sitios web sobre viajes, que son de uso público [3].

Para obtener los resultados se utilizó la red pre entrenada *Resnet-50* para el reconocimiento de objetos, se utilizaron tres modelos *Fixed-Object*, *Fixedscence* y un modelo propio, usaron las representaciones de características fijas y se refieren a ellas como *Fixed-Object* y *Fixedscence*. El modelo propio usa el conjunto de entrenamiento llamado *Hotels-50k* como entrada para ajustar un modelo pre entrenado *Resnet-50* para *ImageNet*, como salida usaron las características de 256-Dimensiones. Entrenaron este modelo usando la variante llamada *Triplet loss* [3].

Se obtuvieron mejores resultados usando el modelo desarrollado por ellos mismos, a diferencia de los otros dos modelos, su modelo puede codificar información sobre los colores importantes y objetos de la habitación. En la Ilustración 1 se pueden observar algunos ejemplos de los resultados obtenidos con algunas imágenes clasificadas.

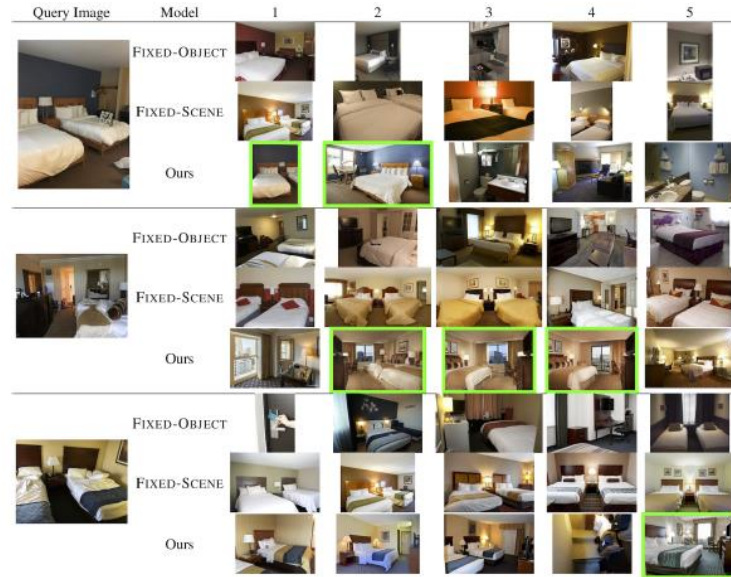


Ilustración 1 Resultados obtenidos en el estudio de Stylianou Abby [3].

El trabajo publicado por Trideep Rath en [4] tiene como objetivo clasificar imágenes de hotel mediante técnicas de *machine learning* para entrenar múltiples modelos: por ejemplo, uno para clasificar en categorías como cuartos, baños y comida, y otro para piscinas y playas. En el desarrollo se utilizó *InceptionV3* y la técnica *transfer learning*. Como resultados, algunas categorías como “baños”, “cuartos” y “planta baja” tuvieron un 90% de efectividad. Es interesante ver que el modelo funciona bien para “planta baja” que contiene la menor cantidad de imágenes. Categorías como “gimnasio”, “lavandería”, “comida”, “restaurante” y “piscina” tienen una precisión superior al 80%. Categorías como “salón de banquetes”, “playa”, “conferencia”, “vestíbulo” y “spa” no funcionan bien debido al ruido en sus etiquetas, la falta de imágenes para la capacitación y la naturaleza de múltiples etiquetas de las imágenes [4].

El objetivo general de Fatma Bozyiğit en [5] es diseñar, implementar y evaluar modelos basados en redes convolucionales que encuentren automáticamente etiquetas relevantes en imágenes de hoteles. Esto con un *dataset* que incluye 20,000 imágenes de hoteles en la página web “Make my trip”. Todas las imágenes se clasifican en 215 subcategorías basadas en las etiquetas del conjunto de datos de *ImageNet* utilizando los modelos *VGG-16*, *VGG19* y *Inception-V3*. Se evaluó y comparó los resultados de los métodos de *deep learning*, donde el resultado del *VGG-16* es superior a los modelos *VGG-19* e *InceptionV3* con un valor de precisión del 84.86% [5].

1.2 Definición del problema

Los traficantes publican regularmente fotografías de sus víctimas en habitaciones de hotel para publicarlas en línea. Estas fotografías son evidencia que se puede utilizar para encontrar y enjuiciar a los autores de estos crímenes. Sin embargo, para poder utilizar estas fotos, los investigadores deben poder determinar en qué hotel se tomaron [3].

Reconocer un hotel a partir de la imagen de una habitación de hotel es importante para las investigaciones de trata de personas. Las imágenes vinculan directamente a las víctimas con los lugares y pueden ayudar a verificar dónde han sido víctimas de trata y adónde sus traficantes podrían moverlas a ellas o a otras personas en el futuro. Reconocer el hotel a partir de sus habitaciones es un desafío debido a la baja calidad de la imagen, las perspectivas de la cámara poco comunes, las grandes oclusiones (a menudo la víctima) y la similitud de los objetos (por ejemplo, muebles, arte, ropa de cama) en diferentes habitaciones de hotel [3].

Los estándares de la decoración en las grandes cadenas de hoteles generan que muchas habitaciones pueden parecer bastante similares a primera vista. Esto puede ayudar a la identificación de la cadena hotelera, para así poder reducir el rango de posibilidades, lo cual ayudaría a reconocer la ubicación donde se encuentra la víctima de una manera más rápida.

1.3 Objetivos

En esta sección se describe el objetivo general del proyecto y los objetivos específicos que se deben cumplir mediante el desarrollo del proyecto.

1.3.1 Objetivo general

Desarrollar un clasificador de imágenes de habitaciones hoteleras para identificar la cadena del hotel usando algoritmos de clasificación híbridos.

1.3.2 Objetivos específicos

- Analizar los distintos algoritmos de clasificación de imágenes utilizados en la literatura.
- Realizar la extracción de características de los datos usando una red neuronal convolucional.
- Entrenar un modelo de *machine learning* en base a las características extraídas.
- Evaluar el algoritmo utilizando métricas ya existentes en la literatura.

1.4 Pregunta de investigación

¿Qué tan efectivo puede llegar a ser un algoritmo híbrido de clasificación de imágenes para el reconocimiento de habitaciones de hoteles?

1.5 Justificación

Con el desarrollo de este proyecto se pretende realizar un clasificador de imágenes de habitaciones de hotel que ayude a colaborar con las investigaciones relacionadas a la trata de personas identificando la ubicación donde fueron tomadas las fotografías de las víctimas; se buscará reducir los tiempos de identificación de la ubicación donde fue tomada la fotografía y así poder agilizar el proceso de investigación.

II. Marco Referencial

2.1 Marco teórico

En este capítulo se describirán los temas relacionados con el desarrollo de este proyecto. Se empezará hablando de la clasificación de imágenes y se tomarán algunos temas de la inteligencia artificial.

2.1.1 Clasificación de imágenes

Una imagen tomada con un dispositivo con una cámara suele presentar una diversificación de sombras y tonos. Las imágenes con estas propiedades se denominan imágenes analógicas. Para poder utilizar una imagen analógica en escala de grises o a color en una computadora o dispositivo inteligente, primero se debe convertir al formato digital [6].

Las dos fuentes principales de pérdida de información al tomar imágenes digitales son la naturaleza discreta de los píxeles y el rango limitado de valores de intensidad de luz que puede tener cada elemento. Estas dos razones conducen a los conceptos de resolución espacial y profundidad de color. Una resolución espacial se refiere a la cantidad de detalles visibles en un píxel de imagen, mientras que la profundidad de color se refiere a la cantidad de bits precisos para codificar y almacenar información de color para cada píxel de una imagen [6]. Por lo anteriormente mencionado es necesario que las imágenes que se utilicen para el desarrollo de un clasificador de imágenes cuenten con una resolución lo suficientemente alta para poder apreciar los detalles que hay en ella.

Una de las labores más significativas del procesamiento de imágenes es clasificar cada píxel como perteneciente a una clase. Después de la clasificación de imágenes se obtiene una lista de categorías cartográficas. Por ejemplo, se puede ver la cantidad de píxeles, y el área asignada a cada categoría. La imagen multibanda se transforma en otra imagen, con las mismas dimensiones y forma que la original. Sin embargo, la superficie digital asignada a cada píxel está relacionada con la etiqueta de la imagen, que es el rango asignado a cada píxel, en lugar de la luz detectada por el sensor [6].

2.1.2 Análisis y procesamiento de imágenes

El análisis de imágenes digitales es un campo técnico relacionado con la búsqueda de medidas, datos o información en imágenes. El objetivo de incluir estos métodos es hacer más fácil la recuperación e interpretación de la información disponible, esta información está relacionada con el contenido de la imagen de entrada. El análisis de imágenes es diferente porque tiene una imagen como entrada y el resultado suele ser numérico en lugar de otra imagen [7].

En [7] se menciona que, la transformación de imágenes sin procesar a conjuntos de parámetros y datos implica muchos procedimientos de procesamiento y filtrado. Se requiere de un análisis de imágenes y adaptación a aplicaciones específicas. Esto significa que el procesamiento de imágenes depende del problema. Según [7] existen tres métodos de procesamiento de imágenes, algoritmos en el dominio espacial, algoritmos en el dominio de la frecuencia y algoritmos de extracción de características.

Las herramientas del procesamiento de imágenes se configuran de acuerdo con el grado de proceso que se desea realizar para examinar la información en la imagen. Estos niveles crean la cadena de procesamiento que se muestra en la Ilustración 2 [8].



Ilustración 2 Niveles de procesamiento [10].

Los niveles de la Ilustración 2 se describen de la siguiente manera según [8]:

- Preprocesamiento: Esta operación se utiliza para ajustar la información de la imagen y analizarla mejor en el siguiente paso. Algunos ejemplos de preprocesamiento incluyen ajustar el contraste o el brillo de la imagen.
- Segmentación: Agrupa imágenes en áreas que brindan la información necesaria para resolver problemas.
- Detección de objetos y clasificación: Identificación de objetos en imágenes, para después realizar una clasificación de imágenes en base a los objetos detectados.
- Análisis de imagen: Recupera información avanzada sobre lo que se muestra en una imagen.

2.1.3 Inteligencia artificial

La definición de la Inteligencia Artificial (IA) es una respuesta complicada de interpretar, es por esto por lo que se podrían encontrar diferentes definiciones en la literatura. en [9] la IA se describe como la capacidad de una máquina para cultivarse de los datos mediante algoritmos y manejar lo que ha estudiado para tomar decisiones humanas.

Las personas ya están utilizando tecnologías impulsadas por inteligencia artificial para mejorar significativamente la eficiencia en algunas de las áreas de sus vidas [9]. La inteligencia artificial se puede utilizar en una variedad de situaciones. Uno de los principales objetivos de este proyecto es detectar, clasificar y etiquetar imágenes fijas. Para comprender mejor el tema de la inteligencia artificial en el contexto de la clasificación de imágenes, debemos comprender qué son el aprendizaje automático y el aprendizaje profundo en este contexto.

2.1.4 Machine learning

El aprendizaje automático (*Machine learning*) es de las tecnologías más importantes en IA. De hecho, permitir que las computadoras y las máquinas aprendan sin programar es parte de la informática. El aprendizaje automático utiliza algoritmos para aprender de los patrones que se encuentren en los datos. Por ejemplo, un filtro de spam utiliza esta formación para identificar los mensajes de spam y distinguirlos de los mensajes que no son spam. Este es un ejemplo simple del uso de un algoritmo para aprender un patrón y usar la información detectada para tomar una decisión. [9].

Existen distintitos subconjuntos de *machine learning*, algunos de ellos son supervisado, no supervisado y de refuerzo, a continuación, se explicará la diferencia entre estos tres subconjuntos según lo descrito en [9]. Si se cuenta con 10,000 fotografías y los algoritmos deben identificar las fotos en las que aparezca un gato:

- Aprendizaje supervisado: Los algoritmos muestran a los usuarios como insertar nueva información en datos previamente etiquetados u ordenados. Este método requiere la intervención humana para proporcionar una retroalimentación similar a la de un gato.
- Aprendizaje no supervisado: En lugar de usar datos previamente etiquetados u organizados para mostrar como revelar nueva información, los algoritmos deben

encontrar una manera de revelarse la información a sí mismos. Por lo tanto, este subconjunto no requiere de seres humanos.

- Aprendizaje por refuerzo: Los algoritmos aprenden por experiencia. siempre que lo hagan bien deben proporcionar un refuerzo positivo. Por ejemplo, la forma en la que aprenden estos algoritmos se puede comparar con la forma en que se recompensa a los perros por aprender a sentarse.

2.1.5 Deep learning

El aprendizaje profundo (Deep Learning) es un subcampo del aprendizaje automático que a menudo se usa al trabajar con problemas que requieren de una gran cantidad de datos para su resolución. El aprendizaje profundo utiliza redes neuronales jerárquicas para identificar patrones y relaciones complejos en los datos [9]. Estas aplicaciones requieren de una gran capacidad de procesamiento para poder manejar cantidades altas de información.

2.1.5.1 Redes neuronales

Una Red Neuronal Artificial es un modelo informático diseñado para formular matemáticamente la estructura y el comportamiento del cerebro humano. Se basa en el aprendizaje experiencial y la consiguiente extracción de conocimiento [10]. Actualmente se conocen muchos métodos de categorización de imágenes con un buen rendimiento, pero tienen limitaciones en los datos de entrada. Por el contrario, las redes neuronales se consideran no paramétricas, es decir, no se requiere una distribución estática de la información de entrada. Durante la fase de aprendizaje, la red “estudia” las regularidades de los datos registrados y crea reglas que pueden extenderse a datos desconocidos [6]. Para comprender el significado de una imagen se utiliza algo llamada *features* (características), las cuales se utilizan para interpretar las propiedades de los objetos con el fin de dar sentido a las imágenes más allá de las representaciones numéricas proporcionadas por las máquinas. Antes de representar el concepto de imagen, existen características en diferentes niveles de abstracción, como bordes, características de color y esquinas [10].

2.1.5.2 Redes neuronales convolucionales

Existen varios tipos de redes neuronales, pero la más utilizada para la clasificación de imágenes es una red neuronal convolucional. Estos son de la familia Deep Learning dedicada a aprender y comprender las características de las imágenes [10].

En [11] describen una red neuronal convolucional como una capa de una red neuronal densa compuesta por elementos como entrada, convolución, pooling, ReLU, una capa totalmente conectada y una capa de salida. Los filtros de convolución y reducción se pueden usar varias veces para obtener la imagen más representativa de las propiedades. Una capa combinada completa une los resultados de la clasificación.

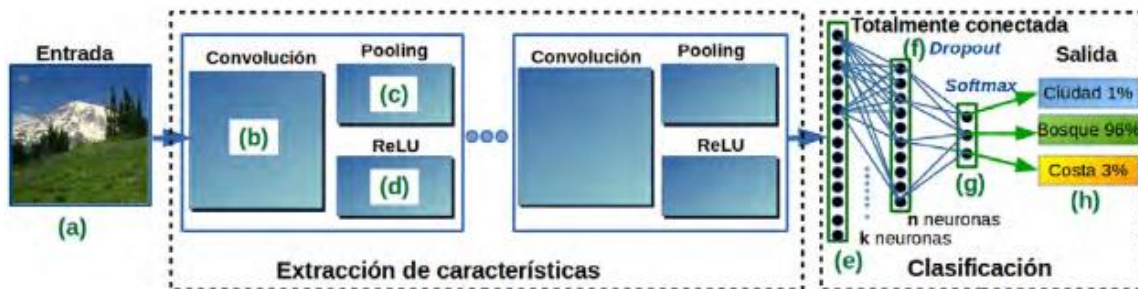


Ilustración 3 Proceso que sigue la CNN [11].

A continuación, se describen los pasos vistos en la Ilustración 3 según [11]:

- Capa de entrada: La imagen se puede figurar como un vector bidimensional de tamaño $M \times N$, donde N es el alto y M el ancho, donde cada valor de la matriz representa un píxel. Puede tener diferentes matrices según el espacio de color con el que se trabaje. Por ejemplo, si está trabajando en un espacio RGB, es un total de 3 matrices de $M \times N$, una para cada canal.
- Convolución: Si los datos se normalizan y se muestran como una matriz, se puede emplear una convolución la cual trata principalmente con dos matrices. Una es la matriz normalizada y la otra se llama Kernel(K) o Kernel. Esta matriz es de $3 \times 3, 5 \times 5$, típicamente $k \times k$. La Ilustración 4 muestra un ejemplo de una operación de convolución.

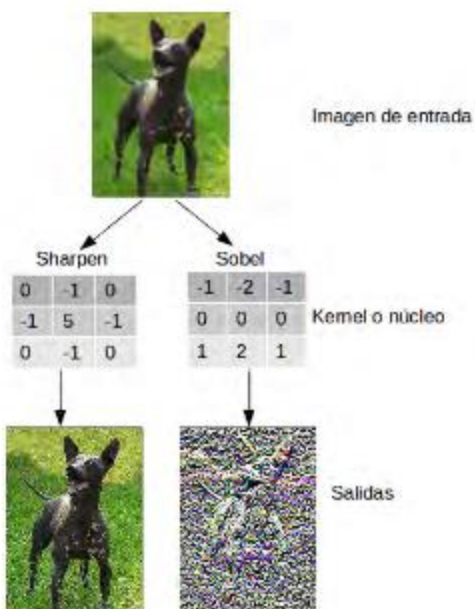


Ilustración 4 Ejemplo de convolución [11].

- c) Pooling: El objetivo principal de la capa de pooling es reducir la salida de la capa convolucional para reducir la cantidad de objetos y mantener la información de la imagen más representativa. Hay distintas agrupaciones, siendo las más comunes la agrupación media y la máxima. La agrupación promedio toma el promedio de los elementos en el área R, y la agrupación máxima toma el valor máximo en el área R. La Ilustración 5 muestra cómo funciona este tipo de agrupación.

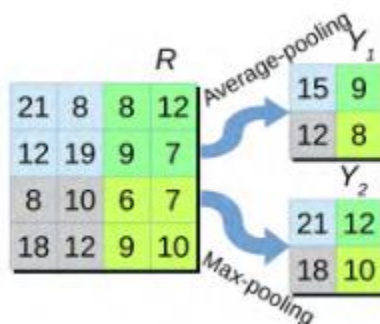


Ilustración 5 Ejemplo de max-pooling y average-pooling [11].

- d) Activación no lineal de la red: Para que una red funcione, dada sus entradas, necesita una función de activación que determine la salida de las neuronas en la red. Tiene varias características, incluida una Unidad lineal rectificadora (ReLU). Cuando se procesan grandes cantidades de datos, estas características son

importantes para reducir el tiempo de cálculo y acortar los tiempos de clasificación y entrenamiento. Al pasar por 5 capas de convolución, reducción y ReLU, se produce el resultado que se muestra en la Ilustración 6.

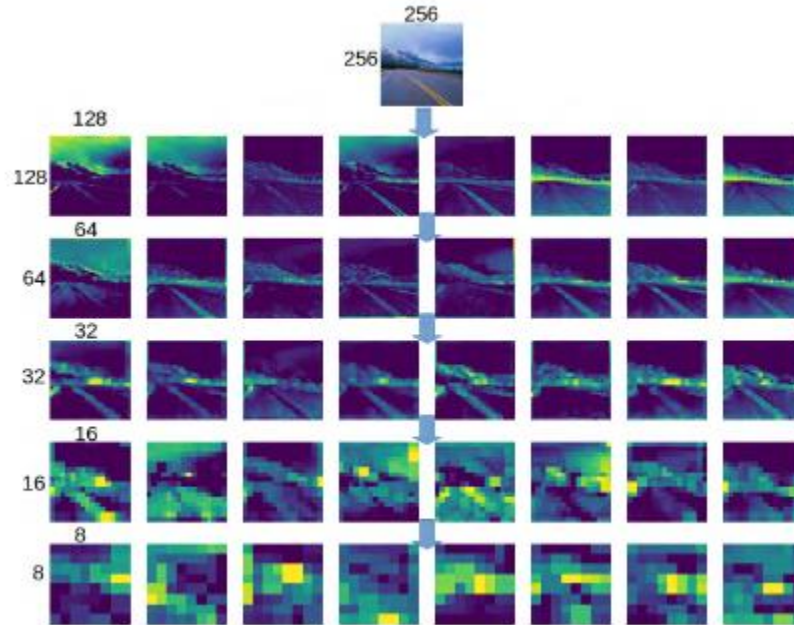


Ilustración 6 Resultados utilizando cinco capas de convolución, reducción y ReLU [11].

- e) Capa totalmente conectada: El objetivo de la capa totalmente conectada es clasificar los datos obtenidos en las capas anteriores, para llegar a una última capa la cual consta de las características extraídas convertidas a un vector.

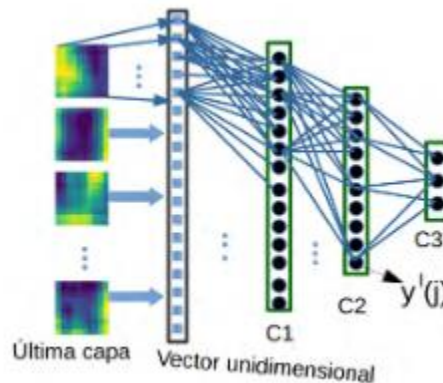


Ilustración 7 Ejemplo de capas totalmente conectadas [11].

- f) Dropout: Elimina aleatoriamente una cantidad fija de neuronas en cada iteración de entrenamiento. Esto ayuda a los nodos activos a hacer que la red sea más específica. Se utiliza para evitar sobre entrenamiento.

- g) Softmax: Función utilizada ordinariamente en la última capa totalmente conectada para predecir la imagen de entrada con un valor entre 0 y 1.
- h) Capa de salida: La tarea de esta capa es descubrir los resultados del análisis realizado en la capa anterior. Por ejemplo, una imagen de entrada tiene un 90 % de posibilidades de éxito y un perro o un gato tiene un 10 % de posibilidades de éxito. Es importante enfatizar que puede haber múltiples salidas con diferentes tasas de éxito según la arquitectura de la red neuronal convolucional.

2.2 Marco tecnológico

En esta sección se describirán las herramientas que se utilizarán durante el desarrollo de este proyecto.

2.2.1 Python

La definición de Python se describe en [12] como un “lenguaje de programación versátil, multiplataforma y multiparadigma que se destaca por su código legible y limpio”. y tiene una licencia de código abierto que se puede usar de forma gratuita en una variedad de entornos. Su objetivo es automatizar procesos que faciliten enormemente una tarea. Con esto en mente, Python escribe código fácil de leer que ahorra tiempo y recursos. Uno de sus puntos fuertes es corregir si afecta la memoria o no, mantener la integridad de la matriz y evitar la codificación compleja [12].

Aunque Python existe desde hace 30 años, en la actualidad es uno de los lenguajes más utilizados. Las tendencias del mercado laboral muestran que la demanda de experiencia en Python crece cada año. La inteligencia artificial es uno de sus aliados más importantes [12].

2.2.2 Keras

Keras es una API (*Application Programming Interface*) diseñada para el uso de humanos, no para máquinas. Keras proporciona una API simple y consistente en minimizar las acciones requeridas por el usuario en casos de uso comunes y reduce la carga cognitiva al proporcionar mensajes de error entendibles. También incluye una extensa documentación y guías de desarrollo [13].

Keras es el marco de aprendizaje profundo más utilizado entre los cinco mejores equipos ganadores de Kaggle. Keras facilita experimentar con cosas nuevas y probar más ideas más rápido que la competencia. [13].

2.2.3 Tensorflow

TensorFlow es una biblioteca de código abierto de punta a punta para el aprendizaje automático. Existe un ambiente grande y flexible de herramientas y recursos comunitarios que permiten a los investigadores innovar en el aprendizaje automático y a los desarrolladores crear e implementar fácilmente aplicaciones [14].

2.2.4 Anaconda

Anaconda es una versión única de código abierto que ofrece la forma más fácil de ejecutar el aprendizaje automático y ciencia de datos de Python en una sola computadora. Diseñado para autónomos, este conjunto de herramientas le permite utilizar miles de paquetes y bibliotecas de código abierto [15].

2.2.5 Jupyter Notebook

Jupyter Notebook es una aplicación web que permite crear y compartir documentaciones con código en vivo, texto explicativo, ecuaciones y visualizaciones. Las aplicaciones incluyen el limpiar datos, transformar los datos, modelado estadístico, simulación numérica, visualización de datos, aprendizaje automático y más [16].

2.2.6 Pandas

Según lo descrito en [17] Pandas es una librería de Python dedicada a analizar y administrar estructuras de datos. Las principales características de esta librería son:

- Leer y crear de una manera sencilla archivos en formato Excel, bases de datos SQL y CSV.
- Proporciona acceso a los datos mediante índices o nombres para filas y columnas.
- Proporciona métodos para reorganizar, dividir y fusionar registros.
- Proporciona la capacidad de manipular series de tiempo.
- Ejecuta todas estas operaciones de manera eficaz.

2.2.7 Matplotlib

En [18] se describe a Matplotlib como una librería de Python especializada en la creación de gráficos en dos dimensiones. Permite crear y personalizar los tipos de gráficos más comunes.

III. Desarrollo del proyecto

En este capítulo se describirá el producto propuesto en este proyecto, se enlistarán algunas limitaciones y delimitaciones con las que se contó, por último, se describirá la metodología con la que se trabajó en el proyecto y cómo se realizó cada una de las actividades.

3.1 Producto propuesto

En este proyecto se desarrolló un clasificador, utilizando imágenes predefinidas de una base de datos de imágenes de habitaciones de hotel de diferentes cadenas hoteleras del mundo. La base de datos proviene de una aplicación móvil llamada *TraffickCam* la cual tiene el propósito de crear una base de datos sobre imágenes de habitaciones de hotel. Esta aplicación obtiene las imágenes mediante los usuarios que cargan fotos de las habitaciones en las que se hospedaron cuando realizan un viaje a cualquier parte del mundo. La base de datos cuenta con los siguientes metadatos: Imagen, id de cadena de hotel, id de hotel y fecha de cuando se tomó la foto.

Se propuso desarrollar un algoritmo de clasificación de imágenes híbrido. Se menciona que es híbrido ya que se realizó una extracción de características mediante una red neuronal convolucional y, dadas estas características, se entrenó un modelo estándar de *machine learning*, con las características obtenidas. En la Ilustración 8 se puede apreciar la arquitectura que se utilizó para entrenar a los múltiples modelos desarrollados, donde primero se realiza el preprocesamiento de imágenes, después se dividen las imágenes en dos conjuntos, los cuales son *test* y *train*, Donde las imágenes de *train* se utilizarán para realizar el entrenamiento de la red y las imágenes de *test* para evaluar los resultados. Después de dividir las imágenes se realiza la extracción de características. Para realizarla existen dos alternativas (Dependerá del modelo), una es mediante una *Convolutional Neural Network* (CNN) entrenada desde cero y la otra alternativa mediante la CNN ya pre entrenada *VGG16*. Después de la extracción de características se realiza la clasificación y finalmente se evalúa al modelo entrenado con las imágenes de prueba. Para el desarrollo de este clasificador se trabajó con el lenguaje de programación Python y algunas herramientas como *Keras*, *TensorFlow*, *Sklearn*, etc.

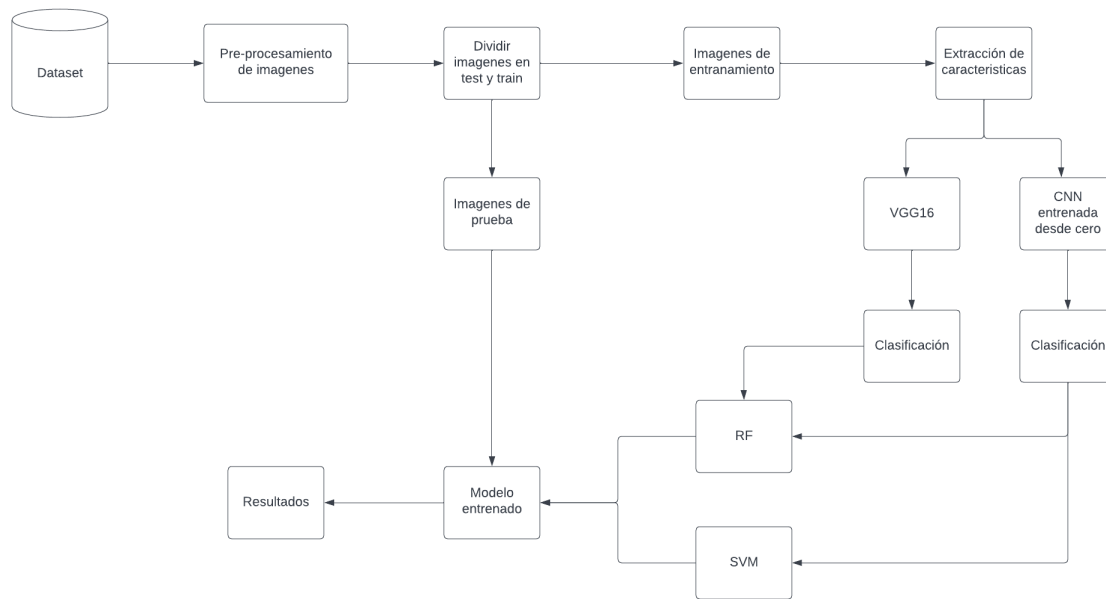


Ilustración 8 Diagrama de la arquitectura propuesta.

3.1.1 Delimitaciones y limitaciones

En esta sección primero se describen las delimitantes y limitantes que enmarcaron el desarrollo de este proyecto.

3.1.1.1 Delimitaciones

Las delimitantes o alcance del proyecto son:

- Todas las imágenes del *dataset* se encuentran en formato JPG.
- Solo considerar las cadenas con las que se entrenó al modelo.

3.1.1.2 Limitantes

Las limitantes del proyecto son:

- Contar con un equipo de cómputo con las especificaciones necesarias para poder procesar una gran cantidad de datos.
- Si la imagen cuenta con oclusión o una baja calidad, es posible que el algoritmo no funcione de una manera correcta.
- No se puede obtener exactamente la ubicación del hotel donde fue tomada la imagen.
- No se cuenta con el nombre del hotel, solo se cuenta con un id de hotel.

3.1.2 Metodología de desarrollo

Para el desarrollo de este proyecto se utilizó CRISP-DM (*Cross Industry Standard Process for Data Mining*), esta metodología cuenta con 6 fases las cuales se pueden observar en la Ilustración 9. Se decidió utilizar esta CRISP-DM ya que está relacionada al trabajo con grandes cantidades de datos, además en [21] se menciona que es una metodología amigable de usar si es la primera vez que se utiliza, ya que la fase de comprensión de negocio no parece muy diferente de un análisis de requisitos, la fase de evaluación es similar a pruebas de aceptación de usuario y la implementación es plenamente un despliegue de software.



Ilustración 9 Fases del modelo CRISP-DM [23].

En [22] se definen las fases de CRISP-DM como las siguientes:

- Comprensión del negocio (*Business Understanding*).
- Comprensión de los datos (*Data Understanding*).
- Preparación de los datos (*Data Preparation*).
- Modelado (*Modeling*).
- Evaluación (*Evaluation*).
- Implementación (*Deployment*).

En las siguientes secciones se describirán cada una de estas fases y las tareas realizadas durante el desarrollo del proyecto, esto en base a lo descrito en [23]. La fase de evaluación

se describirá en el Capítulo 4. La fase de implementación no se llevó a cabo debido a que no se cuenta con un cliente real con el cual se pueda implementar el modelo.

3.2 Comprensión del negocio

En esta sección se realizaron las tareas que corresponden a la fase 1 de la metodología CRISP-DM según lo descrito por Galán Cortina en [21]. Esta fase tiene como finalidad describir los objetivos y requerimientos del proyecto desde una perspectiva de negocio, para más adelante convertirlos en objetivos desde el punto de vista técnico y en un plan de proyecto [20].

Determinar objetivos del proyecto

El objetivo de ese proyecto es el de hacer una clasificación lo más precisa posible a partir de las imágenes que ya se disponen sobre las distintas habitaciones de hotel.

En base al conjunto de datos con el que se cuenta se buscó desarrollar un algoritmo híbrido que cumpla con los siguientes objetivos:

- Clasificar una imagen dentro de una cadena hotelera.
- Obtener el id de la cadena hotelera de la imagen a clasificar.

Si se logra identificar la cadena hotelera en base a una imagen de una habitación de hotel podría beneficiar a las investigaciones sobre la trata de personas, es por esto por lo que se buscó desarrollar este clasificador de imágenes.

3.3 Entendimiento de los datos

En esta fase de CRISP-DM se realizó la recolección de datos con el objetivo de establecer una relación entre los datos y el planteamiento del problema, familiarizarse con los datos y analizar la calidad de estos.

3.3.1 Recolectar los datos

Los datos que se utilizaron en este proyecto provienen de una aplicación para smartphones llamada *TraffickCam*, esta aplicación tiene el propósito de crear un conjunto de imágenes de habitaciones de hotel que puedan ser utilizadas en investigaciones relacionadas con el tráfico de personas. Actualmente estos datos se encuentran disponibles en la página web *Kaggle* para uso libre de ellos y suelen ser actualizados una vez al año.

3.3.2 Descripción de los datos

Los datos se encuentran divididos en dos carpetas (*train* y *test*) y una tabla *train* con datos sobre las imágenes de la carpeta *train*. La carpeta *train* cuenta con la mayoría de las imágenes, un aproximado de 97 mil imágenes, en la cual las imágenes están divididas en carpetas donde cada carpeta está identificada con un número que representa la cadena de hotel a la cual pertenecen las imágenes guardadas dentro de ella. La carpeta *test* cuenta con tres imágenes las cuales tienen la finalidad de usarse como prueba al momento de desarrollar el algoritmo de clasificación. En la tabla *train* se cuenta con los siguientes campos en cada registro:

- Image: este campo cuenta con una serie de caracteres los cuales representan al nombre de la imagen.
- Chain: Este campo es de tipo numérico y representa el número de cadena al cual pertenece el hotel donde fue tomada la fotografía.
- Hotel_id: Este campo es de tipo numérico y representa el número de identificación a cada hotel dentro de cada cadena de hoteles.
- Timestamp: Este campo contiene la fecha y hora en la cual fue tomada la fotografía.

3.3.3 Exploración de los datos

En [21] se menciona que la exploración de datos implica aplicar pruebas estadísticas que permitan revelar propiedades de los datos y poder generar reportes que sean de utilidad para determinar la consistencia y completitud de los datos. Se decidió realizar un mapa de árbol para identificar con cuántas imágenes se contaban por cada cadena de hotel. En el mapa de árbol (Ilustración 10) se logra apreciar la diferencia que existe en la cantidad de imágenes entre las distintas cadenas de hotel. La cadena que cuenta con más imágenes es la 0, la cual cuenta con un total de 20,259 imágenes, y la cadena que cuenta con menos imágenes es la cadena 56 la cual cuenta con solo 8 imágenes.



Ilustración 10 Mapa de árbol.

En la Ilustración 11 se muestra la cantidad de hoteles con los que cuenta cada cadena de hotelera, se logra apreciar que la cadena 0 tiene más de 1500 hoteles mientras que a partir de la cadena 70 suelen contar con menos de 50.

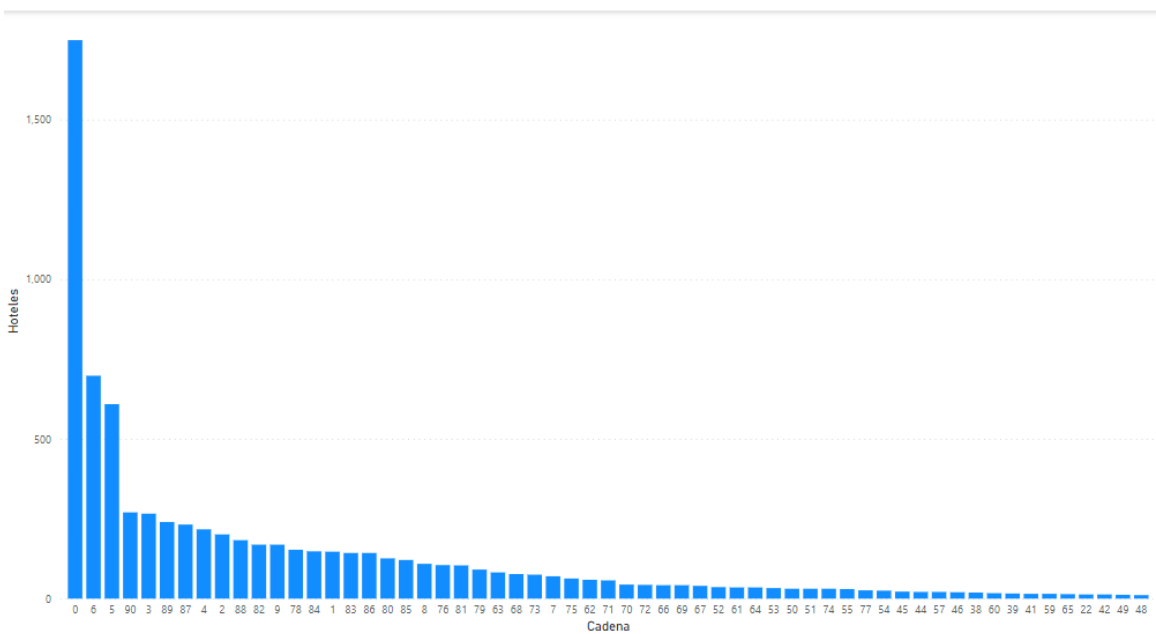


Ilustración 11 Cantidad de hoteles por cadena.

Se decidió explorar las dimensiones de las imágenes para poder visualizar si estas contaban con un solo tamaño, en la Ilustración 12 se puede observar la distribución de dimensiones (en píxeles) de las imágenes de una sola clase, dejando ver que las imágenes no cuentan con un solo tamaño.

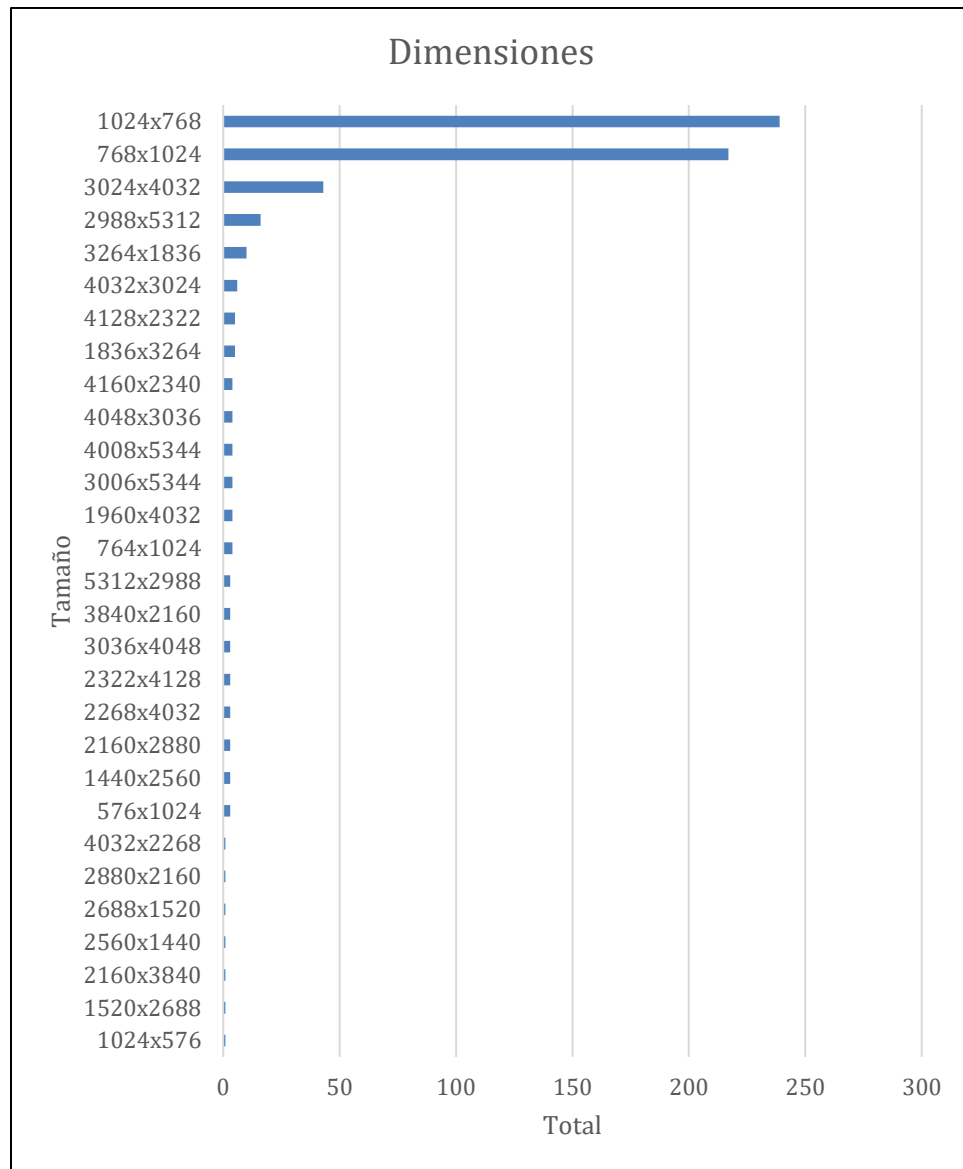


Ilustración 12 Dimensiones de imágenes en una clase.

3.3.4 Verificar calidad de los datos

Una vez realizada la exploración de los datos se descubrió que al tratarse de imágenes obtenidas de una aplicación como *TraffickCam*, estas ya cuentan con un filtro el cual evita encontrar imágenes borrosas, sin luz o sin sentido (como una pared sin objetos, solo el techo, solo el piso, etc.). Es por esto por lo que al revisar el conjunto de datos no se encontraron imágenes dañadas, con errores o que no fueran de utilidad.

3.4 Preparación de los datos

En esta fase se prepararon los datos para que fueran útiles durante el desarrollo del algoritmo. Esto sobrellevó a la selección de los subconjuntos de datos que serían utilizados del *dataset*, si era necesario realizar limpieza sobre ellos para mejorar su calidad, añadir datos de otras fuentes y darles el formato necesario.

En términos de datos, se utilizaron un total de cuatro clases, con un total de 3300 imágenes por clase. De los campos proporcionados por la base de datos solo se utilizó la imagen y el campo llamado *chain* para el desarrollo del algoritmo. Se decidió excluir los datos *Hotel_id* y *Timestamp*, ya que no se consideró de relevancia para la clasificación el identificador de cada hotel dentro de su cadena y la fecha en la que se tomó cada imagen. Se utilizaron cuatro clases debido a la limitante del poder de cómputo ya que no se contó con el suficiente poder de procesamiento para entrenar un modelo con más de 15000 imágenes. Se utilizaron las clases 0, 6, 5 y 90 ya que como se puede observar en la Ilustración 11 estas cadenas cuentan con una mayor cantidad de hoteles, por lo tanto, tienen una mayor distribución de hoteles sobre distintas partes del mundo.

En la construcción del modelo se inició con la parte de preprocesamiento de las imágenes, en la Ilustración 12 se puede observar que no todas las imágenes contaban con un mismo tamaño por lo cual primero se reajustó el tamaño de todas las imágenes a 128 x 128 píxeles. Para finalizar el preprocesamiento de imágenes se normalizaron los datos dividiendo el valor de los píxeles entre 255, esto para conseguir que cada píxel tenga un valor entre 0 y 1, ya que las entradas con valores grandes pueden ocasionar un proceso de aprendizaje más lento.

3.5 Modelado

En esta fase de la metodología se describirá la técnica de modelado utilizada, el plan de prueba y se describirá como se aplicó la técnica sobre los datos para generar el modelo.

3.5.1 Técnica de modelado

En esta sección se describirá la técnica de modelado que se utilizó para probar distintos algoritmos de clasificación de imágenes híbridos. Se decidió modelar cuatro algoritmos con distintas características, para que de esta manera se pueda compararlos entre sí y distinguir cuál funciona mejor al trabajar con imágenes de habitaciones de hotel. Los algoritmos que serán comparados serán cuatro en total, siendo el primero modelo una red neuronal convolucional entrenada desde cero. Los otros tres modelos son algoritmos híbridos que están conformados por tres partes: preprocesamiento, extracción de características y clasificación. En la clasificación es donde se puede diferenciar el desempeño de cada algoritmo. El primer algoritmo híbrido que se desarrolló utiliza el primer modelo para la extracción de características y el modelo SVM (*Support Vector Machine*) para la clasificación. En el segundo algoritmo al igual que el primero se utilizó el primer modelo para la extracción de características y se utilizó el modelo *Random forest* para la clasificación. Para el último algoritmo se decidió utilizar la red pre entrenada *VGG16* para la extracción de características y *Random forest* para la clasificación.

3.5.3 Construcción del modelo

En esta sección se describirán las características de los modelos desarrollados.

3.5.3.1 CNN entrenada desde cero

A continuación, se describirán los parámetros aplicados para la extracción de características al ejecutar los modelos sobre el conjunto de datos descrito anteriormente. En los modelos desarrollados las imágenes pasan a través de cuatro capas de convolución, donde la primera capa recibe las imágenes con un tamaño de 128 x 128 píxeles. En cuestión de filtros la primera capa cuenta con un total de 32 filtros, la segunda capa 64 filtros, la tercera capa 128 filtros y la cuarta capa 256 filtros. Las cuatro capas de convolución cuentan con una activación *relu*, una inicialización de *kernel he_uniform*

Después de las cuatro capas de convolución las imágenes pasarán por una capa densa de 128 neuronas con una activación *relu* y una inicialización de *kernel he_uniform*, para finalmente llegar a una capa densa con 4 neuronas y una activación *softmax* para realizar la clasificación multiclase. Todas las características descritas anteriormente se pueden apreciar en la Ilustración 13.

Layer (type)	Output Shape	Param #
conv2d_input (InputLayer)	[(None, 128, 128, 3)]	0
conv2d (Conv2D)	(None, 128, 128, 32)	896
batch_normalization (Batch Normalization)	(None, 128, 128, 32)	128
conv2d_1 (Conv2D)	(None, 128, 128, 64)	18496
batch_normalization_1 (Batch Normalization)	(None, 128, 128, 64)	256
max_pooling2d (MaxPooling2D)	(None, 64, 64, 64)	0
conv2d_2 (Conv2D)	(None, 64, 64, 128)	73856
batch_normalization_2 (Batch Normalization)	(None, 64, 64, 128)	512
conv2d_3 (Conv2D)	(None, 64, 64, 256)	295168
batch_normalization_3 (Batch Normalization)	(None, 64, 64, 256)	1024
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 256)	0
flatten (Flatten)	(None, 262144)	0
dense (Dense)	(None, 128)	33554560
dense_1 (Dense)	(None, 4)	516

Ilustración 13 Características de la CNN entrenada desde cero.

3.5.3.2 SVM

Este modelo utilizando la misma configuración que la CNN entrenada desde cero para la extracción de características, la diferencia se encuentra en la última capa densa donde se utilizó *Keras* para agregar un regularizador L2 para realizar una clasificación SVM

3.5.3.3 Random Forest

Este modelo también utilizó la configuración de la CNN entrenada desde cero para la extracción de características. la diferencia en este modelo es que utilizó la librería *Sklearn* para realizar una clasificación *Random Forest* en base a las características extraídas.

3.5.3.3 VGG16

Para este modelo se utilizó la CNN pre entrenada VGG16 para realizar la extracción de características. las características de esta red se pueden apreciar en la Ilustración 14. Se realizó el mismo proceso de preprocesamiento de imágenes descrito anteriormente solo que en esta ocasión se les dio una dimensión de 256x256 pixeles a las imágenes. esto debido a que es el tamaño de imágenes que espera recibir en la capa inicial la red VGG16. Se utilizó el modelo VGG16 sin las capas de clasificación ya que solo se busca trabajar con los pesos pre entrenados de la red para obtener la extracción de características. Una vez realizada la extracción de características, se realizó una clasificación utilizando *Random Forest* utilizando la librería *Sklearn*.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 256, 256, 3)]	0
block1_conv1 (Conv2D)	(None, 256, 256, 64)	1792
block1_conv2 (Conv2D)	(None, 256, 256, 64)	36928
block1_pool (MaxPooling2D)	(None, 128, 128, 64)	0
block2_conv1 (Conv2D)	(None, 128, 128, 128)	73856
block2_conv2 (Conv2D)	(None, 128, 128, 128)	147584
block2_pool (MaxPooling2D)	(None, 64, 64, 128)	0
block3_conv1 (Conv2D)	(None, 64, 64, 256)	295168
block3_conv2 (Conv2D)	(None, 64, 64, 256)	590080
block3_conv3 (Conv2D)	(None, 64, 64, 256)	590080
block3_pool (MaxPooling2D)	(None, 32, 32, 256)	0
block4_conv1 (Conv2D)	(None, 32, 32, 512)	1180160
block4_conv2 (Conv2D)	(None, 32, 32, 512)	2359808
block4_conv3 (Conv2D)	(None, 32, 32, 512)	2359808
block4_pool (MaxPooling2D)	(None, 16, 16, 512)	0
block5_conv1 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block5_pool (MaxPooling2D)	(None, 8, 8, 512)	0

Ilustración 14 Características de VGG16

IV. Resultados y Discusiones

En esta sección se describirán los resultados obtenidos al evaluar los modelos desarrollados durante este proyecto.

4.1 Resultados

Se evaluaron en total 4 modelos para poder identificar qué modelo funciona mejor al clasificar habitaciones de hotel. los modelos son los siguientes: una *CNN* (*Convolutional Neuronal Network*) entrenada desde cero utilizando *softmax* (esta es la CNN utilizada en los siguientes modelos para la extracción de características), *CNN* con una clasificación *SVM* (*Support Vector Machine*), *CNN* con una clasificación *RF* (*Random Forest*) y *VGG16* con clasificación *RF*.

Se utilizaron los valores Verdadero Positivo (VP), Verdadero Negativo (VN), Falso Positivo (FP) y Falso Negativo (FN) de la matriz de confusión de cada modelo para calcular las métricas de la Tabla 1.

Se utilizó la Exactitud para poder observar la proporción de predicciones clasificadas correctamente, en la siguiente ecuación se puede observar la manera en la que fue calculada esta métrica.

$$Exactitud = \frac{VP + VN}{VP + VN + FN + FP}$$

Se hizo uso de la Precisión para identificar qué proporción de predicciones positivas fue realmente correcta utilizando la siguiente ecuación.

$$Precision = \frac{VP}{VP + FP}$$

Para identificar la proporción de positivos reales identificados correctamente se utilizó la Sensibilidad la cual fue calculada haciendo uso de la siguiente ecuación.

$$Sensibilidad = \frac{VP}{VP + FN}$$

Para medir la proporción de negativos se usó la Especificidad la cual fue calculada con la siguiente ecuación.

$$Especificidad = \frac{VN}{VN + FP}$$

Por último, se hizo uso de la precisión f1 la cual combina la precisión y la sensibilidad para obtener una media armónica de ambos valores, esta métrica se calculó utilizando la siguiente ecuación.

$$F1 = \frac{2 * (\text{precision} * \text{sensibilidad})}{\text{precision} + \text{sensibilidad}}$$

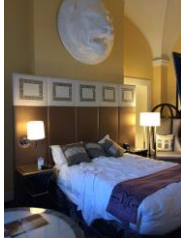

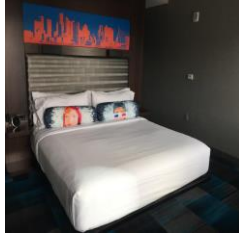

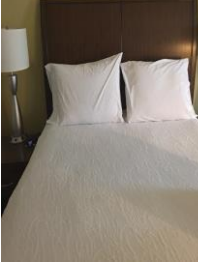



En la Tabla 1 se pueden observar los resultados obtenidos de las métricas que se utilizaron para evaluar a los algoritmos.

Tabla 1 Resultados obtenidos de las métricas utilizadas para la evaluación.

ALGORITMO	EXACTITUD	PRECISION	SENSIBILIDAD	ESPECIFICIDAD	PUNTAJE F1
CNN	66%	33%	33%	78%	33%
CNN + SVM	78.7%	57.5%	57.5%	86%	55.2%
CNN + RF	84%	68%	70.7%	89%	68%
VGG16 + RF	83%	65%	70%	88%	65.7%

En la Tabla 2 se muestran algunos ejemplos al realizar la clasificación con el algoritmo *CNN + RF* el cual fue el que obtuvo mejores resultados en las métricas, se logra apreciar que tipo de imágenes el algoritmo suele obtener buenos resultados al clasificar y con que imágenes tiene problemas.

Tabla 2 Ejemplos de resultados al clasificar.

Clasificado correctamente				
Clasificado incorrectamente				

En la Tabla 3 se puede observar la matriz de confusión del modelo *CNN + RF*, se muestra esta matriz debido a que es la del modelo con el mejor rendimiento.

Tabla 3 Matriz de confusión del modelo CNN + RF.

	CADENA 0	CADENA 6	CADENA 5	CADENA 90
CADENA 0	23%	0%	0%	3%
CADENA 6	8%	13%	3%	3%
CADENA 5	5%	0%	20%	0%
CADENA 90	5%	3%	5%	13%

4.2 Discusiones

En esta sección se discutirán los resultados obtenidos al evaluar los modelos desarrollados, las discusiones estarán centradas principalmente sobre el algoritmo que obtuvo mejores resultados.

En la *Tabla 1* se puede observar que el algoritmo con un mejor rendimiento sería el que cuenta con una clasificación *RF* utilizando la *CNN* entrenada desde cero ya que cuenta con un mayor porcentaje en todas las métricas que se utilizaron para evaluar en comparación con los demás algoritmos. Se obtuvo un 84% de exactitud lo cual se podría considerar aceptable ya que el algoritmo realiza un gran porcentaje de predicciones correctas frente al total. La métrica más alta con la que se cuenta es la especificidad con un 89% lo cual evita en gran medida la obtención de falsos positivos, ayudando a descartar las cadenas hoteleras a las que no pertenece la imagen. En el puntaje f1 se obtuvo un 68%, al obtener esta métrica con tal porcentaje se puede decir que el algoritmo es poco sensible y preciso, esto afecta al algoritmo ya que este pasa por alto una gran parte de los casos positivos.

En la *Tabla 1* se logra apreciar que al momento de clasificar imágenes de habitaciones de hotel se obtienen mejores resultados al utilizar un algoritmo híbrido de clasificación ya que donde se obtuvieron los resultados más bajos fue al realizar la clasificación solo con una *CNN* entrenada desde cero, al agregar una clasificación híbrida a la *CNN* los resultados de las métricas fueron en aumento.

En la Tabla 2 se puede observar que el algoritmo logra clasificar en su mayoría las imágenes en las que se aprecia algún tipo de decoración de la habitación como colores específicos en las paredes, cabeceras de camas, lámparas, alfombras o diseño del piso, muebles, etc. Mientras que las imágenes que no logra clasificar de manera correcta son en las que se logra apreciar muy poco sobre la decoración de la habitación en la imagen o cuentan con una decoración que no cuenta con diseños en paredes o pisos, no cuenta con cuadros en las paredes, ni con diseños en accesorios que van sobre las camas o solo se muestra una parte del baño.

Al observar la Tabla 3 se puede apreciar que las clases que no logra identificar con tanta eficacia el algoritmo son la cadena 6 y la cadena 90, al analizar las imágenes de estas clases se identificó que la clase 6 era la clase que contenía el mayor número de imágenes sobre baños, mientras que la cadena 90 es la clase en la que las habitaciones de hotel cuentan con pocos detalles de decoración.

V. Conclusiones

En esta sección se plasmarán las conclusiones en base a lo descrito en los capítulos anteriores de este proyecto.

El objetivo principal de este proyecto se cumplió al desarrollar múltiples algoritmos de clasificación híbridos. se analizaron los distintos algoritmos de clasificación ya existentes en la literatura y se decidió optar por realizar una extracción de características con una *CNN* (*convolutional neural network*) entrenada desde cero y clasificar con los métodos *RF* (*Random Forest*) y *SVM* (*support vector machine*), cumpliendo de esta manera con los objetivos específicos del proyecto los cuales fueron, “Analizar los distintos algoritmos de clasificación de imágenes utilizados en la literatura”, “Realizar la extracción de características de los datos usando una red neuronal convolucional”, “Entrenar un modelo de *machine learning* en base a las características extraídas” y “Evaluar los modelos utilizando las métricas ya existentes en la literatura”. Al concluir con los resultados se logró observar que es de gran ayuda realizar una clasificación híbrida al trabajar con una clasificación de imágenes sobre habitaciones de hotel ya que al aplicar una clasificación *RF* o *SVM* se obtienen mejores resultados que al utilizar solo una *CNN* entrenada desde cero.

Respondiendo a la pregunta de investigación de este proyecto la cual fue ¿Qué tan efectivo puede llegar a ser un algoritmo híbrido de clasificación de imágenes para el reconocimiento de habitaciones de hoteles?, en base a los resultados se puede concluir que se logró obtener un algoritmo que descarta en un 89% la obtención de falsos positivos. aunque esto es bueno el algoritmo aún se tiene un gran camino por mejorar ya que se obtuvo un porcentaje del 68% en la precisión f1, por lo que si se logra aumentar este porcentaje el algoritmo sería más sensible y preciso haciéndolo más robusto.

Este proyecto podría representar un aporte para futuras investigaciones relacionadas a la clasificación de imágenes de habitaciones de hotel. En los resultados se puede observar que el algoritmo depende de cuan visible es la decoración, por lo cual se podría analizar más a fondo la idea de conservar únicamente las imágenes en las que se logre apreciar una gran parte de la decoración de la habitación, haciendo un *dataset* más pequeño y que probablemente mejore los resultados del algoritmo. Además, también se podría considerar la idea de separar el *dataset* en dos, una parte en donde solo aparezcan imágenes de los

baños y la otra parte con las imágenes del resto de la habitación, ya que en los resultados se logró observar que el algoritmo tenía dificultad al identificar la cadena hotelera mediante la imagen del baño de la habitación, aunque algunas de las imágenes sobre los baños si contuvieran la decoración del hotel. Además de la separación del *dataset*, se podría utilizar este proyecto para futuras investigaciones al desarrollar un algoritmo híbrido utilizando diferentes modelos de clasificación, una *CNN* pre entrenada diferente a *VGG16* o una *CNN* entrenada desde cero con diferentes configuraciones, ya que en la Tabla 1 se logra apreciar que el mejor algoritmo fue el que utilizó la *CNN* entrenada desde cero utilizando *RF*, pero obtuvo unos resultados similares con el algoritmo que utilizó *VGG16* y *RF*, por lo cual también se puede decir que *RF* es mejor trabajando con imágenes de habitaciones de hotel que *SVM*.

Referencias

- [1] M. R. P. Malloch, Human Trafficking : The Complexities of Exploitation, Edinburgh: Edinburgh University Press, 2016.
- [2] A. Paraskevas y M. Brookes, «Human Trafficking in Hotels: An ‘Invisible’ Threat for a», International Journal of Contemporary Hospitality Management, 2018.
- [3] A. Stylianou, X. Hong, S. Maya, B. Jonathan, S. Richard y P. Robert, «Hotels-50K: A Global Hotel Recognition Dataset,» Julio 2019. [En línea]. Available: <https://doi.org/10.1609/aaai.v33i01.3301726>. [Último acceso: 22 agosto 2021].
- [4] T. Rath, «medium.com,» 19 Julio 2018. [En línea]. Available: <https://medium.com/kayak-tech/hotel-image-categorization-with-deep-learning-ffa8429e55b5>. [Último acceso: 20 agosto 2021].
- [5] F. Bozyigit, «A Deep Learning-Based Hotel Image Classifier for Online Travel Agencies,» Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi, İzmir, 2021.
- [6] F. Mihaich, «Aplicacion de redes neuronales en la clasificación de imágenes,» Córdoba, 2014.
- [7] N. A. Dobernack, «Procesamiento de imágenes,» pp. 42-72.
- [8] J. Ramírez y M. Chacón, Redes neuronales artificiales para el procesamiento de imagenes, una revision a la ultima decada, Revista de Ingeniería, Eléctrica, Electrónica y Computación, 2011.
- [9] L. Rouhiainen, Inteligencia artificial, Madrid: Aliente Editorial, 2018.
- [10] P. D. Pusiol, «Redes Convolucionales en Comprensión de Escenas,» Córdoba, 2014.
- [11] F. López-Saca, Clasificación de imágenes usando redes neuronales convolucionales, México, 2019.

- [12] P. y. Digital, «capacitarte.org,» 15 abril 2021. [En línea]. Available: <https://www.capacitarte.org/blog/nota/que-es-y-para-que-sirve-python>. [Último acceso: 16 octubre 2021].
- [13] «Keras.io,» [En línea]. Available: <https://keras.io/>. [Último acceso: 16 octubre 2021].
- [14] «tensorflow.org,» [En línea]. Available: <https://www.tensorflow.org/?hl=es-419>. [Último acceso: 16 octubre 2021].
- [15] «anaconda.com,» [En línea]. Available: <https://www.anaconda.com/products/individual>. [Último acceso: 16 octubre 2021].
- [16] «jupyter.org,» [En línea]. Available: <https://jupyter.org/>. [Último acceso: 16 octubre 2021].
- [17] «aprendeconolaf.es,» 14 mayo 2021. [En línea]. Available: <https://aprendeconalf.es/docencia/python/manual/pandas/>. [Último acceso: 16 octubre 2021].
- [18] «aprendeconolaf.es,» 4 octubre 2020. [En línea]. Available: <https://aprendeconalf.es/docencia/python/manual/matplotlib/>. [Último acceso: 16 octubre 2021].
- [19] 19 enero 2019. [En línea]. Available: <https://sitiobigdata.com/2019/01/19/machine-learning-metrica-clasificacion-parte-3/#>. [Último acceso: 10 septiembre 2021].
- [20] N. S. Chauhan, «datasource.ia,» 02 septiembre 2020. [En línea]. Available: <https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico>. [Último acceso: 11 septiembre 2021].
- [21] I. Gavilán, «ignaciogavilan.com,» 28 junio 2021. [En línea]. Available: <https://ignaciogavilan.com/metodologia-para-machine-learning-i-crisp-dm/>. [Último acceso: 28 septiembre 2021].

- [22] P. Chapman, C. Julian, K. Randy, K. Thomas, R. Thomas y S. R. W. Colin, CRISP-DM 1.0, SPSS, 2000.
- [23] V. Galán Cortina, APLICACIÓN DE LA METODOLOGÍA CRISP-DM A UN PROYECTO DE MINERÍA DE DATOS EN EL ENTORNO UNIVERSITARIO, Madrid, 2015.
- [24] F. Celis, «forbes mexico,» forbes, 31 Enero 2019. [En línea]. Available: <https://www.forbes.com.mx/estos-son-los-miedos-de-los-mexicanos-para-comprar-en-linea/>. [Último acceso: 10 Agosto 2021].
- [25] H. M. Vargas y M. P. Cerna, «eumed.net,» revista de investigación en turismo y desarrollo local, junio 2012. [En línea]. Available: <https://www.eumed.net/rev/turydes/12/mvpc.html>. [Último acceso: 13 Agosto 2021].
- [26] T. Rath, «medium.com,» 19 Julio 2018. [En línea]. Available: <https://medium.com/kayak-tech/hotel-image-categorization-with-deep-learning-ffa8429e55b5>. [Último acceso: 20 agosto 2021].
- [27] T. Page, Prospects for the Design of Electronic Products in Second Life, Journal of Studies in Informatics and Control, 2011.
- [28] «traffickcam.com,» [En línea]. Available: <http://traffickcam.com/about>. [Último acceso: 22 agosto 2021].

Apéndices

Apéndice A

En esta sección se agregará el espacio para poder acceder al código utilizado para el funcionamiento de los algoritmos. El código estará disponible en un repositorio público de GitHub al cual se puede acceder a través del siguiente enlace: <https://github.com/DiegoAlmanza/Clasificacion-de-iman-es-de-habitaciones-de-hotel> .