

Creación de herramienta de aprendizaje automático basada en imágenes satelitales, como insumo para futuros casos de uso relacionados a penetración de paneles solares en Costa Rica

Autor: Carlos Andrés Rodríguez Trigueros
Máster Universitario en Computación.
Inteligencia Artificial.

Docente Guía: Carlos Gaitán Potayos.
Noviembre de 2022.



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-

SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Creación de herramienta de aprendizaje automático, basada en imágenes satelitales, como insumo para futuros casos de uso relacionados a penetración de paneles solares en Costa Rica
Nombre del autor:	Carlos Andrés Rodríguez Trigueros
Nombre del consultor/a:	Carlos Gaitán Potayos
Nombre del PRA:	Carlos Gaitán Potayos
Fecha de entrega (mm/aaaa):	12/2022
Titulación o programa:	Máster Ingeniería en Computación
Área del Trabajo Final:	Inteligencia Artificial
Idioma del trabajo:	Español
Palabras clave	Computer Vision Solar panels Costa Rica
Resumen del Trabajo	
<p>El presente trabajo está basado en la elaboración de una herramienta de aprendizaje automático de tipo supervisado, que permite la detección de paneles solares utilizando imágenes satelitales de Costa Rica, Centroamérica.</p> <p>La intención de esta herramienta es la de facilitar un instrumento que, a través de la detección de paneles solares, posibilite estudios de esta área, donde estos tengan objetivos, tales como: conocer la adaptación de paneles solares en el país ya mencionado o calcular la cantidad de energía producida utilizando proyecciones de Mercator, entre otros.</p>	

Para lograr este trabajo, se creó un conjunto de datos y se procesó el mismo de forma manual, diferenciando los paneles solares de diferentes imágenes que los contuvieran. Así mismo se capturaron imágenes con y sin paneles solares, esto con el objetivo de permitir la generación de modelos de machine learning, basado en una técnica llamada segmentación y otra de clasificación a fin de llevar a cabo la funcionalidad principal de la aplicación, la cual es la detección.

A nivel de resultados, se lograron métricas aceptables a nivel de clasificación, por ende, la aplicación es capaz de discernir si hay o no un papel solar en un área geográfica, y en cuanto a mostrar exactamente dónde se encuentra el panel solar, en términos de alcance del proyecto los resultados obtenidos fueron satisfactorios.

Posteriormente con los modelos generados se implementó una arquitectura software que permitiese a través de un mapa interactivo la detección de paneles solares, arquitectura basada en Google Cloud Platform y que permite fácilmente ajustarse a diferentes necesidades.

Abstract

The work was based on the development of a supervised machine learning tool, which allows the detection of solar panels using satellite images of Costa Rica, Central America.

The intention of this tool is to provide an instrument that, through the detection of solar panels, enables studies in this area, where these, for example, have objectives such as: knowing the adaptation of solar panels in the aforementioned country or calculating the amount of energy produced using Mercator projections, among others.

To achieve this work, a data set was created and manually processed, differentiating the solar panels from different images that contained them. Likewise, images were captured with and without solar panels, this with the aim of allowing the generation of machine learning models, one of a technique called segmentation and another of classification to achieve the main functionality of the application, which is the detection.

At the level of results, quite good metrics were achieved at the classification level, therefore the application is capable of discerning whether there is a solar role in a geographical area, and in terms of showing exactly where the solar

panel is located, they were achieved. Acceptable metrics for the scope of the project, without being excellent, given that the data set used was small.

Once with the generated models, a software architecture was implemented that would allow the detection of solar panels through an interactive map, an architecture based on Google Cloud Platform and that allows it to be easily adjusted to different needs.

Índice

1.	Introducción	1
1.1.	Contexto y justificación del Trabajo	1
1.2.	Objetivos del Trabajo	3
1.3.	Impacto en sostenibilidad, ético-social y de diversidad.....	3
1.4.	Enfoque y método seguido	7
1.5.	Planificación del Trabajo	9
1.6.	Análisis de riesgos.	11
2.	Estado del arte.....	15
2.1	HyperionSolarNet, 2021	15
2.2	Automatic detection and mapping of solar photovoltaic arrays with deep convolutional neural networks in high resolution satellite images, 2020	17
2.3	Automatic Detection of Solar Photovoltaic Arrays in High Resolution Aerial Imagery, 2016.	19
2.4	Conclusiones estado del arte.	22
3.	Materiales y métodos	23
3.1	Materiales utilizados en desarrollo:	23
4.	Resultados	24
4.1	Captura de datos.....	24
4.2	Etiquetado de datos.	25
4.2	Exploración temprana de modelo de segmentación con aumentación de datos	27
4.3	Creación de modelo de aprendizaje automático	30
4.4	Refinamiento de modelo de aprendizaje automático.	30
4.5	Creación de sitio web	34
4.5	Integración de modelo de aprendizaje automático y exploración complementos.....	37
5.	Conclusiones y trabajos futuros	40
6.	Glosario.....	43
7.	Bibliografía	44
8.	Anexos.....	46

1. Introducción

1.1. Contexto y justificación del Trabajo

El presente trabajo se desarrollará en Costa Rica, un país centroamericano de aproximadamente 51.100 km², con una población de alrededor de 5 millones de habitantes, su forma de gobierno corresponde a una democracia y se ha hecho reconocido por ser líder mundial en conservación del medio ambiente. En cuanto a su matriz energética, el siguiente informe menciona:

“Costa Rica es un país destacado internacionalmente por utilizar fuentes de energía renovables para satisfacer casi la totalidad de su demanda interna de electricidad, sin embargo, el consumo de productos derivados del petróleo ha representado alrededor del 60% de toda la energía consumida... En el caso de Costa Rica incluye biomasa (leña, residuos vegetales como bagazo, cascarilla de café y otros) en un 25%, geotermia en un 43%, hidroenergía con 28%, y las energías eólicas, solar y biogás en un 4% de todos los terajulios aprovechables.”
(García, 2016)

Respecto a lo anterior, el uso de petróleo y derivados corresponde al sector transporte que consume 50% de la energía, no obstante, es importante destacar que el sector residencial, servicios, comercial y una cantidad significativa del sector industrial, hacen uso de energía eléctrica producida por fuentes renovables, como energía hidráulica, geotérmica, donde como señala García, la energía solar y eólica como fuentes representan un fragmento pequeño en la generación.

Si bien, el porcentaje es bastante bajo al compararlo con el uso de hidroenergía o geotérmica, el potencial eléctrico en Costa Rica es bastante alto, como menciona un informe de 2006, por Gilmore, generado para Instituto Costarricense de Electricidad (que es la empresa pública más grande de servicios eléctricos) donde se recomienda a esta institución del país ampliar la red solarimétrica. (GILMORE, 2006)

En otro estudio realizado sobre el potencial del uso de la Energía Solar en Costa Rica, creado por el ingeniero Rolando Portilla del ICE (Instituto Costarricense de Electricidad), en la página 3 del mismo, establece que el

potencial de energía derivada del sol en Costa Rica en el 2014 es el siguiente, ver cuadro 1.

Cuadro 1.

Potencia	Energía Anual
576.747 MW	656.195 GW/año

Nota: Tomado de Portilla, 2014

Con relación al potencial de energía anteriormente mencionado es posible afirmar que se puede satisfacer 65 veces la demanda anual de electricidad de toda Costa Rica. Este mismo estudio, establece las zonas geográficas del país y el potencial de energía solar que tiene cada una. (Portilla, 2014)

Por otro lado, en cuanto a costos de instalación de paneles solares y como parte de la contextualización, se denota que el costo de paneles solares en Costa Rica ronda \$7.000 y \$8.000, ya con un sistema de baterías de respaldo puede subir a los \$20.000, lo cual hace que muchos hogares de percentiles bajos de ingresos económicos dichos sistemas sean inaccesibles. Se espera que más adelante en unos años, estos sistemas bajen su valor, debido a los altos precios de los combustibles fósiles, llegando a los paneles solares más accesibles a un segmento mayor de la población y conociendo el potencial solar previamente descrito, será de suma importancia la generación de una herramienta como la planteada en este Trabajo Final de Máster (TFM). (Greenenergy, n.d.)

En cuanto a herramientas similares, no se logró detectar en una búsqueda preliminar herramientas como la que se propone en el país, más, sí, artículos que hablan de cómo realizar detección de paneles solares con aprendizaje automático; en todo caso, se hará una herramienta donde se delimita el alcance a una zona geográfica específica y con ello generar un elemento diferenciador claro; aprovechando las múltiples ventajas que ofrece esta tecnología e investigar qué resultados se obtendrán a partir de un set de datos menor en cantidad al de herramientas similares en el ámbito de la detección de paneles solares.

Finalmente, si bien Costa Rica ya cuenta con una matriz de energía renovable bastante significativa y es ejemplo para muchos países en cuanto a este tema; el uso de combustibles fósiles es predominante para sector transporte,

lograr un cambio en esta dinámica implicara aumentar la matriz energética, por ende, la promoción de una fuente renovable como lo es energía solar , constituye uno de los principales elementos motivadores de este trabajo final de master y que se abordara más a detalle en secciones posteriores.

1.2. Objetivos del Trabajo

1.2.1 Objetivos Generales:

Crear herramienta de aprendizaje automático basada en imágenes satelitales, como insumo para futuros casos de uso relacionados a penetración de paneles solares en Costa Rica.

1.2.3 Objetivos específicos:

- Crear un conjunto de datos basado en imágenes satelitales utilizando Google Maps API.
- Etiquetar un conjunto de datos para identificar paneles solares
- Investigar posibles conjuntos de técnicas y estrategias disponibles sobre las que entrenar modelos de aprendizaje automático.
- Entrenar un modelo de aprendizaje automático sobre conjunto de datos creado.
- Generar un sitio web y servidor que usando un modelo de aprendizaje automático sea capaz de identificar paneles solares.
- Listar posibles casos de uso donde la herramienta a crear sea de utilidad.

1.3. Impacto en sostenibilidad, ético-social y de diversidad

El trabajo final del máster y la aplicación: “Costa Rica, Solar Panel Detector” busca abordar una temática que combina elementos eléctricos con generación de energías renovables. La energía solar es una fuente de energía renovable y limpia que se puede utilizar para generar electricidad. Es importante por una serie de razones, incluyendo sus beneficios ambientales, su abundancia y disponibilidad, así como su sostenibilidad a largo plazo.

1.3.1 Dimensión sostenibilidad.

Impactos positivos.

El proyecto al abordar una aplicación que explora el uso de paneles solares se asocia con el uso y/o promoción de energías renovables, pues detectar paneles solares sobre imágenes satelitales posibilita varios impactos positivos en lo inmediato y en lo futuro. Por ejemplo:

- Se podría utilizar la herramienta con el fin de facilitar un estudio de energía renovable basado en paneles solares y su relación costo-beneficio, por ejemplo: se puede utilizar la herramienta para detectar qué tantos paneles existen en una región al ajustarla para que ejecute la predicción en zonas extensas geográficas, con relación en esta información se daría a conocer qué tantas empresas, hogares o regiones cuentan con paneles solares y proceder con base a esta información realizar un sondeo que permita conocer, a través de entrevistas el resultado a nivel de costo-beneficio de estos usuarios.
- El modelo de segmentación retorna la cantidad de píxeles en los que se ha detectado un potencial panel solar, utilizando esta información, se puede generar proyecciones que complementen un estudio sobre uso de energía solar en diferentes regiones. Para ello, se podría, por ejemplo, calcular la cantidad de energía eléctrica producida en la zona basado en fórmulas que permitan relacionar los píxeles con métricas de distancia para calcular el área.
- El aprendizaje por máquina es un tema que despierta mucho interés, en especial porque se visualiza como un área en crecimiento; además de crear un proyecto que está basado en estudiar un elemento vital para la producción de energía eléctrica solar por sí solo y que actúa como una herramienta de promoción de esta temática y que perfectamente puede ser presentado a diferentes audiencias a fin de promover nuevos casos de uso sobre la herramienta.
- Gobiernos o compañías eléctricas interesados en implementar planes para promover energía eléctrica solar, podrían hacer uso de

esta herramienta para detectar regiones en las cuales ya existen paneles solares y por ende donde podría ser más efectivo llevar a cabo estos planes.

- Compañías eléctricas podrían detectar instalaciones donde existan paneles solares a fin de ajustar la demanda eléctrica o utilidades requeridas para hacer frente a la adopción de esta fuente de energía.

Como se observa en los distintos puntos, los usuarios que se podrían llegar a beneficiar son muy variados, incluye instituciones, gobiernos, estudiantes, investigadores, entre otros.

Impactos negativos

Dado que el proyecto produce una aplicación web, el mismo utilizo diversos recursos computacionales para su generación y disponibilidad, sin embargo, no se midió a nivel de consumo eléctrico cuánto se requirió, ya que el impacto no se consideró como uno altamente negativo, pues se contó desde un inicio con recursos limitados, entre ellos una computadora personal con GPU de 8GB y a nivel de horas de ejecución, no se sobrepasó las 48 horas de funcionamiento agrupando las diferentes pruebas en las que estuvo en máxima carga.

También se utilizaron recursos cloud y en cuanto a la sección de la captura de los datos, se utilizaron una cantidad significativa de horas sin exceder las 24 horas, pero la tarea se ejecutó sobre un servidor pequeño diseñado para este fin (Sagemaker groundtruth),

En cuanto al lanzamiento de la aplicación y mantener esta en funcionamiento, se tomó en cuenta el compromiso sostenible y factores económicos asociados y se utilizan recursos de bajo costo y aquellos componentes que son de más alto consumo funcionan solo sobre demanda.

Además, es importante destacar que el uso de Google Cloud Platform, que fue la plataforma utilizada para esta tarea, posee estadísticas de sostenibilidad importantes, incluyendo el uso de energías renovables y emisión de carbono neutral. (Google, 2022)

1.3.2 Dimensión comportamiento ético y de responsabilidad social

El proyecto, hace uso de imágenes satelitales como principal fuente de información, se utilizan tanto como elemento de entrada para la detección y se utilizaron varias imágenes para la generación de los modelos. El uso de imágenes satelitales tiene un elemento de comportamiento ético y de responsabilidad social pues a nivel mundial la utilización de estas ha generado controversias.

Impactos negativos

El tema de la privacidad es un tema complejo y que con el avance de los años la escala o zoom de las imágenes ha mejorado bastante, al punto de observar personas en la calle desde una imagen capturada en el espacio afectando así temas de privacidad y es por ello que ya países han empezado a crear legislación y han solicitado restricciones como es el caso de Estados Unidos que solicitó no proveer generación de imágenes satelitales a distancias menores a 0.31 metros del suelo. (Coffer, 2020)

En el caso de Costa Rica, el país seleccionado para el trabajo final de máster, luego de una búsqueda a través de internet, no se encontró nada relacionado a legislación al respecto, existe legislación correspondiente a privacidad, pero nunca se especifica nada al respecto sobre imágenes satelitales o abarcar estas en algún artículo de la ley. (LA ASAMBLEA LEGISLATIVA DE LA REPÚBLICA DE COSTA RICA, 2011)

1.3.3 Dimensión diversidad, género y derechos humanos.

Debido a las características del proyecto y al estar enfocado al área de machine learning, no se detectó un impacto negativo o positivo a nivel de la dimensión de este subtema. Dado que la aplicación está enfocada sobre objetos que no tienen en cuenta diversidad, género o derechos humanos. Se reconoce que existe un factor económico importante asociado a la adquisición de paneles solares, pero este factor no es objeto de estudio del trabajo final de máster.

1.4. Enfoque y método seguido

Como punto de inicio se buscó comprender la situación eléctrica a nivel país, revisando generalidades de marco regulatorios y estudios realizados sobre energía solar, posteriormente se desarrollará un mapa interactivo de Costa Rica, posteriormente en este mismo mapa, el usuario podrá identificar paneles solares instalados sobre los techos de viviendas, esto para crear un conjunto de datos.

Una vez obtenido el conjunto de datos, se procedió a hacer una tarea de etiquetado que permite a un modelo de aprendizaje procesar la información y generar los pesos respectivos que logren este “aprendizaje¹” sobre estos datos.

Se logró capturar un mínimo de 120 imágenes con paneles solares identificables los cuales se dividieron cuidadosamente, no obstante, la limitada cantidad de imágenes se debe a la cantidad observada de paneles solares y al esfuerzo en la creación de esta tarea en términos de tiempo. Ver Tabla #1

Tabla #1.

División de conjunto de datos.

Fase	Imágenes	Imágenes con anotaciones
Entrenamiento/Validación	103	103
Prueba o Test	10	10

Nota: Creación propia.

Posterior a este ejercicio de captura, se procedió a crear los elementos programáticos necesarios para una aumentación de datos a fin de aprovechar los recursos al máximo y obtener mejores resultados finales; se hizo uso de librerías especializadas y una vez obtenidos estos subproductos, se procedió a validar si las limitadas imágenes con las que se cuenta serían suficientes para lograr un modelo de segmentación, lo cual se alcanzó, se logró generar modelos de segmentación con bajo nivel de refinamiento de hiper parámetros que resultaron

¹ Cuando se menciona el termino aprendizaje, se hace dado que la técnica emula aprendizaje en cuanto a detección de patrones en base a información previa

exitosos, para posteriormente pasar al refinamiento de los elementos de la red neuronal y parámetros que maximizan el uso de los pocos datos.

Para llegar al modelo final de segmentación, se utilizaron librerías que permitiesen la fácil adaptación de nuevos modelos, una vez obtenido una primer versión satisfactoria, se procedió al refinamiento del modelo para lo cual se utilizó una herramienta de rastreo de experimentos y se realizaron variaciones en la experimentación a fin de obtener un modelo que produjera mejores resultados; en esta sección, no se logró alcanzar resultados perfectos, pero esto debido a la baja cantidad de imágenes y estas no poseían una cantidad significativa de paneles solares.

Posteriormente se procedió a la creación del sitio web, donde se implementó una arquitectura software que se detallará más a continuación en los capítulos correspondientes y alcanzado un sitio web básico, se prosiguió con las tareas que involucraba la exploración de un modelo de clasificación a fin de ayudar al modelo de segmentación a obtener menos resultados falsos.

Este modelo fue bastante satisfactorio de realizar, se encontró código fuente de licencia abierta, que ahorró una cantidad impresionante de tiempo y que fue muy cómodo de adaptar. Las métricas de este modelo fueron bastante altas y con el modelo generado, se crearon variaciones a nivel de código fuente para implementarlo como parte de uno de los productos finales del máster.

Finalmente, se hizo un esfuerzo por mejorar el sitio web a partir de experiencias de usuarios consultados, incluyendo el tutor y se incluyeron instrucciones y un diseño más moderno y amigable.

1.5. Planificación del Trabajo

Tareas desglosadas por objetivos:

- Primer objetivo específico:
 - Identificar áreas potenciales donde puedan existir hogares o empresas con paneles solares.
 - Recolección de coordenadas para su posterior uso en Google Maps API.
- Segundo objetivo específico:
 - Uso de herramienta de etiquetado, por ejemplo: Sagemaker GroundTruth.
 - Uso de técnica de etiquetado ya sea por puntos o cajas dependiendo de resultados de segundo objetivo específico.
- Tercer objetivo específico:
 - Investigación a través de búsqueda en línea, sobre artículos o papers relacionados a detección de objetos usando imágenes satelitales.
 - Investigación sobre amazon rekognition para detección de objetos.
 - Investigar sobre uso de segmentación semántica y Efficient Net.
- Cuarto objetivo específico:
 - Utilización de arquitectura conocida para red neuronal sobre la cual crear de modelo de aprendizaje automático.
 - Afinamiento de modelo y posible re-etiquetado de imágenes basado en resultado para posible aumentación de dataset.
 - utilización de aumentación de conjunto de datos variando propiedades, por ejemplo: variación de colores, orientación, entre otros.
- Quinto objetivo específico:
 - Creación de página web, utilizando Javascript con su librería React.
 - Investigación sobre uso de Amazon Lambda para llamado de modelo de predicción.
 - En caso de Amazon Lambda no ser viable, creación de servidor utilizando NodeJs o Java para hacer lanzamiento de modelo predictivo.

1.5.1 Planificación del Trabajo

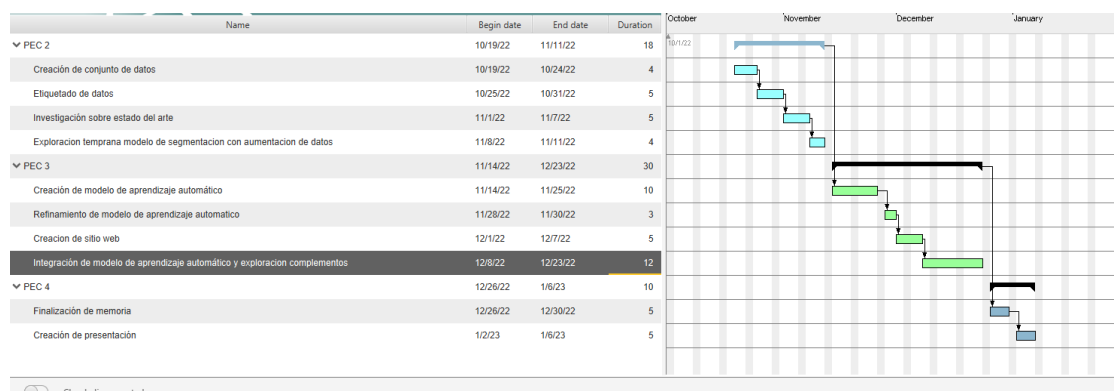
A continuación, se muestra el cronograma de trabajo con el que se trabaja hasta la fecha.

Figura 1.
Cronograma de trabajo.

▼ PEC 2	10/19/22	11/11/22	18
Creación de conjunto de datos	10/19/22	10/24/22	4
Etiquetado de datos	10/25/22	10/31/22	5
Investigación sobre estado del arte	11/1/22	11/7/22	5
Exploracion temprana modelo de segmentacion con aumentacion de datos	11/8/22	11/11/22	4
▼ PEC 3	11/14/22	12/23/22	30
Creación de modelo de aprendizaje automático	11/14/22	11/25/22	10
Refinamiento de modelo de aprendizaje automatico	11/28/22	11/30/22	3
Creacion de sitio web	12/1/22	12/7/22	5
Integración de modelo de aprendizaje automático y exploracion complementos	12/8/22	12/23/22	12
▼ PEC 4	12/26/22	1/6/23	10
Finalización de memoria	12/26/22	12/30/22	5
Creación de presentación	1/2/23	1/6/23	5

Nota: Creación propia.

Figura 2.
Diagrama de Gantt.



Nota: Creación propia.

1.6. Análisis de riesgos.

En la siguiente sección, se han ha generado un listado con los riesgos identificados en el proyecto:

Riesgo	Obtención de métricas de calidad satisfactorias
Factores	Calidad de conjunto de datos recolectado
Descripción	Datos recolectados no suficientes como para producir buenas métricas en el modelo de aprendizaje automático.
Nivel de riesgo	Medio
Mitigación	Modificar variables de modelo de aprendizaje automático, expansión de set de datos de forma artificial.
Resultado:	Se aplicó la mitigación y se generaron más de 50 experimentos donde se modificaron variables del modelo y se llegó a métricas satisfactorias para el alcance del proyecto.

Riesgo	Imágenes satelitales, en país con clima tropical y en desarrollo.
Factores	Popularidad de paneles solares en áreas residenciales
Descripción	Se espera que no se puedan llegar a encontrar muchos paneles solares y que existan muchas imágenes, donde cuestiones relacionadas a economía y clima afecten negativamente al

	modelo de aprendizaje automático, alterando por lo tanto los resultados.
Nivel de riesgo	Medio
Mitigación	Expansión de set de datos de forma artificial, delimitar o reconocer variables que afectan negativamente el modelo, por ende, asumir casos donde predicción no sea exitosa.
Resultado:	Se recolectó y se ha encontrado que no muchas edificaciones cuentan con paneles solares, se ha hecho uso de librerías de aumentación de datos y ha resultado exitoso.

Riesgo	Limitación en cuanto a uso de recursos computacionales
Factores	Recursos computacionales y propiedades de imágenes.
Descripción	Dependiendo de conjunto de datos y calidad de imágenes, tiempo para entrenamiento puede ser significativo y requerir uso de recursos computacionales con GPU, que poseen un costo significativo si se llega a extender por varias horas.
Nivel de riesgo	Medio
Mitigación	Aceptar posibles repercusiones a nivel de métricas o revisar planes gratuitos u opciones para casos académicos. Explorar posibilidad de generar entrenamiento en computadora propia por múltiples días para así disminuir costo.
Resultado:	Se presentó dicho riesgo y se optó por recomendación de profesor guía

	<p>explorar opciones cloud, por ende, se descubrió que el paquete de Google Cloud para machine learning, da bastantes recursos gratuitos y se hizo uso de ellos. Igualmente, recursos computacionales propios ayudaron mucho.</p>
--	---

Riesgo	Posible inclusión de modelo de clasificación.
Factores	Limitación en tiempo sobre alcance proyecto.
Descripción	Estado del arte recomienda la incorporación de modelo de clasificación, aparte de modelo de segmentación.
Nivel de riesgo	Alto
Mitigación	Captura de imágenes sin paneles solares a fin de dar posibilidad a esta opción, adelanto de actividades de cronograma e inclusión de modelo de clasificación para actividades de segundo PEC en caso de contar con disponibilidad de tiempo.
Resultado:	Se capturaron imágenes extras que serán parte de entrega parcial de resultados y se planea iniciar modelo de clasificación como parte de la tarea actualizada llamada: "integración de modelo de aprendizaje automático y exploración de complementos"

1.7. Actualización: Breve resumen de productos obtenidos

- Creación de herramienta para facilitar captura de imágenes basado en parámetros deseados y que permita detección de paneles solares, cuyo objetivo será permitir al usuario identificar un área y proveerle un resultado donde se le muestren en un cuadro los paneles solares de dicha imagen satelital.
 - Herramienta para captura de imágenes satelitales.
 - Métricas y resumen de metodología.
 - Herramienta para detección de paneles solares al seleccionar área en mapa.
- Documentación y instrucciones de uso de herramienta (incluidas en código fuente o son parte de aplicación web).
- Memoria de trabajo, detallando todos los hallazgos.
- Presentación de Power Point para defensa de TFM.

2. Estado del arte

Artículos científicos y páginas web analizados relacionados con detección de paneles solares utilizando algoritmos supervisadas que logran como método.

Trabajos de referencia hallados:

2.1 HyperionSolarNet, 2021

Publicado en 2022 y creado por un grupo de científicos de datos de la Universidad de Berkeley, el artículo describe una aplicación web, que busca de forma interactiva detectar paneles solares usando el servicio de Google Maps. En dicho estudio utilizan un modelo de clasificación y posteriormente de segmentación para llevarlo a cabo, de allí que el TFM propuesto, tome este artículo y aplicación como ejemplo principal para basarse en los hallazgos encontrados y ajustarse en cuanto a tiempo y ámbito, por tanto, HyperionSolarNet cuenta con muchos más recursos.

Según consta el artículo, el set de datos fue creado por ellos, utilizaron localidades de Estados Unidos como región meta y haciendo uso de imágenes con una escala en el servicio de 20, 21, que corresponde a los niveles más granulares o cercanos a tierra y un tamaño de imagen de 600x600 pixeles.

También señalan que se creó un set de datos para la ciudad de Berkeley que fueron usados como evaluación; a nivel de manejo de datos estos fueron divididos para un modelo de clasificación y de segmentación de la siguiente forma:

Figura 3.

Descripción distribución imágenes,

Table 1: Classification dataset				Table 2: Semantic segmentation dataset	
	solar	no_solar	Total		Total
Training	668	1295	1963	Training	668
Validation	168	324	492	Validation	168
Berkeley Test Set	321	1922	2243	Berkeley Test Set	321

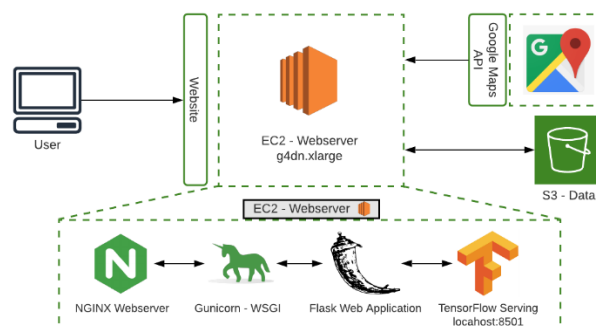
Nota: Tomado de HyperionSolarNet. (Poonam Parhar, HyperionSolarNet Solar Panel Detection from Aerial Images, 2021)

En la tabla en la Figura 3, se denota que para el entrenamiento contaron con un set de datos significativo y que en lo que corresponde al modelo de clasificación, usaron un set de datos desbalanceado donde la cantidad de imágenes que no contienen paneles solares casi llega al doble de los que si posee.

En cuanto al modelo selecto, para clasificación, se utiliza una red neuronal, EfficientNet-B7 donde realizaron variaciones para obtener mejor eficiencia, como por ejemplo cambiar capas, igualmente a nivel de los datos, generación aumentación de datos aplicando a las imágenes rotaciones, variaciones de contraste, recortes, entre otros. La aplicación utilizó un set de datos de 836 anotadas manualmente y a través de proyecciones de Mercator, calculan la cantidad de energía eléctrica generada por los resultados de la segmentación.

En cuanto a los resultados, en lo que se refiere a clasificación logran una precisión media de 0.96 sobre la ciudad de Berkeley y para la segmentación una intersección sobre unión del 0.82; como parte de su sitio web, proveen un ejemplo de la arquitectura que siguieron para generar la aplicación.

Figura 4.
Arquitectura HyperionSolarNet.



Nota: Tomado de página web HyperionSolarNet. (Nathan Nusaputra, 2022)

En la figura anterior, se observa que hacen hosting de su aplicación en un servidor de Amazon y que su inferencia es realizada sobre este mismo servidor. Igualmente utilizan GoogleMapsAPI para navegación y recolección de datos y un S3-Bucket para salvar datos, basándose en esta arquitectura, es que se ha modelado parte del proyecto.

2.2 Automatic detection and mapping of solar photovoltaic arrays with deep convolutional neural networks in high resolution satellite images, 2020

Creado en 2020, bajo autores procedentes de la Universidad de Manchester, UK buscaron estudiar la detección de paneles enfocándose en el uso de redes neuronales en imágenes de alta resolución satelitales. El set de datos consta de 6990 imágenes con paneles fotovoltaicos tomados de Google Earth. La distribución de los datos corresponde a 5873 para ser utilizadas en entrenamiento y 1117 para la tarea de validación y dichas imágenes fueron anotadas manualmente con el uso de polígonos.

A nivel de experimentación, se experimenta con dos tipos de redes neuronales una Mask R-CNN y una SegNet. En cuanto a la Mask R-CNN, A nivel de arquitectura de la red neuronal utilizada consta de dos fases, una donde se detectan los objetos y posteriormente se segmenta la imagen.

Figura 5.

Arquitectura red neuronal

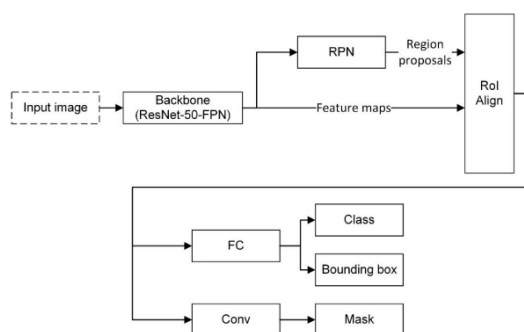


Fig. 2. Overall structure of Mask R-CNN [8]. The bottom boxes are network head where FC stands for fully connected layers and Conv stands for convolutional layers.

Nota: Tomado de introducción de artículo científico, pagina 3. (Kaiji He, 2020)

La figura 5 nos muestra con más detalle cómo funciona esta red neuronal, en la cual en su primera fase una red de tipo ResNET-50-FPN extraerá las características de la imagen para luego pasar a una red llamada “Region Proposal Network” que como su nombre en inglés indica, busca proponer regiones en la

imagen; la segunda fase consta del alineamiento de la región de interés, donde al final se genera una región que indica la localización del objeto.

Y en cuanto a la Segnet, consta de una sola fase que es básicamente la segmentación pues es su especialización y sigue una arquitectura ya definida; a nivel de entrenamiento incluyendo todas sus fases, para ambas redes neuronales se utiliza un GPU Tesla de 16GB, donde en promedio cada imagen tardó sintetizando ambas redes menos de 2 segundos y en cuanto a métricas se presenta la siguiente grafica.

Figura 6.

Curvas de precisión y exhaustividad.

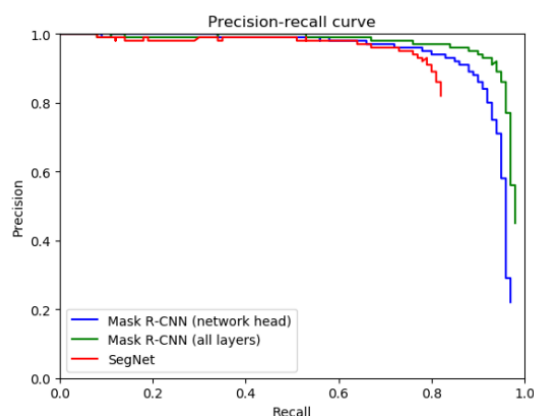


Fig. 5. PR curves of validation results.

Nota: Tomado de resultados de experimentos, pagina 5. (Kaiji He, 2020)

En la figura 6, se denota un comportamiento deseable en lo que corresponde a modelos de clasificación y con un rendimiento no tan superior al de clasificación, pero igualmente bastante significativo en lo que se refiere a segmentación. A nivel de resultados generales y/o conclusiones, el artículo menciona que el detector creado, alcanza en promedio una precisión de 95.66%, aunque no aclaran si esta métrica involucra también la posible precisión de la red de segmentación, en todo caso, se afirma que es totalmente posible crear un detector con red neuronal para paneles solares.

A modo de cierre, se observa el uso de bounding box, una técnica que se considera para el proyecto y con base a los resultados de las fases, se irá determinando que tan posible es o no esta técnica, de forma prematura si se puede afirmar que en HyperionSolarNet, se hacen cálculos sobre posible energía

producida en base a la segmentación donde el modelo previo, uno de clasificación no genera un “bounding box” ² como si lo hace este artículo, por lo tanto, se explorará si es necesario esta fase.

2.3 Automatic Detection of Solar Photovoltaic Arrays in High Resolution Aerial Imagery, 2016.

Trabajo referenciado exclusivamente por su aproximación al problema utilizando una técnica poco convencional, pero aún muy relevante por su técnica que es utilizada hoy en día en miles de problemas de inteligencia en computadora, este artículo publicado en 2016 y creado por instituciones de la Universidad de Duke, Durham, nos describe un método donde utilizaron un set de datos públicos de imágenes satelitales de la ciudad de Fresno, California, cubriendo un área total de 135km cuadrados. Todas las imágenes capturadas del 2013.

Figura 7.

Distribución de conjunto de datos y anotaciones.

TABLE 1
SUMMARY OF FRESNO COLOR ORTHOIMAGERY DATASET

Designation	Area of Imagery	Number of PV Annotations
Fresno Training	90 km ²	1780
Fresno Testing	45 km ²	1014

Nota: Tomado de distribución de conjunto de datos, pagina 3. (Jordan M. Malof, 2016)

El set completo consiste en 601 imágenes, de 5000x5000 pixeles, dichas imágenes fueran anotadas para así utilizar un modelo de tipo supervisado logrando generar 2700 anotaciones y en términos de anotaciones nos presentan una distribución de casi 2 a 1 en lo que respecta a lo utilizado en entrenamiento y testeo como se muestra en la figura 7. En cuanto al método utilizado para la detección, el proyecto se basa en una arquitectura en la cual no se detecta una red neuronal per se, quizá por el año de creación.

² Bounding box, se le conoce a una forma geométrica, usualmente un rectángulo que encapsula una región de interés.

Figura 8.
Progreso de fases aplicadas a imagen.

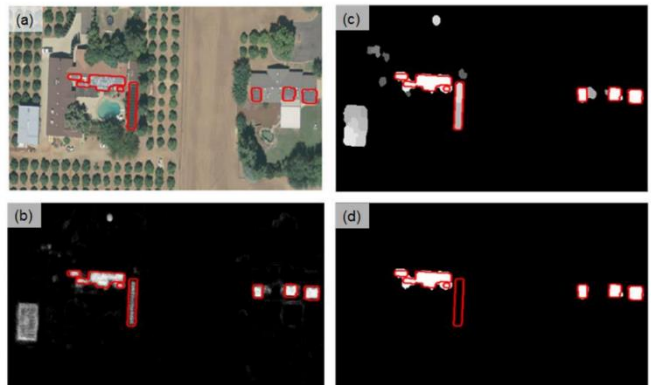
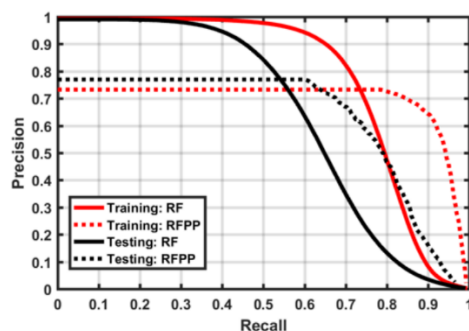


Fig. 6. Example output of the rooftop PV detection algorithm after several of the major processing steps. Four different images are shown, (a)-(d), and each image shows the human PV annotations in red. (a) is the original RGB image. (b) is the confidence map output from the Random Forest classifier, without post-processing; brighter pixels indicate higher confidence. (c) is the confidence map after post-processing. (d) shows the objects detected after the object detection stage of processing. Given the detection rate and false alarm rate employed in this example, the detector correctly removes all of the false alarms, while losing one of the true panel regions.

Nota: Tomado de descripción de algoritmo y fases, pagina 6, los recuadros rojos corresponden a la anotación real y las áreas blancas a la predicción. (Jordan M. Malof, 2016)

A modo ilustrativo, el artículo nos presenta una serie de imágenes, como se muestra en la Figura 8, que corresponde a la mejora en términos de detección donde B) corresponde al resultado obtenido del Random Forest y la imagen d) corresponde al post procesamiento. Al aplicar el post procesamiento se denota un ejercicio de balance, pues se ganan y se pierden pixeles. En la figura posterior (Figura 8) se entiende más esto a nivel de datos.

Figura 9.
Curva de precisión y exhaustividad.



Nota: RF corresponde a Random Forest y RFPP, corresponde a Random Forest con post procesamiento. (Jordan M. Malof, 2016)

En las curvas de la figura 9, al aplicar la capa de post procesamiento se pierde en la exhaustividad claramente, esto se puede argumentar si es o no correcto, pues es un ejercicio donde se decide si se quiere un número mayor de positivos falsos versus falsos negativos, por el contexto de paneles solares y el artículo nos indica que establecer el balance dependerá de la necesidad de la aplicación, por ende no nos presentan métricas de precisión más comunes, como intersección de la curva o dice score.

Figura 10.

Curva de exhaustividad y precisión en base a Índice de Jaccard.

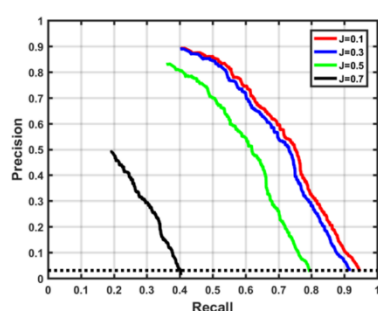


Fig. 10. PR curves for the object-based performance of the rooftop PV detector on the Fresno Testing dataset. Each PR curve corresponds to a different setting of the Jaccard index, J , during scoring. The left-most point of the curves represents the performance when classifying every object with confidence of one (i.e., the maximum RF output) to a detection. With object-based scoring, the detectors are not guaranteed to place objects over all true PV array locations, and indeed, none of the detectors reach $R = 1$.

Nota: Tomado de pagina 9. (Jordan M. Malof, 2016)

Lo que se desprende del análisis visual gráfico de la figura 10, el índice de jaccard al ser más alto o estricto tanto en recall como en precisión se ven altamente afectados, pues estamos estableciendo cuándo tomaremos como positivo un píxel o no, por ende, un índice de 0.7 resulta demasiado estricto, mientras que uno de 0.1 nos podría estar arrojando resultados con muchos pixeles que no corresponde a las anotaciones reales. A modo de análisis final, se decidió incluir este artículo porque random forest es una modelo que permite entender de forma estadística las decisiones que toma el algoritmo para la clasificación de los pixeles y que podría resultar interesante en análisis de imágenes satelitales a nivel científico o de argumentación.

2.4 Conclusiones estado del arte.

En los 3 artículos analizados se detecta que el uso de aprendizaje por computadora para detección de paneles solares es lo que corresponde claramente al estado del arte, donde la presencia de redes neuronales es la técnica actual para utilizar en este caso de uso.

En cuanto a arquitecturas, los artículos más recientes nos muestran una clara inclinación a la utilización de modelos pre entrenados y que requieren de aumentación de datos. También se detecta una presencia de dos fases en lo que corresponde al entrenamiento e inferencia/predicción, la primera una de clasificación y otra de segmentación o detección de objetos.

En cuanto a recursos, los artículos fueron creados por equipos especializados en universidades con GPU's potentes y con set de datos significativos, donde el trabajo final de máster tendrá menos recursos y por ende las expectativas no son las mismas, pero sí demostrar lo que se puede lograr con menos recursos bajo objetivos similares.

Otro punto encontrado es que el uso de la arquitectura de HyperionSolarNet, es similar a la arquitectura que se idealizó en un inicio, por lo tanto, el alineamiento es reafirmante en la implementación y se espera incluso utilizar herramientas que son conocidas y pueden llegar a escalar mejor en lo que respecta a cuestiones de experiencia de usuario como lo es React y Typescript con uso de Material UI como herramienta de componentes grafica.

El último artículo científico mencionado en el estado del arte arroja resultados y una técnica bastante interesante, pero no se espera hacer uso de Random Forest como herramienta de solución, esto debido a que dista de aproximaciones más recientes, debido a la alta popularidad que han obtenido las redes neuronales en los últimos años.

3. Materiales y métodos

3.1 Materiales utilizados en desarrollo:

- Hardware, elementos más importantes:
 - Procesador: AMD Ryzen 7 5800X 8-Core Processor, 3801 Mhz, 8 Core(s), 16 Logical Processor(s)
 - GPU: NVIDIA GeForce RTX 3060 Ti 8GB.
 - Memoria RAM: 16GB.
- Software, más relevantes:
 - Sistema operativo: Windows 11.
 - NodeJs V16.16.0
 - React, Typescript
 - Python 3.9
 - Jupyter Notebooks
 - Pytorch
 - Albumentations
- Servicios en la nube:
 - Weights & Biases (WanDB)
 - AWS
 - Sagemaker
 - Google Cloud Platform.
 - Google Colab.
 - Tesla T4 - 14GB, paquete gratuito.
 - 2 Cloud Run
 - Uso de Container Registry
 - 1 App Engine

4. Resultados

4.1 Captura de datos.

El objetivo propuesto para esta sección fue la de capturar imágenes satelitales que contuvieran paneles solares y obtener al menos 100 de estas donde se evidencie claramente la presencia de estos objetos sobre superficies, siendo específicos un total de 120 imágenes.

Para lograr este cometido, se exploraron alternativas que permitieran este ejercicio de forma sencilla y rápida, pero no se encontraron herramientas ya creadas con este fin, por lo tanto, se creó una, haciendo uso de aplicaciones que facilitasen entonces la interacción con Google Maps API, librería mencionada en nuestros objetivos y fuente principal de los datos. Esta herramienta propia fue creada con React y Typescript, ambos lenguajes considerados estado del arte en lo que respecta a aplicaciones web modernas y altamente escalables y hace uso de la librería React-Google-Maps-API. (Alexey Lyakhov, 2022).

En cuanto al uso de la librería, se ajustaron coordenadas para apuntar a una región dentro de Costa Rica y se configuraron parámetros que mostraran de forma inicial una interfaz lista para la rápida captura de imágenes. Se conectaron eventos específicos como lo es interacciones de mouse, que permiten un ajuste correcto del zoom, que corresponde al nivel planificado con la que se consumirán las imágenes y otra acción de mouse para salvar la imagen.

En cuando a salvar la imagen, se notó que dicha funcionalidad, requiere de un permiso especial por parte de Google, por lo tanto, se creó una cuenta en Google Cloud, y se crearon dos llaves API, una que permite la ejecución de la librería de Google Maps API previamente referenciada y otra llave API que permite generar imágenes con base a parámetros específicos, los cuales son facilitados por la aplicación web creada.

Las imágenes salvadas cuentan con una escala según Google Maps API de tipo 2, que ofrecen más pixelaje y con una resolución de 1280x1280, estas configuraciones fueron seleccionadas pues se quiso un nivel de detalle muy alto,

ya que no se cuenta con los recursos de tiempo extenso para hacer un conjunto de datos que exceda los miles, como sí sucede con artículos del estado del arte, pues se quiere alcanzar el mayor provecho a los paneles solares detectados.

4.2 Etiquetado de datos.

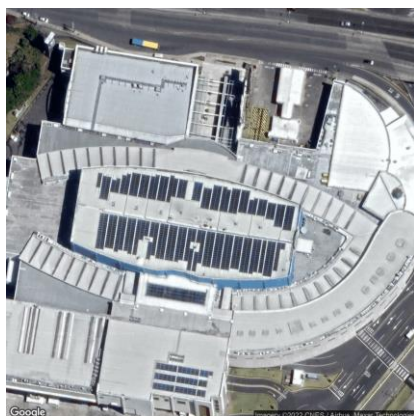
Conseguidas todas las imágenes con paneles solares identificables, se procedió a la tarea más manual en lo que respecta a este trabajo final de máster y vital para cualquier tipo de modelo automático de aprendizaje supervisado, en el proceso que se le conoce como etiquetado.

Para la realización de esta tarea se decidió utilizar Sagemaker, Groundtruth, de la empresa Amazon Web Services (AWS), esta decisión meramente justificada por herramientas previas laborales con el uso de esta. El primer paso realizado fue el de transferir las imágenes capturadas a un contenedor de objetos de AWS (S3 Bucket). Si bien, el guardado puede ir conectado directamente al contenedor, la planeación no permitía un desvío tan significativo, pues involucra adelantar uno de los pasos finales; por ende, una transferencia manual fue aceptable para la tarea.

Una vez ubicados en el lugar adecuado, se generó una tarea de etiquetado, donde se optó en una primera fase por una tarea de tipo detección de objetos con bounding box, pero al avanzar en las imágenes, se decidió detener esta aproximación de forma prematura.

Figura 11.

Muestra imagen satelital con paneles solares.



Nota: Creación propia, 2022.

En la figura 11 y muchas otras imágenes, se encontró que los paneles solares, cuentan con formas muy irregulares al contar estos con una cercanía de centímetros unos de otros, crear un bounding box genera dos efectos:

1. El bounding box creado abarca otros paneles solares, por ende, se da mucho solapamiento, pues hay que recordar que se crean rectángulos, por lo tanto, se tiene poco nivel de margen e inclusive cuesta ajustar el rectángulo.
2. El bounding box creado abarca muchos paneles solares y regiones de superficie sin paneles solares.

Se optó, con base en las observaciones realizadas, realizar un etiquetado para segmentación en lugar de detección de objetos, esto como riesgo asumido y conociendo que existen técnicas para reutilizar anotaciones de segmentación y convertirlas en bounding box.

Figura 12.
Sagemaker GroundTruth.



Nota: Creación propia, 2022.

La herramienta de la Figura 12, ground truth, es bastante fácil de usar, simplemente basta con generar puntos en las esquinas de los paneles solares e ir generando la forma deseada, una vez obtenida la forma deseada se da click en "Submit" y la herramienta pasa a la siguiente imagen, hasta completar todas las imágenes.

Lo que se puede mencionar en cuanto a aprendizaje de este ejercicio, y en específico con esta herramienta, es que salva las imágenes en una localización poco común, por ende, fue necesario utilizar un script para mover las anotaciones

a una carpeta y nombre más adecuado, se adjunta como parte del repositorio, el script utilizado.

4.2 Exploración temprana de modelo de segmentación con aumentación de datos

Para esta fase, se exploran en primer lugar aproximaciones a la aumentación de datos y se detectó una librería llamada Albumentations, con la que no se tenía familiaridad. Esta librería permite programáticamente generar un “pipeline” que transforma imágenes con base a una serie de parámetros, luego esta librería provee las imágenes en sus diferentes variaciones que pueden llegar a ser consumidas por un modelo de aprendizaje automático. Se provee un notebook en Jupyter en Anexo #1, llamado data_explorer, que muestra el funcionamiento de dicha librería como evidencia la figura 13.

Figura 13.

Aumentación de imágenes.



Nota: Creación propia, 2022.

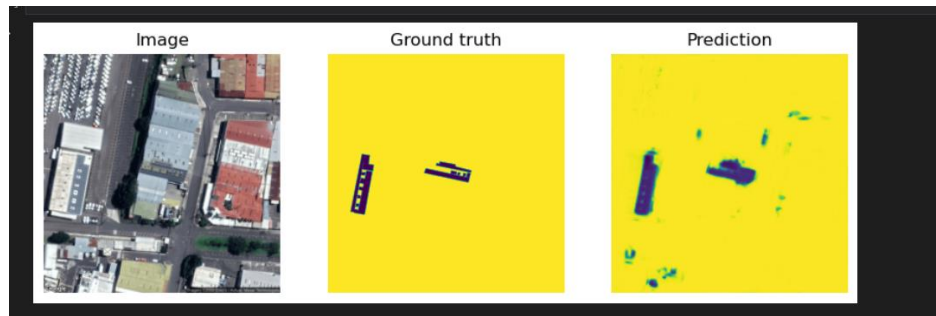
Para la exploración temprana de modelo de segmentación, se espera seguir las instrucciones del artículo “Semantic segmentation data labeling and model training using Amazon SageMaker” de Kara Yang. (Yang, 2022), al inicio se utilizó como guía para extraer las anotaciones lo cual resultó un poco diferente, pues el de datos contaba con nombres un tanto distintos, pero se hicieron los ajustes del caso. Los problemas surgieron cuando se iba a abordar la sección de entrenamiento, pues, todo esto se realizó en un Sagemaker Notebook, que es básicamente una versión de Jupyter Server que te da acceso a instancias o servidores donde entrenar el modelo, similar a Google Colab, pero de pago. Al final de la sección, se encuentra un subcapítulo titulado “Problemas encontrados” que detalla porqué se decide a partir del entrenamiento, abandonar la plataforma de AWS para este trabajo.

El entorno de programación cambió, como parte de acciones de remediación, y se habilitó una computadora personal como herramienta de programación y entrenamiento con la desventaja de que todo será entrenado bajo el sistema operativo Windows por limitaciones de la computadora, sin embargo, no debería significar un problema mayor, pues la arquitectura final será en la nube y en la instalación se evaluará el uso de Docker o documentación como método de portabilidad.

Para el modelo de segmentación temprana, se utilizó una librería que permite el uso de modelos pre-entrenados, llamada “Segmentation Models” (Iakubovskii, 2022). Esta librería está adaptada para Tensorflow y para Pytorch y por experiencias previas, se optó por Pytorch, principalmente por su manejo de versiones que tiende a ser un poco más amigable que Tensorflow. Una de las razones más importantes de la creación temprana del modelo, fue la de comprobar que el set de datos fuese decente en términos de cantidad para poder generar predicciones, aspecto que se logró, pues a nivel visual se observan buenos resultados pese a las limitantes antes mencionadas que afectan sobre todo la creación del set de datos.

Figura 14.

Predicción temprana modelo de segmentación, entrenado con



Nota: Creación propia, 2022

En la figura 14, se obtuvo con un modelo de precisión de 0.98 en intersección de la curva sobre un set de datos de prueba compuesto por 9 imágenes, únicamente la imagen a nivel visual y no arroja resultados excelentes pese a la métrica reflejar algo muy significativo, pero se debe de tomar en cuenta que la mayoría de los píxeles eran área sin paneles solares, por lo cual el área de los paneles es muy reducida.

4.2.1 Problemas encontrados

Sagemaker y el artículo previamente referenciado, hacen uso de instancias con recursos computacionales altos, aspecto que es en cierto modo esperable por el tipo de problema al tratarse de imágenes, aun así, se abordó el riesgo, desde una perspectiva de facilitar los recursos económicos para lo mismo.

Sin embargo, las cuentas de AWS de tipo personal vienen altamente restringidas, por lo cual no tuve acceso temprano a ninguna instancia con GPU, fue necesario crear un tiquete en soporte para solicitar una instancia con GPU. Se procede a realizar la solicitud con la sorpresa de que es declinada, pues no se ha hecho uso suficiente de la cuenta, ver Anexo #2, más adelante ya pasados los días se daría acceso, pero no se hizo uso de ella, por el riesgo que implicó y desvió de expectativa.

Otro aspecto importante para hacer notar, es que los precios de instancias de GPU son significativos, el costo promedio de la instancia mencionada en el

artículo es de 3,825 USD por hora, para el día 14/11/2022, pero se contemplaba utilizar una instancia de tipo ml.g4dn.xlarge, con un costo por hora de 0,7364 USD.

Últimamente, todo esto conllevó obviamente a tomar las previsiones del caso y se descartó continuar con Sagemaker para evitar complicaciones similares y utilizar recursos locales y posteriormente como se presenta en otros capítulos recursos de Google Cloud Platform.

4.3 Creación de modelo de aprendizaje automático

Para este capítulo, dado que la tarea anterior dio buenos resultados, la labor del cronograma se centró en utilizar los hallazgos obtenidos en la primera exploración donde la mayor parte del tiempo se utilizó en robustecer el código y prepararlo para su eventual implementación en el sitio web, así como generar un código fuente que facilitara ejecutar acciones de refinamiento accesibles y fáciles de rastrear. Para ello, se implementó la solución de Weight and Biases (<https://wandb.ai/>), en la cual se generaron dos proyectos, uno para un modelo de segmentación que fue el foco del capítulo y posteriormente se generaría un segundo modelo; uno de clasificación.

Una de las ventajas principales de esta herramienta es que optimiza todo lo relativo al análisis del modelo y de los datos, ahorrando las tareas relacionadas a visualización. En el siguiente capítulo se abordan algunas de las figuras relacionadas con este punto. Otra de las cuestiones que se abordó, fue el tema de portabilidad, para ello la librería Pytorch provee más de una forma de salvar el modelo, por lo cual se optó por guardar el diccionario de datos y utilizarlo en conjunto con la red neuronal creada.

4.4 Refinamiento de modelo de aprendizaje automático.

El abordaje de este punto se dio casi de forma paralela con la creación del modelo de aprendizaje, pero para efectos del cronograma dado que mucho del refinamiento ocurrió sin intervención manual, sino más bien a través de la ejecución de experimentos que tardan horas en entrenarse, se decidió no modificar el cronograma, pues el esfuerzo real se dio en la cantidad de tiempo propuesto.

En lo relativo al refinamiento y variados entrenamientos se modificaron múltiples métricas, las principales siendo:

- Variación de épocas: Es un parámetro bastante frecuente de modificar en tareas relacionadas a machine learning, básicamente se modificó varias veces, supervisando el progreso del parámetro loss, se crearon épocas en exceso, pero esto no tuvo nunca un impacto negativo en ningún aspecto, más que el de consumir recursos computacionales de forma innecesaria, sin tener un impacto significativo o importante.
- Variación de batch-size: Este fue un parámetro bastante problemático, uno de los mayores retos se centró en poder incrementar este parámetro sin exceder los límites de memoria de la GPU con la que se contaba, inclusive en conjunto con el parámetro de resolución, se abordó el uso de Google Colab para ejecutar combinaciones más complejas. Al final se optó por batch-sizes pequeños pues no se encontró una mejora al aumentarlos. En algunos casos inclusive por el batch-size siendo de 1, el algoritmo “Stochastic gradient descent” jugó un rol importante, y se analizaron las ventajas y desventajas que este tipo de algoritmo involucrado; se detectó que no es lo más óptimo, pues provoca que el entrenamiento sea más lento y que presente variaciones significativas durante el entrenamiento, pero se tomó en cuenta como variables principales el test_daset_ioc y test_per_image_iou y las pruebas realizadas con batch-size, mostraron mejores resultados visuales por ende se tomaron las desventajas como unas aceptables para el trabajo. (Brownlee, 2017)
- Variación en tamaño de imagen: Fue el parámetro más significativo para el modelo, se denoto que cuanto más pixelaje posee las imágenes, la mayoría de las modelos obtienen mejores resultados, por ende, se hicieron muchas variaciones de este parámetro a fin de ajustarlo a las capacidades del GPU en términos de RAM, al final se encontró que 960 o 1280 en anchura y altura que es la resolución ideal para los modelos de segmentación probados.
- Variación de arquitectura: A fin de aprovechar recursos existentes, se optó por utilizar redes neuronales de arquitecturas conocidas y pre entrenadas, principalmente se exploraron las redes: mobilenet, efficientnet y por último resnet. Todas estas redes neuronales se detectaron en el estado del arte al ser las más utilizadas, donde todas tienen rendimientos bastante

- Variación de threshold de probabilidad: A fin de determinar que píxel es o no es un píxel sin panel solar, se ajusta el rango de probabilidad, siendo más o menos laxo en esta métrica, y se detectó que con un threshold de 0.7, los paneles solares eran visualmente confirmados de forma más precisa.
- Variación de porcentaje de entrenamiento y validación: Este punto se tuvo que explorar pues el conjunto de datos es bastante reducido, 120 imágenes para un modelo de segmentación trae consigo limitantes en las métricas que se pueden lograr, por ende se buscó diferentes combinaciones, conociendo de previo que lo ideal es 80% utilizado para entrenamiento, debido a la necesidad de ajustar el modelo a fin de que este se ajustara lo máximo posible se optó por utilizar 70% de los datos de entrenamiento/validación (alrededor de 70 imágenes) y 30% de validación.

Reporte de experimentos WanDB, sin filtros



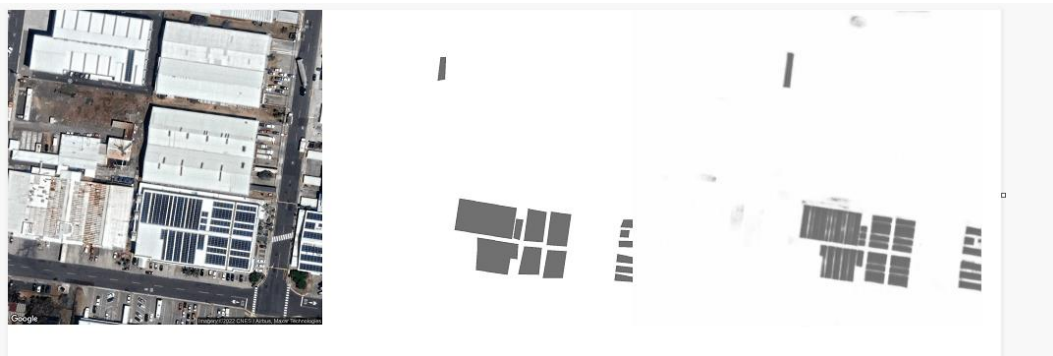
El reporte presente en la Figura 15, es uno de los reportes que se incorporó al sitio web, es una interfaz interactiva que muestra una serie de visualizaciones para observar los diferentes experimentos realizados, que a la creación de esta minuta se acerca a los 50 experimentos, donde se intentaron muchas combinaciones a fin de sacarle el máximo provecho a la baja cantidad de

imágenes. A nivel de métricas, el modelo de segmentación presento métricas muy elevadas, debido a que a las imágenes poseen pocos pixeles con paneles solares, por lo tanto, tiende a predecir de forma correcta cuando no existen paneles solares con mayor facilidad y las variaciones entre experimentos son muy bajas tanto así que los mejores experimentos, poseen métricas que se diferencian en decimales.

Una de las mayores lecciones que se observó en el ejercicio de refinamiento, se da a través de la figura 16:

Figura 16.

Imagen sin modificar vs mascara generada vs predicción



Nota: Creación propia.

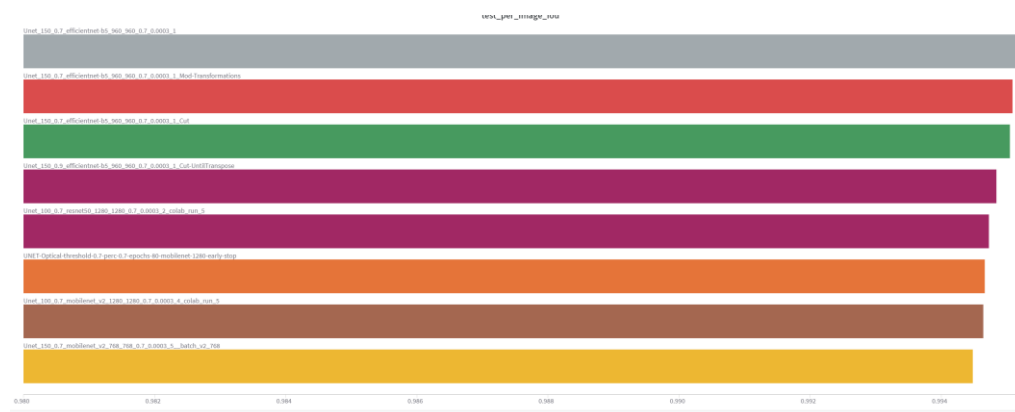
En la imagen anterior, se apreció que la predicción realmente se ajusta mejor que la máscara creada a través del etiquetado; se puede afirmar que la máscara creada en etiquetado puede ser considerada de menor calidad que la predicción en la imagen de la derecha, este fallo se debió a que es muy subjetivo determinar la diferencia entre un arreglo de paneles solares vs un panel solar individual.

Un elemento que pudo haber ayudado fue el haber sido el ser más minucioso en la captura con la clara afectación en que la labor de etiquetado hubiese sido aún más costosa o bien, implementar una tercer etiqueta llamada arreglo de paneles solares.

Pese a ello, los resultados se mantienen satisfactorios; igualmente en el refinamiento, se eliminaron ciertas imágenes, que se determinó, no eran lo suficiente claras al ojo humano.

Figura 17.

Diferencia área sobre la curva, en los mejores experimentos.



Nota: Creación propia, tomado de WanDB.

En la figura 17, se observa un rango que comprende de los 0.980 a los 0.994 correspondientes al área sobre la curva calculado sobre imágenes, el rango es sumamente bajo, reflejo de lo anteriormente expuesto. Para determinar cuál modelo escoger, se tomaron los 10 modelos mas altos, se visualizaron los resultados y a modo de comparación contra la máscara real se determinó cuál es la más apropiada; para esta tarea WanDB posee una interfaz que facilita esta labor, por ejemplo: mantiene record de los resultados, muestra imágenes de predicciones, entre otros.

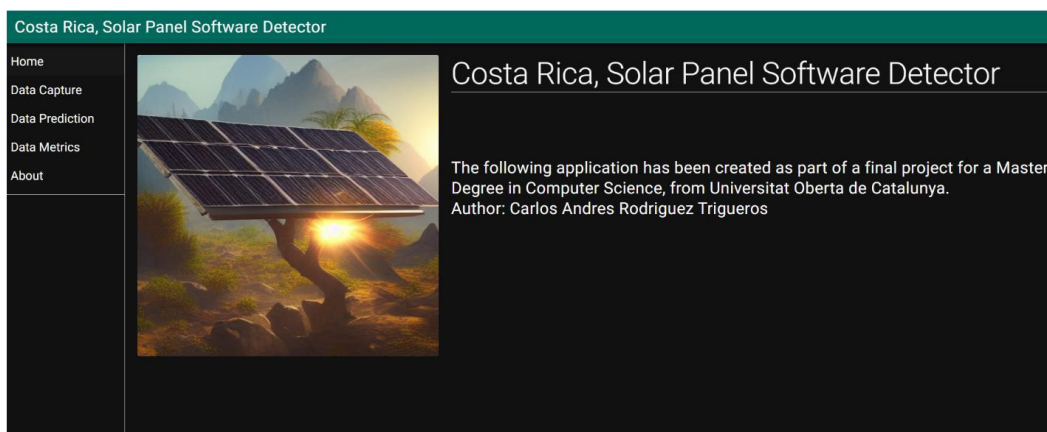
Se eligió un modelo cuyas características son: Arquitectura EfficientNet, entrenado con imágenes a resolución de 960 pixeles en anchura y altura, con un threshold de 0.7 en cuanto a aceptación para que sea o no pixel y 70 imágenes de entrenamiento y 30 de validación.

4.5 Creación de sitio web

Para la creación del sitio web, se optó por utilizar React con Typescript y la librería Material UI. Dichas tecnologías son altamente utilizadas en la industria y son aptas para expandir en caso de nuevos requerimientos. Se produjeron dos versiones, donde la primera sirvió de prototipo para la integración del modelo de segmentación y que contaba con una interfaz sencilla.

Figura 18.

Primera versión sitio web.

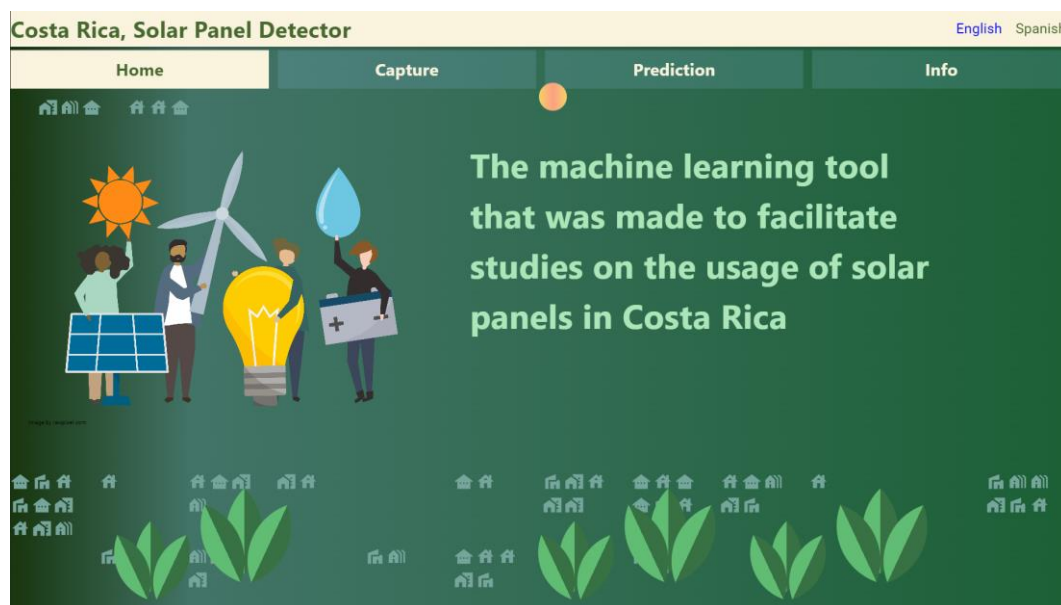


Nota: Creación propia.

Este sitio de la figura 18, no se adaptaba bien a diferentes resoluciones por lo tanto se optó por mejorarlo posteriormente, además crear un sitio que fuera más amigable en términos de UI/UX igualmente, al considerar mejor paletas de colores, tipografía y organización del sitio web.

Figura 19.

Versión final sitio web para Desktop.



Nota: Creación propia.

En la figura 19, se muestra un sitio web con un diseño más complejo a nivel visual y más llamativo, donde el elemento ecológico y de energías verdes es destacado. Igualmente se agregaron elementos de CSS y animaciones, donde se incluye un elemento que simula un sol al acceder a la página y esto se logra a través de un elemento de CSS llamado keyframes.

Figura 20.

Versión final sitio web, versión móvil.



Nota: Creación propia.

En la figura 20 por otro lado se observa que el sitio web, se ajusta a diferentes resoluciones, incluyendo versiones móviles, sin llegar a una optimización perfecta a raíz de alcance del proyecto, pero permite igualmente ejecutar todas las funciones, con la desventaja que se pierde un poco de texto y se le informa al usuario que fue creado con la idea de ser usado en PC.

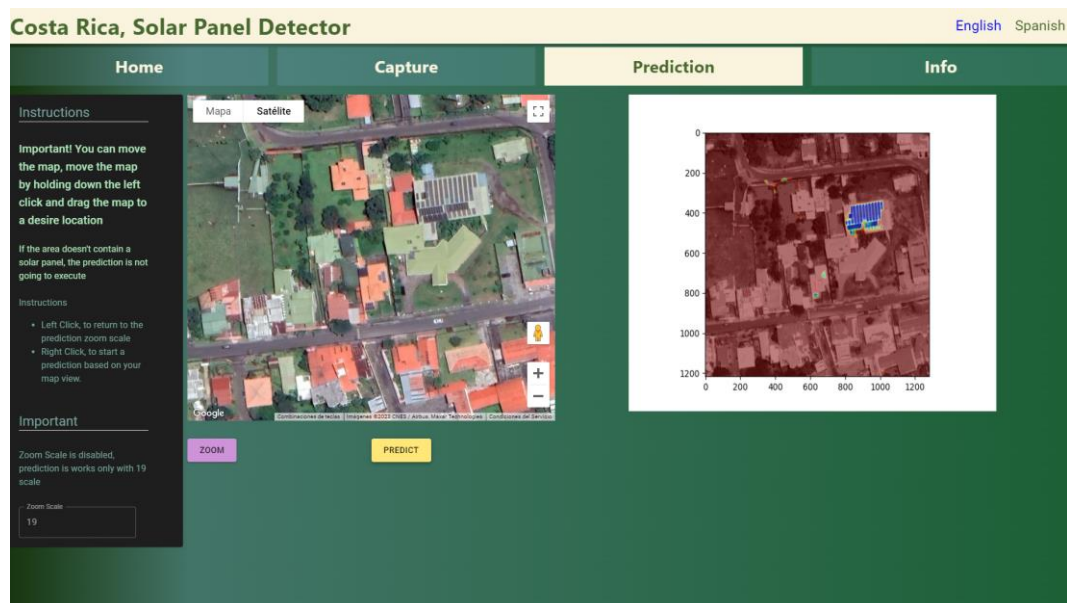
A nivel de diseño “responsive”, se utilizaron las capacidades de la librería Material UI, que permite detectar la resolución y ajustar los diferentes elementos de acuerdo con la pantalla que se utiliza para navegar en el sitio web. Otro elemento visualmente llamativo se encuentra en las casas que se observan en la página principal, pues estas son generadas en un arreglo único, cada vez que consulta la página web.

4.5 Integración de modelo de aprendizaje automático y exploración complementos.

El sitio web cuenta con dos secciones, una que permite la captura de datos, tema que se aborda en un capítulo anterior y que permite la descarga de imágenes y otra página llamada “Prediction” que posee la implementación del modelo de segmentación y de clasificación, para esta página se presenta un mapa de Google Maps, que permite navegar y al darle click derecho o presionar el botón predict, y al realizar esta acción genera una llamada al modelo de aprendizaje automático, donde con base al resultado del modelo de clasificación, el modelo de segmentación será generado.

Figura 21.

Página de predicción.



Nota: Creación propia.

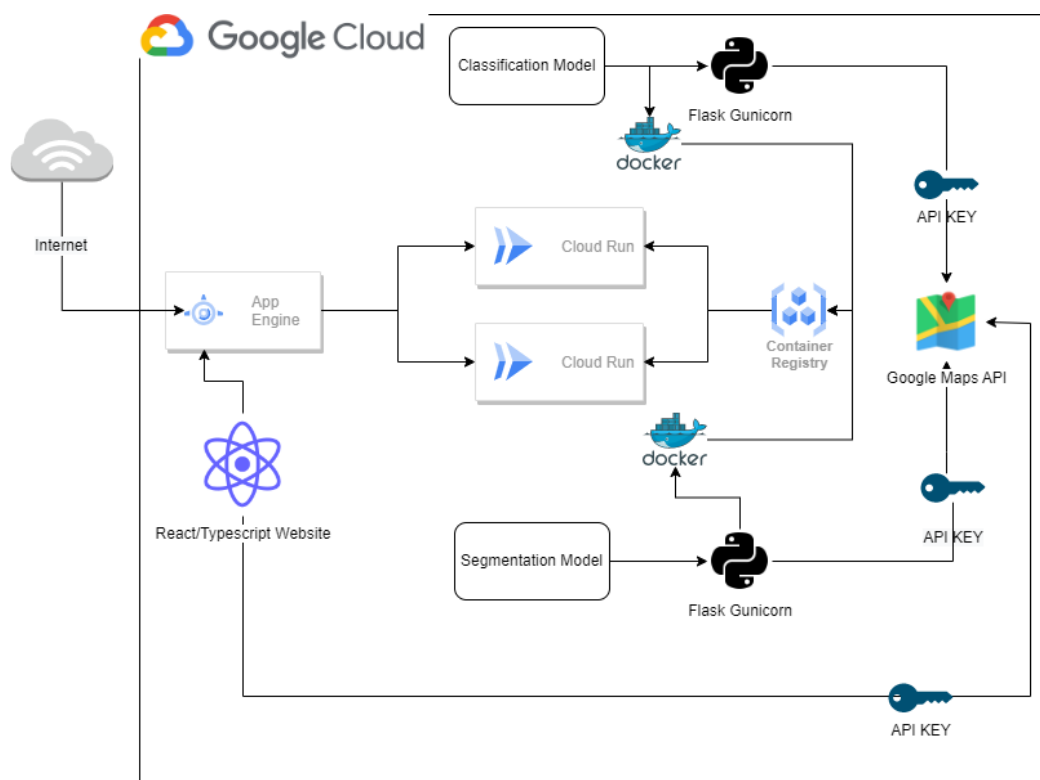
En la figura 21, se observa como el modelo de segmentación fue capaz de detectar el panel zonal del área del mapa seleccionada, donde lo azul representa el panel solar y lo rojo, el área del mapa en el cual no se detectó uno. Para lograr esta visualización, se utilizó la librería Matplotlib, que permitió la interpolación de la imagen.

Dentro la página de “Capture” y “Predict” se incorporaron en ambas una sección de instrucciones para facilitar el uso de la página web. Igualmente se agregó una página de “Info” que contiene los reportes generados de WanDB que puedan servir para exponer los diferentes experimentos realizados.

Uno de los elementos más importantes de la creación del sitio web y altamente relacionado con la integración del modelo automático consiste en la implementación de una arquitectura software que permita mostrar las capacidades de lo desarrollado en el trabajo final de máster. Para poder acceder al sitio, el Anexo #4 presenta la URL disponible para la consulta y utilización de la aplicación.

Figura 22.

Arquitectura software.



Nota: Creación propia, utilizando Drawio.

En la figura 22, se observan varios elementos, en primer lugar que la aplicación funciona sobre Google Cloud Platform, en ella se ha configurado una cuenta de servicio asociada a mi propia cuenta de usuario, no se consideró crear una compartida, pues contiene tarjeta de crédito y el esfuerzo por crear una cuenta segura a ser compartida con evaluadores, sería uno significativo y que nunca se contempló como parte del alcance, en todo caso el código contiene

documentación que explica cómo se realiza el lanzamiento de la aplicación, incluyendo referencias a las guías utilizadas para el desarrollo de la misma.

Entre los componentes se tiene:

- App Engine: En este servicio se ubica la aplicación de React/Typescript que hace uso de la librería Material UI que facilita elementos visuales y diseño que se puede ajustar a diferentes resoluciones. Otras de las librerías utilizadas son React Google Maps (<https://www.npmjs.com/package/@react-google-maps/api>), la cual es pieza clave para navegación y obtención de coordenadas geográficas, así como ajustes sobre el mapa que permiten obtener el mapa en su versión satelital y no de navegación per se (mapa gris con calles).
- Cloud Run: Existen dos funciones, una que se encarga del modelo de segmentación y otra del modelo de clasificación, ambas funcionan con un servidor de Flask – Gunicorn y poseen una única llamada de tipo REST API. Estos servidores, son dockerizados y el archivo resultante (imagen de Docker) es cargado al Container Registry, desde el cual se crean las funciones de Cloud Run. Cloud Run ofrece múltiples opciones, se ajusta parámetros de cantidad máxima de funciones que trabaja de forma paralela, cantidad de tiempo en espera antes de fallar un request, entre otros.
- Se hace uso de API Key durante todo el proceso, esto debido a que el servicio de Google Maps requiere de una y cobra sobre request realizados sobre mapas cuando estos se ejecutan desde sitios web y en especial cuando se requiere descargar las imágenes satelitales.

En cuanto a la sección de exploración de complementos, como se anticipaba y se halló en el estado del arte y como se mencionó varias veces anteriormente, se generó un modelo de clasificación, para este, se utilizó código de un repositorio de Github con licencia MIT, creado por Takumi Karasawa (Karasawa, 2022), este repositorio permitió generar un modelo inicial en cuestión de minutos y con pequeñas modificaciones a nivel de código se logró un modelo de clasificación con una precisión del 93% al clasificar imágenes con y sin paneles solares. Para el modelo de clasificación se contó con la siguiente distribución de datos:

Tabla #3

Distribución de datos.

Categoría	Imagen con Panel	Imagen sin Panel
Entrenamiento	76	379
Validación	27	24
Test	9	18

Nota: Creación propia

En la tabla 3, se observa cómo se contó con una cantidad más significativa de datos, en comparación con el modelo de segmentación, esto se debe a que la labor de generar este conjunto resulta mucho más sencilla, pues no requiere de la tarea de etiquetado. Es importante también denotar que, pese a que existió un capítulo de captura de datos, la captura de datos se realizó para esta sección también, pero fue un proceso mucho más sencillo, dado que como se observó anteriormente, no existen muchos paneles solares en las zonas cubiertas.

En cuanto a los resultados obtenidos fueron también rastreados utilizando la herramienta WanDB, donde solo hicieron falta 6 experimentos para elegir un modelo de clasificación, este utiliza la red neuronal resnet50, pero hubo uno muy cercano utilizando efficientnet-b0, de ahí que se optó por resnet, pues, se halló que era un modelo más rápido en predicción, donde la diferencia se midió en segundos de diferencia (menor a 10 segundos).

5. Conclusiones y trabajos futuros

El trabajo realizado produjo una cantidad de aprendizaje importante que involucran múltiples áreas de la ingeniería de software, pese a que el área de este es machine learning. A nivel de alcance, se plantearon objetivos generales muy ambiciosos que conllevaron a la creación final de un producto de inicio a fin con elementos bases provenientes de modelos de aprendizaje automático. Las áreas abordadas incluyen subáreas de la ingeniería software conocidas como: Machine Learning, Full stack development, DevOps, UI/UX, entre otras, sin llegar a un nivel de perfección en cada una de ellas.

Para la sección de captura de datos, el proceso involucró la creación de un ambiente de etiquetado, por ende se inició con una herramienta basada en React (la primera versión del sitio web), la cual permitió la captura de las imágenes a utilizar, posteriormente se procedió con el etiquetado, ambas tareas dando resultados inmediatos positivos, no obstante sí se halló que la elección de Sagemaker ground truth fue excesiva, pues extraer las máscaras de las imágenes no es un proceso sencillo y se tuvieron que crear scripts para extraer los resultados.

Por lo tanto, una de las recomendaciones en este aspecto para trabajos futuros, es reconsiderar la herramienta de etiquetado y solo utilizar Sagemaker en caso de que el conjunto de datos que se ambiciona tomar involucre miles de imágenes.

A nivel de resultados, se obtuvieron dos modelos de aprendizaje automático ambos utilizando PyTorch, uno de ellos es uno de clasificación, donde a partir de un set de datos de prueba se eligió un modelo con una precisión de modelo de 93% de efectividad detectando paneles solares, esta métrica se obtuvo con un set de datos que consto de 18 imágenes sin presencia de paneles solares y 9 imágenes con presencia de paneles solares, el modelo seleccionado está basado en un efficientnet y bastó con 6 experimentos, donde se cambiaron variables para llegar a una métrica satisfactoria.

En cuanto al modelo de segmentación que fue uno de los objetivos específicos principales; el set de datos representó un elemento muy significativo; se conocía desde un inicio que este elemento traería riesgos al proyecto por el reducido tamaño (120 imágenes), pero parte de las mitigaciones consistían en aceptar las métricas que se obtuviesen y así se hizo, en especial porque el modelo

de clasificación, por sí solo, ya llega a satisfacer el objetivo general, igualmente es importante aclarar que el modelo de segmentación posee métricas aceptables y que se asemejan a resultados encontrados en el estado del arte, donde se observó exactamente lo mismo, en cuanto al modelo de clasificación siendo superior al modelo de segmentación, pero por la naturaleza de ambos, donde el último resulta más complejo a nivel de resultados.

En cuanto a las métricas del modelo de segmentación, se calculan con base a cantidad de píxeles que se predijeron de forma correcta, donde el tipo de modelo es uno de segmentación binaria (es o no es píxel de panel). Se halló que las imágenes, al tener una resolución tan alta o lejana al suelo impacta en el modelo de segmentación, en la imagen la cantidad de píxeles que poseen paneles solares son bajos, haciendo que las imágenes estén altamente desbalanceadas, también se descubrió que arreglos de paneles solares y paneles solares individuales se tuvieron que haber tomado más en cuenta a la hora del etiquetado, pues afectan en precisión al notar líneas en medio de paneles, que potencialmente afectan precisión.

Pese a todo ello, el área sobre la curva obtenida es de 0.995 en el modelo de segmentación, métrica obtenida a partir de un set de datos de solo 9 imágenes de resolución de 960 x 960 en altura y anchura, obviamente los resultados no son estadísticamente representativos, y a nivel visual se observa elementos que no son paneles solares, representados como que lo fueran, pero en términos de alcance de proyectos, tiempo y recursos los resultados fueron impresionantes, aspecto que demuestra el gran avance de las redes neuronales, pues bastaron 100 imágenes con paneles solares para llegar a una herramienta que se puede afirmar es capaz de detectar paneles solares.

Para el análisis de las métricas, este se dio a nivel de decimales y se dependió mucho de validaciones visuales, considerando que las métricas podían ser aceptadas como parte del análisis de riesgo en cualquier momento. El modelo de segmentación elegido fue basado en uno que utiliza la arquitectura de red neuronal llamada “EfficientNet-b5”, el mismo resultó tener las mejores métricas de precisión pese a ser un poco más lento, pero este último factor fue solventado con una buena planificación; a nivel de predicción resulta lento, pero al asignar mejores configuraciones en la plataforma donde se implementa, se determinó que

este factor se puede mitigar; en cuanto al modelo de clasificación, se eligió uno basado en ResNet-50 basado en los resultados de los experimentos.

A nivel de creación del sitio web, el uso de Google Cloud Platform fue bastante positivo, los tutoriales que se siguieron fueron precisos y el proceso sencillo de seguir y entender, igualmente la cantidad de posibilidades que se ofrecen para implementar soluciones facilita ajustar la aplicación para escalar en caso de ser necesario y acelerar la predicción. Los elementos elegidos para la aplicación fueron App Engine y Cloud Run principalmente, ambos son de bajo costo/consumo.

En general, la metodología propuesta se ajustó bastante a lo planificado, se realizaron cambios en el primer avance del proyecto a fin de incluir la incorporación de un modelo de clasificación lo cual resultó de bastante beneficio para el producto final y que vino a complementar bastante el objetivo general planteado del proyecto.

Para futuras líneas de trabajo, se propone expandir la arquitectura a fin de agregar nuevas funciones que permitan, por ejemplo, procesar áreas extensas del mapa. Así mismo, se propone expandir el set de datos, utilizando para ello inclusive, la misma herramienta, pues la detección de imágenes con paneles al utilizar el modelo de clasificación podría simplificar la detección de estos para luego proceder a tareas manuales de etiquetado.

Como líneas futuras recomendadas, se propone la introducción de proyecciones de Mercator que agregarían funcionalidad a la aplicación y permitirá cálculos eléctricos potenciales, también la inclusión de un servicio que permita la detección de paneles solares en zonas geográficas específicas en un área mas extensa.

6. Glosario

- Machine learning: traducción en inglés de la palabra aprendizaje automático o aprendizaje por máquina.
- Aprendizaje supervisado: Técnica de aprendizaje automático en la cual se le provee a la solución, información previa donde se le indica de forma categórica o cuantitativa, ejemplos de soluciones para que la computadora los utilice para generar nuevas soluciones ante elementos desconocidos de la misma clase.
- Épocas: O también conocido como epochs más comúnmente por la predominancia del inglés. Se refiere a la cantidad de ciclos que el modelo de aprendizaje automático usara para entrenarse una y otra vez con el set de datos de entrenamiento.
- Hiperparámetros: Se refiere a las configuraciones que se pueden ajustar dentro de una modelo de aprendizaje automático a fin de controlar la forma en que el algoritmo funciona.
- Docker: Herramienta que permite portabilidad al encapsular aplicaciones y sus sistemas operativos.
- EfficientNet, MobileNet, ResNet: Se refiere a arquitecturas de redes neuronales ampliamente utilizadas y que permiten el entrenamiento de algoritmos de aprendizaje automático sobre redes ya previamente entrenadas.
- Intersección sobre la curva: Se refiere a una métrica utilizada a nivel estadístico, que, en términos de uso del documento, nos indica que tantos píxeles corresponden a los píxeles reales que queremos encontrar.
- Google Cloud Platform, Amazon Web Services: Se refiere a plataformas web, mundialmente conocidas por proveer servicios digitales que facilita tareas diversas a nivel computacional.

7. Bibliografía

- Alexey Lyakhov, U. K. (2022). *React Google Maps Api Style Guide*.
Obtenido de <https://react-google-maps-api-docs.netlify.app/>
- Brownlee, J. (2017). *machinelearningmastery*. Obtenido de
<https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/>
- García, D. Z. (Octubre de 2016). *Matriz Energética de Costa Rica*. Obtenido de <https://library.fes.de/pdf-files/bueros/fesamcentral/12979.pdf>.
- GILMORE, M. J. (Noviembre de 2006). *ESTUDIO DEL POTENCIAL SOLAR EN COSTA RICA*. Obtenido de <http://www.acesolar.org/wp-content/uploads/2016/03/INFORME-FINAL-CONSULTORIA-SOLAR.pdf>
- Greenenergy. (s.f.). *Paneles solares para casas y el precio*. Obtenido de <https://greenenergy.cr/paneles-solares-para-casas-y-el-precio/#:~:text=Un%20sistema%20convencional%20interconectado%20a%20la%20red%20para%20una%20casa,entre%20los%2050%2D60mil%20colone>
- Iakubovskii, P. (2022). *Segmentation Models*. Obtenido de https://github.com/qubvel/segmentation_models
- Jordan M. Malof, K. B. (2016). *Automatic Detection of Solar Photovoltaic Arrays in High Resolution Aerial Imagery*. Obtenido de <https://arxiv.org/ftp/arxiv/papers/1607/1607.06029.pdf>
- Kaiji He, L. Z. (2020). *Automatic detection and mapping of solar photovoltaic arrays with deep convolutional neural networks in high resolution satellite images*. Obtenido de https://www.research.manchester.ac.uk/portal/files/178629759/PV_CNN.pdf
- Karasawa, T. (2022). *Github - pytorch-image-classification*. Obtenido de <https://github.com/karasawatakumi/pytorch-image-classification>
- LA ASAMBLEA LEGISLATIVA DE LA REPÚBLICA DE COSTA RICA. (07 de 07 de 2011). *Ley de Protección de la Persona frente al tratamiento de sus datos personales*. Obtenido de http://www.pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm_texto_completo.aspx?param1=NRTC&nValor1=1&nValor2=70975&nValor3=85989

- Nathan Nusaputra, P. P. (2022). *HyperionSolarNet*. Obtenido de <https://groups.ischool.berkeley.edu/HyperionSolarNet/>
- Poonam Parhar, R. S. (2021). *HyperionSolarNet Solar Panel Detection from Aerial Images*. Obtenido de <https://arxiv.org/pdf/2201.02107.pdf>
- Portilla, R. (2014). *Energía solar para generación eléctrica: un recurso abundante pero con limitaciones*. Obtenido de https://www.ambientico.una.ac.cr/wp-content/uploads/tainacan-items/5/26190/250-251_29-35.pdf
- Yang, K. (28 de 06 de 2022). *Semantic segmentation data labeling and model training using Amazon SageMaker*. Obtenido de <https://aws.amazon.com/es/blogs/machine-learning/semantic-segmentation-data-labeling-and-model-training-using-amazon-sagemaker/>

8. Anexos

Anexo 1: Repositorio código fuente:

<https://github.com/carlosan1708/MasterTFM>

Anexo 2: Imagen denegación de instancia de entrenamiento de AWS, en primera solicitud.

Hi there,

After an extensive review, the SageMaker team have denied your limit increase request. Meaning, as of now I'm unable to process the limit increase you requested for ml.g4dn.2xlarge for spot training job usage in the Northern Virginia region at this time.

The team have noted that the reason for this and why we are unable to approve your increase request in particular, is because your account has minimal usage up to this point. We will need you to first utilize your current instance capacity before granting such a request.

Quotas help decrease the likelihood of large bills for customers. After we have a broader window of usage on your account to review, we can reassess any request.

Here are some recommendations in case you'd like request a limit increase in the future:

- Use your services about 90% of usage in that way we can request a limit increase.
- You can wait for the next billing cycle then we can request again the limit increase.
- You can select resources that are not much expensive and scale down your usage.
- Change the credit or debit card on file. Pre-paid cards are not recommended.
- Take advantage of current resources in the mentioned regions and later on attempt to use a different area.

Should you have any further question, feel free to reach out to me and I will address your concerns ASAP.

We value your feedback. Please share your experience by rating this and other correspondences in the AWS Support Center. You can rate a correspondence by selecting the stars in the top right corner of the correspondence.

Best regards,

Anexo 3:

Reportes de métricas, en sitio web que facilita visualizaciones:

<https://wandb.ai/carlosan1708/Segmentation-TFM/reports/Summary-Solar-Panel-Segmentation--VmlldzoyOTk3NTk0?accessToken=429cugi5ex8ojjuiewxn8w613mfngraz5j7het6wahbacksx29ya7tqzr1741w46>

Anexo 4: URL de sitio web.

<https://costa-rica-solar.uc.r.appspot.com/>

