

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**DETERMINACIÓN Y SEGMENTACIÓN DEL NIVEL
SOCIO-ECONÓMICO A PARTIR DEL ANÁLISIS AUTOMÁTICO
DE IMÁGENES DE SATÉLITE**

Tesis para obtener el título profesional de Ingeniero Informático

AUTOR:

Carlos Alberto Sosa Pezo

ASESOR:

Dr. César Armando Beltrán Castañón

Lima, Febrero, 2022

Resumen

Este proyecto tiene como objetivo desarrollar un modelo de análisis de imágenes satelitales basado en redes neuronales profundas para poder identificar de forma automática el nivel socioeconómico de zonas urbanas, a través de la utilización de las imágenes obtenidas mediante el satélite estatal peruano PerúSat-1, las cuales fueron provistas al proyecto por la Comisión Nacional de Investigación y Desarrollo Aeroespacial (CONIDA). De esta manera, se buscó proveer una fuente alternativa y complementaria a los métodos manuales de recolección de datos demográficos para los casos en los que esta metodología tradicional no resulta conveniente de ejecutar por factores de tiempo y presupuesto.

Para ello, el proyecto inicia con la conformación del corpus de datos utilizados para el entrenamiento del modelo de análisis de imágenes satelitales. Este conjunto de datos incorpora, además de imágenes satelitales de PerúSat-1 de la ciudad de Lima capturadas entre 2016 y 2020, una serie de etiquetas que indican el nivel de ingresos de cada manzana en dicha ciudad. Estas etiquetas son el resultado de un estudio elaborado por el Instituto Nacional de Estadística e Informática en el año 2019, tomando como base los datos obtenidos durante el Censo Nacional de Población y Vivienda de 2017. En el caso de las imágenes satelitales, este proceso consiste en la composición en imágenes de cuatro bandas (rojo, azul, verde e infrarrojo) de alta resolución y su posterior división en bloques cuadrados de 512 píxeles en cada dimensión; mientras que las etiquetas censales son organizadas en una base de datos relacional, de forma que puedan ser accedidas por el *script* para la integración de ambas fuentes de datos. Finalmente, se asigna a cada bloque de 512 x 512 píxeles una máscara con las etiquetas censales correspondientes a la zona capturada.

Luego, se define un protocolo de preprocesamiento de los datos, en el que normalizan los parches de las imágenes satelitales, además de incrementar el tamaño del conjunto de datos, de modo que sean adecuados para el entrenamiento de los modelos desarrollados. Respecto a estos, se dispone de dos modelos de redes neuronales convolucionales para el análisis socioeconómico de las imágenes satelitales: el primero, encargado de segmentar semánticamente los techos observados en las imágenes; y el segundo que, recibiendo como entradas la imagen satelital original y la máscara de techos generada por el modelo anterior, detecta el nivel socioeconómico de cada manzana en dicha imagen. Así, se obtiene como resultado final una máscara de bits en la que se indica el nivel socioeconómico de cada manzana observada en la imagen satelital.

Finalmente, para facilitar su acceso hacia el usuario final, se desarrolla una plataforma web, compuesta por una interfaz gráfica y una interfaz de programación de aplicaciones, que permite seleccionar, con la ayuda de un mapa en la interfaz gráfica, imágenes satelitales para que sean procesadas por los modelos de redes neuronales y puedan ser las máscaras generadas visualizadas en conjunto con la imagen original.

Tema FCI

FACULTAD DE
**CIENCIAS E
INGENIERÍA**
ESPECIALIDAD DE
INGENIERÍA INFORMÁTICA



TEMA DE TESIS PARA OPTAR POR EL TÍTULO DE INGENIERO INFORMÁTICO

TÍTULO : Determinación y segmentación del nivel socio-económico a partir del análisis automático de imágenes de satélite

ÁREA : Ciencia de la Computación

ASESOR : Dr. César Beltrán Castañón

ALUMNO(S) : Carlos Alberto Sosa Pezo

CÓDIGO : 20161310

FECHA : San Miguel, Noviembre de 2021

DESCRIPCIÓN:

De acuerdo con la Asociación Estadounidense de Psicología, el nivel socioeconómico es el estatus social, o clase de un individuo o grupo. Si bien su forma de determinación no es única y los criterios tienden a cambiar de acuerdo con el investigador, esta puede consistir, a nivel de individuos, en una composición de indicadores como el nivel educativo alcanzado, los ingresos (personales o familiares) o la ocupación. La obtención de estos datos y su entendimiento es crucial para desarrollar políticas públicas adecuadas para promover y sostener el desarrollo económico de un país, por lo que varios países desarrollados y organizaciones internacionales invierten recursos financieros y humanos en recolectar estos datos.

En Perú, estos datos son recolectados y publicados por el Instituto Nacional de Estadística e Informática (INEI) a través de métodos como los censos de población y vivienda. Sin embargo, este proceso resulta costoso a nivel financiero y requiere de la capacitación de voluntarios para su ejecución, por lo que la frecuencia de toma de datos es de varios años y no se dispone de información actualizada. Esto también lleva a que la información socioeconómica disponible no concuerde con la realidad vigente, dificultando las labores del Estado.

Ante esta situación, el uso de imágenes satelitales surge como una alternativa para la obtención de datos, debido a la cobertura geográfica, prontitud y el alto nivel de detalle que las caracterizan. Perú cuenta con cinco satélites artificiales, de los cuales tres pertenecen a universidades privadas; uno, a una universidad pública; y uno, al Estado peruano, a través de la Agencia Espacial del Perú - CONIDA, conocido como PeruSat-1. Dicho satélite captura imágenes de alta resolución del área terrestre, incluyendo el territorio peruano, cuyo procesamiento busca servir de fuente para otros organismos del Estado peruano. Sin embargo, no se cuenta con criterios de evaluación precisos para los resultados de determinación del nivel socioeconómico, siendo estos dependientes de la experiencia de cada especialista que realiza dicha evaluación, por lo que la calidad de las respuestas obtenidas resulta comprometida.

Para lidiar con la falta de criterios uniformes para la evaluación de imágenes satelitales, se ha propuesto anteriormente el empleo de técnicas de análisis automático. No obstante, una serie de condiciones impiden el empleo de dicha información obtenida mediante las imágenes capturadas por dicho satélite estatal en la determinación automática del nivel socioeconómico de áreas urbanas. En primer lugar,

la información censal y visual no se encuentran organizadas correctamente, de forma que no es posible su uso como referencia en algún proceso de análisis automatizado sin integrarlas anteriormente. Asimismo, no existen métodos de análisis automáticos que sean apropiados para la resolución y número de bandas de las que está compuesta cada imagen satelital capturada por el satélite PeruSat-1 y correspondan a clasificaciones compatibles con los resultados obtenidos de forma tradicional para su comparación con dichas fuente. Finalmente, no existe una plataforma que permita el uso de dichos métodos por parte de usuarios finales que no necesariamente conozcan el funcionamiento interno de los mismos.

Dada la problemática identificada, este proyecto de fin de carrera propone desarrollar un modelo basado en redes neuronales profundas para identificar de forma automática el nivel socioeconómico de zonas urbanas, empleando para dicho análisis imágenes satelitales del satélite PeruSat-1.

Para ello, se desarrollará un sistema basado en redes neuronales convolucionales que procese dichas imágenes. Este sistema, en una primera etapa, realizará una segmentación semántica de techos; en una segunda etapa, en cambio, identificará el nivel socioeconómico (tanto en un sistema de dos y de tres niveles) en dichos techos delimitados en la etapa anterior. Para su entrenamiento y evaluación, se conformará un corpus que agrupe de forma organizada las imágenes satelitales producidas por el satélite PeruSat-1 y datos censales socioeconómicos georreferenciados obtenidos como resultado del XII Censo de Población y VII de Vivienda, organizado por el INEI en 2017.

Para el acceso de los resultados de este sistema por parte de los usuarios interesados en dicha información socioeconómica, se desarrollará un sistema web, el cual consiste de una interfaz de programación de aplicaciones (componente *back-end*) que permitirá la comunicación con los modelos y una interfaz gráfica de usuario (componente *front-end*) que brindará una forma de manejo oportuno de las imágenes satelitales de referencia, el modelo de redes neuronales (a través de la interfaz de programación de aplicaciones) y el producto final de dichos modelos.

OBJETIVO GENERAL:

Desarrollar un modelo de análisis de imágenes satelitales basado en redes neuronales profundas para la identificación automática del nivel socioeconómico de zonas urbanas

OBJETIVOS ESPECÍFICOS:

1. Integrar datos visuales con datos censales
2. Elaborar un criterio automático de los resultados de determinación del nivel socioeconómico
3. Desarrollar una plataforma para el aprovechamiento oportuno de la información socioeconómica obtenida a través del criterio automático



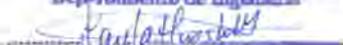
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
Departamento de Ingeniería

Dr. LAYLA HERSH M.
Coordinadora de Especialidad
Ingeniería Informática

Tabla de Contenido

Índice de Figuras	8
Índice de Tablas	11
Generalidades	12
Problemática	12
Árbol de Problemas	12
Descripción	13
Problema seleccionado	15
Objetivos	16
Objetivo general	16
Objetivos específicos	16
Resultados esperados	16
Mapeo de objetivos, resultados y verificación	17
Métodos y Procedimientos	19
Herramientas, métodos y procedimientos a usar	19
Descripción de las herramientas, métodos y procedimientos a usar	20
Marco Conceptual	27
Objetivos del marco conceptual	27
Desarrollo del marco	27
Nivel socioeconómico	27
Políticas públicas	28
Imágenes satelitales	29
Redes neuronales convolucionales	30
Estado del Arte	32
Introducción	32
Objetivos de revisión	32
Preguntas de revisión	33
Estrategia de búsqueda	33
Motores de búsqueda a usar	33
Cadenas de búsqueda a usar	33
Documentos encontrados	35
Criterios de inclusión/exclusión	35
Formulario de extracción de datos	35

Resultados de la revisión	37
Forma de uso de las técnicas de visión por computador en el análisis automático de imágenes satelitales (Pregunta 1)	38
Ventajas del análisis automático de imágenes satelitales respecto a técnicas tradicionales (Pregunta 2)	39
Características de una imagen satelital relevantes para el análisis (Pregunta 3)	40
Conclusiones	40
Integrar datos visuales con datos censales	42
Introducción	42
Resultado alcanzado: Conjunto de imágenes satelitales con regiones etiquetadas en base a los datos censales	42
Transformación de la base de datos censales	42
Tratamiento de imágenes satelitales	45
Conformación del conjunto de imágenes con datos censales integrados	48
Discusión	49
Elaborar un criterio automático de los resultados de determinación del nivel socioeconómico	51
Introducción	51
Resultado alcanzado	51
Protocolo de preprocesamiento de imágenes satelitales etiquetadas para su procesamiento por parte de los modelos de red neuronal	51
Modelo de arquitectura de red neuronal para la segmentación semántica de regiones urbanas en base a techos implementado	52
Arquitectura	52
Entrenamiento	54
Modelo de arquitectura basado en red neuronal convolucionales para la extracción de variables socioeconómicas en imágenes satelitales implementado.	56
Arquitectura	56
Entrenamiento	57
Discusión	62
Desarrollar una plataforma para el aprovechamiento oportuno de la información socioeconómica obtenida a través del criterio automático	64
Introducción	64
Resultado alcanzado	64
Interfaz de Programación de Aplicaciones (API) para el uso automático del modelo de redes convolucionales	64

Interfaz Gráfica de Usuario (GUI) para la visualización de los resultados del modelo	66
Integración de los componentes del sistema (back-end y front-end)	69
Discusión	71
Conclusiones y trabajos futuros	73
Conclusiones	73
Trabajos futuros	75
Referencias	77
Anexos	
Anexo A: Formularios de extracción	1
Anexo B: Plan de Proyecto	5
Anexo C: Declaración de resultado alcanzado	19
Anexo D: Repositorios de código	20
Anexo E: Arquitectura de los modelos de red neuronal	20
Anexo F: Documentación de la Interfaz de Programación de Aplicaciones	28
Anexo G: Secuencia del prototipo	42
Anexo H: Manual de uso de la aplicación web	45
Anexo I: Reportes de pruebas unitarias y funcionales	53
Anexo J: Capturas de pantalla de las pruebas de ejecución de la plataforma	64

● Índice de Figuras

Figura 1. Árbol de problemas. Elaboración propia	12
Figura 2. Bandas de una imagen satelital Landsat. Fuente: Open Geospatial Education	29
Figura 3a. Visualización en QGIS de la base de datos espacial de la publicación «Planos estratificados de Lima Metropolitana a nivel de manzanas según Ingreso Per Cápita del Hogar» y las características asociadas a una etiqueta. Fuente: Elaboración propia	43
Figura 3b. Ventana de QGIS para la importación del directorio como una capa de vectores. Fuente: Elaboración propia	44
Figura 3c. Figura 3c. Ventana de QGIS para la exportación de la capa de vectores en un archivo SQLite. Fuente: Elaboración propia	44
Figura 4. De izquierda a derecha: imagen multiespectral, imagen pancromática e imagen tras refinado pancromático. Fuente: CONIDA	45
Figura 5a. Código fuente del «script» de refinado pancromático. Fuente: Elaboración propia	46
Figura 5b. Imágenes satelitales tras el refinado pancromático. Fuente: Elaboración propia	46
Figura 5c. Ejemplo de carpeta (producto de PeruSat-1) esperada por el «script» de refinamiento pancromático. Fuente: Elaboración propia	47
Figura 6a. De izquierda a derecha: bloque (fragmento) de una imagen satelital, etiqueta de manzanas y etiqueta segmentada por dos niveles socioeconómicos (en verde, «nivel bajo o medio»; en azul, «nivel medio o alto»). Fuente:Elaboración propia	48
Figura 6b. Izquierda: bloque (fragmento) de una imagen satelital. Derecha: etiqueta segmentada por dos niveles socioeconómicos (en verde, «nivel bajo o medio»; en azul, «medio alto»; en amarillo, «alto»). Fuente: Elaboración propia	48
Figura 7. Arquitectura de TernaNet. Fuente: Iglovikov, Shvets	53

Figura 8. Evolución de la función de pérdida Jaccard a través de las épocas para el modelo de segmentación semántica de techos. Fuente: Elaboración propia	55
Figura 9. De izquierda a derecha: imágenes satelitales, etiquetas de manzanas pertenecientes al conjunto de datos, predicción de las etiquetas generada por el modelo de segmentación semántica de techos. Fuente: Elaboración propia	56
Figura 10. Evolución de la función de pérdida Jaccard a través de las épocas para el modelo de extracción de variables socioeconómicas para dos niveles socioeconómicos. Fuente: Elaboración propia	58
Figura 11. De izquierda a derecha: predicción del nivel socioeconómico original, predicción del modelo de segmentación de techos para la imagen, predicción original tras ser aplicado la predicción de techos como máscara. Fuente: Elaboración propia	59
Figura 12. De izquierda a derecha: imágenes satelitales, etiquetas de nivel socioeconómico, predicción de las etiquetas generada por el modelo de extracción de variables socioeconómicas. Fuente: Elaboración propia	59
Figura 13. Evolución de la función de pérdida Jaccard a través de las épocas para el modelo de extracción de variables socioeconómicas para tres niveles socioeconómicos. Fuente: Elaboración propia	61
Figura 14. De izquierda a derecha: imágenes satelitales, etiquetas de nivel socioeconómico, predicción de las etiquetas generada por el modelo de extracción de variables socioeconómicas para tres niveles socioeconómicos. Fuente: Elaboración propia	62
Figura 15. Esquema de funcionamiento de la API. Fuente: Gallardo	65
Figura 16. Prototipo de la interfaz gráfica de usuario, vista inicial. Fuente: Elaboración propia	67
Figura 17. Diagrama de flujo de la interacción entre cliente y servidor. Fuente: Elaboración propia	67
Figura 18. Carpeta de imágenes del servidor SFTP de la plataforma web, vista desde Bash. Fuente: Elaboración propia	69
Figura 19. Prototipo de la interfaz gráfica de usuario, selección del área de interés. Fuente: Elaboración propia	A43

Figura 20. Prototipo de la interfaz gráfica de usuario, lista de imágenes candidatas. Fuente: Elaboración propia	A44
Figura 21. Prototipo de la interfaz gráfica de usuario, selección de una imagen para procesar. Fuente: Elaboración propia	A44
Figura 22. Prototipo de la interfaz gráfica de usuario, imagen recibida. Fuente: Elaboración propia	A45
Figura 23. Ejecución de la plataforma web, vista inicial Fuente: Elaboración propia	A50
Figura 24. Ejecución de la plataforma web, selección del Área de Interés Fuente: Elaboración propia	A51
Figura 25. Ejecución de la plataforma web, selección de imagen a procesar. Fuente: Elaboración propia	A51
Figura 26. Ejecución de la plataforma web, ubicación de la imagen satelital y máscaras de predicción en el mapa. Fuente: Elaboración propia	A52
Figura 27. Ejecución del servidor web en el servidor remoto, visto desde Bash. Fuente: Elaboración propia	A64
Figura 28. Ejecución de la plataforma web, ubicación de la imagen satelital en el mapa. Fuente: Elaboración propia	A66

Índice de Tablas

Tabla 1. Resultados esperados, indicadores y medios de verificación del objetivo específico 1.	17
Tabla 2. Resultados esperados, indicadores y medios de verificación del objetivo específico 2.	17
Tabla 3. Resultados esperados, indicadores y medios de verificación del objetivo específico 3.	18
Tabla 4. Resultados esperados, herramientas a usar y métodos a utilizar en el proyecto	19
Tabla 5.1. Criterios PICOC	32
Tabla 5.2. Términos derivados de PICOC	34
Tabla 5.3. Formulario de extracción de datos	35
Tabla 6.1. Valores de los hiperparámetros para el entrenamiento	54
Tabla 6.2. Reporte de rendimiento del modelo de segmentación semántica de regiones urbanas en base a techos	54
Tabla 7.1. Valores de los hiperparámetros para el entrenamiento del modelo de extracción de variables socioeconómicas para dos niveles socioeconómicos	57
Tabla 7.2. Reporte de rendimiento del modelo de extracción de variables socioeconómicas para dos niveles socioeconómicos	58
Tabla 8.1. Valores de los hiperparámetros para el entrenamiento del modelo de extracción de variables socioeconómicas para dos niveles socioeconómicos	60
Tabla 8.2. Reporte de rendimiento del modelo de extracción de variables socioeconómicas para tres niveles socioeconómicos	60
Tabla 9. Funcionalidades de la API	65

Capítulo 1. Generalidades

1.1 Problemática

1.1.1 Árbol de Problemas

En el presente proyecto de fin de carrera, se empleó la técnica del árbol de problemas (Figura 1) para estructurar los problemas tratados en el mismo. Esta técnica consiste en la identificación del problema central de la investigación para, luego, determinar y jerarquizar los problemas causa y consecuencias que se desprenden del principal.

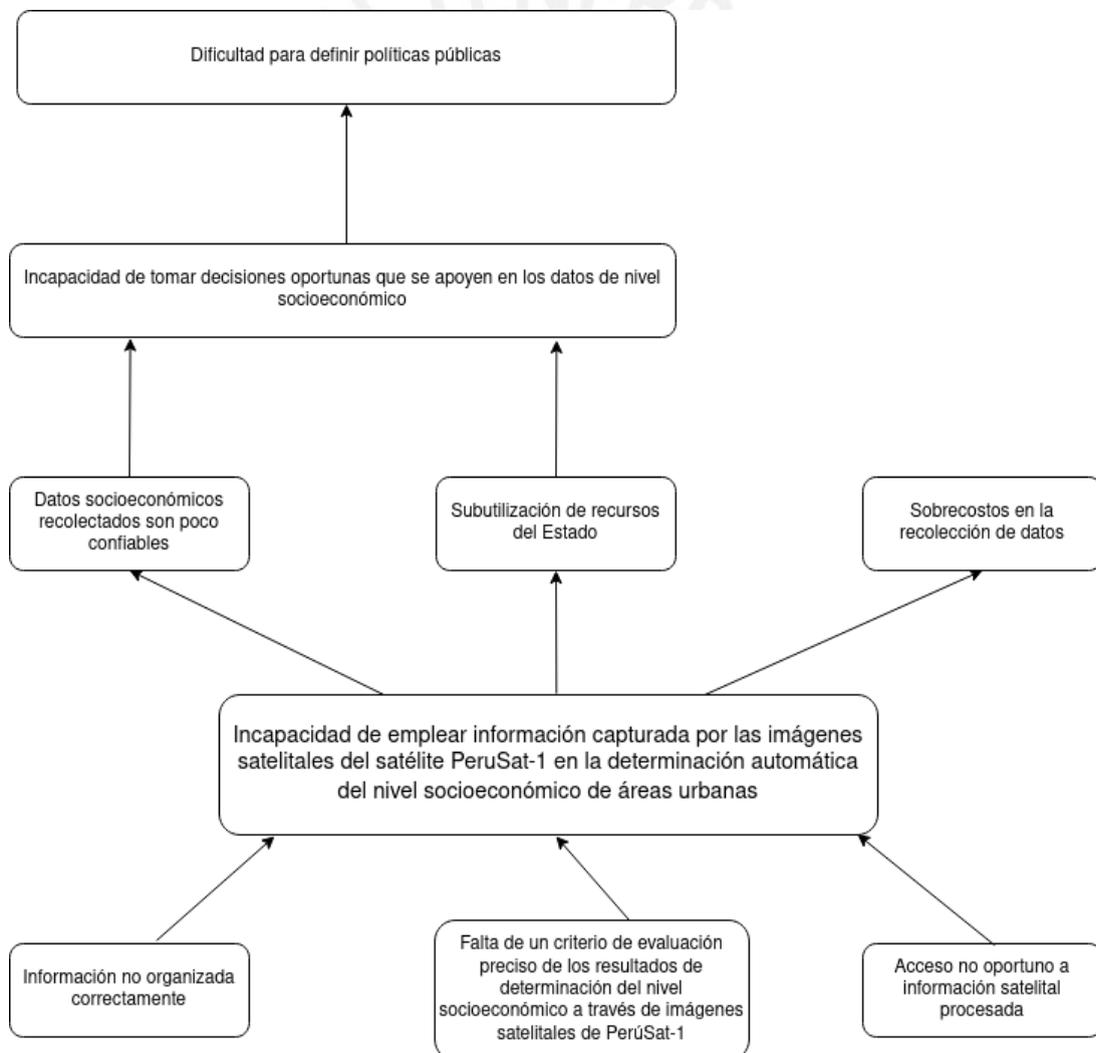


Figura 1. Árbol de problemas.

Fuente: Elaboración propia

1.1.2 Descripción

De acuerdo con la Asociación Estadounidense de Psicología (APA), el nivel socioeconómico (también conocido como estatus socioeconómico) es el estatus social, o clase de un individuo o grupo. Si bien su forma de determinación no es única y los criterios tienden a cambiar de acuerdo con el investigador, esta puede consistir, a nivel de individuos, en una composición de indicadores como el nivel educativo alcanzado, los ingresos (personales o familiares) o la ocupación (Saegert *et al.*, 2006). La obtención de estos datos y su entendimiento es crucial para desarrollar políticas públicas adecuadas para promover y sostener el desarrollo económico de un país, por lo que varios países desarrollados y organizaciones internacionales invierten recursos financieros y humanos en recolectar estos datos (Han *et al.*, 2020).

En Perú, estos datos son recolectados y publicados por el Instituto Nacional de Estadística e Informática (INEI) a través de métodos como los censos de población y vivienda (INEI, 2018). Sin embargo, este proceso resulta costoso a nivel financiero (El Comercio, 2017) y requiere de la capacitación de voluntarios para su ejecución (Sánchez, entrevista con El Peruano, 2017), por lo que la frecuencia de toma de datos es de varios años (INEI, 2018) y no se dispone de información actualizada. Esto también lleva a que la información socioeconómica disponible no concuerde con la realidad vigente, dificultando las labores del Estado (Alva, entrevista con Radio Programas del Perú, como se citó en Gestión, 2020).

Ante esta situación, el uso de imágenes satelitales surge como una alternativa para la obtención de datos, debido a la cobertura geográfica, prontitud y el alto nivel de detalle que las caracterizan (Han *et al.* 2020). Perú cuenta con cinco satélites artificiales, de los cuales tres pertenecen a universidades privadas; uno, a una universidad pública; y uno, al Estado peruano, a través de la Agencia Espacial del Perú - CONIDA, conocido como PeruSat-1 (Radio Programas del Perú, 2018). Dicho satélite captura imágenes de alta resolución (0,7 metros por píxel en la banda pancromática y 2,8, en la banda multiespectral) del área terrestre, incluyendo el territorio peruano, cuyo procesamiento busca servir de fuente para otros organismos del Estado peruano (Guevara, 2018).

Sin embargo, no se cuenta con criterios de evaluación precisos para los resultados de determinación del nivel socioeconómico. Cuando CONIDA recibe el pedido de algún otro organismo del estado de corroborar algún dato demográfico con el que cuenta, el procesamiento de las imágenes del área objeto de estudio es parcialmente automático y sus resultados dependen de la experiencia con la que cuenta cada especialista que participa en la ejecución de la labor (Pasapera, C. (Comunicación personal), 2020), por lo que la calidad de las respuestas obtenidas resulta comprometida.

Para lidiar con la falta de criterios uniformes para la evaluación de imágenes satelitales, se ha propuesto anteriormente el empleo de técnicas de análisis automático; sobre todo, de modelos de redes neuronales convolucionales que emplean como entrada las imágenes satelitales para automatizar el análisis de las mismas y la obtención de respuestas a través de dicho análisis (Tingzon *et al.*, 2019; Bansal *et al.*, 2020).

No obstante, la información censal y visual no se encuentran organizadas correctamente, de forma que no es posible su uso como referencia en algún proceso de análisis automatizado sin integrarlas anteriormente. Así, por ejemplo, el Instituto Nacional de Estadística e Informática (2020) cuenta con información censal procesada y georreferenciada, mas no se encuentra asociada a referencias visuales.

Además, si bien existen métodos que emplean como variables de entrada valores extraídos de una (Tingzon *et al.*, 2019) o más bandas (Bansal *et al.*, 2020) de una imagen satelital para devolver valores de variables socioeconómicas, estos resultados no son devueltos con un sistema jerarquizado por niveles (Tingzon *et al.*, 2019; Han *et al.*, 2020), o corresponden a clasificaciones no compatibles con los resultados obtenidos de forma tradicional, lo que impide una comparación adecuada entre ambas fuentes (Bansal *et al.*, 2020). Asimismo, los métodos ya existentes no resultan adecuados para el tipo de imágenes satelitales capturadas por el satélite PeruSat-1, dada su resolución y número de bandas de las que está compuesta cada imagen (Guevara, 2018).

Por otro lado, la inexistencia de una plataforma que permita el uso de dichos métodos por parte de usuarios finales que no necesariamente conozcan

el funcionamiento interno de los mismos dificulta el acceso oportuno a la información obtenida mediante estos. Esto sería consecuencia de que los trabajos existentes se limitan a presentar el funcionamiento y los resultados de sus modelos, mas no son aprovechados en situaciones reales (Tingzon *et al.*, 2019; Bansal *et al.*, 2020; Han *et al.*, 2020).

El no poder contar con una determinación del nivel socioeconómico actualizada y oportuna trae como consecuencia el lidiar con información poco confiable, que no ha podido ser contrastada con otras fuentes y que puede ser desfasada. Esto también implica un incremento excesivo del costo de la obtención de esta información, debido a la necesidad de realizar nuevos procesos de recopilación de datos para corregir o actualizar los ya disponibles (Alva, entrevista con Radio Programas del Perú, como se citó en Gestión, 2020).

Por otro lado, el hecho de no poder emplear, dadas las circunstancias vigentes, imágenes satelitales como un medio para contrastar la información disponible y determinar su confiabilidad constituye una subutilización de los recursos del Estado, al no estar aprovechando el potencial de las herramientas disponibles, en este caso, el satélite artificial que dispone el Estado, así como el equipo computacional con el que cuenta CONIDA.

En consecuencia, las decisiones tomadas en base a estos datos poco confiables y sin el aprovechamiento total de los recursos no resultan oportunas y pueden llegar a ser desacertadas (Tingzon *et al.*, 2019), como, por ejemplo, el considerar dentro de un grupo económicamente en situación de riesgo a alguien que no cumple con dichos requisitos (Medina, como se citó en La República, 2020).

Finalmente, la incapacidad de tomar decisiones oportunas dificulta la elaboración de políticas públicas, ya que estas no pueden basarse en dichas decisiones, alejadas de la agenda política (Ruíz y Cardenas, s.f.), cuyo entendimiento es importante en el proceso de elaboración de políticas públicas.

1.1.3 Problema seleccionado

Se ha seleccionado el siguiente problema central para el presente proyecto final de carrera: Incapacidad de emplear información capturada por las imágenes

satelitales del satélite PeruSat-1 en la determinación automática del nivel socioeconómico de áreas urbanas.

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar un modelo de análisis de imágenes satelitales basado en redes neuronales profundas para la identificación automática del nivel socioeconómico de zonas urbanas.

1.2.2 Objetivos específicos

- O1. Integrar datos visuales con datos censales
- O2. Elaborar un criterio automático de los resultados de determinación del nivel socioeconómico
- O3. Desarrollar una plataforma para el aprovechamiento oportuno de la información socioeconómica obtenida a través del criterio automático

1.2.3 Resultados esperados

- O 1. **Integrar** datos visuales con datos censales
 - R1. Conjunto de imágenes satelitales con regiones etiquetadas en base a los datos censales
- O 2. **Elaborar** un criterio automático de los resultados de determinación del nivel socioeconómico
 - R1. Protocolo de preprocesamiento de imágenes satelitales etiquetadas para su procesamiento por parte de los modelos de red neuronal
 - R2. Modelo de arquitectura de red neuronal para la segmentación semántica de regiones urbanas en base a techos implementado
 - R3. Modelo de arquitectura basado en red neuronal convolucionales para la extracción de variables socioeconómicas en imágenes satelitales implementado
- O 3. **Desarrollar** una plataforma para el aprovechamiento oportuno de la información socioeconómica obtenida a través del criterio automático

R1. Interfaz de Programación de Aplicaciones (API) para el uso automático del modelo de redes convolucionales

R2. Interfaz Gráfica de Usuario (GUI) para la visualización de los resultados del modelo

R3. Integración de los componentes del sistema (*back-end* y *front-end*)

1.2.4 Mapeo de objetivos, resultados y verificación

En las Tablas 1, 2 y 3 se listan los resultados esperados asociados a cada objetivo específico, así como sus indicadores y sus respectivos medios de verificación.

Tabla 1: Resultados esperados, indicadores y medios de verificación del objetivo específico 1.

Objetivo: Integrar datos visuales con datos censales		
Resultado	Indicador objetivamente verificable	Medio de verificación
R1. Conjunto de imágenes satelitales con regiones etiquetadas en base a los datos censales	<ul style="list-style-type: none"> Al menos 5 distritos de la ciudad con información censal de nivel socioeconómico Un mapa de imagen satelital del mismo periodo censal Un mapa de niveles socioeconómicos etiquetados de acuerdo con información censal 	<ul style="list-style-type: none"> Repositorio de información censal de al menos 5 distritos de la ciudad Base de datos de imágenes satelitales correspondientes al mismo periodo censal Base de datos de imágenes etiquetadas de acuerdo con información censal

Tabla 2: Resultados esperados, indicadores y medios de verificación del objetivo específico 2 (Elaboración propia).

Objetivo: elaborar un criterio automático de los resultados de determinación del nivel socioeconómico		
Resultado	Indicador objetivamente verificable	Medio de verificación
R1. Protocolo de preprocesamiento de imágenes satelitales	<ul style="list-style-type: none"> Protocolo de preprocesamiento de imágenes satelitales 	<ul style="list-style-type: none"> Reporte de descripción del protocolo

etiquetadas para su procesamiento por parte de los modelos de red neuronal	validado por un especialista	
R2. Modelo de arquitectura de red neuronal para la segmentación semántica de regiones urbanas en base a techos implementado	<ul style="list-style-type: none"> Un modelo de red neuronal optimizado de al menos 70,0 % de precisión para la segmentación 	<ul style="list-style-type: none"> Diagrama de la arquitectura del modelo de red neuronal de segmentación implementado Repositorio de código, el cual contiene la implementación del modelo de segmentación Reporte de resultados de ejecución de pruebas de optimización al modelo
R3. Modelo de arquitectura basado en red neuronal convolucionales para la extracción de variables socioeconómicas en imágenes satelitales implementado	<ul style="list-style-type: none"> Un modelo de red neuronal optimizado de al menos 70,0 % de precisión para la determinación del nivel socioeconómico 	<ul style="list-style-type: none"> Diagrama de la arquitectura del modelo de red neuronal de extracción de variables socioeconómicas implementado Repositorio de código del modelo de extracción de variables socioeconómicas Reporte de resultados de ejecución de pruebas de optimización al modelo

Tabla 3: Resultados esperados, indicadores y medios de verificación del objetivo específico 3 (Elaboración propia).

Objetivo: desarrollar una plataforma para el aprovechamiento oportuno de la información socioeconómica obtenida a través del criterio automático		
Resultado	Indicador objetivamente verificable	Medio de verificación
R1. Interfaz de Programación de Aplicaciones (API) para el uso automático del modelo de redes convolucionales	<ul style="list-style-type: none"> Pruebas unitarias aprobadas al 100% Pruebas funcionales aprobadas al 100 % 	<ul style="list-style-type: none"> Repositorio de código fuente del API Reporte de resultados de ejecución de pruebas unitarias y funcionales del API Documentación descriptiva de funcionalidades de la API

R2. Interfaz Gráfica de Usuario (GUI) para la visualización de los resultados del modelo	<ul style="list-style-type: none"> • Pruebas unitarias aprobadas al 100% • Pruebas funcionales aprobadas al 100 % 	<ul style="list-style-type: none"> • Repositorio de código fuente de la GUI • Reporte de resultados de ejecución de pruebas unitarias y funcionales de la GUI • Manual de usuario para la manipulación de la GUI
R3. Integración de los componentes del sistema (<i>back-end</i> y <i>front-end</i>)	<ul style="list-style-type: none"> • Pruebas funcionales aprobadas al 100 % 	<ul style="list-style-type: none"> • Sistema desplegado en un servidor local • Reporte de resultados de ejecución de pruebas funcionales

1.3 Métodos y Procedimientos

En esta sección se mencionan y explican las herramientas, métodos y procedimientos a emplear para el desarrollo de este proyecto de fin de carrera.

1.3.1 Herramientas, métodos y procedimientos a usar

Se menciona, a continuación, las herramientas a emplear y procedimientos a seguir, asociados con cada resultado esperado de este proyecto (ver Tabla 4).

Tabla 4: Resultados esperados, herramientas a usar y métodos a utilizar en el proyecto (Elaboración propia).

Resultado esperado	Herramienta a utilizar	Métodos a utilizar
Conjunto de imágenes satelitales con regiones etiquetadas en base a los datos censales	<ul style="list-style-type: none"> • QGIS • Python • Rasterio • SQLite • NumPy • GitHub • Bash 	<ul style="list-style-type: none"> • Refinamiento pancromático
Modelo de segmentación semántica de regiones urbanas en base a techos	<ul style="list-style-type: none"> • Python • GitHub • PyTorch • Rasterio 	<ul style="list-style-type: none"> • Algoritmos de aprendizaje supervisados
Modelo automático basado en redes	<ul style="list-style-type: none"> • NumPy • Matplotlib 	<ul style="list-style-type: none"> • Algoritmos de aprendizaje no

neuronales convolucionales para la extracción de variables socioeconómicas en imágenes satelitales		supervisado <ul style="list-style-type: none"> ● Intersección sobre Unión (índice Jaccard)
Interfaz de Programación de Aplicaciones (API) para el uso automático del modelo de redes convolucionales	<ul style="list-style-type: none"> ● Python ● GitHub ● PyTorch ● Flask ● Rasterio 	<ul style="list-style-type: none"> ● API REST ● Georreferenciación
Interfaz Gráfica de Usuario (GUI) para la visualización de los resultados del modelo	<ul style="list-style-type: none"> ● JavaScript ● GitHub ● React.js 	

1.3.2 Descripción de las herramientas, métodos y procedimientos a usar

En esta sección, se describirá las herramientas y procedimientos a usar en el presente proyecto, que fueron mencionados en la sección anterior, y se explicará cómo se planea emplear en el mismo.

Herramientas

- **QGIS**

QGIS es un sistema de información de código abierto para la lectura y manipulación de información geográfica. Permite abrir y crear diferentes tipos de archivos georreferenciados, tales como imágenes satelitales o mapas de polígonos (Open Source Geospatial Foundation, s.f.).

En el presente proyecto, se empleará dicha herramienta para la conversión de las etiquetas georreferenciadas en un formato manejable por el resto de las herramientas.

- **Rasterio**

Rasterio es una biblioteca de Python para la extracción y manipulación de imágenes ráster georreferenciadas, tanto los píxeles en sí como los metadatos asociados a la imagen (Mapbox, 2018).

En el presente proyecto, Rasterio cumple la función de transformar las imágenes desde el formato ráster hasta una forma matricial, de forma que sea manipulable durante el etiquetado y durante el procesamiento por los modelos. Asimismo, permitirá leer la ubicación de dicha imagen en la superficie terrestre, lo que ayuda a su visualización en un mapa virtual.

- **SQLite**

SQLite es una biblioteca en lenguaje C que implementa un motor de base de datos SQL pequeño, rápido, autónomo, de alta confiabilidad y con todas las funciones. (Consortio SQLite, s.f)

En el presente proyecto, se empleará esta herramienta para el almacenamiento y acceso a las etiquetas de nivel socioeconómico asociadas a las zonas urbanas del presente proyecto final de carrera.

- **Python**

Python es un lenguaje de programación de alto nivel, interpretado y multiparadigma; es decir, permite la codificación mediante más de un paradigma de programación, sea programación orientada a objetos, programación funcional o programación imperativa; administrado por la Python Software Foundation (Python Software Foundation, s.f.).

Se ha elegido este lenguaje para la elaboración del modelo como de la interfaz de programación de aplicaciones, debido a diferentes motivos. En primer lugar, debido a la existencia de bibliotecas diseñadas para la implementación de modelos de aprendizaje automático, las cuales son compatibles con el tipo de modelos a desarrollar, así como de bibliotecas para la manipulación de imágenes satelitales y sus etiquetas asociadas. Por otro lado, debido a la presencia de los módulos necesarios para la implementación de las API.

- **GitHub**

GitHub es una plataforma de colaborativo que permite mantener un control de versiones del código fuente de un proyecto mientras este se almacena de forma remota y accesible mediante la web (GitHub, s.f.).

El código fuente desarrollado en el marco del presente proyecto será almacenado y controlado empleando dicha herramienta, de forma que su acceso no dependa de algún equipo que se disponga y que pueda manejarse un control de cambios durante la duración del proyecto.

- **Bash**

Bash es un lenguaje de órdenes y *shell* de Unix escrito por Brian Fox como un reemplazo de software libre para el shell Bourne en el proyecto GNU (Ramey en Hamilton, 2011).

En el presente proyecto, esta herramienta permitirá la manipulación y tratamiento previo de las imágenes satelitales de forma que estén aptas para su integración con los datos censales.

- **PyTorch**

PyTorch es un marco de trabajo de código abierto orientado al desarrollo y despliegue de modelos de aprendizaje automático en Python. Este incluye, principalmente, métodos para el procesamiento de matrices, componentes para la construcción de redes neuronales y modelos preentrenados (Facebook, s.f. b).

En el proyecto, se emplea dicho marco de trabajo en la implementación de los modelos de redes neuronales convolucionales en el análisis de las imágenes satelitales para la extracción de variables socioeconómicas.

- **NumPy**

NumPy es un proyecto de código abierto que permite la computación numérica en Python. El código fuente del módulo se encuentra disponible desde GitHub (Numpy, s.f.).

Los módulos mencionados anteriormente emplean estructuras definidas en NumPy para la representación de sus datos numéricos, por lo que, en el proyecto, esta biblioteca permitirá manipular y transformar dichas estructuras.

- **Matplotlib**

Matplotlib es una biblioteca de Python para la visualización de datos estática, dinámica e interactiva (Hunter. 2007).

Se empleará esta biblioteca para el análisis gráfico del comportamiento de los datos y los resultados obtenidos tras el procesamiento de los mismos a través del modelo de redes neuronales propuesto.

- **Flask**

Flask es un marco de trabajo para el desarrollo de aplicaciones web que cumple la función de interfaz de puerta de enlace de servicios web (WSGI). Se caracteriza por consistir en un mínimo de componentes para su funcionamiento y su capacidad de escalar a aplicaciones más complejas (Ronacher, s.f.).

En el proyecto, será empleado para la implementación de las interfaces de programación de aplicaciones (API), uno de los resultados esperados mencionados en la sección anterior, de forma que pueda comunicarse con la interfaz gráfica de usuario, otro resultado esperado de este proyecto de fin de carrera, para la ejecución automática del modelo de extracción del nivel socioeconómico.

- **JavaScript**

JavaScript es un lenguaje de programación interpretado y multiparadigma (permite la programación imperativa, declarativa y orientada a objetos) empleado mayoritariamente como una herramienta de *scripting* para entornos web (Mozilla Foundation, s.f.).

Este lenguaje de programación será empleado en el presente proyecto para la implementación de la interfaz gráfica de usuario planteada como resultado esperado para el objetivo 3, de forma que los

resultados del modelo de extracción de variables socioeconómicas sean posibles de visualizar por sus usuarios finales sin necesitar conocer el funcionamiento del modelo en sí.

- **React**

React es una biblioteca diseñada para la construcción de interfaces gráficas de usuario. Disponible para JavaScript, es una biblioteca declarativa y basada en componentes encapsulados (Facebook, s.f. a).

Para el presente proyecto, este será empleado en la construcción de la interfaz gráfica de usuario descrita anteriormente y que constituye un resultado esperado. Este marco de trabajo se comunicará con la interfaz de programación de aplicaciones descrita y consumirá los servicios que requiera durante su funcionamiento.

Métodos y procedimientos

- **Refinado pancromático**

Conocido también como *pansharpening*, es un tipo de fusión de datos que se refiere al proceso de combinar píxeles de color de menor resolución con los píxeles pancromáticos de mayor resolución para producir una imagen de color de alta resolución (Padwick, Deskevich, Pacifici, Smallwood, 2010).

- **Algoritmos de aprendizaje automático supervisados**

Los algoritmos de aprendizaje automático supervisado son modelos computacionales que para su entrenamiento requieren de datos etiquetados previamente, los cuales sirven como base para el ajuste de los parámetros del modelo (Bishop, 2006).

- **Algoritmos de aprendizaje automático no supervisados**

Los algoritmos de aprendizaje automático no supervisado son algoritmos de aprendizaje automático cuya data de entrenamiento es generada de manera autónoma. Esto debido a ciertos factores, aunque

por lo general se da cuando el proceso de etiquetado resulta particularmente tedioso, o si se necesita empezar a procesar un video al mismo instante en que está disponible (Bishop, 2006).

- **Intersección sobre Unión (índice Jaccard)**

La intersección sobre unión, también conocida como índice Jaccard, es una métrica sobre la similitud de dos conjuntos (Raimundo & Vargas, 1996). Para dos conjuntos A y B, su cálculo se daría de la siguiente manera:

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

en la que $A \cup B$ representa la unión de los conjuntos A y B mientras que $A \cap B$, su intersección.

Se empleará dicha métrica para este proyecto en la evaluación de la eficacia del modelo respecto de la tarea de segmentación de imágenes satelitales de PeruSAT-1. Para ello, los conjuntos a comparar son las etiquetas generadas por el modelo y las etiquetas reales de los datos empleados para el entrenamiento de dichos modelos.

- **API REST**

Una interfaz de programación de aplicaciones de transferencia de estado representacional (API REST) es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de aplicaciones, las cuales siguen un conjunto de principios, mencionados a continuación:

- Arquitectura cliente-servidor, cuyas partes se comunican mediante HTTP
- Cada comunicación entre el cliente y el servidor no presenta estado y es independiente de la otra.
- Los datos son almacenables en caché.
- Transferencia estandarizada de datos entre cliente y servidor
- Un sistema en capas que organiza en jerarquías invisibles (Red

Hat, s.f.)

En el presente proyecto, se empleará REST para la implementación de las interfaces de programación de aplicaciones, debido a su simpleza en manejo y facilidad de entendimiento.

- **Georreferenciación**

De acuerdo con ArcGIS Resources (s.f.):

La georreferenciación es el uso de coordenadas de mapa para asignar una ubicación espacial a entidades cartográficas. Todos los elementos de una capa de mapa tienen una ubicación geográfica y una extensión específicas que permiten situarlos en la superficie de la Tierra o cerca de ella. La capacidad de localizar de manera precisa las entidades geográficas es fundamental tanto en la representación cartográfica como en SIG.

En este proyecto de fin de carrera, se empleará la georreferenciación para ubicar las imágenes satelitales en el mapa, de acuerdo con sus metadatos.

Capítulo 2. Marco Conceptual

2.1 Objetivos del marco conceptual

Esta sección busca realizar una descripción de conceptos que contribuyen con el entendimiento del problema planteado en el presente proyecto de fin de carrera, el cual es que la determinación del nivel socioeconómico de la población peruana es ineficiente temporalmente, es costosa y no se encuentra con información actualizada, y su relación con el empleo de imágenes satelitales. A continuación, se conceptualiza principalmente términos asociados a los siguientes:

- Nivel socioeconómico
- Políticas públicas
- Imágenes satelitales
- Redes neuronales convolucionales

2.2 Desarrollo del marco

2.2.1 Nivel socioeconómico

Se entiende por nivel o estatus socioeconómico al estatus social en el que un individuo o grupo de individuos se encuentra, como parte de una clasificación jerárquica. Existen diferentes acercamientos a la labor de clasificar a las personas en estratos sociales y estos pueden variar de acuerdo al estudio. Una de las estrategias más empleadas es aquella que incluye los ingresos percibidos, el nivel de educación máximo alcanzado (con énfasis en la obtención o no de un bachillerato) y la ocupación (o la ausencia de ocupación) sea de forma individual o a través de un resultado compuesto. Asimismo, también existen estudios que consideran variables cuyo valor es más duradero en el tiempo, esto es, se toman en cuenta los recursos económicos y materiales que dispone un individuo para la clasificación. (Saegert *et al.*, 2018).

Mientras, otro enfoque para la determinación del nivel socioeconómico se basa en el análisis de variables referidas a la población en vez del individuo aislado, tales como la actividad económica, el tamaño del parque automotor, el uso de los suelos y la contaminación del aire, las cuales permiten inferir el nivel de desarrollo

socioeconómico de un grupo de personas asentadas en una región (Han *et al.*, 2020) si bien no constituyen indicadores directos de un estatus social.

En Perú, los grupos de estudio de mercados son unos de los organismos que publican esta información al público. Por ejemplo, Ipsos Perú emplea una escala de niveles socioeconómicos desde la A hasta la E, donde la primera representa el nivel más favorecido, mientras la última representa al nivel más vulnerable (2020).

2.2.2 Políticas públicas

Según el maestro en fiscal Domingo Ruiz y el maestro en políticas públicas Carlos Cadenas, las políticas públicas son las acciones emitidas por el gobierno que buscan cómo dar respuestas a las diversas demandas de la sociedad. El establecimiento de políticas públicas puede resolver problemas concretos que aquejan a la población, así como promover la integración social y la equidad en esta (s.f.). Así, en Perú existen actualmente, por ejemplo, políticas públicas enfocadas al desarrollo de la ciencia, tecnología e innovación tecnológica; a la inclusión financiera; y para asegurar la seguridad alimentaria y nutricional (Centro Nacional de Planeamiento Estratégico, 2020).

Para la elaboración de una política pública, de acuerdo con la Guía de políticas nacionales peruana, debe primero determinarse si es pertinente la elaboración de la misma. Para ello, se requiere saber si el asunto sobre el cual se busca desarrollar una política pública es de interés nacional. Ello debe encontrarse documentado en información recopilada que demuestre la existencia de un estado indeseable de carencia en una población, y que dicha situación puede mejorarse; asimismo, requiere demostrarse la necesidad del Estado de intervenir y si es un problema que ya ha sido abordado anteriormente. En el caso de las políticas nacionales, también debe corroborarse que sea

de interés nacional. Con lo anterior determinado, ya puede procederse a la elaboración de la misma, en la que también será necesaria la recopilación de información existente respecto al asunto de interés de la política, incluyendo información de índole socioeconómica (Centro Nacional de Planeamiento Estratégico, 2018).

2.2.3 Imágenes satelitales

Las imágenes satelitales, también conocidas como imágenes de satélite, son representaciones gráficas de regiones terrestres adquiridas mediante satélites artificiales (Han *et al.*, 2020; Oshri *et al.*, 2018; Tingzon *et al.*, 2019) o sensores de aeronaves (Shermeyer *et al.*, 2020). A diferencia de las imágenes ópticas estándar, estas suelen contar con más de 3 bandas espectrales - a las bandas rojo, verde y azul; se adiciona capas infrarrojas y otras derivadas de las capas anteriores, como puede verse en la figura 2 (Bansal *et al.*, 2020).

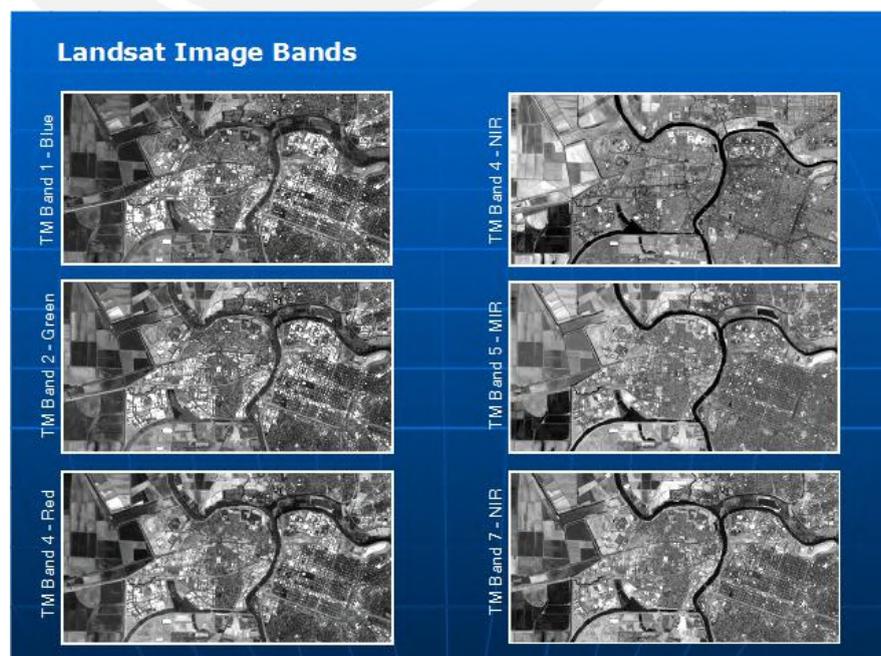


Figura 2. *Bandas de una imagen satelital Landsat. Fuente: Open Geospatial Education*

Debido a su cobertura geográfica extensa, su prontitud y granularidad (nivel de detalle), son consideradas una fuente alternativa para la medición de variables económicas (Han *et al.* 2020). En especial, se ha demostrado que las imágenes satelitales que capturan el nivel de intensidad de luz nocturna ha demostrado estar correlacionada con variables económicas como el producto bruto interno (Tingzon *et al.*, 2019; Bansal *et al.*, 2020). A pesar de ello, también existen estudios que infieren estas variables solamente mediante imágenes satelitales diurnas (Han *et al.*, 2020).

2.2.4 Redes neuronales convolucionales

Las redes neuronales convolucionales son un tipo de red neuronal artificial basada en el funcionamiento de las neuronas pertenecientes al sistema visual de los vertebrados, con el objetivo de reconocer patrones visuales. Su estructura fue propuesta originalmente por Fukushima (1980) en el artículo en el que presenta al Neocognitron. Luego, LeCun *et al.* (1998) añadió al modelo el uso de la propagación hacia atrás, en el cual se propaga el error en la inferencia desde las capas finales hacia las iniciales, como método de aprendizaje para el entrenamiento.

La arquitectura base de una red neuronal convolucional consiste en varias capas convolucionales, es decir, capas que extraen características en forma de matrices a través de filtros entrenables que multiplican a las capas de entrada. Asimismo, tras cada capa convolucional, suele existir una función no lineal que modifica el resultado (Fukushima, 1980). Dependiendo de la labor que la red efectúa, puede incluir solo capas convolucionales en caso se busque segmentar regiones de interés (Yuan *et al.*, 2018) o incluir capas totalmente conectadas en tareas de clasificación (Han *et al.*, 2020).

Debido a su capacidad de reconocer patrones en imágenes y su capacidad de aprender sin necesitar un proceso manual de extracción de características, son muy utilizadas en tareas de visión por computador para el procesamiento de imágenes ópticas convencionales (Simonyan & Zisserman, 2015), pero también para procesar imágenes satelitales (Oshri *et al.*, 2018).



Capítulo 3. Estado del Arte

3.1 Introducción

El presente capítulo comprende la presentación del plan de revisión sistemática y el reporte de los resultados de la misma, de acuerdo con las directrices publicadas por Kitchenham (2007), para el presente proyecto final de carrera. Esta presenta como propósito el conocer e interpretar los trabajos más relevantes respecto al análisis automático de imágenes satelitales y su uso en el problema cubierto en este proyecto de fin de carrera, de forma que estos contribuyan con identificar aspectos relacionados que puedan ser desarrollados más adelante en el mismo. Para ello, a continuación, se identificará el tipo de revisión sistemática y las preguntas de investigación, se presentará la estrategia de búsqueda y se reportará los resultados de la misma, de forma que se respondan las preguntas planteadas.

3.2 Objetivos de revisión

Esta revisión sistemática presenta como objetivo el identificar cómo se está aplicando el análisis automático de imágenes satelitales en la determinación y segmentación del nivel socioeconómico, eso es, se va a realizar una revisión del estado del arte respecto al tema del proyecto. Así, la revisión busca responder a la pregunta de investigación principal siguiente: ¿De qué forma el análisis automático de imágenes satelitales se está empleando para determinar y segmentar el nivel socioeconómico de un área observada?

La estructuración de las preguntas de la revisión se apoyó en los criterios PICOC - *Population, Intervention, Comparison, Outcome and Context* en inglés (Petticrew & Roberts, 2006), tal como se muestra en la Tabla 3.1. Para el presente caso, debido a que no se está realizando comparación alguna con otro método, no aplicaría el elemento de «comparación» de PICOC.

Tabla 5.1: Criterios PICOC

Población (<i>Population</i>)	Trabajos sobre la determinación y segmentación del nivel socioeconómico de un área habitada observada
Intervención (<i>Intervention</i>)	Uso del análisis automático de imágenes satelitales para la determinación de niveles socioeconómicos.

Comparación (<i>Comparison</i>)	<i>No aplica</i>
Resultado (<i>Outcome</i>)	Técnicas automáticas utilizadas para la predicción del nivel socioeconómico de un área observada
Contexto (<i>Context</i>)	Estudios de investigación del área de ciencia de la computación orientados a contribuir con el proceso de determinación del nivel socioeconómico de un área estudiada

3.3 Preguntas de revisión

Partiendo de la pregunta de investigación principal, se plantearon las siguientes preguntas específicas para comprender tanto las técnicas empleadas como su contribución con el objetivo del proyecto.

- P1. ¿Cómo las técnicas de visión por computador se están empleando para el análisis automático de imágenes satelitales?
- P2. ¿Cuáles son las ventajas del análisis automático de imágenes satelitales en la determinación y segmentación del nivel socioeconómico de un área observada?
- P3. ¿Cuáles son las características observables en una imagen satelital que contribuyen con la determinación del nivel socioeconómico?

3.4 Estrategia de búsqueda

3.4.1 Motores de búsqueda a usar

Para la ejecución de la revisión sistemática, se decidió emplear los siguientes motores de búsqueda:

- Scopus
- ACM Digital Library

3.4.2 Cadenas de búsqueda a usar

Para identificar qué términos forman parte de la cadena de búsqueda, se empleó como referencia la tabla de criterios PICOC (Tabla 3.1), de los cuales se extrajeron términos discriminantes y se complementó con términos relacionados a los anteriores, como se muestra en la Tabla 3.2. Como los términos asociados al resultado son similares a los de la intervención, este no se toma en cuenta.

Tabla 5.2: Términos derivados de PICOOC

Población (<i>Population</i>)	socioeconomics , income class, income level
Intervención (<i>Intervention</i>)	satellite image* , overhead imagery computer vision , machine learning , neural network, segmentation, detection, deep learning
Resultado (<i>Outcome</i>)	-
Contexto (<i>Context</i>)	computer science

Teniendo lo anterior en cuenta, se unieron los términos mediante conectores AND y OR, y de forma que uno de cada categoría sea obligatoria, pero que cualquiera de cada categoría pueda aparecer.

Scopus:

TITLE-ABS-KEY ("socioeconomic" OR "income level" OR "income class") AND TITLE-ABS-KEY ("satellite imagery" OR "satellite image" OR "overhead imagery") AND TITLE-ABS-KEY ("computer vision" OR "machine learning" OR "neural network" OR "segmentation" OR "detection" OR "deep learning") AND (LIMIT-TO (SUBJAREA , "COMP"))

ACM Digital Library:

[[Full Text: "socioeconomic"] OR [Full Text: "income level"] OR [Full Text: "income class"]] AND [[Full Text: "satellite imagery"] OR [Full Text: "satellite image"] OR [Full Text: "overhead imagery"]] AND [[Full Text: "computer vision"] OR [Full Text: "machine learning"] OR [Full Text: "neural network"] OR [Full Text: segmentation detection] OR [Full Text: "deep learning"]]

3.4.3 Documentos encontrados

En el motor de búsqueda Scopus, el 13 de agosto de 2020, se encontraron 18 documentos con la cadena de búsqueda descrita previamente. Mientras, se encontró 60 documentos en ACM Digital Library. En ambos casos, estos corresponden a resultados previos a la aplicación de los criterios de inclusión y exclusión.

3.4.4 Criterios de inclusión/exclusión

Se incluyen en la revisión aquellos estudios que cumplen con los siguientes criterios de inclusión:

1. El estudio emplea técnicas de visión por computador sobre imágenes satelitales.
2. El estudio reporta resultados relacionados con factores socioeconómicos.
3. Los resultados de la experimentación son validados con datos reales del área observada por la imagen.
4. El estudio está escrito en español o inglés.
5. El estudio fue publicado en los últimos 5 años.

Para responder a las preguntas planteadas, es necesario que se aplique cualquier método de visión por computadora en imágenes satelitales, y que el estudio se aplique a temas socioeconómicos. Asimismo, que los resultados encontrados se corroboren con información real de las entradas empleadas brinda una visión de su utilidad en el problema de este proyecto.

No se tomaron en cuenta los estudios que cumplieron con el siguiente criterio de exclusión:

1. El estudio no realiza un análisis de una zona habitada. Esto se debe a que carece de sentido medir el nivel socioeconómico de una zona inhabitada.

3.5 Formulario de extracción de datos

Se realizó el siguiente formulario de extracción de datos (Tabla 3.3) para facilitar la síntesis de la información obtenida de los estudios de la revisión, lo cual es parte del análisis para la resolución de las preguntas de investigación.

Tabla 5.3: Formulario de extracción de datos

Campo	Descripción	Pregunta
Id		General
Fecha de extracción		General
Autores		General

Tipo de fuente		General
Fuente	Nombre de la fuente	General
Año de publicación		General
Afiliación	Instituciones de afiliación de los autores	General
País	País(es) de afiliación de los autores	General
Técnica empleada	Técnica(s) de visión por computadora empleada(s) para el análisis automático de imágenes de satélite	P1
Aplicación de la técnica	Forma en cómo la técnica se está empleando en las imágenes satelitales	P1
Ventaja principal	Principal ventaja mencionada del análisis automático de imágenes de satélite respecto a métodos tradicionales en la determinación del nivel socioeconómico de una región	P2
Ventaja secundaria	Ventaja secundaria mencionada del análisis automático de imágenes de satélite respecto a métodos tradicionales en la determinación del nivel socioeconómico de una región (de aplicar)	P2
Característica analizada	Característica o componente propia de una imagen satelital relevante para los objetivos del estudio en cuestión (de aplicar)	P3
Elemento relevante	Elemento observable contenido en las imágenes satelitales considerado relevante para los objetivos del estudio en cuestión (de aplicar)	P3

3.6 Resultados de la revisión

Tras el filtrado de las referencias, inicialmente encontradas a través de las cadenas de búsqueda, mediante los criterios de aceptación y exclusión planteados, se obtuvieron 2 (de 18) documentos en Scopus y 3 (de 60), en la biblioteca digital de la ACM, sin presencia de duplicados, los cuales se presentan a continuación:

- Bansal, C., Jain, A., Barwaria, P., Choudhary, A., Singh, A., Gupta, A., & Seth, A. (2020). Temporal Prediction of Socio-Economic Indicators Using Satellite Imagery. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD* (pp. 73–81). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3371158.3371167>
- Han, S., Ahn, D., Park, S., Yang, J., Lee, S., Kim, J., ... Cha, M. (2020). Learning to Score Economic Development from Satellite Imagery. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2970–2979). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3394486.3403347>
- Oshri, B., Hu, A., Adelson, P., Chen, X., Dupas, P., Weinstein, J., ... Ermon, S. (2018). Infrastructure Quality Assessment in Africa Using Satellite Imagery and Deep Learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 616–625). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3219819.3219924>
- Tingzon, I., Orden, A., Go, K. T., Sy, S., Sekara, V., Weber, I., ... Kim, D. (2019). MAPPING POVERTY in the PHILIPPINES USING MACHINE LEARNING, SATELLITE IMAGERY, and CROWD-SOURCED GEOSPATIAL INFORMATION. In T. M. E. A. R. A. A. Blanco A.C. Sarmiento C.J.S. (Ed.), *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* (Vol. 42, pp. 425–431). International Society for Photogrammetry and Remote Sensing. <https://doi.org/10.5194/isprs-archives-XLII-4-W19-425-2019>

Yuan, J., Roy Chowdhury, P. K., McKee, J., Yang, H. L., Weaver, J., & Bhaduri, B. (2018). Exploiting deep learning and volunteered geographic information for mapping buildings in Kano, Nigeria. *Scientific Data*, 5. <https://doi.org/10.1038/sdata.2018.217>

Con la información obtenida en los presentes artículos, la cual fue organizada en los formularios de extracción (Anexo A), se respondió a las preguntas de investigación planteadas al inicio de la revisión sistemática.

3.6.1 Forma de uso de las técnicas de visión por computador en el análisis automático de imágenes satelitales (Pregunta 1)

¿Cómo las técnicas de visión por computador se están empleando para el análisis automático de imágenes satelitales?

Esta pregunta presenta como objetivo conocer cómo las técnicas de visión por computador se están empleando en el análisis automático de imágenes satelitales. En líneas generales, para esta tarea, se emplea tanto aprendizaje automático tradicional como aprendizaje profundo mediante redes neuronales.

En el primer caso, Tingzon *et al.* (2019) - en una de sus técnicas propuestas - y Bansal *et al.* (2020) plantean la extracción de características numéricas de una o más bandas de las imágenes satelitales para emplearlas como entradas para modelos de regresión (árboles de regresión) y clasificación (árboles de clasificación y máquinas de vectores de soporte), respectivamente.

En el segundo caso, Tingzon *et al.* (2019) - en otra de sus técnicas propuestas - y Oshri *et al.* (2018) aplican aprendizaje por transferencia; específicamente, se valen de modelos de redes neuronales convolucionales preentrenadas en conjuntos de imágenes conocidos para afinarlo, con datos específicos del problema, en la tarea específica planteada (clasificación del nivel de ingresos y del tipo de infraestructura, respectivamente). Por otro lado, Yuan *et al.* (2018) propone el entrenamiento de un modelo de segmentación que emplea datos generados de forma colaborativa en conjunto con las imágenes satelitales, mientras que Han *et al.* (2020) formula un modelo por etapas que combina el aprendizaje autosupervisado en redes neuronales convolucionales, la creación de un ordenamiento guiado mediante intervención humana o datos de baja calidad para los clusters generados en la etapa

autosupervisada y un modelo de puntuación que se basa en el orden relativo generado en la etapa anterior.

3.6.2 Ventajas del análisis automático de imágenes satelitales respecto a técnicas tradicionales (Pregunta 2)

¿Cuáles son las ventajas del análisis automático de imágenes satelitales en la determinación y segmentación del nivel socioeconómico de un área observada?

El objetivo de esta pregunta es identificar cuáles son las ventajas de emplear técnicas de análisis automático en imágenes satelitales en la determinación y segmentación del nivel socioeconómico respecto a técnicas tradicionales de recolección de datos como de análisis de imágenes.

Respecto a las primeras, Tingzon *et al.* (2019) afirma que el uso de redes neuronales preentrenadas para estimar el nivel socioeconómico de un área observada constituye un método menos costoso, más eficiente temporalmente y con menor esfuerzo humano para obtener dicha información.

Respecto a las segundas, tanto Han *et al.* (2020) y Yuan *et al.* (2018) mencionan que la aplicación de redes neuronales convolucionales como las descritas en sus investigaciones en el análisis de imágenes satelitales para esta problemática permite realizar inferencias de la variable objetivo incluso ante la escasez o ausencia de valores reales sobre los cual realizar el entrenamiento. Asimismo, según Oshri *et al.* (2018), el modelo de redes residuales que propone obtiene un mayor rendimiento en la clasificación de la infraestructura respecto a métodos como el de la interpolación vecina natural.

Además, Han *et al.* (2020) y Tingzon *et al.* (2019) mencionan, como una ventaja derivada de las anteriores, que estas técnicas son aplicables a países en vías de desarrollo, donde los métodos tradicionales implican un costo excesivo y difícil de afrontar o no se cuentan con datos preexistentes suficientes.

3.6.3 Características de una imagen satelital relevantes para el análisis (Pregunta 3)

¿Cuáles son las características observables en una imagen satelital que contribuyen con la determinación del nivel socioeconómico?

Esta pregunta cuenta como objetivo el identificar qué características presentes en una imagen satelital son relevantes en la determinación del nivel socioeconómico. Estas incluyen tanto componentes internos de una imagen satelital como elementos observables que fueron capturados en foto por esta.

Sobre los primeros, Tingzon *et al.* (2019) y Bansal *et al.* (2020) mencionan que los valores obtenibles en una o más de las bandas que componen una imagen satelital pueden brindar información de entrada útil para la determinación del nivel socioeconómico, tras un tratamiento previo que convierte estas capas en características numéricas unidimensionales.

Sobre los segundos, Han *et al.* (2020) y Yuan *et al.* (2018) señalan que los edificios por separado como los bloques de edificios en entornos urbanos son indicadores para la identificación de variables socioeconómicas, entre ellas el nivel de desarrollo económico en las regiones observadas.

3.7 Conclusiones

- El presente capítulo presentó los resultados de la ejecución de una revisión sistemática del estado del arte de la aplicación del análisis automático de imágenes satelitales en la determinación y segmentación del nivel socioeconómico. Esta presentó como objetivo el recopilar y analizar investigaciones referidas a dicho tema, enfocándose en la forma en cómo las técnicas de visión por computador han sido empleadas en el análisis automático, los beneficios de este en la resolución de la problemática comparado con otras técnicas y los elementos contenidos en las imágenes satelitales que contribuyen con esta solución.
- Inicialmente, se hallaron 78 trabajos previos entre las dos bases de datos consultadas. Tras la aplicación de los criterios de inclusión y exclusión, se seleccionaron 5 de estos trabajos para su análisis y compilación mediante formularios de extracción, de forma que se respondiera a las preguntas planteadas.
- Sobre la forma de uso de las técnicas mencionadas en el análisis automático de imágenes de satélite, dos de los estudios emplearon la extracción de características numéricas para el entrenamiento de modelos de aprendizaje automático convencional, mientras que cuatro estudios (incluyendo uno de los que

también propuso el uso de aprendizaje automático tradicional) plantearon el uso de aprendizaje profundo mediante redes neuronales convolucionales.

- Respecto a las ventajas del uso del análisis automático en la determinación del nivel socioeconómico, un estudio afirmó que el análisis automático resulta menos costoso en tiempo, recursos y esfuerzo respecto a técnicas tradicionales de recolección de este tipo de datos. Asimismo, otros tres estudios afirmaron que puede brindar resultados coherentes y significativos incluso ante la escasez o ausencia de datos etiquetados previos. Como conclusión de lo afirmado anteriormente, dos de ellos también sostienen que el análisis automático resulta aplicable a países en vías de desarrollo, donde otros métodos son difíciles o imposibles de ejecutar.
- Acerca de las características de las imágenes satelitales relevantes para el problema planteado, dos trabajos mencionan que emplean una o más bandas que componen las imágenes satelitales en su solución. Mientras, otros dos mencionan que la aparición de edificios en estas contribuye con la detección de variables socioeconómicas como la búsqueda.
 - Los resultados mencionados constituyen un resumen del estado del arte del uso del análisis automático de imágenes satelitales en la determinación del nivel socioeconómico, el cual permite entender las formas uso de este método y sus beneficios en la tarea en cuestión, de forma que resulta útil para el desarrollo del presente proyecto de fin de carrera.

Capítulo 4. Integrar datos visuales con datos censales

4.1 Introducción

El presente capítulo comprende los avances correspondientes al primer módulo de resultados de este proyecto de fin de carrera, referido al objetivo de integrar datos visuales con datos censales. Para ello, en el conjunto de las cuatro secciones de este capítulo, se buscará conformar un conjunto de imágenes satelitales de zonas urbanas con regiones etiquetadas en base a datos censales a través del tratamiento de imágenes satelitales y datos censales georreferenciados, pero independientes unos de otros.

Con este primer módulo, se buscar conformar un corpus que agrupe de forma organizada las imágenes satelitales producidas por el satélite PeruSat-1 y datos censales socioeconómicos georreferenciados obtenidos como resultado del XII Censo de Población y VII de Vivienda, organizados por el Instituto Nacional de Estadística e Informática (INEI) en el año 2017, de forma que este resultado sirva de entrada para los modelos de aprendizaje profundo que se propondrán en el módulo siguiente.

4.2 Resultado alcanzado: Conjunto de imágenes satelitales con regiones etiquetadas en base a los datos censales

El contenido de esta sección abarca todo el proceso para la construcción del conjunto de imágenes satelitales etiquetados con información censal, el cual constituye el resultado del presente objetivo, para el futuro entrenamiento del modelo de redes neuronales que será planteado más adelante, desde los tratamientos previos de ambos tipos de datos hasta su integración. Dado el tamaño de los archivos, los medios de verificación para el presente resultado no se encuentran adjuntos a este documento. Una declaración del cumplimiento del resultado se encuentra en el **anexo C (Declaración de resultado alcanzado)**.

4.2.1 Transformación de la base de datos censales

Se cuenta inicialmente con una base de datos espacial (*geodatabase*) en formato GDB correspondiente a todos los distritos de Lima Metropolitana, presentada en el documento «Planos estratificados de Lima Metropolitana a nivel de manzanas según Ingreso Per Cápita del Hogar» (Instituto Nacional de Estadística e Informática, 2020). Esta base de datos comprende 102 406 etiquetas georreferenciadas que

asocian un estrato de nivel de ingresos (bajo, medio bajo, medio, medio alto, alto) a una manzana de la ciudad (figura 3a).

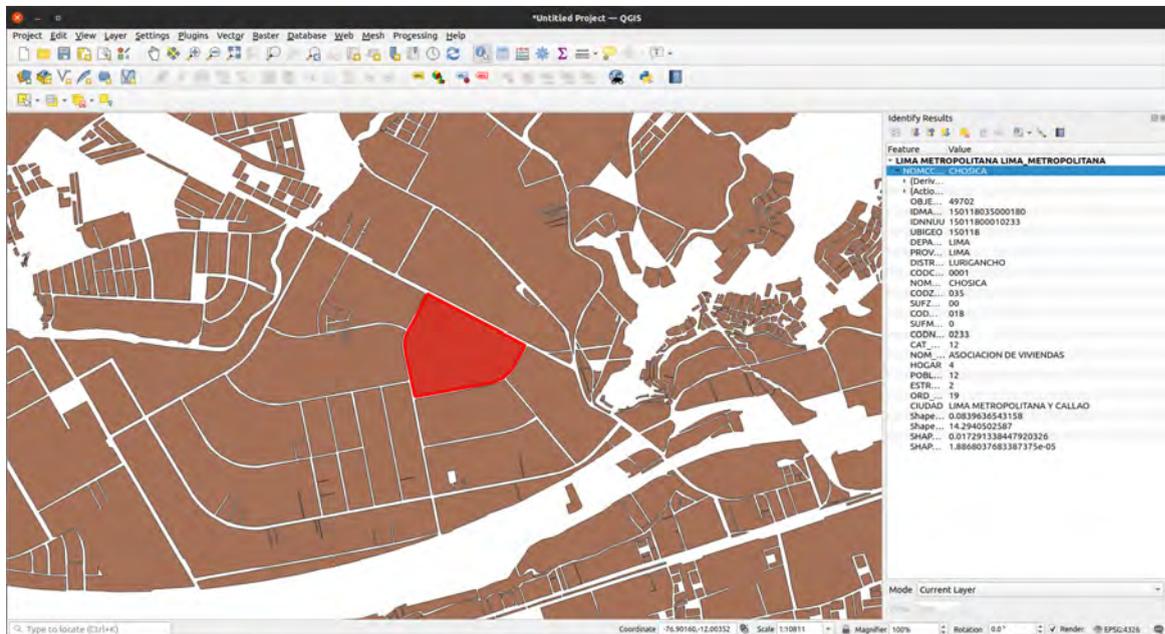


Figura 3a. Visualización en QGIS de la base de datos espacial de la publicación «Planos estratificados de Lima Metropolitana a nivel de manzanas según Ingreso Per Cápita del Hogar» y las características asociadas a una etiqueta.

Fuente: Elaboración propia

Para poder manipular las etiquetas contenidas en la base de datos al momento de integrar los datos y seleccionar solo las etiquetas correspondientes a cada imagen, se exportó esta en una base de datos relacional que funciona mediante el motor de SQLite. Para ello, se importó como capa de vectores al directorio GDB en un proyecto de QGIS (figura 3b) para después exportar dicha capa en el formato especificado (figura 3c).

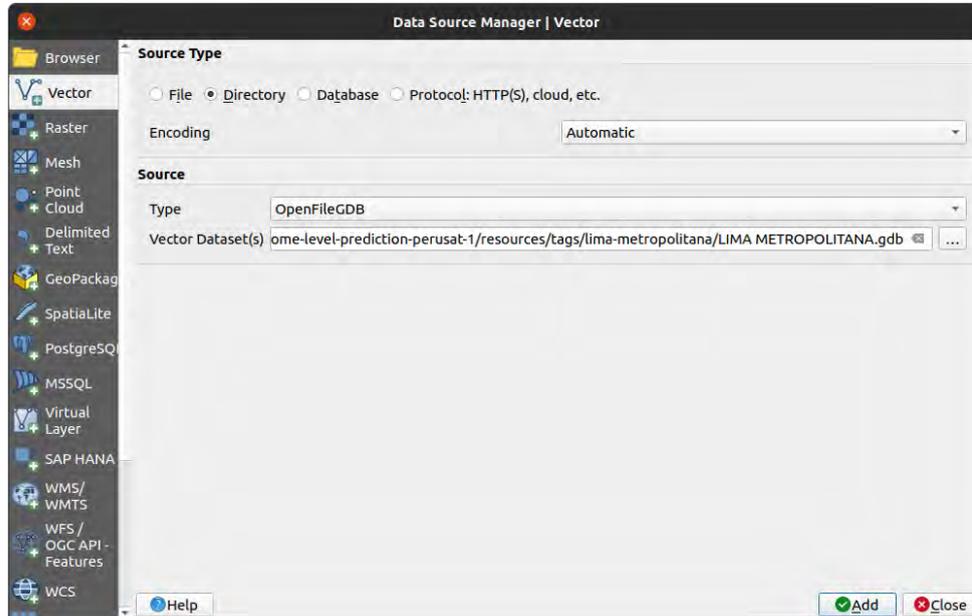


Figura 3b. Ventana de QGIS para la importación del directorio como una capa de vectores.

Fuente: Elaboración propia

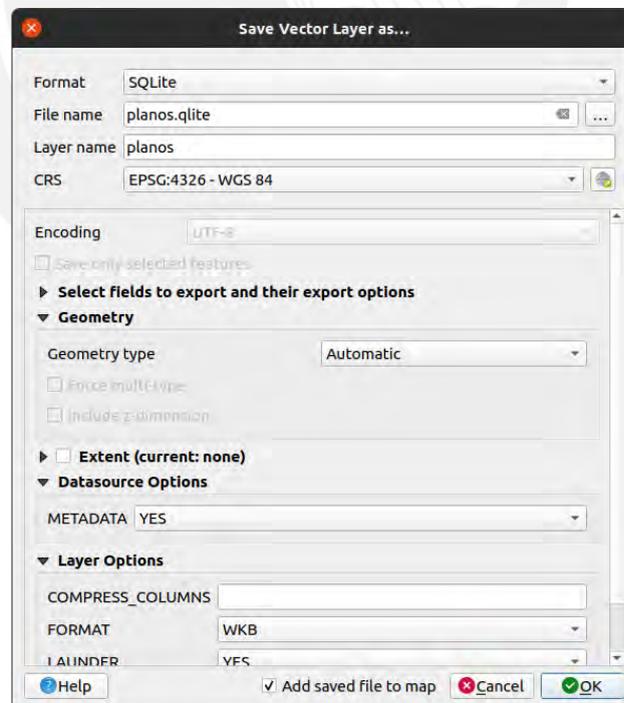


Figura 3c. Ventana de QGIS para la exportación de la capa de vectores en un archivo SQLite.

Fuente: Elaboración propia

4.2.2 Tratamiento de imágenes satelitales

Inicialmente, se solicitó a la Agencia Espacial del Perú - CONIDA 5 productos que contienen imágenes satelitales correspondientes al territorio de Lima Metropolitana durante los años cercanos a la recolección de los datos censales detallados en la subsección anterior. En estos productos, se obtuvo un grupo inicial de 13 imágenes satelitales, tanto en formato pancromático (una sola capa de alta resolución) como multiespectral (multicapa de baja resolución), de Lima Metropolitana capturadas entre diciembre de 2016 y abril de 2020. Tras el descarte de imágenes que cubrían zonas ya cubiertas por otras, así como imágenes satelitales mayormente rurales y con pocas zonas urbanas, quedó un grupo refinado de 8 imágenes satelitales de Lima Metropolitana.

A las imágenes seleccionadas, se les aplicó un proceso de refinado pancromático (*pansharpening*, figura 5) para obtener una imagen que mantenga los colores de la imagen multiespectral y la resolución de la pancromática (Padwick, Deskevich, Pacifici, Smallwood, 2010). Para dicho proceso, se implementó un *script* en bash que automatiza el acceso, la apertura y la ejecución de dicho refinado. A continuación, se muestra dicho código (figura 6a), la estructura de carpetas esperada por el mismo (figura 6b) y la lista de imagen ya tratadas (figura 6c):

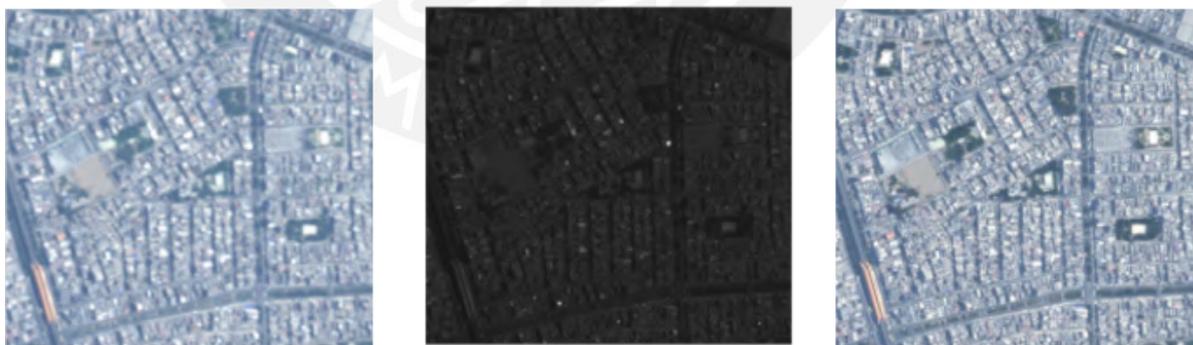


Figura 4. De izquierda a derecha: imagen multiespectral, imagen pancromática e imagen tras refinado pancromático.

Fuente: CONIDA

```
pansharpen.sh
1 #!/bin/bash
2
3 # Variables
4 origin_dir=$1
5 output_dir=$2
6 search_term=$3
7
8 # Search files
9 list=$(find "$origin_dir" -maxdepth 2 -type d -name "$search_term")
10
11 mkdir "$output_dir"
12
13 for folder in $list
14 do
15     ms_tif=$(find "$folder" -name "*ORT_MS*.TIF")
16     p_tif=$(find "$folder" -name "*ORT_P*.TIF")
17
18     echo "Panchromatic: ${p_tif[0]}"
19     echo "Multispectral: ${ms_tif[0]}"
20
21     basename_tif=$(basename "${p_tif[0]}")
22     echo "Storing result in $output_dir/$basename_tif"
23     echo " "
24
25     gdal_pansharpen.py "${p_tif[0]}" "${ms_tif[0]}" "$output_dir/$basename_tif"
26 done
```

Figura 5a. Código fuente del «script» de refinado pancromático.

Fuente: Elaboración propia

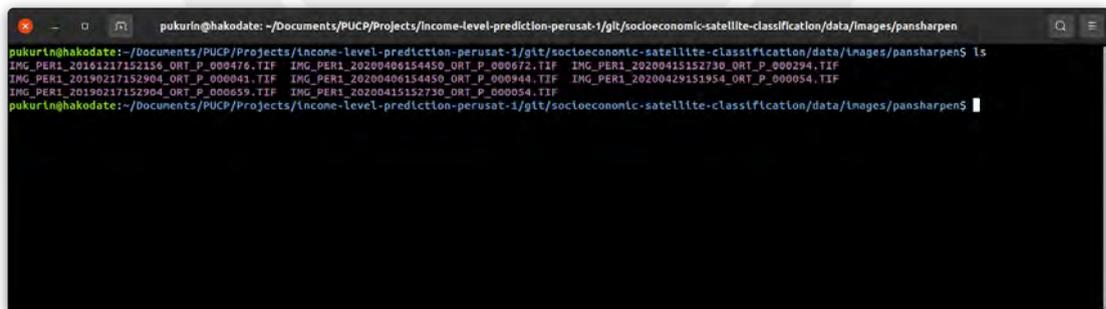


Figura 5b. Imágenes satelitales tras el refinado pancromático.

Fuente: Elaboración propia

```

DELIVERY.PDF
INDEX.HTM
LIBRARY
├── DS_LOGO.JPG
├── DS_STYLE.XSL
LICENSE.PDF
PER_LIST.XML
VOL_PER1_OR1_001_000041
├── IMG_PER1_OR1_MS_000041
│   ├── DIM_PER1_20161217152156_OR1_MS_000041.XML
│   ├── ICON_PER1_20161217152156_OR1_MS_000041.JPG
│   ├── IMG_PER1_20161217152156_OR1_MS_000041.TFW
│   ├── IMG_PER1_20161217152156_OR1_MS_000041.TIF
│   ├── INDEX.HTM
│   ├── LIBRARY
│   │   ├── LOGO.JPG
│   │   └── STYLE.XSL
│   ├── LINEAGE
│   │   └── PROCESSING_PER1_20161217152156_OR1_MS_000041_DIM.XML
│   ├── PREVIEW_PER1_20161217152156_OR1_MS_000041.JPG
│   ├── PREVIEW_PER1_20161217152156_OR1_MS_000041.KMZ
│   └── RPC_PER1_20161217152156_OR1_MS_000041.XML
├── IMG_PER1_OR1_P_000041
│   ├── DIM_PER1_20161217152156_OR1_P_000041.XML
│   ├── ICON_PER1_20161217152156_OR1_P_000041.JPG
│   ├── IMG_PER1_20161217152156_OR1_P_000041.TFW
│   ├── IMG_PER1_20161217152156_OR1_P_000041.TIF
│   ├── INDEX.HTM
│   ├── LIBRARY
│   │   ├── LOGO.JPG
│   │   └── STYLE.XSL
│   ├── LINEAGE
│   │   └── PROCESSING_PER1_20161217152156_OR1_P_000041_DIM.XML
│   ├── PREVIEW_PER1_20161217152156_OR1_P_000041.JPG
│   ├── PREVIEW_PER1_20161217152156_OR1_P_000041.KMZ
│   └── RPC_PER1_20161217152156_OR1_P_000041.XML
├── INDEX.HTM
├── LIBRARY
│   ├── VOL_LOGO.JPG
│   └── VOL_STYLE.XSL
└── PER_VOL.XML
VOL_PER1_OR1_001_000476
├── IMG_PER1_OR1_MS_000476
│   ├── DIM_PER1_20161217152156_OR1_MS_000476.XML
│   ├── ICON_PER1_20161217152156_OR1_MS_000476.JPG
│   ├── IMG_PER1_20161217152156_OR1_MS_000476.TFW
│   ├── IMG_PER1_20161217152156_OR1_MS_000476.TIF
│   ├── INDEX.HTM
│   ├── LIBRARY
│   │   ├── LOGO.JPG
│   │   └── STYLE.XSL
│   ├── LINEAGE
│   │   └── PROCESSING_PER1_20161217152156_OR1_MS_000476_DIM.XML
│   ├── PREVIEW_PER1_20161217152156_OR1_MS_000476.JPG
│   ├── PREVIEW_PER1_20161217152156_OR1_MS_000476.KMZ
│   └── RPC_PER1_20161217152156_OR1_MS_000476.XML
├── IMG_PER1_OR1_P_000476
│   ├── DIM_PER1_20161217152156_OR1_P_000476.XML
│   ├── ICON_PER1_20161217152156_OR1_P_000476.JPG
│   ├── IMG_PER1_20161217152156_OR1_P_000476.TFW
│   ├── IMG_PER1_20161217152156_OR1_P_000476.TIF
│   ├── INDEX.HTM
│   ├── LIBRARY
│   │   ├── LOGO.JPG
│   │   └── STYLE.XSL
│   ├── LINEAGE
│   │   └── PROCESSING_PER1_20161217152156_OR1_P_000476_DIM.XML
│   ├── PREVIEW_PER1_20161217152156_OR1_P_000476.JPG
│   ├── PREVIEW_PER1_20161217152156_OR1_P_000476.KMZ
│   └── RPC_PER1_20161217152156_OR1_P_000476.XML
├── INDEX.HTM
├── LIBRARY
│   ├── VOL_LOGO.JPG
│   └── VOL_STYLE.XSL
└── PER_VOL.XML

```

Figura 5c. Ejemplo de carpeta (producto de PeruSat-1) esperada por el «script» de refinamiento pancromático. Fuente: Elaboración propia

4.2.3 Conformación del conjunto de imágenes con datos censales integrados

Con la base de datos de datos censales e imágenes satelitales preparadas para la integración, se procedió a recortar las imágenes satelitales en bloques de 512x512 píxeles y generar capas máscara para dichos bloques, las cuales contienen etiquetas de manzanas a nivel general (512x512 píxeles) como segmentado por estratos de ingresos, a través de la base de datos censales. Para el presente proyecto; se consideró dos conjuntos de niveles: en el primero, se optó por definir dos niveles, agrupando los niveles «bajo», «medio bajo» y «medio» en un nivel llamado «bajo o medio» y los niveles «medio alto» y «alto», en uno llamado «medio o alto»; en el segundo, se optó por mantener la división entre los niveles «medio alto» y «alto», con lo que se obtiene un total de tres niveles. En las figuras 6a y 6b, puede observarse una muestra del conjunto de datos.

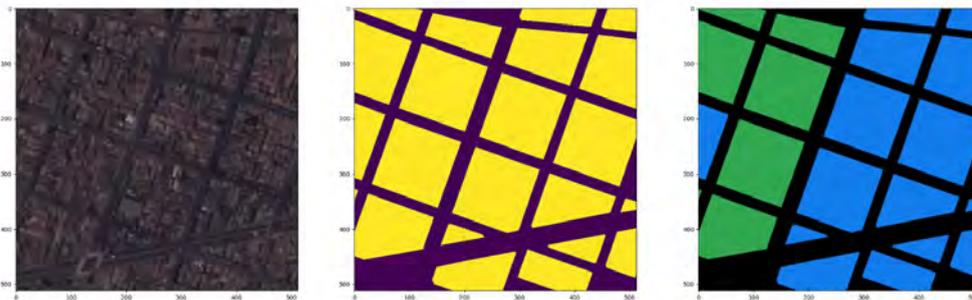


Figura 6a. De izquierda a derecha: bloque (fragmento) de una imagen satelital, etiqueta de manzanas y etiqueta segmentada por dos niveles socioeconómicos (en verde, «nivel bajo o medio»; en azul, «nivel medio o alto»). Fuente: *Elaboración propia*

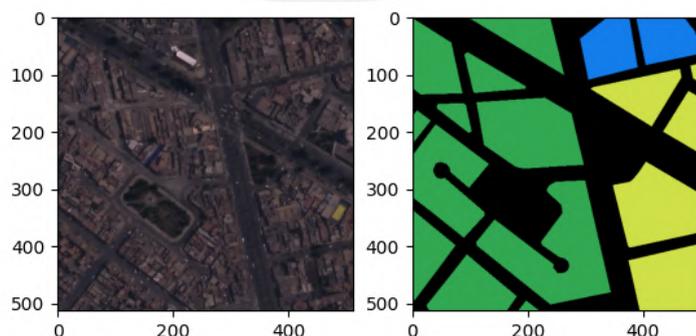


Figura 6b. Izquierda: bloque (fragmento) de una imagen satelital. Derecha: etiqueta segmentada por dos niveles socioeconómicos (en verde, «nivel bajo o medio»; en azul, «medio alto»; en amarillo, «alto»). Fuente: *Elaboración propia*

Durante la integración entre los datos censales y las imágenes satelitales, debido a la diferencia de los sistemas de coordenadas entre ambas fuentes de datos, la ubicación de las etiquetas en la máscara no coincidían inicialmente con su posición en las imágenes, por lo que fue necesario ajustar los valores de las coordenadas de las etiquetas. Dichos valores fueron calculados de forma manual y por cada imagen solicitada. Sin embargo, este ajuste no corrigió el error en todos los casos, por lo que se descartaron aquellos bloques en los que las etiquetas no coincidían con lo mostrado en dicho bloque incluso después del ajuste.

Asimismo, no se consideró para la conformación del conjunto de datos bloques generados que no contaban con etiquetas asociadas (mayormente, cuerpos de agua o zonas montañosas en los límites de Lima urbana), que contaban con un 5 % o más de píxeles negros, que habían capturado nubes o que contaban con viviendas rodeadas de zonas no asfaltadas, cercanas a zonas extensas de tierra, con el objetivo de formar un corpus sin ruido.

Después de ambos criterios de descarte, se cuenta con 118 bloques de imágenes satelitales etiquetados a nivel de manzanas y por nivel socioeconómico.

4.3 Discusión

El resultado esperado en el presente capítulo es el conjunto de imágenes satelitales con regiones etiquetadas con información socioeconómica censal. Esto se obtuvo gracias a la asociación de imágenes satelitales obtenidas mediante el satélite PeruSat-1 y una base de datos de etiquetas georreferenciadas publicada por el Instituto Nacional de Estadística e Informática.

Respecto a integración de los datos censales con los datos visuales, una posible causa de la imposibilidad en el presente proyecto de acoplar completamente las etiquetas a los bloques de imágenes satelitales generadas, dificultad mencionada en la última subsección del resultado esperado, son las aparentes diferencias en la forma de georreferenciar los datos censales, respecto a la georreferenciación de las imágenes satelitales. En futuros trabajos, convendría integrar ambos tipos de datos - censal y satelital - desde el momento de la georreferenciación para evitar dichas diferencias o, de no ser posible, poder conocer cómo ha sido ejecutado el proceso de georreferenciación, de forma que se disponga de toda la información al momento de la integración de los datos.

El presente resultado fue posible de obtener debido a la existencia previa del estudio «Planos estratificados de Lima Metropolitana a nivel de manzanas según Ingreso Per Cápita del Hogar» mencionado anteriormente, que cuenta con datos censales georreferenciados, así como de imágenes satelitales despejadas de Lima Metropolitana. Así, realizar un corpus similar a este enfocado en otras ciudades del país requeriría un preprocesamiento de los datos obtenidos mediante el Censo de Población y Vivienda de 2017. Asimismo, la disponibilidad de imágenes satelitales con baja nubosidad es menor en las ciudades de la costa que en otras regiones del país, por lo que resultaría más difícil generar un corpus como este de algunas ciudades de la costa poco capturadas por el satélite PeruSat-1.



Capítulo 5. Elaborar un criterio automático de los resultados de determinación del nivel socioeconómico

5.1 Introducción

Este capítulo comprende los avances respectivos al segundo módulo de resultados del presente proyecto de fin de carrera, los cuales contribuyen a elaborar un criterio automático de determinación del nivel socioeconómico a través del procesamiento en modelos de redes neuronales de imágenes satelitales. Esto incluye la elaboración de un protocolo de preprocesamiento de un conjunto de datos para el entrenamiento de estos modelos de arquitectura de redes neuronales, la implementación de los mismos y finalmente el propio entrenamiento para que pueda constituirse como criterio.

5.2 Resultado alcanzado

5.2.1 Protocolo de preprocesamiento de imágenes satelitales etiquetadas para su procesamiento por parte de los modelos de red neuronal

El propósito de contar con un protocolo de preprocesamiento de imágenes satelitales es contar con un método para acondicionar las imágenes satelitales y sus etiquetas para servir de entrada para los modelos de redes neuronales elaborados como parte de este objetivo.

Para el presente protocolo, se requiere indicar cómo se va a dividir el conjunto de datos (es decir, qué porcentaje de estos servirán para el entrenamiento de la red y qué porcentaje, para la validación de la misma), así como para qué modelo se está realizando el preprocesamiento (el modelo de segmentación semántica de techos o de extracción de variables socioeconómicas).

Así, este inicia con la división del conjunto de datos en los subconjuntos mencionados en el párrafo anterior, de acuerdo a los porcentajes especificados. Para esta división, se reparten de forma pseudoaleatoria los elementos del conjunto de datos original en los subconjuntos.

A continuación, se calcula los valores máximos, medios y la desviación estándar del subconjunto de entrenamiento, los cuales serán empleados en una transformación posterior en el protocolo de preprocesamiento. Estos valores están representados mediante vectores, cuyos componentes representan dichos valores por

cada capa que compone una imagen satelital (4 bandas). Luego, se completa con los valores medios los espacios con valores nulos que se encuentran en el conjunto de datos, de forma que estos no afecten negativamente el entrenamiento.

Posteriormente, se define las funciones de transformación que se aplicará a las imágenes de los conjuntos de entrenamiento y de pruebas. Para los elementos del subconjunto de entrenamiento, se aplican las siguientes funciones:

- Volteo horizontal
- Volteo vertical
- Rotación pseudoaleatoria
- Normalización: se centran los valores en 0 y con varianza de valor 1, de acuerdo con los valores medios y la desviación estándar

Mientras, al subconjunto de validación solo se aplica la normalización. Finalmente, se define cómo el modelo va a recuperar las imágenes y las etiquetas, y cómo se van a aplicar las transformaciones respectivas.

La implementación del presente protocolo de preprocesamiento se realizó en el lenguaje de programación Python. Para ello, la lectura de las imágenes satelitales del conjunto de datos se realizó desde la biblioteca Rasterio y para la manipulación de estas, además de las máscaras, como matrices se utilizó NumPy. Esta implementación se encuentra referenciada en el **anexo D (Repositorios de código)**, en el apartado «Modelos de arquitectura de redes neuronales y protocolo de preprocesamiento», en la que se encuentra integrada al proceso de entrenamiento.

5.2.2 Modelo de arquitectura de red neuronal para la segmentación semántica de regiones urbanas en base a techos implementado

5.2.2.1 Arquitectura

Para el presente modelo de arquitectura de red neuronal, cuyo objetivo es segmentar los techos presentes en una imagen satelital, se ha optado por partir de una versión modificada de la red neuronal UNet, conocida como TernausNet (figura 8), que consiste en la red base con un codificador VGG11 (Igloukov, Shvets, 2018).

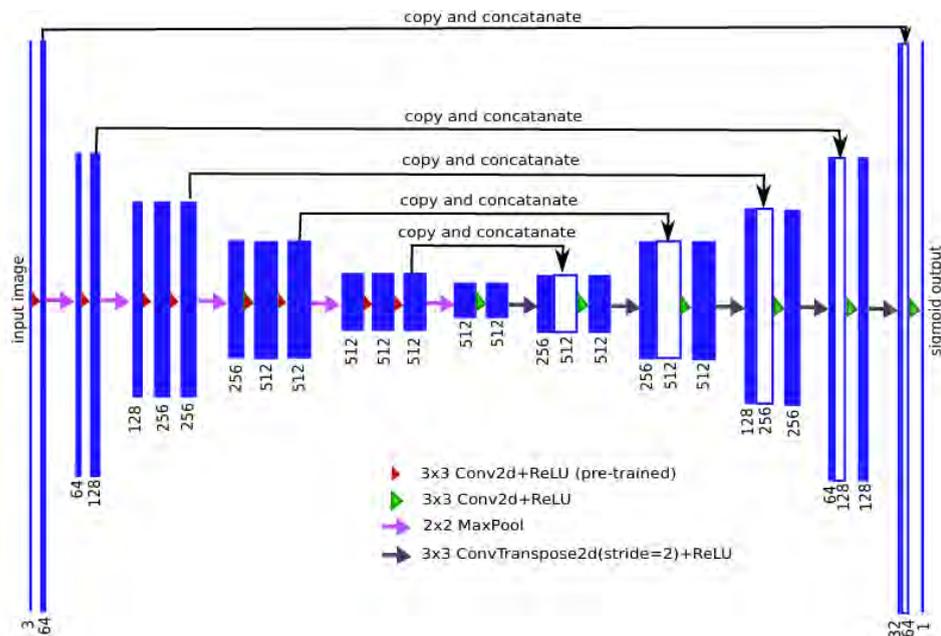


Figura 7. Arquitectura de TeraNet.

Fuente: Iglovikov, Shvets

En esta arquitectura, la entrada original es objeto de sucesivas convoluciones, pasos por funciones de activación y reducciones de dimensionalidad para luego pasar por convoluciones, funciones de activación y aumento de dimensionalidad hasta recuperar las dimensiones de la entrada original, aunque con una cantidad de capas de acuerdo con las dimensiones de la respuesta.

En el caso particular de este proyecto, la red recibe una imagen de 4 x 512 x 512 píxeles (4 bandas de 512 x 512) y devuelve una salida de 1 x 512 x 512, con una única capa que representa los techos existentes en la entrada de referencia.

Para la codificación de la red neuronal, se optó por el lenguaje de programación Python, debido a la existencia de bibliotecas que contienen módulos y clases previamente implementadas para la construcción de redes neuronales. Dentro de dichas bibliotecas, se optó por el marco de trabajo (*framework*) PyTorch, debido a la existencia de trabajos previos en otros problemas de segmentación de imágenes implementados con dicha herramienta. La implementación de la presente red se encuentra referenciada en el **anexo D (Repositorios de código)**, en el apartado «Modelos de arquitectura de redes neuronales y protocolo de preprocesamiento», mientras una representación a nivel de arquitectura, incluyendo capas y cantidad de

parámetros se encuentra en el **anexo E (Estructura de los modelos de arquitectura de red neuronal)**, en el apartado «Modelo de arquitectura de segmentación semántica de regiones urbanas en base a techos».

5.2.2.2 Entrenamiento

Para este primer modelo de red neuronal, se optó por dividir el conjunto de datos con los que cuenta el proyecto en un subconjunto de entrenamiento del 90 %, mientras que el 10 % conforma el subconjunto de validación. Asimismo, como criterio de evaluación del aprendizaje del modelo, se emplea la función de intersección sobre unión (índice Jaccard), a través de su complemento (función de pérdida Jaccard).

Para la presente red, se obtuvo un menor valor en la función de pérdida Jaccard con los siguientes valores para los hiper parámetros a continuación:

Tabla 6.1. Valores de los hiperparámetros para el entrenamiento

Número de épocas	200
Tasa de aprendizaje inicial	0.001
Factor de decaimiento de la tasa de aprendizaje (<i>gamma</i>)	0.1
Número de épocas entre decaimientos (<i>step size</i>)	60
Número de imágenes por bloques (<i>batch size</i>)	4

Para los siguientes valores de hiperparámetros, se obtuvo los siguientes valores de rendimiento:

Tabla 6.2. Reporte de rendimiento del modelo de segmentación semántica de regiones urbanas en base a techos

Criterio de evaluación	Entrenamiento	Validación	Total
Índice Jaccard	70,3861 %	69,5680 %	70,3043 %
Función de pérdida Jaccard	29,6139 %	30,4320 %	29,6957 %

La evolución del aprendizaje, medido a través de la función de pérdida Jaccard, del modelo se encuentra descrito en la figura 8.

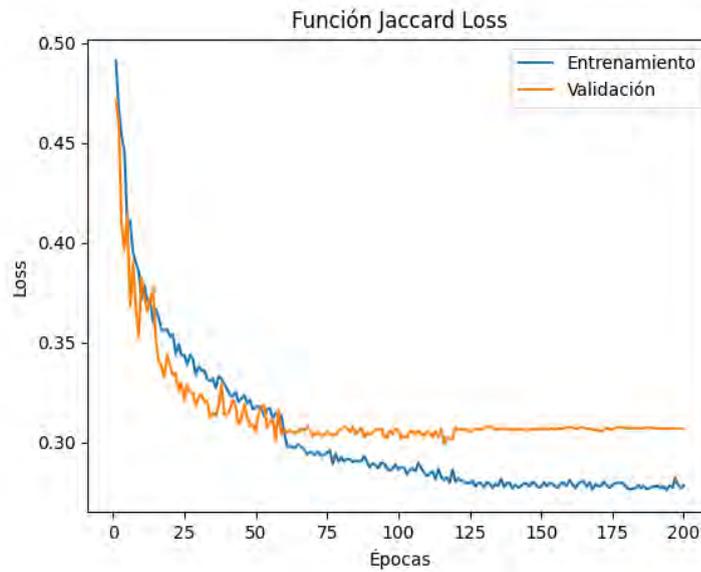


Figura 8. *Evolución de la función de pérdida Jaccard a través de las épocas para el modelo de segmentación semántica de techos.*

Fuente: Elaboración propia

Se incluye en la presente sección una muestra visual de los resultados del entrenamiento del modelo de segmentación semántica, en la cual se compara la etiqueta real y la predicción otorgada por el modelo (figura 9).

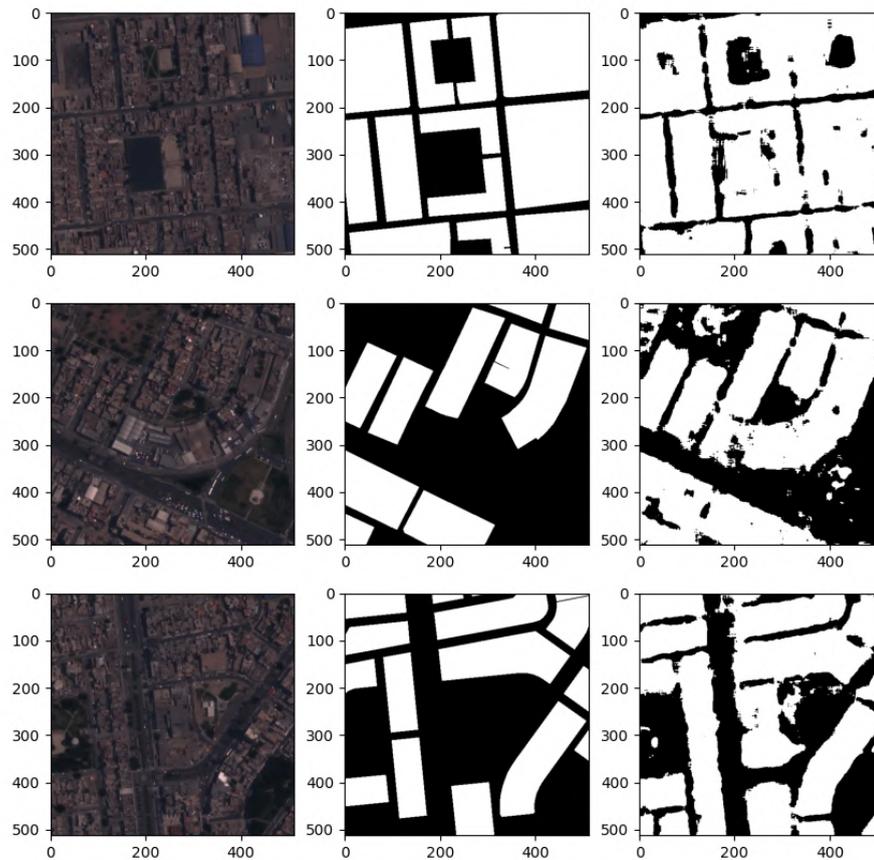


Figura 9. De izquierda a derecha: imágenes satelitales, etiquetas de manzanas pertenecientes al conjunto de datos, predicción de las etiquetas generada por el modelo de segmentación semántica de techos.

Fuente: Elaboración propia

5.2.3 Modelo de arquitectura basado en red neuronal convolucionales para la extracción de variables socioeconómicas en imágenes satelitales implementado.

5.2.3.1 Arquitectura

Para este segundo modelo de arquitectura, enfocado en determinar el nivel socioeconómico de una zona urbana tomando como entradas una imagen satelital y un mapa de techos, generados por el modelo anterior, también se ha optado por utilizar como modelo base la red TerausNet (figura 8), debido a que esta también es una tarea de segmentación, aunque con más de una etiqueta respuesta, es decir, existe más de una clase (niveles socioeconómicos) en la salida.

Dada estas diferencias, la estructura del presente modelo se diferencia del anterior en que la entrada, al considerar también el resultado del modelo anterior, está presenta una dimensión de 5 x 512 x 512 píxeles, mientras que la salida está compuesta por 2 capas de 512 x 512 x 512 (2x512x512), en la que cada capa representa si tal pixel pertenece a un nivel dado.

Debido a las similitudes respecto al modelo de red neuronal anterior y a las herramientas utilizadas para el desarrollo del mismo, se decidió mantener las mismas tecnologías descritas en el resultado anterior para el desarrollo de este modelo. La implementación de la presente red se encuentra referenciada en el **anexo C (Repositorios de código)**, en el apartado «Modelos de arquitectura de redes neuronales y protocolo de preprocesamiento», mientras una representación a nivel de arquitectura, incluyendo capas y cantidad de parámetros se encuentra en el **anexo D (Estructura de los modelos de arquitectura de red neuronal)**, en el apartado «Modelo de arquitectura para la extracción de variables socioeconómicas en imágenes satelitales».

5.2.3.2 Entrenamiento

A diferencia del modelo anterior, se optó por aumentar el tamaño del subconjunto de validación, el cual es de 25 % respecto al total de imágenes del conjunto de datos. Para la medición numérica del aprendizaje, se emplearon los mismos criterios que para el de segmentación semántica de techos.

Para la presente red, con el sistema de dos niveles («medio o bajo» y «medio o alto»), se obtuvo un menor valor en la función de pérdida Jaccard con los siguientes valores para los hiper parámetros a continuación:

Tabla 7.1. Valores de los hiperparámetros para el entrenamiento del modelo de extracción de variables socioeconómicas para dos niveles socioeconómicos

Número de épocas	1000
Tasa de aprendizaje inicial	0.001
Factor de decaimiento de la tasa de aprendizaje (<i>gamma</i>)	0.1
Número de épocas entre decaimientos (<i>step size</i>)	200
Número de imágenes por bloques (<i>batch size</i>)	6

Para los siguientes valores de hiperparámetros, se obtuvo los siguientes valores de rendimiento:

Tabla 7.2. Reporte de rendimiento del modelo de extracción de variables socioeconómicas para dos niveles socioeconómicos

Criterio de evaluación	Entrenamiento	Validación	Total
Índice Jaccard	33,8200 %	30,1934 %	32,9134 %
Función de pérdida Jaccard	66,1800 %	69,8066 %	67,0866 %

La evolución del aprendizaje, medido a través de la función de pérdida Jaccard, del modelo se encuentra descrito en la figura 10.

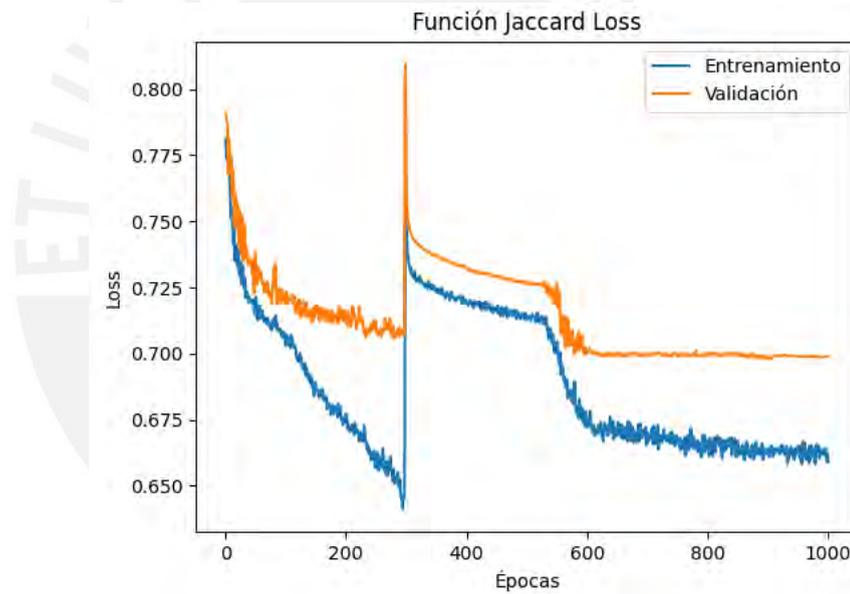


Figura 10. Evolución de la función de pérdida Jaccard a través de las épocas para el modelo de extracción de variables socioeconómicas para dos niveles socioeconómicos.

Fuente: Elaboración propia

En un principio, los resultados obtenidos por el presente modelo no lograban diferenciar con exactitud cuáles partes de la imagen corresponden a techos y cuáles no. Para limitar la predicción a solamente manzanas, se decidió emplear nuevamente la predicción del modelo anterior como máscara de recorte, de forma que solo se emita una predicción de acuerdo a lo que el anterior modelo considera como techos, como se muestra en la figura 11.

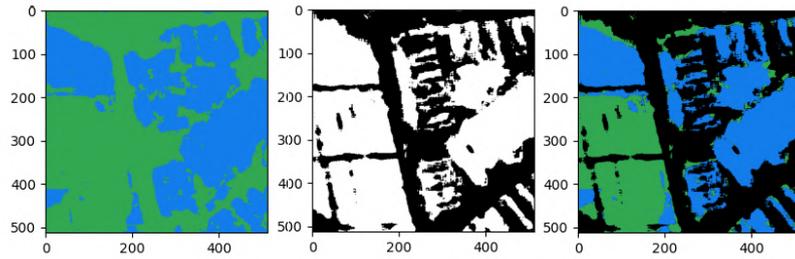


Figura 11. De izquierda a derecha: predicción del nivel socioeconómico original, predicción del modelo de segmentación de techos para la imagen, predicción original tras ser aplicado la predicción de techos como máscara.

Fuente: Elaboración propia

Se incluye en la presente sección una muestra visual de los resultados del entrenamiento del modelo de segmentación semántica, en la cual se compara la etiqueta real y la predicción otorgada por el modelo (figura 12).

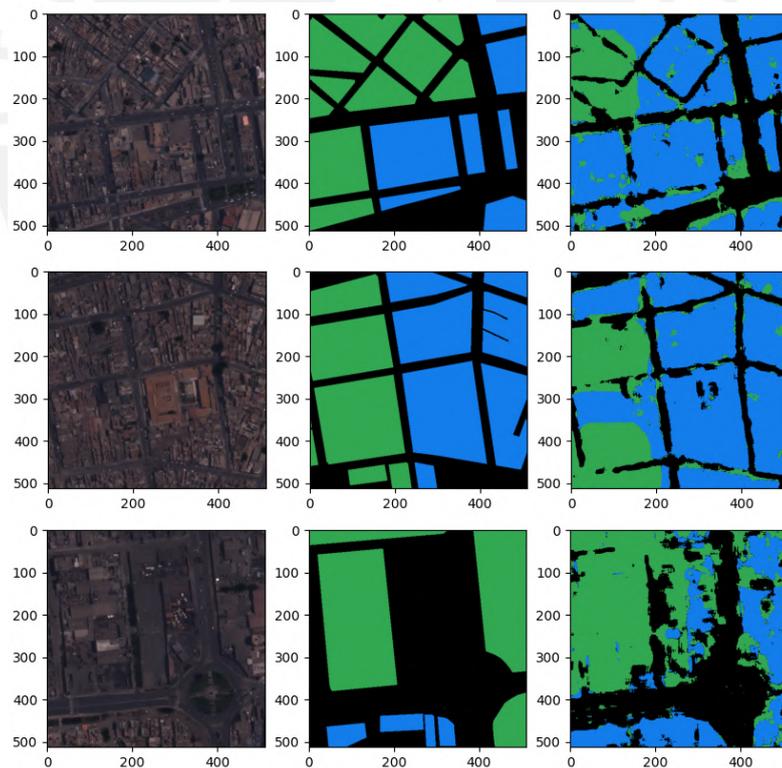


Figura 12. De izquierda a derecha: imágenes satelitales, etiquetas de nivel socioeconómico, predicción de las etiquetas generada por el modelo de extracción de variables socioeconómicas.

Fuente: Elaboración propia

Empleando como referencia el sistema de tres niveles («medio o bajo», «medio alto» y «alto»), se obtuvo un menor valor en la función de pérdida Jaccard con los siguientes valores para los hiper parámetros a continuación:

Tabla 8.1. Valores de los hiperparámetros para el entrenamiento del modelo de extracción de variables socioeconómicas para dos niveles socioeconómicos

Número de épocas	305
Tasa de aprendizaje inicial	0.001
Factor de decaimiento de la tasa de aprendizaje (<i>gamma</i>)	0.1
Número de épocas entre decaimientos (<i>step size</i>)	60
Número de imágenes por bloques (<i>batch size</i>)	4

Para los siguientes valores de hiperparámetros, se obtuvo los siguientes valores de rendimiento:

Tabla 8.2. Reporte de rendimiento del modelo de extracción de variables socioeconómicas para tres niveles socioeconómicos

Criterio de evaluación	Entrenamiento	Validación	Total
Índice Jaccard	27,9006 %	26,3936 %	27,5239 %
Función de pérdida Jaccard	72,0994 %	73,6064 %	72,4761 %

La evolución del aprendizaje, medido a través de la función de pérdida Jaccard, del modelo se encuentra descrito en la figura 10.

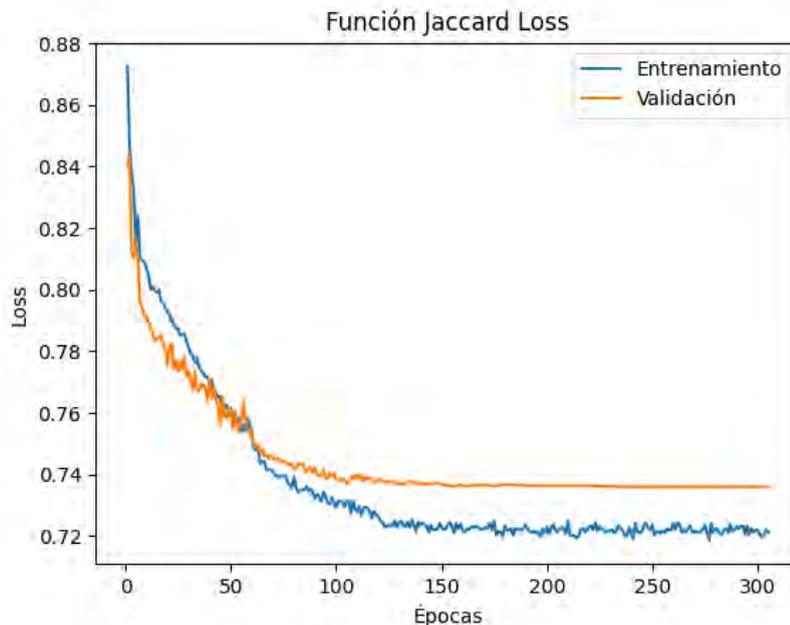


Figura 13. *Evolución de la función de pérdida Jaccard a través de las épocas para el modelo de extracción de variables socioeconómicas para tres niveles socioeconómicos.*

Fuente: Elaboración propia

A diferencia de la experimentación con el modelo de dos niveles socioeconómicos, en esta ocasión se consideró, como una cuarta capa, todo aquello de la imagen que no representa un techo, de forma que el modelo sea capaz de delimitar adecuadamente a qué partes de la imagen corresponde una etiqueta o no,

Se incluye en la presente sección una muestra visual de los resultados del entrenamiento del modelo de segmentación semántica, en la cual se compara la etiqueta real y la predicción otorgada por el modelo (figura 14).

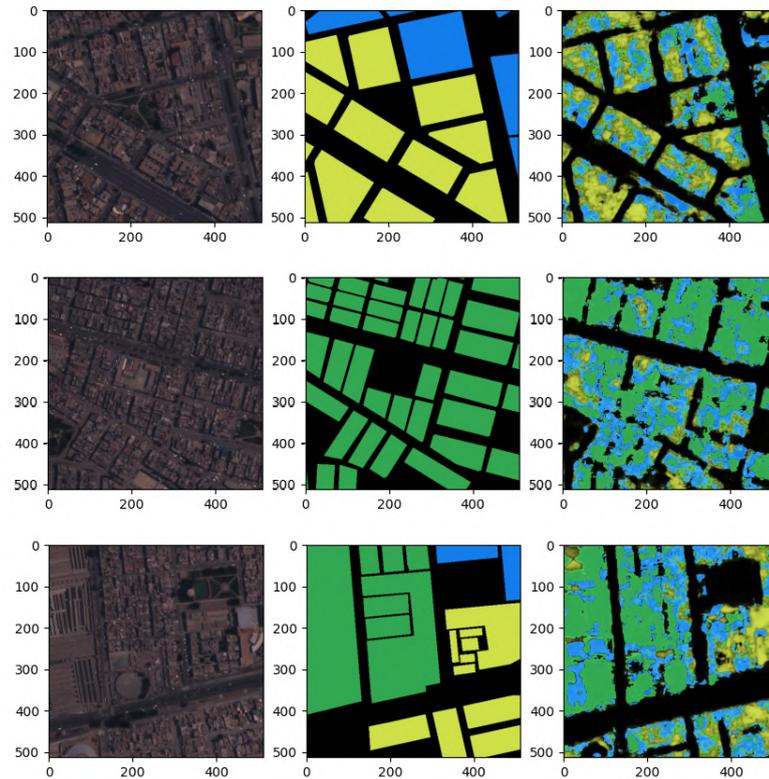


Figura 14. De izquierda a derecha: imágenes satelitales, etiquetas de nivel socioeconómico, predicción de las etiquetas generada por el modelo de extracción de variables socioeconómicas para tres niveles socioeconómicos.

Fuente: Elaboración propia

5.3 Discusión

El presente capítulo cuenta como resultados un protocolo de preprocesamiento de imágenes satelitales para su uso en los modelos de redes neuronales, así como los modelos de arquitectura, tanto el de segmentación semántica de regiones urbanas en base a techos como el de extracción de variables socioeconómicas. Por un lado, el protocolo de preprocesamiento genera subconjuntos de entrenamiento y validación, así como normaliza las imágenes y emplea técnicas de aumento de datos para preparar el conjunto de datos para el procesamiento. Por otro lado, se definió los valores de los hiperparámetros que definen el entrenamiento y se empleó el índice Jaccard como medida de evaluación del aprendizaje de los modelos de redes neuronales. Así, se logró entrenar ambos modelos para la resolución de sus tareas respectivas, segmentación en base a techos y determinar niveles socioeconómicos

(para dos y tres niveles socioeconómicos), con un rendimiento de 70,2 %, 32,9 % y 27,5 %, respectivamente.

En un principio, se intentó emplear para el entrenamiento todas las imágenes generadas cuyas etiquetas coincidían con la ubicación real de las manzanas en la imagen, pero se descubrió que las imágenes de zonas sin asfaltar o parcialmente asfaltadas no eran reconocidas adecuadamente por el modelo, además de que dificultaron el aprendizaje del conjunto de datos en general por parte de los modelos de red neuronal, debido a su baja representación de dichas zonas en el conjunto de datos original. Por ello, se optó por retirar dichas imágenes del conjunto de datos.

La escasa cantidad de imágenes que componen el conjunto de datos, producto de las discrepancias en las etiquetas y las imágenes satelitales, y la limitada variedad de las zonas capturadas - en las que no se incluyen zonas urbanas parcialmente asfaltadas ni zonas sin asfaltar - impide que los modelos entrenados sean generalizables a toda la ciudad. Dicho esto, los modelos entrenados son válidos para zonas urbanas totalmente asfaltadas y sin presencia de grandes zonas de tierra, capturadas en imágenes satelitales, sin presencia de nubes, por el satélite PerúSat-1, si bien los bajos índices Jaccard obtenidos por el modelo de extracción para ambos juegos de niveles y las etiquetas de muestra revelan que las etiquetas generadas por este proceso están sujetas a ciertos errores.

Aparte de las observaciones mencionadas en el capítulo anterior respecto a las etiquetas, beneficiaría a la resolución del problema también que los modelos de redes neuronales aprendan a detectar diferentes tipos de pistas o zonas de la ciudad que no constituyen pistas ni techos, de forma que puedan estas ser filtradas adecuadamente y no perjudiquen el aprendizaje respecto de la tarea principal.

Capítulo 6. Desarrollar una plataforma para el aprovechamiento oportuno de la información socioeconómica obtenida a través del criterio automático

6.1 Introducción

El presente capítulo comprende los avances correspondientes al tercer módulo de resultados de este proyecto de fin de carrera, referido al objetivo de desarrollar una plataforma para el aprovechamiento oportuno de la información socioeconómica obtenida a través del criterio automático. Esto incluye la elaboración de tanto unas interfaces de programación de aplicaciones (API) como de una interfaz gráfica de usuario (GUI), así como su integración en un sistema web.

6.2 Resultado alcanzado

6.2.1 Interfaz de Programación de Aplicaciones (API) para el uso automático del modelo de redes convolucionales

En esta subsección, se presentará el primer resultado de tercer objetivo, el cual consiste en una interfaz de programación de aplicaciones que sirva para el intercambio de datos con el modelo y, asimismo, sirva como componente *back-end* del sistema web que se plantea para este objetivo. En el desarrollo del presente apartado, se detalla el proceso de elaboración, así como su funcionamiento y los módulos que permiten funcionar a esta interfaz.

Esta interfaz de programación de aplicaciones se basa en la implementación de la API para el despliegue de modelos de segmentación de imágenes satelitales del proyecto final de tesis «Reconocimiento y Segmentación en imágenes satelitales de PeruSAT-1 para detectar regiones de interés mediante técnicas de aprendizaje no supervisado» (Gallardo, 2020), por lo que se decidió para la codificación de la misma el uso del marco de trabajo Flask para el lenguaje de programación Python, además de por su simplicidad y la facilidad para el despliegue de servicios web con dicha herramienta; así como Rasterio para la lectura y manipulación de las imágenes satelitales. La implementación de la presente API se encuentra referenciada en el **anexo D (Repositorios de código)**, en el apartado «Interfaz de Programación de Aplicaciones».

La API propuesta para este proyecto final de carrera funge de servidor, recibe peticiones del cliente y las responde mediante el protocolo HTTP. Por otro lado, para acceder al corpus de imágenes satelitales, se comunica con un servidor que almacena dichas imágenes mediante el protocolo FTP, como se muestra en la figura a continuación:

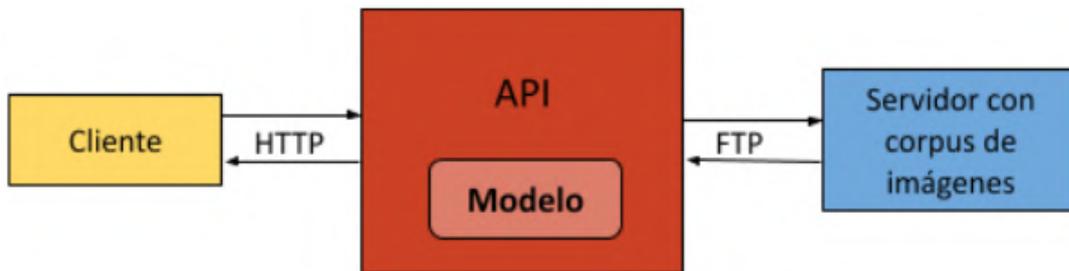


Figura 15. Esquema de funcionamiento de la API.

Fuente: Gallardo

Así, los puntos finales de comunicación con los que la API interactúa con el cliente y el servidor de imágenes, implementados en la interfaz del presente proyecto son los siguientes:

Tabla 9: Funcionalidades de la API

Nombre	Descripción	Entrada	Salida
predict	Ejecuta los modelos de red neuronal para la determinación del nivel socioeconómico	JSON que indica el nombre de la imagen a buscar en el servidor de imágenes y procesar	Referencia de la ubicación de la máscara segmentada por nivel socioeconómico, un nivel por capa (como imagen PNG)
checkFiles	Verifica si existe una máscara ya procesada y almacenada con el nombre brindado	JSON que indica el nombre de la imagen a buscar en la lista de máscaras ya elaboradas	Referencia de la ubicación de la máscara segmentada por nivel socioeconómico, un nivel por capa (de haberla) o un mensaje de máscara no

			encontrada
searchImages	Busca imágenes satelitales que contengan parcial o totalmente el área de búsqueda	JSON que contiene las coordenadas mínima y máxima que definen el área de interés	Lista de imágenes satelitales que cumplen los requisitos de ubicación, definidas por su nombre de archivo y coordenadas

Una documentación detallada de las funcionalidades de la API se encuentra en el **anexo F (Documentación de la Interfaz de Programación de Aplicaciones)**. Asimismo, se encuentra en el **anexo I (Reportes de pruebas unitarias y funcionales)** una tabla con la descripción y el reporte del estado de las pruebas realizadas, bajo las pestañas «Backend».

6.2.2 Interfaz Gráfica de Usuario (GUI) para la visualización de los resultados del modelo

El presente resultado, parte del tercer objetivo, comprende el desarrollo de una interfaz gráfica de usuario para permitir un manejo oportuno de las imágenes satelitales de referencia, los modelos implementados como parte del objetivo anterior, que se comunican con este componente a través de interfaz de programación de aplicaciones propuesta en la subsección anterior, y los productos de los mismos a través de componentes gráficos que esta interfaz incluye. Para el presente proyecto final de carrera, la interfaz gráfica será implementada como una página web.

En una primera instancia, se realizó un prototipado de la página web a desarrollar, la cual se muestra a continuación:

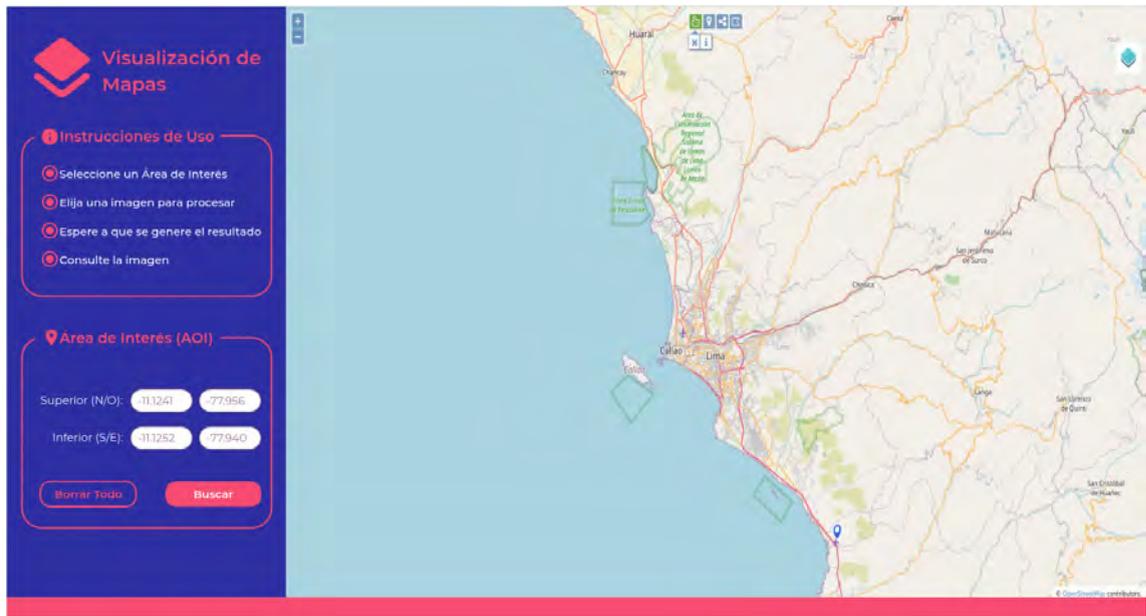


Figura 16. Prototipo de la interfaz gráfica de usuario, vista inicial.

Fuente: Elaboración propia

El flujo completo del prototipo se encuentra descrito en el **anexo G (Secuencia del prototipo)**.

Por otro lado, se definió el flujo del funcionamiento de la interfaz gráfica de usuario y su interacción con el API, según el siguiente diagrama de flujo (figura 11), codificado de acuerdo a los colores de cada componente en la figura 9:

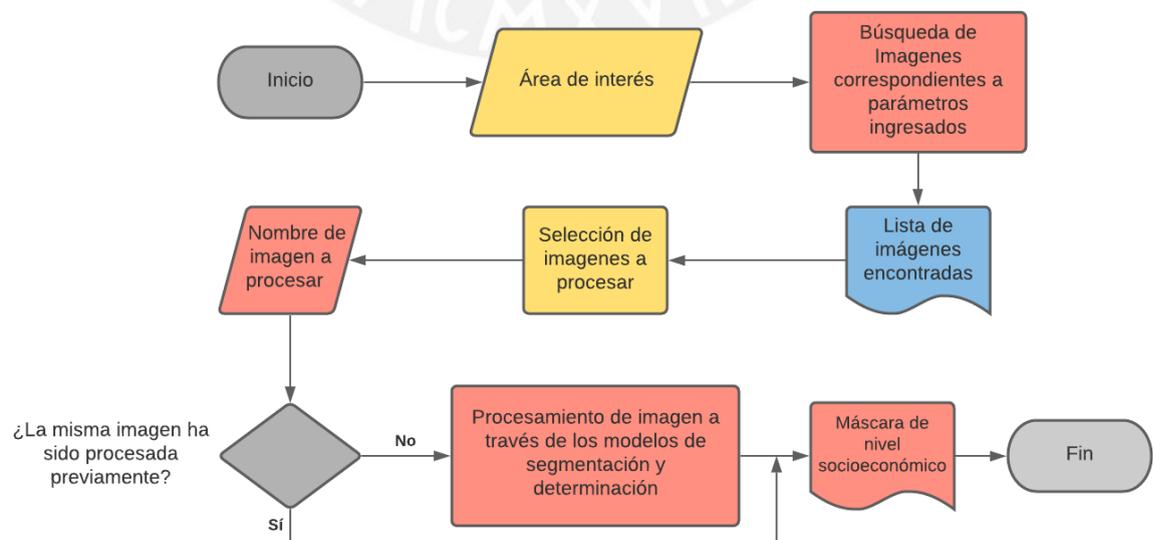


Figura 17. *Diagrama de flujo de la interacción entre cliente y servidor*

Fuente: Elaboración propia

Así, el cliente realiza una consulta al servidor en la que envía como información el área de interés (determinado por los límites norte, sur, este y oeste de una región del mapa) que se busca consultar. El servidor (la API), ante ello, realiza una búsqueda de imágenes satelitales con dichos datos y devuelve una lista de información básica (nombre de la imagen y ubicación en el mapa) de todas las imágenes satelitales que cumplen con los objetivos, de acuerdo a lo almacenado en el servidor de imágenes. Luego, desde el cliente se selecciona una imagen para solicitar la entrega de una máscara, para lo cual el servidor revisará si ya cuenta con la máscara procesada. De ser así, se enviará solamente la máscara en formato PNG (una imagen por nivel socioeconómico) a través de una referencia a la ubicación de la imagen en el servidor web; en caso contrario, se carga la imagen solicitada y se procesa la misma en los modelos para obtener el recurso solicitado.

Para la implementación de la interfaz gráfica de usuario, se optó por emplear el lenguaje de programación JavaScript para la estructura y el funcionamiento de la misma, mientras que el apartado visual se maneja mediante hojas de estilo. Específicamente, se eligió React como biblioteca para la construcción de la interfaz en JavaScript, debido a estar basada en componentes, lo que permite una mayor organización y legibilidad del código. Asimismo, debido a que se requiere visualizar la información procesada por el servidor en un mapa, se decidió usar la biblioteca Leaflet, que permite tanto ubicar un mapa en la interfaz gráfica como superponer capas vectoriales (para mostrar áreas de interés y ubicación en el mapa de una imagen), así como capas de imágenes (para superponer la máscara con los niveles socioeconómicos). La implementación del presente cliente web se encuentra referenciada en el **anexo C (Repositorios de código)**, en el apartado «Interfaz Gráfica de Usuario».

Asimismo, se cuenta con una manual de uso para la presente herramienta, que se encuentra en el **anexo H (Manual de uso de la aplicación web)**. Además, se encuentra en el **anexo I (Reportes de pruebas unitarias y funcionales)** una tabla con la descripción y el reporte de estado de las pruebas realizadas, bajo las pestañas «Frontend».

6.2.3 Integración de los componentes del sistema (back-end y front-end)

En este apartado, se tratará el desarrollo del último resultado del tercer objetivo, el cual refiere a la integración de los dos componentes desarrollados también como resultados del presente objetivo, de forma que se cuente con un sistema funcional para que los usuarios finales puedan interactuar con los modelos, mediante el componente *back-end*, de forma sencilla y a través del cliente web que conforma el componente *front-end*. La integración de los componentes también incluye el acondicionamiento de un servidor accesible mediante SFTP para las imágenes satelitales.

Respecto al servidor de imágenes, se experimentó con un servidor remoto Linux (Ubuntu 18.04 LTS), propiedad del Grupo de Inteligencia Artificial de la PUCP (IAPUCP), accesible mediante SFTP, en el que se incluyó, dentro de la carpeta «home» del usuario con el que se accede al mismo, una carpeta con todos los bloques de imágenes satelitales que han sido generados durante la formación del conjunto de datos, tal como se muestra en la figura 18.

```
csosa@discover:~/imagenes $ ls
ING_PERI_20190217152904_ORI_P_000659_subtittle_13312-7680.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_17408-9216.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19948-9728.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_13824-7680.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_17408-9728.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20480-10240.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_14336-6656.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_17920-5120.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20480-10752.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_14336-7680.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_17920-6656.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20480-6144.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_14848-8704.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18432-7168.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20480-6656.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_14848-9192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_17920-8192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20480-7680.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_14848-6656.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18432-8192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20480-8192.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_14848-7168.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_17920-9728.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20480-8704.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_14848-7680.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18432-10240.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20480-9216.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_14848-8192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18432-8192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20480-9728.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_14848-8704.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18432-6656.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20992-10240.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15360-6144.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18432-7168.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20992-10752.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15360-6656.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18432-7680.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20992-11264.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15360-7168.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18432-8192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20992-3584.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15360-7680.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18944-10240.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20992-5120.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15360-8192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18944-5120.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20992-6656.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15360-8704.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18944-6144.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20992-7168.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15872-5632.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18944-6656.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20992-8192.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15872-6656.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18944-7168.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20992-9216.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15872-7168.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18944-7680.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_20992-9728.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15872-7680.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18944-8192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_21504-4608.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15872-8192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18944-8704.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_21504-5120.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15872-8704.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18944-9216.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_21504-9216.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_15872-9216.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_18944-9728.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_21504-9728.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16384-6144.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19456-10240.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_22016-11264.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16384-6656.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19456-10752.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_22016-11776.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16384-7168.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19456-6144.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_22016-3584.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16384-7680.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19456-6656.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_22016-4608.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16384-8192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19456-7168.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_22016-6656.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16384-9216.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19456-7680.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_22016-7168.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16896-4608.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19456-8192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_22528-10240.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16896-6656.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19456-8704.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_22528-10240.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16896-7168.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19456-9728.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_22528-1264.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16896-7680.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19968-10240.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_22528-11776.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16896-8192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19968-5120.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_22528-3584.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16896-8704.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19968-6144.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_22528-5120.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_16896-9216.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19968-7680.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_23040-10752.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_17408-5120.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19968-8192.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_24064-4096.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_17408-6656.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19968-8704.tif
ING_PERI_20190217152904_ORI_P_000659_subtittle_17408-8704.tif  ING_PERI_20190217152904_ORI_P_000659_subtittle_19968-9216.tif
```

Figura 18. Carpeta de imágenes del servidor SFTP de la plataforma web, vista desde

Bash

Fuente: Elaboración propia

Por otro lado, respecto a la configuración y despliegue del componente *back-end*, se acondicionó un entorno virtual de Python 3, en el cual se instaló las dependencias del código del servidor web, de forma que su entorno de ejecución sea independiente de la versión de Python del servidor en el que se encuentra alojado. Además, se creó un conjunto de archivos de configuración para definir el comportamiento del mismo, los cuales son los siguientes:

- un archivo que describe el puerto, host, clave secreta y encabezados CORS del servidor para cada etapa de su desarrollo, además de una variable lógica que indica si se inicia la aplicación en modo de depuración o no
- un archivo que indica con cuántos niveles socioeconómicos se está trabajando; lo cual influye en la arquitectura del modelo de red neuronal para la extracción de variables socioeconómicas y, por consiguiente, en el resultado final que devuelve el sistema; así como con qué colores se va a representar cada nivel socioeconómico, descrito a través de sus valores en rojo, verde y azul
- un archivo que indica la ubicación de los modelos entrenados, tanto de segmentación semántica de techos como de extracción de variables socioeconómicas, dentro del directorio del sistema
- un archivo con las credenciales de acceso (host, usuario, contraseña) para la conexión con el servidor de imágenes

Durante el arranque del servidor web, se analiza los archivos y se extrae las variables mencionadas - salvo por el archivo de conexión con el servidor de imágenes, que es leído durante cada consulta a los servicios que lo requieren.

Durante la experimentación y ajuste del sistema, se habilitó el modo de depuración en la aplicación, se empleó el puerto 9997, y se emplearon los modelos de redes neuronales entrenados y los niveles socioeconómicos- incluyendo su paleta de colores - descritos en los capítulos 6 y 5, respectivamente. Asimismo, debido a no disponer de unidades de procesamiento gráfico para la ejecución de los modelos de redes neuronales, se optó por ejecutar la plataforma desde un servidor Linux (Ubuntu 18.04 LTS) remoto, también provisto por IAPUCP.

Finalmente, respecto a la plataforma web (interfaz gráfica de usuario), esta fue ejecutada de forma local en un ordenador portátil con sistema operativo Ubuntu 20.04 LTS y accedida mediante el navegador web Firefox 89.0 durante la experimentación

con el sistema. Debido a las limitaciones de hardware descritas anteriormente, para permitir la conexión entre los componentes cliente y servidor del sistema, se redirigió el puerto 9997 del equipo desde el que se ejecutó el cliente web hacia el puerto del mismo número en el servidor remoto de IAPUCP, el cual recepcionaba las solicitudes al servidor.

Se encuentra en el **anexo J (Capturas de pantalla de las pruebas de ejecución de la plataforma)** un conjunto de imágenes que demuestran el funcionamiento del sistema web, de acuerdo a las pruebas realizadas. Asimismo, se encuentra en el **anexo I (Reportes de pruebas unitarias y funcionales)** una tabla con la descripción y el reporte de estado de las pruebas realizadas, bajo la pestaña de «Plataforma».

Discusión

El presente capítulo cuenta como resultados un servidor web para la ejecución de los modelos de redes neuronales, un cliente web que comunique al usuario final con los servicios del servidor, así como la integración y despliegue de los mismos. Por un lado, se desarrolló un servidor Flask (Python) que ejecuta ambos modelos de arquitectura de redes neuronales y devuelve las etiquetas por nivel socioeconómico generadas por el modelo de extracción de variables socioeconómicas a través de una referencia a su ubicación dentro del servidor web. Asimismo, se desarrolló un cliente web en JavaScript que realiza peticiones y consume los recursos del servidor web. Finalmente, se realizó el despliegue local (cliente web) y remoto (servidor) del sistema, y se realizó la conexión de ambos componentes y entre el servidor web y el servidor de imágenes.

Debido a las limitaciones generadas por los conjuntos de datos generados, descritos en el capítulo 4, el servidor web solo maneja el procesamiento de bloques de 512 píxeles de ancho y largo, por lo que imágenes de mayor tamaño necesitarían ser divididas y luego reconstruidas para ser devueltas. Esa última funcionalidad no ha sido considerada en el presente proyecto.

Por otro lado, el manejo de un archivo de configuración que define los niveles socioeconómicos considerados y los colores con los que se van a representar permite que estos servicios sean compatibles con una cantidad arbitraria de niveles socioeconómicos, de acuerdo al archivo referido.

Asimismo, el manejo de las etiquetas de niveles socioeconómicos como capas PNG independientes permite que la plataforma web pueda ser también compatible con una cantidad arbitraria de niveles, siempre que el nombre de cada nivel sea indicado en la respuesta del servidor.

La presente plataforma web constituye una demostración de cómo sería una plataforma que permita procesar imágenes satelitales de Lima Metropolitana para determinar el nivel socioeconómico de todas las manzanas descritas en la imagen. Al ser una demostración, no se ha tomado en cuenta pruebas en un ambiente de ejecución similar a un hipotético ambiente real, por lo que no se ha realizado pruebas de acceso por parte de varios usuarios o múltiples peticiones en corto tiempo.



Capítulo 7. Conclusiones y trabajos futuros

7.1 Conclusiones

El presente proyecto de fin de carrera ha desarrollado un modelo para, a través del procesamiento de imágenes satelitales provenientes del satélite PerúSat-1, determinar de forma automática el nivel socioeconómico en zonas urbanas, para lo cual se contó con tres objetivos específicos.

El primer objetivo consistió en la integración de datos visuales (imágenes satelitales) con datos censales. Para ello, se generó un conjunto de datos compuesto de imágenes satelitales de 4 bandas y alta resolución del satélite PerúSat-1, correspondientes a Lima metropolitana, divididas en bloques de 512 píxeles de largo y 512 píxeles de ancho; los cuales se asociaron a matrices de igual ancho y largo, y con 1, 2 y 3 capas, generadas a partir de la extracción de etiquetas georreferenciadas de una base de datos espacial, provenientes de unos planos estratificados por nivel de ingresos a nivel de manzanas, publicados por el Instituto Nacional de Estadística e Informática. La matriz de capa única representa la distribución de manzanas existentes en el bloque asociado, mientras que las otras matrices representan esa distribución de manzanas, distribuidas de acuerdo a su nivel socioeconómico, de acuerdo a un sistema de 2 o 3 niveles socioeconómicos, respectivamente. Este conjunto de datos sirvió de entrada para el entrenamiento de los modelos de redes neuronales encargados de realizar la determinación del nivel socioeconómico en imágenes satelitales.

Si bien se ejecutó una integración entre ambas fuentes de datos y se conformó un corpus que pudo luego ser empleado para el desarrollo de los siguientes objetivos, diferencias entre los sistemas de referencia espacial de las imágenes satelitales y las etiquetas censales generó un desfase en la ubicación de las manzanas entre la imagen y la etiqueta, lo que redujo considerablemente el corpus de datos originalmente considerado al tener que descartar imágenes en las que el desfase no fue posible de eliminar o reducir. Ello supuso, asimismo, que un intento de determinar los 5 niveles socioeconómicos que originalmente componen el estudio de INEI devolvería resultados con precisión sumamente baja, por lo que no resultaría información con utilidad alguna, por lo que se optó por trabajar con sistemas de 2 y 3 niveles, como se describió en el párrafo anterior.

El segundo objetivo consistió en la elaboración de un criterio automático para la determinación del nivel socioeconómico. En primer lugar, se desarrolló un protocolo de preprocesamiento (Resultado 2.1) para transformar las imágenes satelitales y a sus etiquetas de forma que puedan ser procesadas por los modelos de redes neuronales propuestos en este objetivo. Este protocolo incluye la división del conjunto de datos en subconjuntos para el entrenamiento de cada modelo y su validación, así como la normalización de los datos. Asimismo, se desarrolló y entrenó con el corpus de datos preprocesado dichos modelos de arquitectura, uno de segmentación semántica en base a techos (Resultado 2.2) y otro, de extracción de variables socioeconómicas (Resultado 2.3). Ambos modelos toman como base la arquitectura del modelo TerausNet. Respecto al primer modelo, después de un entrenamiento de 200 épocas, una tasa de aprendizaje inicial de 0,001 y 4 imágenes por lote, se consigue un modelo que cuenta con un índice Jaccard de más de aproximadamente 70,3 %. Mientras, para el segundo modelo, se experimentó con 2 y 3 niveles socioeconómicos. En el primer caso, tras 1 000 épocas, la misma tasa y 5 imágenes por lote, se obtuvo un índice Jaccard de 32,9 %; en el segundo, tras 305 épocas y 4 imágenes por lote, 27,5%. El reducido tamaño del conjunto de datos y poca variedad en las etiquetas obtenidas como parte del objetivo anterior influye significativamente en el resultado de este objetivo, limitando la generalización de resultados a algunas zonas urbanas de Lima, en las que no existen montículos de tierra y las calles se encuentran correctamente asfaltadas, además de no tomar en cuenta los 5 niveles descritos por INEI, como se menciona en el párrafo anterior; e incluso así, si bien el modelo de segmentación semántica de techos cuenta con una alta precisión y resultados que logran segmentar los techos, los resultados del índice Jaccard del modelo de extracción de variables socioeconómicas tras el entrenamiento son menores al 70 % incluso el 50 %, por lo que los resultados del segundo modelo tampoco resultarían tan significativos como lo esperado.

Como tercer objetivo, se contó con el desarrollar una plataforma para aprovechar oportunamente la información socioeconómica provista por los modelos de redes neuronales. Para ello, se desarrolló en Python unas interfaces de programación de aplicaciones (Resultado 3.1) que permiten conectarse a un servidor de imágenes, listar su contenido, obtener una imagen arbitraria, procesarla a través de los modelos de redes neuronales y devuelva una referencia de la ubicación de la máscara de nivel socioeconómico generada, de forma que pueda servir a un cliente web (Resultado

3.2), que los usuarios finales manipularían, que pueda mostrar dichos datos en un mapa, el cual también ha sido desarrollado desde cero como parte de este objetivo. Finalmente, estos dos componentes, junto con el servidor de imágenes, se integraron y desplegaron (Resultado 3.3): el cliente web, de forma local, mientras que el servidor de imágenes y las API, de forma remota. Con el sistema ya desplegado y en funcionamiento, se realizó algunas pruebas para probar su funcionamiento.

En este grupo de resultados, la única consecuencia de contar con un conjunto de datos reducido es que las API manejan solamente bloques de 512 píxeles de ancho y largo, y no manejan imágenes enteras, lo cual era la idea original. Mientras, el cliente, al solo recibir referencias de la ubicación de las máscaras que se generan, puede adaptarse a cualquier tamaño de imagen. Por otro lado, ambos componentes del sistema pueden manejar una cantidad variable de niveles socioeconómicos.

Expuesto ello, durante el presente proyecto de fin de carrera se logró un sistema que determina con un error no despreciable el nivel socioeconómico, dentro de una escala simplificada de hasta 3 niveles, de bloques de imágenes, los cuales son de 512 píxeles de ancho y largo, por lo que este sistema no resulta adecuado para obtener información que sirva en una toma de decisiones. Sin embargo, los resultados de este proyecto abren la posibilidad de, con un corpus de imágenes y etiquetas más numeroso, contar con resultados más precisos y generalizables a toda la ciudad.

7.2 Trabajos futuros

A partir del presente proyecto, se propone las siguientes investigaciones futuras:

- Elaborar un corpus de información censal que sea georreferenciado mediante el mismo sistema de referencia espacial (CRS) que usan las imágenes satelitales que conforman su corpus de imágenes satelitales asociadas; y entrenar modelos de arquitectura de redes neuronales de determinación del nivel socioeconómico; de forma que se incremente considerablemente el conjunto de datos sobre los que el modelo trabaja y los modelos entrenados puedan generar resultados con una mayor precisión, generalizables a todo tipo de imágenes satelitales de zonas

urbanas de una ciudad (por ejemplo, Lima metropolitana) y aplicables a imágenes satelitales completas

- Emplear otras arquitecturas de segmentación semántica de imágenes (como LinkNet o SegNet) como base para resolver el problema y realizar una comparación del rendimiento a nivel cuantitativo y cualitativo entre los diferentes modelos estudiados y, así, definir cuál resulta más adecuado para la resolución del problema
- Emplear arquitecturas de segmentación de instancias (como Mask R-CNN) como base para la resolución del problema, de forma que sea posible manejar cada manzana como un ente separado y poder generar etiquetas independientes en vez de mapas de bits por nivel.



● Referencias

ArcGIS Resources. (s.f.). Georreferenciación y sistemas de coordenadas.

Consulta: 3 de noviembre de 2020.

<https://resources.arcgis.com/es/help/getting-started/articles/026n0000000s000000.htm>

Bansal, C., Jain, A., Barwaria, P., Choudhary, A., Singh, A., Gupta, A., & Seth, A.

(2020). Temporal Prediction of Socio-Economic Indicators Using Satellite

Imagery. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD* (pp.

73–81). New York, NY, USA: Association for Computing Machinery.

<https://doi.org/10.1145/3371158.3371167>

Centro Nacional de Planeamiento Estratégico (2018). *Guía de Políticas*

Nacionales. Lima, Perú: Preciso Agencia de Contenidos.

Centro Nacional de Planeamiento Estratégico (2020). *Políticas Nacionales de los*

Sectores. Consulta: 8 de diciembre de 2020.

<https://www.ceplan.gob.pe/politicas-nacionales-y-sectoriales/>

Consortio SQLite (s.f.). *What is SQLite?*. Consulta: 2 de abril de 2021.

<https://www.sqlite.org/index.html>

El Comercio (2017). *¿Cuánto invirtió el Gobierno en el Censo 2017?*. Consulta: 2

de septiembre de 2020.

<https://elcomercio.pe/peru/censo-2017-invertira-proceso-noticia-463178-noticia/>

El Peruano (2017). *Se necesitan más de 500,000 voluntarios para censos 2017*.

Consulta: 4 de octubre de 2020.

<https://elperuano.pe/noticia-se-necesitan-mas-500000-voluntarios-para-censos-2017-58757.aspx>

Facebook (s.f. a). React - A JavaScript library for building user interfaces.

Consulta: 18 de abril de 2021. <https://reactjs.org/>

Facebook (s.f. b). PyTorch. Consulta: 3 de noviembre de 2020. <https://pytorch.org/>

Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets* (pp. 267-285). Springer, Berlin, Heidelberg.

Gallardo, A. (2020). *Reconocimiento y segmentación en imágenes satelitales de PeruSAT-1 para detectar regiones de interés mediante técnicas de aprendizaje no supervisado* [Tesis de bachillerato sin publicar]. Pontificia Universidad Católica del Perú

Gestión (2020). *INEI realizará estudio para ampliar padrón de beneficiarios del bono de S/ 380*. Consulta: 4 de octubre de 2020.

<https://gestion.pe/economia/coronavirus-peru-mef-inei-realizara-estudio-para-ampliar-padron-de-beneficiarios-del-bono-de-s-380-nndc-noticia/>

Github (s.f.). About. Consultado: 3 de noviembre de 2020.

<https://github.com/about>

Guevara, G. (2018), *Capacidades del Sistema Satelital Peruano "PeruSAT-1"*.

Lima, Perú: CONIDA.

Hamiltom, N. (2011). *The A-Z of Programming Languages: BASH/Bourne-Again*

Shell. Consulta: 5 de abril de 2021.

https://web.archive.org/web/20110706103704/http://www.computerworld.com.au/article/222764/a-z_programming_languages_bash_bourne-again_shell/?pp=2&fp=16&fpid=1

Han, S., Ahn, D., Park, S., Yang, J., Lee, S., Kim, J., ... Cha, M. (2020). Learning to Score Economic Development from Satellite Imagery. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2970–2979). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3394486.3403347>

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), 90-95. <https://doi.org/10.1109/mcse.2007.55>

Iglovikov, V., & Shvets, A. (2018). Terausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. arXiv preprint arXiv:1801.05746.

Instituto Nacional de Estadística e Informática (2018). *INEI difunde Base de Datos de los Censos Nacionales 2017 y el Perfil Sociodemográfico del Perú*. Consulta: 3 de octubre de 2020.

Instituto Nacional de Estadística e Informática (2020). *Planos estratificados de Lima Metropolitana a nivel de manzanas 2020*. Consulta: 24 de enero de 2021.

Ipsos Perú (2020). *Características de los niveles socioeconómicos en el Perú*. Consulta: 23 de noviembre de 2020.

Kitchenham, B., & Charters S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (Vol. 5). Technical report, Ver. 2.3 EBSE Technical Report. EBSE.

La República (2020). *Magaly Medina indignada por ser afiliada al SIS: "De repente también tengo bono"*. Consulta: 4 de octubre de 2020.
<https://larepublica.pe/espectaculos/2020/06/16/magaly-medina-critica-al-gobierno-por-ser-afiliada-al-sis-y-presume-que-tiene-bono-familiar-por-figurar-en-pobreza/>

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

Mapbox (2018). Rasterio: access to geospatial raster data. Consulta: 3 de noviembre de 2020. <https://rasterio.readthedocs.io/en/latest/>

Ministerio de Transportes y Comunicaciones (2018). Mapeo con imágenes del satélite peruano PeruSAT-1 y clasificación de la carretera central.
http://portal.mtc.gob.pe/transportes/caminos/normas_carreteras/Imagen_Satelital/10%20Mapeo_Imag_PeruSat-1_PE-22_Morococha_Pachachaca_Feb-2018.pdf

Mozilla Foundation (s.f.). JavaScript. Consulta: 3 de noviembre de 2020.
<https://developer.mozilla.org/es/docs/Web/JavaScript>

NumPy. About us. Consulta: 3 de noviembre de 2020. <https://numpy.org/about/>

Open Source Geospatial Foundation (s.f.). Acerca de QGIS.
<https://www.qgis.org/es/site/about/index.html>

- Oshri, B., Hu, A., Adelson, P., Chen, X., Dupas, P., Weinstein, J., ... Ermon, S. (2018). Infrastructure Quality Assessment in Africa Using Satellite Imagery and Deep Learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 616–625). New York, NY, USA: Association for Computing Machinery.
<https://doi.org/10.1145/3219819.3219924>
- Padwick, C., Deskevich, M., Pacifici, F., & Smallwood, S. (2010). WORLDVIEW-2 PAN-SHARPENING. In *ASPRS 2010 Annual Conference*. San Diego, CA, USA: American Society for Photogrammetry and Remote Sensing.
<https://www.asprs.org/a/publications/proceedings/sandiego2010/sandiego10/Padwick.pdf>
- Petticrew, M., & Roberts, H. (2006). *Systematic Reviews in the Social Sciences*.
https://books.google.com.pe/books/about/Systematic_Reviews_in_the_Social_Science.html?id=ZwZ1_xU3E80C&redir_esc=y
- Python Software Foundation (s.f). About. Consulta: 3 de noviembre, 2020.
<https://www.python.org/about/>
- Radio Programas del Perú (2018). *Perú SAT-1 | Los otros cuatro satélites peruanos que también orbitan la Tierra*. Consulta: 3 de octubre de 2020.
<https://rpp.pe/peru/actualidad/peru-sat-1-los-otros-cuatro-satelites-peruanos-que-tambien-orbitan-la-tierra-noticia-1171287?ref=rpp>
- Real, Raimundo & Vargas, J.. (1996). The Probabilistic Basis of Jaccard's Index of Similarity. *Systematic Biology - SYST BIOL.* 45. 380-385.
10.1093/sysbio/45.3.380.

Red Hat. (s.f.). ¿Qué es una API de REST?. Consulta: 3 de noviembre, 2020.

<https://www.redhat.com/es/topics/api/what-is-a-rest-api>

Ronacher, A. (2015). Flask. Consulta: 3 de noviembre de 2020.

<https://palletsprojects.com/p/flask/>

Ruiz, D., Cadenas, C. (s.f.) ¿Qué es una política pública?. In *Revista Jurídica IUS*.

Morelia, México: Universidad Latina de América.

Saegert, S., Adler, N., Bullock, H., Cauce, A., Ming Liu, W., Wyche, K. (2006).

Report of the APA Task Force on Socioeconomic Status. Washington D.C., USA: American Psychological Association.

Shermeyer, J., Hogan, D., Brown, J., Van Etten, A., Weir, N., Pacifici, F., ... &

Lewis, R. (2020). SpaceNet 6: Multi-Sensor All Weather Mapping Dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 196-197).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556

Tingzon, I., Orden, A., Go, K. T., Sy, S., Sekara, V., Weber, I., ... Kim, D. (2019).

MAPPING POVERTY in the PHILIPPINES USING MACHINE LEARNING, SATELLITE IMAGERY, and CROWD-SOURCED GEOSPATIAL

INFORMATION. In T. M. E. A. R. A. A. Blanco A.C. Sarmiento C.J.S. (Ed.),

International Archives of the Photogrammetry, Remote Sensing and Spatial

Information Sciences - ISPRS Archives (Vol. 42, pp. 425–431). International

Society for Photogrammetry and Remote Sensing.

<https://doi.org/10.5194/isprs-archives-XLII-4-W19-425-2019>

Yuan, J., Roy Chowdhury, P. K., McKee, J., Yang, H. L., Weaver, J., & Bhaduri, B. (2018). Exploiting deep learning and volunteered geographic information for mapping buildings in Kano, Nigeria. *Scientific Data*, 5. <https://doi.org/10.1038/sdata.2018.2>



Anexo A: Formularios de extracción

Learning to Score Economic Development from Satellite Imagery

Campo	Contenido
Id	1
Fecha de extracción	18 de septiembre de 2020
Autores	Han, Sungwon Ahn, Donghyun Park, Sungwon Yang, Jeasurk Lee, Susang Kim, Jihee Yang, Hyunjoo Park, Sangyoon Cha, Meeyoung
Tipo de fuente	Artículo en actas de sesión
Fuente	26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining
Año de publicación	2020
Afiliación	School of Computing, KAIST, Universidad de Singapore, Universidad de Hong Kong, Universidad de Sogang
País	Corea del Sur, Singapur, China
Técnica empleada	Conjunción de un modelo de aprendizaje profundo no supervisado, categorización de clústeres y una red neuronal de puntuación
Aplicación de la técnica	El primer modelo categoriza en clústeres las imágenes satelitales; luego, se genera un grano parcialmente ordenado, categorizando los clústeres de acuerdo a su grado de urbanización vs ruralidad. Finalmente, el modelo de puntuación asigna puntajes a las imágenes de acuerdo al grado.
Ventaja principal	Posibilidad de realizar una categorización del nivel socioeconómico de regiones observadas ante la ausencia de datos etiquetados
Ventaja secundaria	El método es aplicable en países en desarrollo
Característica analizada	-
Elemento relevante	Bloques urbanos y tipos de edificios en estos

MAPPING POVERTY in the PHILIPPINES USING MACHINE LEARNING,
SATELLITE IMAGERY, and CROWD-SOURCED GEOSPATIAL
INFORMATION

Campo	Contenido
Id	2
Fecha de extracción	16 de septiembre de 2020
Autores	Tingzon, I Orden, A Go, K T Sy, S Sekara, V Weber, I Fatehkia, M García-Herranz, M Kim, D
Tipo de fuente	Conferencia
Fuente	International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives
Año de publicación	2019
Afiliación	Twenty-Four Seven McKinley, UNICEF Office of Innovation, Qatar Computing Research Institute
País	Filipinas, Estados Unidos, Catar
Técnica empleada 1	Aprendizaje por transferencia
Aplicación de la técnica 1	Se entrenan las capas densas una red neuronal convolucional de arquitectura VGG16 preentrenada en ImageNet en la clasificación de imágenes satelitales en 5 niveles socioeconómicos.
Técnica empleada 2	Árbol de regresión
Aplicación de la técnica 2	Se extrae características de un mapa de intensidad de luz nocturna y, junto con datos generados por la comunidad de Open Street Map, se entrena un árbol de regresión.
Ventaja principal	Clasificación del nivel socioeconómico de forma menos costosa en recursos, tiempo y personas, que los métodos tradicionales de recolección de datos económicos
Ventaja secundaria	Aplicable a países en desarrollo
Característica analizada	Intensidad de luz nocturna
Elemento relevante	-

Exploiting deep learning and volunteered geographic information for mapping buildings in Kano, Nigeria

Campo	Contenido
Id	3
Fecha de extracción	16 de septiembre de 2020
Autores	Yuan, J Roy Chowdhury, P K McKee, J Yang, H L Weaver, J Bhaduri, B
Tipo de fuente	Artículo
Fuente	Scientific Data
Año de publicación	2018
Afiliación	Urban Dynamics Institute, Oak Ridge National Laboratory; University of Tennessee
País	Estados Unidos
Técnica empleada	Red neuronal convolucional
Aplicación de la técnica	Se entrena una red neuronal convolucional con imágenes satelitales y etiquetas de Open Street Map para generar datos que, tras ser corregidos manualmente, sirven para ampliar el conjunto de entrenamiento.
Ventaja principal	Permite mapear la distribución de edificios en regiones en los que los datos preexistentes son escasos
Ventaja secundaria	-
Característica analizada	-
Elemento relevante	Edificios en zonas urbanas y parcialmente urbanas

Temporal Prediction of Socio-Economic Indicators Using Satellite Imagery

Campo	Contenido
Id	4
Fecha de extracción	18 de septiembre de 2020

Autores	Bansal, Chahat Jain, Arpit Barwaria, Phaneesh Choudhary, Anuj Singh, Anupam Gupta, Ayush Seth, Aaditeshwar
Tipo de fuente	Artículo en acta de sesión
Fuente	7th ACM IKDD CoDS and 25th COMAD
Año de publicación	2020
Afiliación	Institute of Technology Delhi, Phaneesh Barwaria Vidyalandkar Institute of Technology
País	India
Técnica empleada	Aprendizaje automático tradicional
Aplicación de la técnica	Extracción de características basadas en histogramas por cada banda. Estas características son luego la entrada para diferentes modelos de clasificación (SVM, árboles de clasificación). Para mejorar su precisión temporal, los resultados del primer modelo en un año base y años posteriores (incluyendo el año que se quiere predecir) son procesados en otro modelo de clasificación para mejorar la predicción.
Ventaja principal	-
Ventaja secundaria	-
Característica analizada	Valores numéricos de cada banda
Elemento relevante	-

Infrastructure Quality Assessment in Africa using Satellite Imagery and Deep Learning

Campo	Contenido
Id	5
Fecha de extracción	18 de septiembre de 2020

Autores	Oshri, Barak Hu, Annie Adelson, Peter Chen, Xiao Dupas, Pascaline Weinstein, Jeremy Burke, Marshall Lobell, David Ermon, Stefano
Tipo de fuente	Artículo en acta de sesión
Fuente	24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining
Año de publicación	2018
Afiliación	Stanford University
País	Estados Unidos
Técnica empleada	Aprendizaje por transferencia
Aplicación de la técnica	Se entrena una red neuronal residual ResNet preentrenada en ImageNet en la clasificación de la infraestructura urbana
Ventaja principal	Mayor rendimiento en la inferencia de calidad de infraestructura de zonas con pocas muestras (imágenes) disponible respecto de métodos como la interpolación vecina natural.
Ventaja secundaria	-
Característica analizada	-
Elemento relevante	-

Anexo B: Plan de Proyecto

• Justificación

El presente proyecto propone un modelo algoritmo para la determinación del nivel socioeconómico de áreas urbanas mediante imágenes satelitales. Esta herramienta permitirá obtener dicha información demográfica rápidamente, de forma que sea útil para el análisis de la población con el objetivo de la elaboración de políticas públicas.

Uno de los motivos para la realización de este proyecto es que permite aprovechar la información contenida en las imágenes satelitales de PerúSAT-1 para el análisis de variables que podrían inferirse con elementos capturados en las mismas, como el nivel socioeconómico (Han et al. 2020). Esto se debe a la capacidad de los modelos basados en redes neuronales convolucionales de detectar y analizar patrones (Fukushima, 1980). Así, el modelo desarrollado en este proyecto podría servir como base para otros proyectos que busquen determinar otras variables demográficas.

Por otro lado, el proyecto permite la automatización de las labores de organismos encargados de recabar dicha información en la población peruana, como el Instituto Nacional de Estadística e Informática (INEI). De este modo, no se dependerá exclusivamente de los datos recolectados de forma tradicional, los cuales requieren un mayor tiempo para ser obtenidos (El Comercio, 2017).

- **Viabilidad**

Para el presente proyecto de fin de carrera, se evaluó la viabilidad del mismo en cuatro perspectivas en las que se soporta el proyecto: técnica, temporal, económica y disponibilidad de recursos.

Desde la perspectiva técnica, los conceptos y técnicas que se emplean en el proyecto son conocimientos ya manejados con anterioridad por el tesista, tanto en cursos anteriores como en sus prácticas preprofesionales. Asimismo, las herramientas utilizadas para la ejecución del proyecto son de código abierto, han sido empleadas anteriormente por el tesista y cuentan con documentación adecuada para el correcto entendimiento de sus funcionalidades.

En el aspecto temporal, se plantea que la ejecución del proyecto ocurra entre los meses de enero y julio, aunque sin exceder la extensión del semestre académico 2021-1. Este periodo se considera adecuado para la realización del proyecto dada la experiencia previa del tesista, lo que implica que no será necesario un periodo de aprendizaje previo.

Referido a lo económico, debido al carácter libre de las herramientas a utilizar, no se incurre en gastos de licencias de *software*. Asimismo, dado el convenio de la universidad con CONIDA y con INEI, tampoco existen gastos por la obtención de los datos de entrenamiento para el sistema. Por otro lado, dado el estado de emergencia sanitaria y el cierre del campus universitario, todas las reuniones con el asesor de tesis se realizan de forma remota, por lo que no se considera la existencia de gastos de transporte.

Finalmente, respecto a la disponibilidad de recursos, se ha identificado con anterioridad las fuentes tanto de las imágenes satelitales como de la información censal de dichas áreas. Además, en caso de que no se reciba suficientes imágenes para el proyecto, se cuenta con un plan de contingencia, por lo que la disponibilidad de recursos de forma oportuna está garantizada en el presente proyecto.

- **Alcance**

Se busca con el presente proyecto el desarrollar y desplegar un modelo que determine el nivel socioeconómico de zonas urbanas mediante el análisis automático de imágenes satelitales del satélite PerúSAT-1.

Como primer paso, se armará un conjunto de imágenes segmentadas y con etiquetas respecto del nivel socioeconómico, el cual servirá como base para el entrenamiento del modelo. Para ello, es necesario recolectar imágenes satelitales provenientes del satélite PerúSAT-1, previamente procesadas por especialistas del CONIDA. De igual manera, se obtendrá los niveles socioeconómicos de las áreas capturadas en las imágenes obtenidas para generar las etiquetas del conjunto de

datos. Adicionalmente, se dividirán las imágenes en parches de dimensionalidad reducida para que puedan ser manejadas por el modelo.

Después de ello, para su posterior uso en el entrenamiento, se va a aumentar el número de imágenes del conjunto mediante transformaciones tales como rotaciones, así como se normalizará el color de las imágenes del conjunto.

A continuación, se implementará el modelo de segmentación semántica de imágenes en base a techos mediante el uso del paquete PyTorch. Mediante el entrenamiento, se ajustarán los parámetros necesarios para la realización de dicha tarea. Durante el entrenamiento, las estructuras que representan las imágenes serán almacenadas en GPU para reducir el tiempo de procesamiento. Para medir la eficacia del modelo y analizar posibles mejoras en los parámetros, se ejecutará la validación del modelo mediante la métrica de intersección sobre unión (índice Jaccard). El mismo proceso se seguirá con la implementación del modelo de clasificación del nivel socioeconómico, el cual tomará como entrada la salida del modelo anterior.

Luego, se implementará una interfaz de programación de aplicaciones (API) para desplegar un servicio web que permita el uso del modelo por parte del usuario final. Esto se realizará mediante el uso del marco de trabajo Flask de lenguaje de programación Python y solo se incluirá actividades relacionadas a esta tarea. Durante la implementación, se realizarán pruebas unitarias y de integración a las interfaces.

Tras ello, se implementará una interfaz gráfica para que los usuarios finales realicen consultas al modelo y puedan visualizar los resultados del mismo. Esta se implementará mediante JavaScript, usando VueJS como marco de trabajo. La interfaz gráfica incluirá como funcionalidades la búsqueda de imágenes por coordenadas y fecha de captura, así como de la visualización de los resultados como capas del mapa. Debido a que el proyecto solo trabaja con imágenes del satélite Perú-SAT1, la interfaz no permitirá seleccionar para una búsqueda regiones externas al territorio nacional.

Finalmente, se integrará localmente ambos componentes desarrollados en los pasos anteriores para su posterior uso. Para corroborar el correcto funcionamiento, se realizará pruebas funcionales que midan el funcionamiento integral del sistema.

El proyecto no incluirá la predicción de otras variables demográficas ni la predicción del nivel socioeconómico a nivel histórico, limitándose solamente a imágenes satelitales capturadas recientemente, aunque el modelo puede servir como base para otros sistemas que busquen resolver el problema para otras épocas.

- **Restricciones**

Debe considerarse que, debido a que el satélite pasa por un mismo lugar cada 21 días (Ministerio de Transportes y Comunicaciones, 2018), puede que no se pueda solicitar una imagen satelital de una zona para una fecha en cuestión. Ante eso, en ocasiones se optará por emplear imágenes satelitales de fechas cercanas a las inicialmente planteadas.

Por otro lado, para garantizar la concordancia entre los datos censales y las imágenes satelitales que serán empleados durante el entrenamiento y validación, se

deberá seleccionar imágenes satelitales cuyas fechas sean cercanas a las de recolección de datos censales.

Finalmente, el nivel de precisión de la inferencia del nivel socioeconómico depende de la granularidad de las etiquetas obtenidas en base a la información censal para el entrenamiento y validación de los modelos.

● **Identificación de los riesgos del proyecto**

Se detallan a continuación dos riesgos asociados a la realización del presente proyecto de fin de carrera, los cuales son necesarios considerar. Además de describirlos, se mencionan los síntomas del riesgo, las acciones de mitigación para evitar la aparición de los síntomas y las acciones de contingencia por si el riesgo se materializa. Asimismo, se ha asignado dos valores, de acuerdo a una escala ascendente de 1 al 4, que describen la probabilidad de ocurrencia del riesgo como el nivel del impacto de su ocurrencia. Dados estos valores, se calcula también la severidad del riesgo, que se define como el producto de la probabilidad y el impacto. Así, se puede estimar la importancia del riesgo.

	Riesgo 1	Riesgo 2
Descripción	No disponibilidad de <i>hardware</i> en el ambiente de entrenamiento y validación de los modelos	Escasez de información debido a la negativa del CONIDA a entregar suficientes imágenes satelitales para el entrenamiento y validación
Síntomas	<ul style="list-style-type: none"> • Insuficientes recursos de <i>hardware</i> (GPU/CPU) • Uso del <i>hardware</i> por parte de terceros 	<ul style="list-style-type: none"> • Negativa de CONIDA de entregar imágenes
Probabilidad	2	2
Impacto	2	4
Severidad	4	8
Mitigación	Elaborar el cronograma del proyecto tomando en cuenta el uso del <i>hardware</i> compartido por parte del resto de proyectos que lo usan.	Firmar un acuerdo con CONIDA para garantizar el acceso a un determinado número mínimo de imágenes satelitales
Contingencia	Emplear recursos de algún servicio de procesamiento en la nube o emplear un entorno local	Realizar un aumento de elementos del conjunto de datos mediante técnicas de incremento de datos,

	de procesamiento para realizar las actividades respectivas	tales como la rotación, volteo horizontal o vertical.
--	--	---

- **Estructura de descomposición del trabajo (EDT)**

1. **Conjunto de imágenes satelitales con regiones etiquetadas en base a los datos censales**
 - 1.1. Base de datos de imágenes satelitales
 - 1.2. Información censal
 - 1.3. Base de datos de imágenes etiquetadas
2. **Criterio automático de los resultados de determinación del nivel socioeconómico**
 - 2.1. Modelo de segmentación semántica de regiones urbanas en base a techos
 - 2.1.1. Preprocesamiento
 - 2.1.2. Código del modelo
 - 2.1.3. Modelo optimizado
 - 2.2. Modelo automático basado en redes neuronales convolucionales para la extracción de variables socioeconómicas en imágenes satelitales
 - 2.2.1. Preprocesamiento
 - 2.2.2. Código del modelo
 - 2.2.3. Modelo optimizado
3. **Aplicación web de extracción de variables socioeconómicas en imágenes satelitales**
 - 3.1. Interfaz de Programación de Aplicaciones (API) para el uso automático del modelo de redes convolucionales
 - 3.1.1. Pruebas unitarias
 - 3.1.2. Pruebas funcionales
 - 3.1.3. Software
 - 3.2. Interfaz Gráfica de Usuario (GUI) para la visualización de los resultados del modelo
 - 3.2.1. Pruebas unitarias
 - 3.2.2. Pruebas funcionales
 - 3.2.3. Software
 - 3.3. Despliegue de la aplicación

- **Lista de tareas**

Actividad	Duración estimada (días)	Esfuerzo (horas-persona)	Costo (\$)
1. Elaborar un conjunto de imágenes satelitales con regiones etiquetadas en base a los datos censales	-		
1.1 Generar una base de datos de imágenes satelitales	30	75	375
1.2 Analizar información censal	30	75	375
1.3 Generar una base de datos de imágenes etiquetadas	7	14	70

2. Elaborar un criterio automático de los resultados de determinación del nivel socioeconómico			
2.1 Desarrollar un modelo de segmentación semántica de regiones urbanas en base a techos	25	40	200
2.2 Desarrollar un modelo automático basado en redes neuronales convolucionales para la extracción de variables socioeconómicas en imágenes satelitales	25	40	200
3. Desarrollar una aplicación web de extracción de variables socioeconómicas en imágenes satelitales	-		
3.1 Desarrollar una Interfaz de Programación de Aplicaciones (API) para el uso automático del modelo de redes convolucionales	60	100	500
3.2 Desarrollar una Interfaz Gráfica de Usuario (GUI) para la visualización de los resultados del modelo	60	100	500
3.3 Desplegar la aplicación	10	20	100
4. Elaborar el documento de tesis	100	100	500
5. Coordinar con el asesor de tesis	20	20	400
6. Actividades del curso Proyecto de Tesis 2	15	30	0

- **Cronograma del proyecto**

Se detalla en esta sección el cronograma que se planea seguir durante la ejecución de este proyecto de fin de carrera. En este, se menciona las fechas de inicio, de fin, las relaciones de dependencia entre actividades y una duración estimada.

Actividad	Inicio	Fin	Precedencia	Febrero	Marzo	Abril	Mayo	Junio
1	20/2/2021	5/4/2021						
1.1	20/2/2021	20/3/2021						
1.2	20/2/2021	20/3/2021						
1.3	21/3/2021	5/4/2021	1.1, 1.2					
2	7/4/2021	1/6/2021	1					
2.1	7/4/2021	18/5/2021						
2.2	7/4/2021	18/5/2021						
2.3	12/4/2021	1/6/2021						
3	7/4/2021	7/6/2021						
3.1	7/4/2021	4/5/2021						
3.2	7/4/2021	4/5/2021						
3.3	28/4/2021	7/6/2021	3.1, 3.2					
4	29/3/2021	7/6/2021						
5	1/2/2021	4/6/2021						
6	4/6/2021	8/6/2021						

Título de tesis: Determinación y segmentación del nivel socio-económico a partir del análisis automático de imágenes de satélite					
Semana	Entrega	Componente		Avance propuesto	
		Componete	Subcomponete		
1	04/04	1	1,1	Generar una base de datos de imágenes satelitales	
			1,2	Analizar información censal	
			1,3	Generar una base de datos de imágenes satelitales etiquetadas	
		4	4,1	Ajustes finales al plan de proyecto de fin de carrera (cronograma)	
				Redacción del capítulo 4 del documento de tesis	
		5	5,1	Reunión con asesor – Sábado 11:00	
6	6,1	Desarrollo de experimentación			
2	06/04	2	2.1	Modelo de segmentación de techos – preprocesamiento de corpus al 50%	
		3	3,1	Desarrollo de API al 30% para para el uso automático del modelo de redes convolucionales	
			3,2	Desarrollo inicial de Interfaz Gráfica al 25% - Prototipado	
		4	4,1	Ajustes finales al plan de proyecto de fin de carrera (metodología)	
				Redacción del capítulo 6 del documento de tesis	
		5	5,1	Reunión con asesor – Martes/Sábado 11:00	
6	6,1	Desarrollo de experimentación			
3	13/04	2	2.2	Implementación del modelo de segmentación de techos al 25% en python/pytorch – Modelo base	
			2.3	Implementación del modelo de predicción del nivel socioeconómico al 25% en python/pytorch – Modelo base	
		3	3,1	Desarrollo de API al 50% para despliegue de modelos de segmentación	
			3,2	Desarrollo de Interfaz Gráfica al 50% - Implementación inicial	

		4	4,1	Ajustes finales al plan de proyecto de fin de carrera (metodología)
				Redacción de capítulos 5 y 6 del documento de tesis
		5	5,1	Reunión con asesor – Martes/Sábado 11:00
		6	6,1	Desarrollo de experimentación
4	20/04	2	2.2	Implementación del modelo de segmentación de techos al 50% en python/pytorch
			2.3	Implementación del modelo de predicción del nivel socioeconómico al 50% en python/pytorch
		3	3,1	Desarrollo final de API al 80% para despliegue de modelos de segmentación
			3,2	Desarrollo de Interfaz Gráfica al 80%
		4	4,1	Ajustes finales al plan de proyecto de fin de carrera (metodología)
				Redacción de capítulos 5 y 6 del documento de tesis
		5	5,1	Reunión con asesor – Martes/Sábado 11:00
6	6,1	Desarrollo de experimentación		
5	27/04	2	2.2	Implementación del modelo de segmentación de techos al 75% en python/pytorch
			2.3	Implementación del modelo de predicción del nivel socioeconómico al 75% en python/pytorch
		3	3,1	Desarrollo final de API al 100% para despliegue de modelos de segmentación – pruebas
			3,2	Desarrollo final de Interfaz Gráfica al 100% - Pruebas
			3,3	Despliegue de aplicación web - Integración inicial de frontend y backend
		4	4,1	Redacción de capítulos 5 y 6 del documento de tesis
		5	5,1	Reunión con asesor – Martes/Sábado 11:00
6	NO HAY ENTREGA	2	2.2	Implementación del modelo de segmentación de techos al 100%
			2.3	Implementación del modelo de predicción del nivel socioeconómico al 100%
			2.1	Modelo de segmentación de techos – preprocesamiento de corpus al 100%

			2,2	Modelo de segmentación de techos optimizado – Entrenamiento
		4	4,1	Redacción de capítulos 5 y 6 del documento de tesis
		5	5,1	Reunión con asesor – Martes/Sábado 11:00
		6	6,1	Desarrollo de experimentación
7	11/05	2	2.1	Modelo de predicción del nivel socioeconómico – preprocesamiento de corpus al 50%
			2.2	Modelo de segmentación de techos optimizado – Entrenamiento
		3	3,3	Despliegue de aplicación web - Integración de frontend y backend, pruebas
		4	4,1	Redacción de capítulos 5 y 6 del documento de tesis
		5	5,1	Reunión con asesor – Martes/Sábado 11:00
		6	6,1	Desarrollo de experimentación
8	18/05	2	2.1	Modelo de predicción del nivel socioeconómico – preprocesamiento de corpus al 100%
			2.3	Modelo de predicción del nivel socioeconómico optimizado – Entrenamiento
		3	3,3	Despliegue de aplicación web - Integración de frontend y backend, pruebas
		4	4,1	Redacción de capítulos 4 y 5 del documento de tesis
		5	5,1	Reunión con asesor – Martes/Sábado 11:00
		6	6,1	Desarrollo de experimentación
9	25/05	2	2.3	Modelo de predicción del nivel socioeconómico optimizado – Entrenamiento
		3	3,3	Despliegue de aplicación web – pruebas con imágenes satelitales
		4	4,1	Redacción de capítulos 5 y 6 del documento de tesis
		5	5,1	Reunión con asesor – Martes/Sábado 11:00
		6	6,1	Desarrollo de experimentación
10	01/06	3	3,3	Despliegue de aplicación web – pruebas con imágenes satelitales

		4	4,1	Redacción de capítulos 5 y 6 del documento de tesis
		5	5,1	Reunión con asesor – Martes/Sábado 11:00
		6	6,1	Desarrollo de experimentación

- **Lista de recursos**

A continuación, se listan los recursos necesarios para la realización del presente proyecto final de tesis, tanto humanos, materiales e inmateriales.

- **Personas involucradas y necesidades de capacitación**

- **Personas involucradas**

- Tesista
- Asesor de tesis

- **Materiales requeridos para el proyecto**

- Imágenes satelitales de zonas urbanas
- Información censal correspondientes a dichas zonas urbanas

- **Equipamiento requerido**

- Computadora personal
- Servidor con capacidad de computación en paralelo

- **Herramientas requeridas**

- **Software**

- Python
- JavaScript
- Visual Studio
- PyCharm

- **Costeo del Proyecto**

En la siguiente tabla, se detalla los costos del presente proyecto, desglosado por recurso, cantidad y valor unitario. Para calcular el valor por mes del portátil, se ha tomado en cuenta un valor inicial de S/ 12 000, un valor residual de S/ 3 000 y un periodo de vida útil de 3 años.

Cabe resaltar que los montos listados a continuación no son monetarios, sino que corresponden a un estimado en caso el proyecto no estuviese bajo el contexto de un proyecto de pregrado. Debido a ello y a acuerdos con las entidades proveedoras de la información a utilizar, el gasto real - como se mencionó anteriormente - es nulo.

Ítem	Descripción	Unidad	Cantidad	Valor Unidad (S/.)	Monto Parcial (S/.)	Monto Total (S/.)
0	Costo total del proyecto	---	---	---	---	6,520
1.	Estudiantes o tesis	---	---	---	---	890

1.1	Tesista			Horas	600	5	3000	
2.	Otros participantes			---	---	---	---	440
2.1	Asesor de tesis			Horas	40	20	800	
3.	Materiales e insumos			---	---	---	---	200 000
3.1	Imágenes satelitales de zonas urbanas			Unidad	5	40 000	200 000	
3.2	Información censal			Unidad	1	0	0	
4.	Bienes y equipos	Unid1	Cant1	Unid2	Cant2	-	-	3250
4.1	Computadoras	Equipo	1	Meses	7	250	1750	
4.2	Servidor	Equipo	1	Meses	5	300	1500	
4.3	Python	Unidad	1	Horas	300	0	0	
4.5	JavaScript	Unidad	1	Horas	150	0	0	
4.6	QGis	Unidad	1	Horas	100	0	0	
4.7	Visual Studio	Unidad	1	Horas	150	0	0	
4.8	PyCharm	Unidad	1	Horas	300	0	0	

Anexo C: Declaración de resultado alcanzado

Lima, 03 de Mayo del 2021

Profesor de Curso de Tesis 2

Por el presente doy mi conformidad de los avances realizados en el trabajo de fin de curso de mi asesorado Carlos Alberto Sosa Pezo, Cód. 20161310, siendo que para el objetivo 1 se cumplió con desarrollar el entregable correspondiente, con los verificables que a continuación se detalla:

Entregable 1	Verificables
Integrar datos visuales con datos censales	<ul style="list-style-type: none">- Se ha generado un conjunto de datos de imágenes satelitales refinadas pancromáticamente- Se ha obtenido un conjunto de datos de información socioeconómica georreferenciada- Se ha generado un conjunto de datos de imágenes satelitales etiquetadas en base a información censal

Cabe indicar que la información obtenida de INEI, así como las imágenes satelitales otorgadas por CONIDA se encuentran dentro del servidor del grupo de Inteligencia Artificial, dado el tamaño y espacio que demandan los mismos.

Sin otro particular,



Dr. César A. Beltrán Castañón
Asesor

Anexo D: Repositorios de código

El presente anexo contiene los accesos a los repositorios de código que implementan este proyecto final de carrera

- [Modelos de arquitectura de redes neuronales y protocolo de preprocesamiento](#)
- [Interfaz de Programación de Aplicaciones](#)
- [Interfaz Gráfica de Usuario](#)

Anexo E: Arquitectura de los modelos de red neuronal

- **Modelo de arquitectura de segmentación semántica de regiones urbanas en base a techos**

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 512, 512]	2,304
ReLU-2	[-1, 64, 512, 512]	0
ReLU-3	[-1, 64, 512, 512]	0
ReLU-4	[-1, 64, 512, 512]	0
ReLU-5	[-1, 64, 512, 512]	0
ReLU-6	[-1, 64, 512, 512]	0
ReLU-7	[-1, 64, 512, 512]	0
MaxPool2d-8	[-1, 64, 256, 256]	0
Conv2d-9	[-1, 128, 256, 256]	73,856
Conv2d-10	[-1, 128, 256, 256]	73,856
ReLU-11	[-1, 128, 256, 256]	0
ReLU-12	[-1, 128, 256, 256]	0
ReLU-13	[-1, 128, 256, 256]	0
ReLU-14	[-1, 128, 256, 256]	0
ReLU-15	[-1, 128, 256, 256]	0
ReLU-16	[-1, 128, 256, 256]	0
MaxPool2d-17	[-1, 128, 128, 128]	0
Conv2d-18	[-1, 256, 128, 128]	295,168
Conv2d-19	[-1, 256, 128, 128]	295,168
ReLU-20	[-1, 256, 128, 128]	0

ReLU-21	[-1, 256, 128, 128]	0
ReLU-22	[-1, 256, 128, 128]	0
ReLU-23	[-1, 256, 128, 128]	0
ReLU-24	[-1, 256, 128, 128]	0
ReLU-25	[-1, 256, 128, 128]	0
Conv2d-26	[-1, 256, 128, 128]	590,080
Conv2d-27	[-1, 256, 128, 128]	590,080
ReLU-28	[-1, 256, 128, 128]	0
ReLU-29	[-1, 256, 128, 128]	0
ReLU-30	[-1, 256, 128, 128]	0
ReLU-31	[-1, 256, 128, 128]	0
ReLU-32	[-1, 256, 128, 128]	0
ReLU-33	[-1, 256, 128, 128]	0
MaxPool2d-34	[-1, 256, 64, 64]	0
Conv2d-35	[-1, 512, 64, 64]	1,180,160
Conv2d-36	[-1, 512, 64, 64]	1,180,160
ReLU-37	[-1, 512, 64, 64]	0
ReLU-38	[-1, 512, 64, 64]	0
ReLU-39	[-1, 512, 64, 64]	0
ReLU-40	[-1, 512, 64, 64]	0
ReLU-41	[-1, 512, 64, 64]	0
ReLU-42	[-1, 512, 64, 64]	0
Conv2d-43	[-1, 512, 64, 64]	2,359,808
Conv2d-44	[-1, 512, 64, 64]	2,359,808
ReLU-45	[-1, 512, 64, 64]	0
ReLU-46	[-1, 512, 64, 64]	0
ReLU-47	[-1, 512, 64, 64]	0
ReLU-48	[-1, 512, 64, 64]	0
ReLU-49	[-1, 512, 64, 64]	0
ReLU-50	[-1, 512, 64, 64]	0
MaxPool2d-51	[-1, 512, 32, 32]	0
Conv2d-52	[-1, 512, 32, 32]	2,359,808
Conv2d-53	[-1, 512, 32, 32]	2,359,808
ReLU-54	[-1, 512, 32, 32]	0

ReLU-55	[-1, 512, 32, 32]	0
ReLU-56	[-1, 512, 32, 32]	0
ReLU-57	[-1, 512, 32, 32]	0
ReLU-58	[-1, 512, 32, 32]	0
ReLU-59	[-1, 512, 32, 32]	0
Conv2d-60	[-1, 512, 32, 32]	2,359,808
Conv2d-61	[-1, 512, 32, 32]	2,359,808
ReLU-62	[-1, 512, 32, 32]	0
ReLU-63	[-1, 512, 32, 32]	0
ReLU-64	[-1, 512, 32, 32]	0
ReLU-65	[-1, 512, 32, 32]	0
ReLU-66	[-1, 512, 32, 32]	0
ReLU-67	[-1, 512, 32, 32]	0
MaxPool2d-68	[-1, 512, 16, 16]	0
Upsample-69	[-1, 512, 32, 32]	0
Conv2d-70	[-1, 512, 32, 32]	2,359,808
ReLU-71	[-1, 512, 32, 32]	0
ConvRelu-72	[-1, 512, 32, 32]	0
Conv2d-73	[-1, 256, 32, 32]	1,179,904
ReLU-74	[-1, 256, 32, 32]	0
ConvRelu-75	[-1, 256, 32, 32]	0
DecoderBlock-76	[-1, 256, 32, 32]	0
Upsample-77	[-1, 768, 64, 64]	0
Conv2d-78	[-1, 512, 64, 64]	3,539,456
ReLU-79	[-1, 512, 64, 64]	0
ConvRelu-80	[-1, 512, 64, 64]	0
Conv2d-81	[-1, 256, 64, 64]	1,179,904
ReLU-82	[-1, 256, 64, 64]	0
ConvRelu-83	[-1, 256, 64, 64]	0
DecoderBlock-84	[-1, 256, 64, 64]	0
Upsample-85	[-1, 768, 128, 128]	0
Conv2d-86	[-1, 512, 128, 128]	3,539,456
ReLU-87	[-1, 512, 128, 128]	0
ConvRelu-88	[-1, 512, 128, 128]	0

Conv2d-89	[-1, 128, 128, 128]	589,952
ReLU-90	[-1, 128, 128, 128]	0
ConvRelu-91	[-1, 128, 128, 128]	0
DecoderBlock-92	[-1, 128, 128, 128]	0
Upsample-93	[-1, 384, 256, 256]	0
Conv2d-94	[-1, 256, 256, 256]	884,992
ReLU-95	[-1, 256, 256, 256]	0
ConvRelu-96	[-1, 256, 256, 256]	0
Conv2d-97	[-1, 64, 256, 256]	147,520
ReLU-98	[-1, 64, 256, 256]	0
ConvRelu-99	[-1, 64, 256, 256]	0
DecoderBlock-100	[-1, 64, 256, 256]	0
Upsample-101	[-1, 192, 512, 512]	0
Conv2d-102	[-1, 128, 512, 512]	221,312
ReLU-103	[-1, 128, 512, 512]	0
ConvRelu-104	[-1, 128, 512, 512]	0
Conv2d-105	[-1, 32, 512, 512]	36,896
ReLU-106	[-1, 32, 512, 512]	0
ConvRelu-107	[-1, 32, 512, 512]	0
DecoderBlock-108	[-1, 32, 512, 512]	0
Conv2d-109	[-1, 32, 512, 512]	27,680
ReLU-110	[-1, 32, 512, 512]	0
ConvRelu-111	[-1, 32, 512, 512]	0
Conv2d-112	[-1, 1, 512, 512]	33

=====
Total params: 32,146,593

Trainable params: 32,146,593

Non-trainable params: 0

Input size (MB): 4.00

Forward/backward pass size (MB): 5087.00

Params size (MB): 122.63

Estimated Total Size (MB): 5213.63

- **Modelo de arquitectura para la extracción de variables socioeconómicas en imágenes satelitales**

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 512, 512]	2,880
ReLU-2	[-1, 64, 512, 512]	0
ReLU-3	[-1, 64, 512, 512]	0
ReLU-4	[-1, 64, 512, 512]	0
ReLU-5	[-1, 64, 512, 512]	0
ReLU-6	[-1, 64, 512, 512]	0
ReLU-7	[-1, 64, 512, 512]	0
MaxPool2d-8	[-1, 64, 256, 256]	0
Conv2d-9	[-1, 128, 256, 256]	73,856
Conv2d-10	[-1, 128, 256, 256]	73,856
ReLU-11	[-1, 128, 256, 256]	0
ReLU-12	[-1, 128, 256, 256]	0
ReLU-13	[-1, 128, 256, 256]	0
ReLU-14	[-1, 128, 256, 256]	0
ReLU-15	[-1, 128, 256, 256]	0
ReLU-16	[-1, 128, 256, 256]	0
MaxPool2d-17	[-1, 128, 128, 128]	0
Conv2d-18	[-1, 256, 128, 128]	295,168
Conv2d-19	[-1, 256, 128, 128]	295,168
ReLU-20	[-1, 256, 128, 128]	0

ReLU-21	[-1, 256, 128, 128]	0
ReLU-22	[-1, 256, 128, 128]	0
ReLU-23	[-1, 256, 128, 128]	0
ReLU-24	[-1, 256, 128, 128]	0
ReLU-25	[-1, 256, 128, 128]	0
Conv2d-26	[-1, 256, 128, 128]	590,080
Conv2d-27	[-1, 256, 128, 128]	590,080
ReLU-28	[-1, 256, 128, 128]	0
ReLU-29	[-1, 256, 128, 128]	0
ReLU-30	[-1, 256, 128, 128]	0
ReLU-31	[-1, 256, 128, 128]	0
ReLU-32	[-1, 256, 128, 128]	0
ReLU-33	[-1, 256, 128, 128]	0
MaxPool2d-34	[-1, 256, 64, 64]	0
Conv2d-35	[-1, 512, 64, 64]	1,180,160
Conv2d-36	[-1, 512, 64, 64]	1,180,160
ReLU-37	[-1, 512, 64, 64]	0
ReLU-38	[-1, 512, 64, 64]	0
ReLU-39	[-1, 512, 64, 64]	0
ReLU-40	[-1, 512, 64, 64]	0
ReLU-41	[-1, 512, 64, 64]	0
ReLU-42	[-1, 512, 64, 64]	0
Conv2d-43	[-1, 512, 64, 64]	2,359,808
Conv2d-44	[-1, 512, 64, 64]	2,359,808
ReLU-45	[-1, 512, 64, 64]	0
ReLU-46	[-1, 512, 64, 64]	0

ReLU-47	[-1, 512, 64, 64]	0
ReLU-48	[-1, 512, 64, 64]	0
ReLU-49	[-1, 512, 64, 64]	0
ReLU-50	[-1, 512, 64, 64]	0
MaxPool2d-51	[-1, 512, 32, 32]	0
Conv2d-52	[-1, 512, 32, 32]	2,359,808
Conv2d-53	[-1, 512, 32, 32]	2,359,808
ReLU-54	[-1, 512, 32, 32]	0
ReLU-55	[-1, 512, 32, 32]	0
ReLU-56	[-1, 512, 32, 32]	0
ReLU-57	[-1, 512, 32, 32]	0
ReLU-58	[-1, 512, 32, 32]	0
ReLU-59	[-1, 512, 32, 32]	0
Conv2d-60	[-1, 512, 32, 32]	2,359,808
Conv2d-61	[-1, 512, 32, 32]	2,359,808
ReLU-62	[-1, 512, 32, 32]	0
ReLU-63	[-1, 512, 32, 32]	0
ReLU-64	[-1, 512, 32, 32]	0
ReLU-65	[-1, 512, 32, 32]	0
ReLU-66	[-1, 512, 32, 32]	0
ReLU-67	[-1, 512, 32, 32]	0
MaxPool2d-68	[-1, 512, 16, 16]	0
Upsample-69	[-1, 512, 32, 32]	0
Conv2d-70	[-1, 512, 32, 32]	2,359,808
ReLU-71	[-1, 512, 32, 32]	0
ConvRelu-72	[-1, 512, 32, 32]	0

Conv2d-73	[-1, 256, 32, 32]	1,179,904
ReLU-74	[-1, 256, 32, 32]	0
ConvRelu-75	[-1, 256, 32, 32]	0
DecoderBlock-76	[-1, 256, 32, 32]	0
Upsample-77	[-1, 768, 64, 64]	0
Conv2d-78	[-1, 512, 64, 64]	3,539,456
ReLU-79	[-1, 512, 64, 64]	0
ConvRelu-80	[-1, 512, 64, 64]	0
Conv2d-81	[-1, 256, 64, 64]	1,179,904
ReLU-82	[-1, 256, 64, 64]	0
ConvRelu-83	[-1, 256, 64, 64]	0
DecoderBlock-84	[-1, 256, 64, 64]	0
Upsample-85	[-1, 768, 128, 128]	0
Conv2d-86	[-1, 512, 128, 128]	3,539,456
ReLU-87	[-1, 512, 128, 128]	0
ConvRelu-88	[-1, 512, 128, 128]	0
Conv2d-89	[-1, 128, 128, 128]	589,952
ReLU-90	[-1, 128, 128, 128]	0
ConvRelu-91	[-1, 128, 128, 128]	0
DecoderBlock-92	[-1, 128, 128, 128]	0
Upsample-93	[-1, 384, 256, 256]	0
Conv2d-94	[-1, 256, 256, 256]	884,992
ReLU-95	[-1, 256, 256, 256]	0
ConvRelu-96	[-1, 256, 256, 256]	0
Conv2d-97	[-1, 64, 256, 256]	147,520
ReLU-98	[-1, 64, 256, 256]	0

ConvRelu-99	[-1, 64, 256, 256]	0
DecoderBlock-100	[-1, 64, 256, 256]	0
Upsample-101	[-1, 192, 512, 512]	0
Conv2d-102	[-1, 128, 512, 512]	221,312
ReLU-103	[-1, 128, 512, 512]	0
ConvRelu-104	[-1, 128, 512, 512]	0
Conv2d-105	[-1, 32, 512, 512]	36,896
ReLU-106	[-1, 32, 512, 512]	0
ConvRelu-107	[-1, 32, 512, 512]	0
DecoderBlock-108	[-1, 32, 512, 512]	0
Conv2d-109	[-1, 32, 512, 512]	27,680
ReLU-110	[-1, 32, 512, 512]	0
ConvRelu-111	[-1, 32, 512, 512]	0
Conv2d-112	[-1, 5, 512, 512]	165

=====

Total params: 32,147,301

Trainable params: 32,147,301

Non-trainable params: 0

Input size (MB): 5.00

Forward/backward pass size (MB): 5095.00

Params size (MB): 122.63

Estimated Total Size (MB): 5222.63

Anexo F: Documentación de la Interfaz de Programación de Aplicaciones

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**DETERMINACIÓN Y SEGMENTACIÓN DEL NIVEL
SOCIO-ECONÓMICO A PARTIR DEL ANÁLISIS AUTOMÁTICO
DE IMÁGENES DE SATÉLITE**

Documentación de la Interfaz de Programación de Aplicaciones

AUTOR:

Carlos Alberto Sosa Pezo

ASESOR:

Dr. César Armando Beltrán Castañón

Lima, Abril, 2021

TABLA DE CONTENIDOS

Lista de imágenes	32
Recursos	32
Bounding box	32
left	32
bottom	32
right	32
top	32
Diccionario de información de una imagen satelital	33
path	33
name	33
bounding_box	33
Diccionarios de información de imágenes satelital	34
images	34
Endpoint	35
Listar imágenes satelitales	35
Solicitud	35
Respuesta	36
Máscaras	37
Recursos	37
Máscara	37
url	37
level	37
Lista de máscaras	38
bounding_box	38
masks	38
Lista de máscaras asignadas a una zona urbana	39
is_located	39
images	39
Endpoint	40
Revisar si una máscara fue procesada anteriormente	40
Solicitud	40
Respuesta	40
Procesar una máscara	41
	30

Solicitud	41
Respuesta	42

Lista de imágenes

Recursos

Bounding box

Coordenadas que delimitan un área en el mapa terrestre (área de interés)

Ejemplo

```
{  
  "left": -77.1010,  
  "bottom": -12.0664,  
  "right": -77.0249,  
  "top": -12.0112,  
}
```

left

- **Tipo:** float
- **Ejemplo:** -77.1010
- Límite oeste del área de interés

bottom

- **Tipo:** float
- **Ejemplo:** -12.0664
- Límite sur del área de interés

right

- **Tipo:** float
- **Ejemplo:** -77.0249
- Límite este del área de interés

top

- **Tipo:** float
- **Ejemplo:** -12.0112
- Límite norte del área de interés

Diccionario de información de una imagen satelital

Información básica de una imagen satelital almacenada en un servidor de imágenes

Ejemplo

```
{
  "path":
  "./data/PER_001/IMG_PER1_20190131153309_ORT_P_000621.TIF",
  "name": "IMG_PER1_20190131153309_ORT_P_000621.TIF",
  "bounding_box": {
    "left": -77.1010,
    "bottom": -12.0664,
    "right": -77.0249,
    "top": -12.0112,
  }
}
```

path

- **Tipo:** string
- **Ejemplo:** ./data/PER_001/IMG_PER1_20190131153309_ORT_P_000621.TIF
- Ruta del servidor de imágenes en la que se encuentra un archivo de imagen satelital

name

- **Tipo:** string
- **Ejemplo:** IMG_PER1_20190131153309_ORT_P_000621.TIF
- Nombre del archivo que contiene la imagen satelital

bounding_box

- **Tipo:** objeto del tipo [Bounding Box](#)
- Diccionario con coordenadas que describen la ubicación de la imagen satelital

Diccionarios de información de imágenes satelital

Lista que detalla la información básica de un número de imágenes satelitales almacenadas en un servidor de imágenes

Ejemplo

```
{
  "images": [
    {
      "path":
"/data/PER_001/IMG_PER1_20190131153309_ORT_P_000621.TIF",
      "name": "IMG_PER1_20190131153309_ORT_P_000621.TIF",
      "bounding_box": {
        "left": -77.1010,
        "bottom": -12.0664,
        "right": -77.0249,
        "top": -12.0112,
      }
    }
  ]
}
```

images

- **Tipo:** array de [diccionarios de información de imágenes satelital](#)

Endpoint

Listar imágenes satelitales

POST	api/searchImages/
-------------	-------------------

Obtiene la información básica de todas las imágenes satelitales del servidor que contengan parcial o totalmente un área de interés.

Solicitud

Content-Type	application/json
---------------------	------------------

Cuerpo de la solicitud

left

- **Tipo:** float
- **Ejemplo:** -77.1010
- Límite oeste del área de interés

bottom

- **Tipo:** float
- **Ejemplo:** -12.0664
- Límite sur del área de interés

right

- **Tipo:** float
- **Ejemplo:** -77.0249
- Límite este del área de interés

top

- **Tipo:** float
- **Ejemplo:** -12.0112
- Límite norte del área de interés

Respuesta

400 (BAD REQUEST)

- **Content-Type:** application/json
- **Tipo:** mensaje de error
- Retorna un mensaje de error alertando no haber recibido datos

200 (OK)

- **Content-Type:** application/json
- **Tipo:** [diccionarios de información de imágenes satelital](#)
- Retorna la lista de información básica de las imágenes que cumplen con el área de interés.

Máscaras

Recursos

Máscara

Mapa de bits que representa los espacios urbanos asignados a determinado nivel socioeconómico.

Ejemplo

```
{  
  "url": "http://2.44.66.194:8008/MASK_4.0.PNG",  
  "level": "4.0"  
}
```

url

- **Tipo:** string
- **Ejemplo:** http://2.44.66.194:8008/MASK_4.0.PNG
- URL de la máscara, provista por la API

level

- **Tipo:** string
- **Ejemplo:** 4.0
- Etiqueta del nivel socioeconómico al que pertenece la máscara

Lista de máscaras

Lista que máscaras de espacios urbanos asignados a cada nivel socioeconómico.

Ejemplo

```
{
  "bounding_box": {
    "left": -77.1010,
    "bottom": -12.0664,
    "right": -77.0249,
    "top": -12.0112,
  },
  "masks": [{
    "url": "http://2.44.66.194:8008/MASK_4.0.PNG",
    "level": "4.0"
  }]
}
```

bounding_box

- **Tipo:** objeto del tipo [Bounding Box](#)
- Diccionario con coordenadas que describen la ubicación de la imagen satelital

masks

- **Tipo:** array de [máscaras](#)

Lista de máscaras asignadas a una zona urbana

Lista que máscaras de espacios urbanos asignados a cada nivel socioeconómico, asociados a un espacio geográfico.

Ejemplo

```
{
  "is_located": True,

  "image": {
    "bounding_box": {
      "left": -77.1010,
      "bottom": -12.0664,
      "right": -77.0249,
      "top": -12.0112,
    },

    "masks": [{
      "url": "http://2.44.66.194:8008/MASK_4.0.PNG",
      "level": "4.0"
    }]
  }
}
```

is_located

- **Tipo:** bool
- Variable lógica que indica si está disponible información de las máscaras

images

- **Tipo:** [lista de máscaras](#)
- array de máscaras georreferenciadas, en caso haberlas

Endpoint

Revisar si una máscara fue procesada anteriormente

GET	api/checkFiles/
------------	-----------------

Corroborar si determinada imagen fue procesada anteriormente por los modelos de redes neuronales. De ser así, entregar una referencia a la imagen.

Solicitud

Content-Type	application/json
---------------------	------------------

Parámetros

name

- **Tipo:** string
- **Ejemplo:** IMG_PER1_20190131153309_OR_T_P_000621.TIF
- Nombre del archivo de una imagen satelital a consultar

Respuesta

400 (BAD REQUEST)

- **Content-Type:** application/json
- **Tipo:** mensaje de error
- Retorna un mensaje de error alertando no haber recibido datos

200 (OK)

- **Content-Type:** application/json
- **Tipo:** [lista de máscaras asignadas a una zona urbana](#)
- Retorna, de existir, las máscaras socioeconómicas de una zona urbana, previamente procesada

200 (OK)

- **Content-Type:** application/json
- **Tipo:** [lista de máscaras asignadas a una zona urbana](#)
- Retorna una variable que indica la no existencia de máscaras previamente procesadas

Procesar una máscara

POST	api/predict/
-------------	--------------

Obtiene desde el servidor de imágenes una imagen satelital solicitada, la procesa mediante los modelos de redes neuronales para determinar los niveles socioeconómicos a los que pertenecen cada manzana de la imagen y devuelva las máscaras asociada a cada nivel

Solicitud

Content-Type	application/json
---------------------	------------------

Parámetros

filepath

- **Tipo:** string
- **Ejemplo:** ./data/PER_001/IMG_PER1_20190131153309_ORT_P_000621.TIF
- Ruta del archivo de una imagen satelital a consultar

Respuesta

400 (BAD REQUEST)

- **Content-Type:** application/json
- **Tipo:** mensaje de error
- Retorna un mensaje de error alertando no haber recibido la ruta de la imagen

400 (BAD REQUEST)

- **Content-Type:** application/json
- **Tipo:** mensaje de error
- Retorna un mensaje de error alertando que la ruta no corresponde a una imagen

400 (BAD REQUEST)

- **Content-Type:** application/json
- **Tipo:** mensaje de error
- Retorna un mensaje de error alertando que la imagen en la ruta no cuenta con la cantidad correcta de bandas (4)

400 (BAD REQUEST)

- **Content-Type:** application/json
- **Tipo:** mensaje de error
- Retorna un mensaje de error alertando que la imagen en la ruta no puede ser dividida en parches de 512 x 512 píxeles (por ser de un menor tamaño)

200 (OK)

- **Content-Type:** application/json
- **Tipo:** [lista de máscaras asignadas a una zona urbana](#)
- Retorna, de existir, las máscaras socioeconómicas de una zona urbana procesada

Anexo G: Secuencia del prototipo

El presente anexo consta de las vistas diseñadas durante el prototipado del cliente web, de forma que sirvan como referencia para la programación de las interfaces finales funcionales.

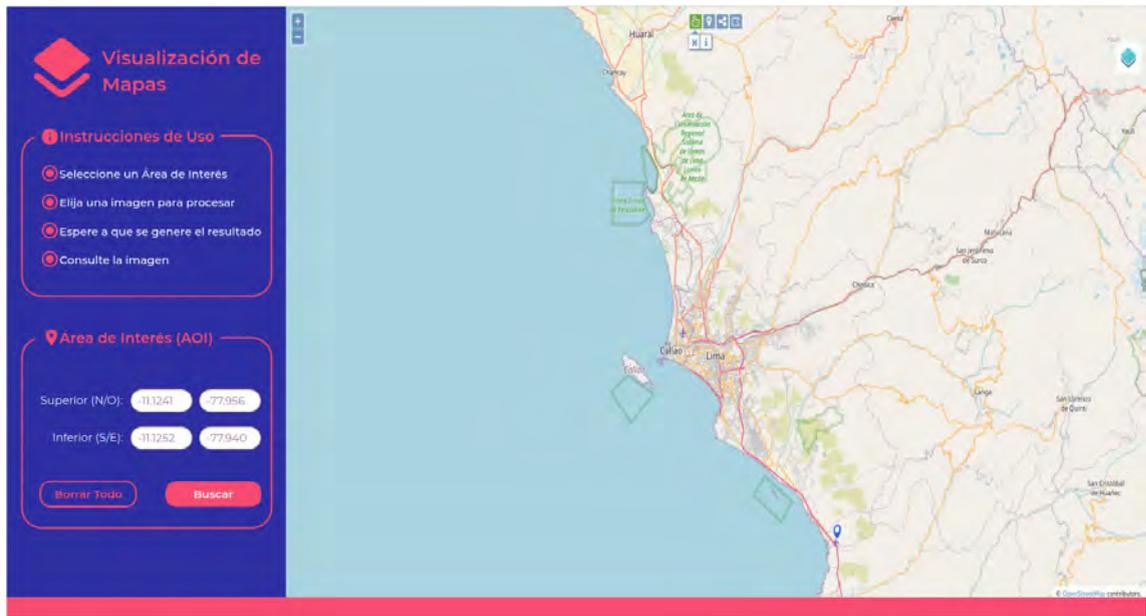


Figura 16. Prototipo de la interfaz gráfica de usuario, vista inicial.

Fuente: Elaboración propia

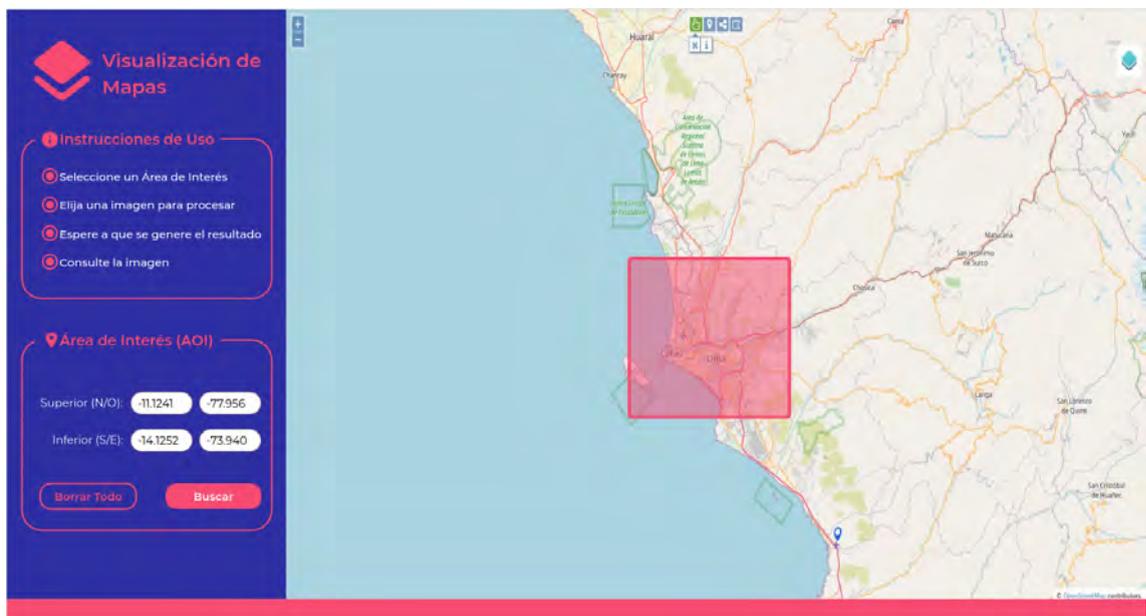


Figura 19. Prototipo de la interfaz gráfica de usuario, selección del área de interés.

Fuente: Elaboración propia

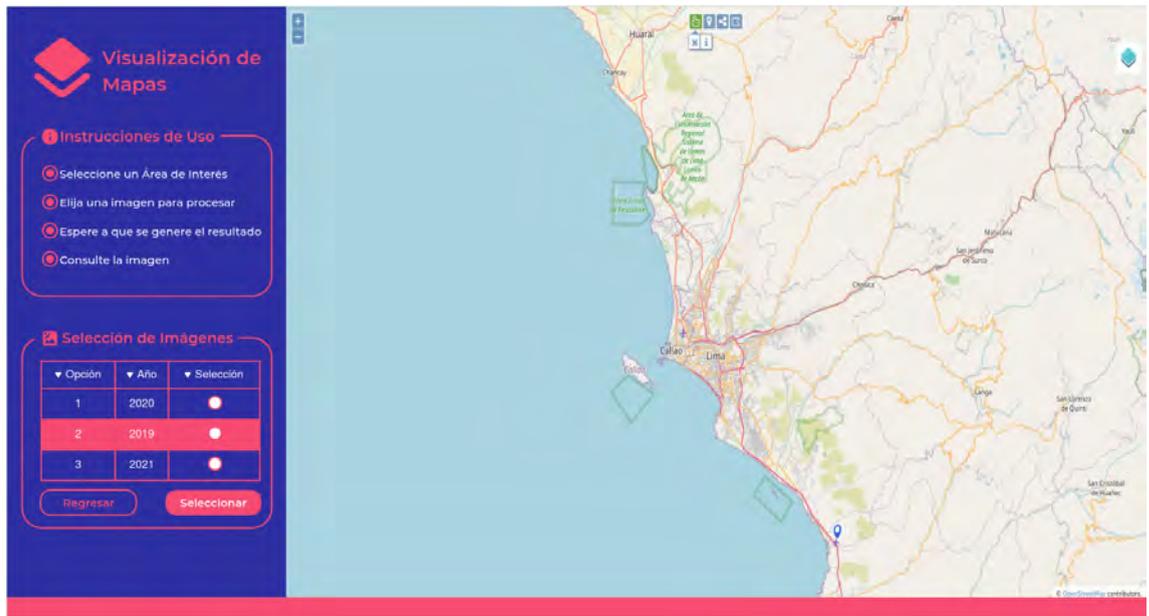


Figura 20. Prototipo de la interfaz gráfica de usuario, lista de imágenes candidatas.

Fuente: Elaboración propia

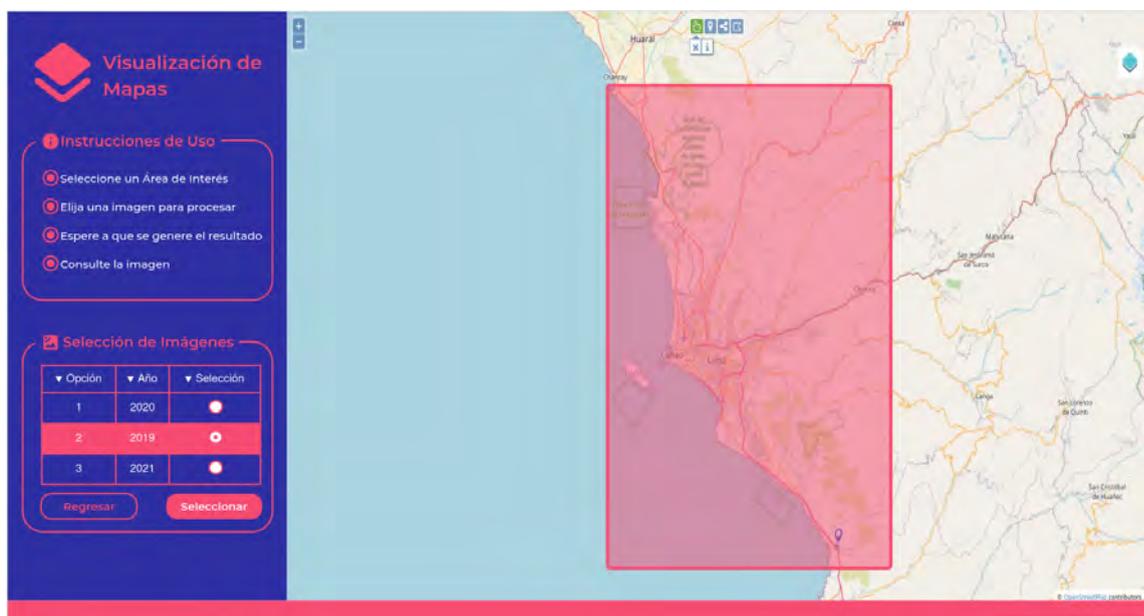


Figura 21. Prototipo de la interfaz gráfica de usuario, selección de una imagen para procesar.

Fuente: Elaboración propia

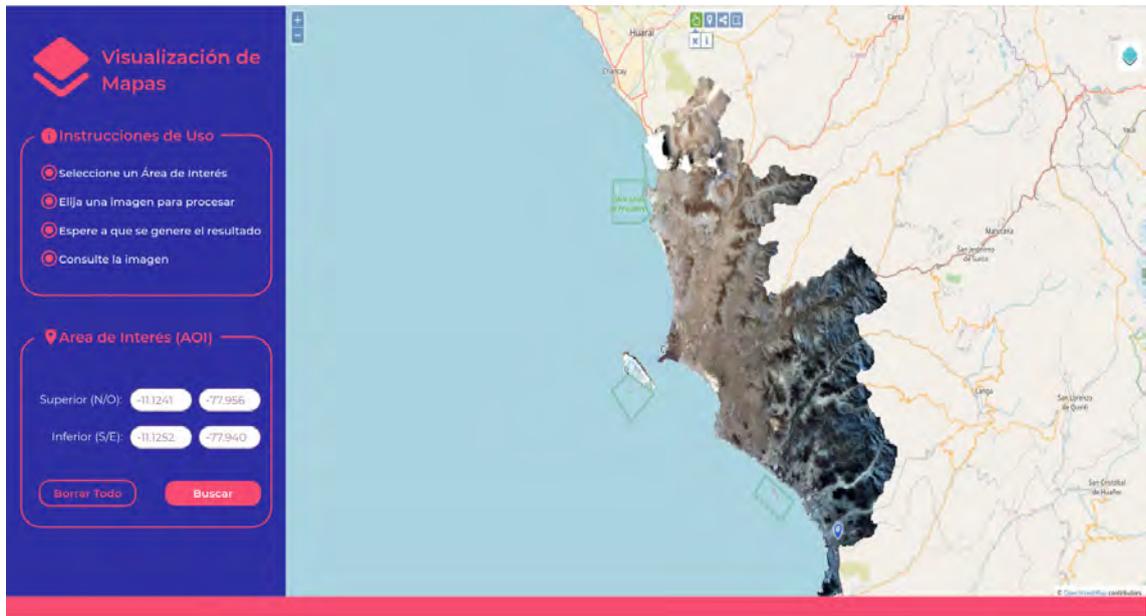


Figura 22. Prototipo de la interfaz gráfica de usuario, imagen recibida.

Fuente: Elaboración propia

Anexo H: Manual de uso de la aplicación web

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**Visualización de Mapas: Aplicación web para la determinación y
segmentación del nivel socioeconómico en zonas urbanas a través del
procesamiento de imágenes satelitales**

Manual de usuario

AUTOR:

Carlos Alberto Sosa Pezo

ASESOR:

Dr. César Armando Beltrán Castañón

Lima, mayo de 2021

Historial de revisiones

Versión	Fecha	Realizado por	Descripción
1.0	14/06/2021	Carlos Alberto Sosa Pezo	Versión completa inicial
0.1	3/05/2021	Carlos Alberto Sosa Pezo	Versión preliminar

Tabla de contenidos

Manual de uso de aplicación web	49
1. Introducción	49
2. Visualización de Mapas	49
3. Presentación de vistas	50
3.1. Vista 1: Selección de región de interés	50
3.2. Vista 2: Selección de imagen satelital a procesar	51
3. Contacto y soporte técnico	52

Manual de uso de aplicación web

1. Introducción

El objetivo del presente documento es la presentación de la aplicación web para la determinación y segmentación del nivel socioeconómico en zonas urbanas a través del procesamiento de imágenes satelitales, así como indicar el correcto uso de la plataforma en cuestión, de forma que exista un documento de referencia para su posterior uso por sus usuarios finales.

2. Visualización de Mapas

«Visualización de Mapas» es una herramienta web desarrollada con el fin de proveer un servicio que determine el nivel socioeconómico a nivel de manzanas de una zona urbana, empleando como referencia imágenes satelitales de la zona en cuestión. Este sistema cuenta con una interfaz gráfica, de modo que el usuario final pueda acceder a este servicio de forma sencilla. Además, el usuario final puede consultar las máscaras generadas por este servicio tanto de forma individual como en conjunto.

Así, la presente herramienta consiste en 2 vistas principales:

- Selección de región de interés
- Selección de imagen satelital a procesar

El uso de estas será explicado en las secciones a continuación.

3. Presentación de vistas

3.1. Vista 1: Selección de región de interés

Al acceder a la aplicación, se visualizará una vista como la siguiente:

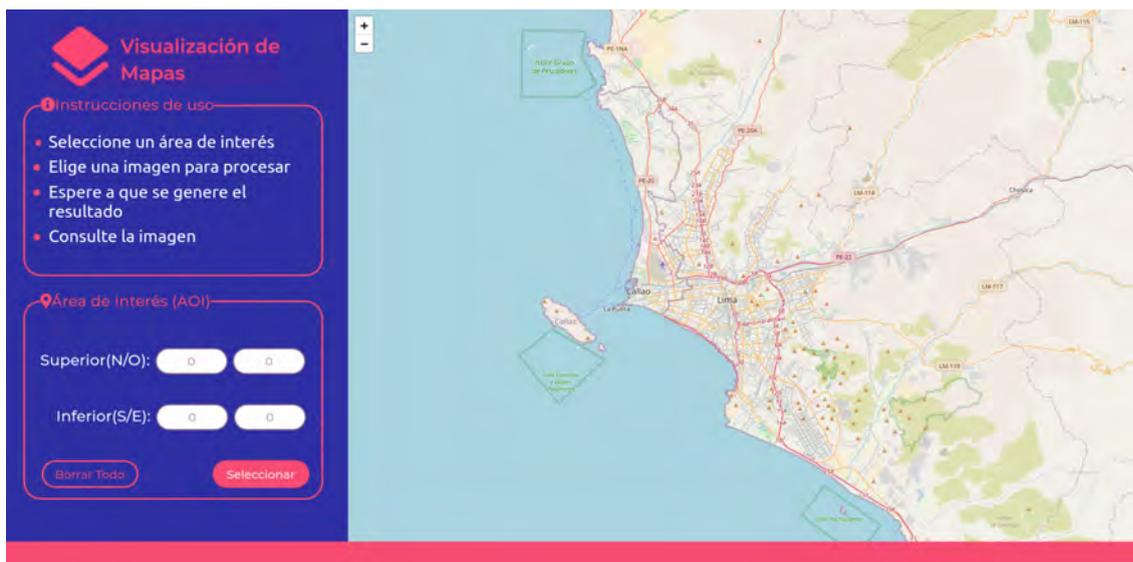


Figura 23. Ejecución de la plataforma web, vista inicial

Fuente: Elaboración propia

Esta vista, tal como se muestra en la columna izquierda de la Figura 1, permite al usuario seleccionar a través del mapa un área que se pretende consultar, la cual se ha denominado «área de interés» (AOI).

Para seleccionar un AOI, se debe seleccionar con el puntero del ratón una región visible del mapa del lado izquierdo de la pantalla y dar un clic. A continuación, se habrá seleccionado un rectángulo de 100 km² de área centrado en la posición pulsada.

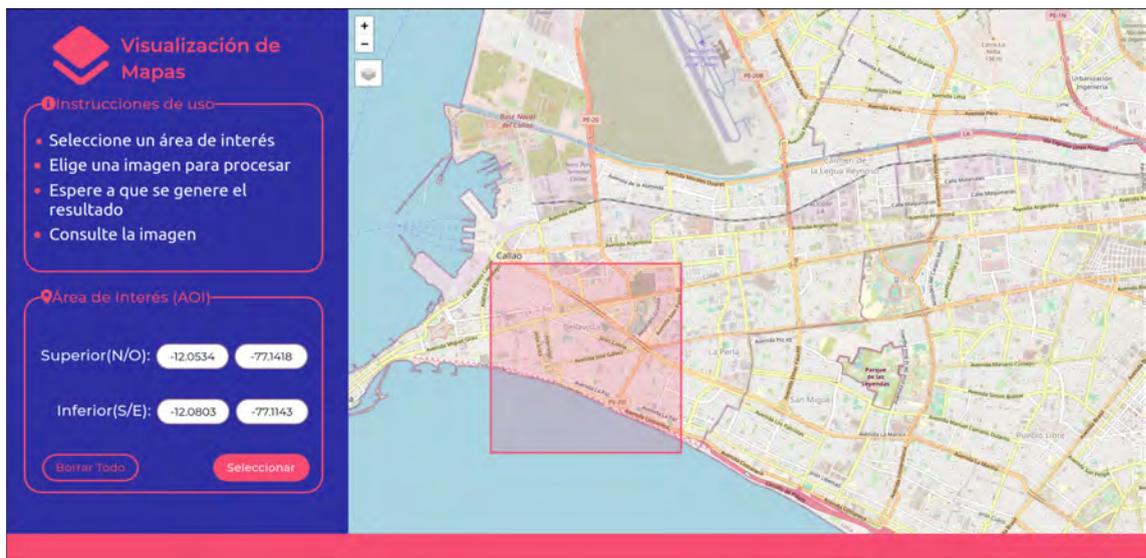


Figura 24. Ejecución de la plataforma web, selección del Área de Interés
Fuente: Elaboración propia

Tras ello, se da un clic en el botón **Seleccionar** para iniciar con la búsqueda de imágenes satelitales (Tiempo estimado: entre 10 y 15 segundos)

3.2. Vista 2: Selección de imagen satelital a procesar

Una vez terminada la carga, se mostrará una lista de imágenes candidatas de acuerdo al AOI elegido.

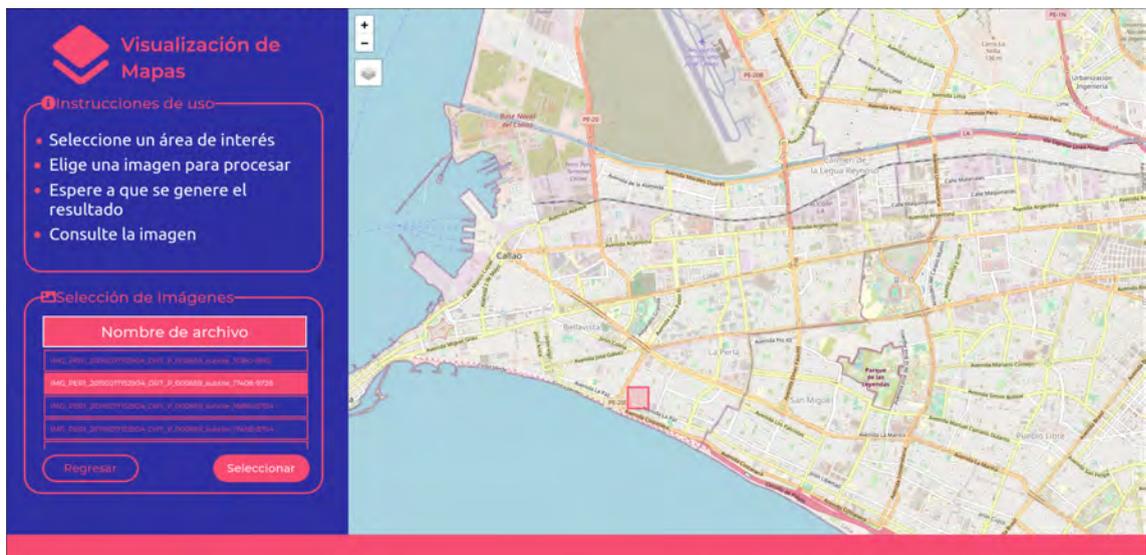


Figura 25. Ejecución de la plataforma web, selección de imagen a procesar

Fuente: *Elaboración propia*

Para iniciar el procesamiento, se debe seleccionar (hacer clic) en el botón una imagen de dicha lista. Finalmente, podemos dar clic al botón **Seleccionar** para confirmar la selección y ejecutar el procesamiento

De haber sido la imagen seleccionada procesada con anterioridad, el resultado de la misma se mostrará en un tiempo breve. Caso contrario, el proceso puede tardar 5 segundos aproximadamente. El resultado se visualiza superpuesto al mapa, en la ubicación designada.

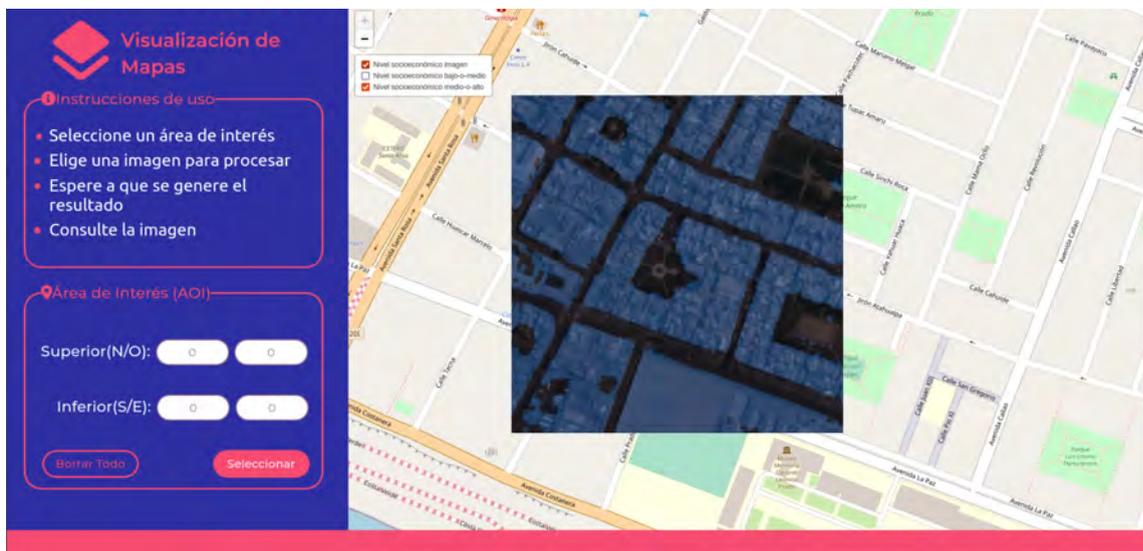


Figura 26. Ejecución de la plataforma web, ubicación de la imagen satelital y máscaras de predicción en el mapa

Fuente: *Elaboración propia*

3. Contacto y soporte técnico

De encontrarse algún error en la aplicación web o requerir asistencia adicional, por favor, contáctese con Carlos Alberto Sosa Pezo (c.sosa@pucp.edu.pe / +51999943247).

Anexo I: Reportes de pruebas unitarias y funcionales

El presente anexo consiste en un reporte de los resultados de las pruebas realizadas al sistema web, tanto a cada uno de sus componentes como al sistema en funcionamiento. De ser el caso, se menciona qué función (ubicada en el repositorio de su componente, mencionada en el **Anexo D (Repositorios de código)**) implementa la prueba y a qué archivo corresponde.

Backend (pruebas unitarias)

Código	Nombre	Objetivo	Datos de entrada	Resultado esperado	Archivo	Función	Estado
BE01	Lectura modo depuración (verdadero)	Revisar si se asigna «verdadero» al modo depuración en caso de declararlo así en el archivo de configuración	[DEVELOPMENT] DEBUG = True PORT = 5000 HOST = 0.0.0.0 SECRET_KEY = 70e34e3bce494540edafd078c6109645 CORS_HEADERS = Access-Control-Allow-Origin	test_config.DEBUG = True	test_app_config.py	test_debug_true_is_read()	Resultado satisfactorio
BE02	Lectura modo depuración (falso)	Revisar si se asigna «falso» al modo depuración en caso de declararlo así en el archivo de configuración	[DEVELOPMENT] DEBUG = False PORT = 5000 HOST = 0.0.0.0 SECRET_KEY = 70e34e3bce494540edafd078c6109645 CORS_HEADERS = Access-Control-Allow-Origin	test_config.DEBUG = True	test_app_config.py	test_debug_true_is_read()	Resultado satisfactorio

BE03	Lectura de host (aplicación)	Revisar si el host designado en el archivo de configuración es leído	[DEVELOPMENT] DEBUG = False PORT = 5000 HOST = 0.0.0.0 SECRET_KEY = 70e34e3bce494540edafd078c6109645 CORS_HEADERS = Access-Control-Allow-Origin	test_config.HOST = '0.0.0.0'	test_app_config.py	test_host_is_read()	Resultado satisfactorio
BE04	Lectura de puerto	Revisar si el puerto designado en el archivo de configuración es leído	[DEVELOPMENT] DEBUG = False PORT = 5000 HOST = 0.0.0.0 SECRET_KEY = 70e34e3bce494540edafd078c6109645 CORS_HEADERS = Access-Control-Allow-Origin	test_config.PORT = 9997	test_app_config.py	test_port_is_read()	Resultado satisfactorio
BE05	Lectura de clave secreta	Revisar si la clave secreta designada en el archivo de configuración es leída	[DEVELOPMENT] DEBUG = False PORT = 5000 HOST = 0.0.0.0 SECRET_KEY = 70e34e3bce494540edafd078c6109645 CORS_HEADERS = Access-Control-Allow-Origin	test_config.SECRET_KEY = '70e34e3bce494540edafd078c6109645'	test_app_config.py	test_secret_key_is_read()	Resultado satisfactorio

BE06	Lectura de CORS header	Revisar si los CORS headers designados en el archivo de configuración son leídos	[DEVELOPMENT] DEBUG = False PORT = 5000 HOST = 0.0.0.0 SECRET_KEY = 70e34e3bce494540edafd078c6109645 CORS_HEADERS = Access-Control-Allow-Origin	test_config.CORS_HEADERS = 'Access-Control-Allow-Origin'	test_app_config.py	test_cors_headers_is_read()	Resultado satisfactorio
BE07	Lectura de niveles socioeconómicos	Revisar si los niveles socioeconómicos designados en el archivo de configuración son leídos	[INFO] '0' = 0 '1' = 1 '2' = 2 [0] 'R' = 100 'G' = 100 'B' = 100 [1] 'R' = 100 'G' = 100 'B' = 100 [2] 'R' = 100 'G' = 100 'B' = 100	len(test_config) = 3	test_income_levels.py	test_income_level_is_read	Resultado satisfactorio
BE08	Lectura de la ruta del modelo de segmentación semántica en	Revisar si la ubicación en el servidor del modelo	[ROOF] STATE_DICT = foo_path.h5 [INCOME] STATE_DICT = bar_path.h5	test_config.roof_model_state_dict = 'foo_path.h5'	test_model_credentials.py	test_roof_segmentation_path_is_read	Resultado satisfactorio

	base a techos	preentrenado de segmentación semántica en base a techos es leída					
BE09	Lectura de la ruta del modelo de extracción	Revisar si la ubicación en el servidor del modelo preentrenado de extracción de variables socioeconómicas es leída	[ROOF] STATE_DICT = foo_path.h5 [INCOME] STATE_DICT = bar_path.h5	test_config.income_model_state_dict = 'bar_path.h5'	test_model_credentials.py	test_income_determination_is_read	Resultado satisfactorio
BE10	Lectura de host (servidor de imágenes)	Revisar si el host designado en el archivo de configuración es leído	[SFTP] HOSTNAME = 0.0.0.0 USERNAME = testusername PASSWORD = testpassword	test_config.sftp_hostname = '0.0.0.0'	test_sftp_credentials.py	test_hostname_is_read	Resultado satisfactorio
BE11	Lectura del nombre de usuario	Revisar si el nombre de usuario designado en el archivo de configuración es leído	[SFTP] HOSTNAME = 0.0.0.0 USERNAME = testusername PASSWORD = testpassword	test_config.sftp_username = 'testuser'	test_sftp_credentials.py	test_username_is_read	Resultado satisfactorio

BE12	Lectura de contraseña	Revisar si la contraseña designado en el archivo de configuración es leída	[SFTP] HOSTNAME = 0.0.0.0 USERNAME = testusername PASSWORD = testpassword	test_config.sftp_password = 'testpassword'	test_sftp_credentials.py	test_password_is_read	Resultado satisfactorio
BE13	Dimensiones de la máscara refinada	Revisar si, tras el refinamiento de la máscara de nivel socioeconómico, la dimensión de la misma concuerda con la de la máscara original, tras reducirle un nivel	mask = np.random.rand(1, 2, 512, 512) roof_mask = np.random.rand(1, 1, 512, 512)	result.shape == (2, 512, 512)	test_image_utils.py	test_define_image_has_valid_shape	Resultado satisfactorio
BE14	Mínimo de la máscara refinada	Revisar si el mínimo de la máscara tras refinar es 0	mask = np.random.rand(1, 2, 512, 512) roof_mask = np.random.rand(1, 1, 512, 512)	result.min() == 0	test_image_utils.py	test_define_image_min	Resultado satisfactorio
BE15	Máximo de la máscara refinada	Revisar si el máximo de la máscara tras refinar es 1	mask = np.random.rand(1, 2, 512, 512) roof_mask = np.random.rand(1, 1, 512, 512)	result.max() == 1	test_image_utils.py	test_define_image_max	Resultado satisfactorio

BE16	Tipo de dato de la máscara refinada	Revisar si la máscara refinada está compuesta de enteros	mask = np.random.rand(1, 2, 512, 512) roof_mask = np.random.rand(1, 1, 512, 512)	result.dtype == int	test_image_utils.py	test_define_image_type	Resultado satisfactorio
[*] Solo han sido objetos de prueba funciones nuevas implementadas para el presente proyecto y que no forman parte del proyecto que usa este como base							

Backend (prueba de servicios)

Código	Nombre	Objetivo	Precondición	Datos de entrada	Resultado esperado	API	Estado
BE17	Buscar imágenes en el servidor	Revisar el funcionamiento del servicio de búsqueda de imágenes en el servidor		{ "left": -77.1418, "bottom": -12.0803, "right": -77.1143, "top": -12.0534 }	{images: [{"filepath": IMG_PER1_20190217152904_ORT_P_000659_subtile_15360-8192.tif, "filename": IMG_PER1_20190217152904_ORT_P_000659_subtile_15360-8192, "bounding_box": {"top": top, "bottom": bottom, "left": left, "right": right}, ...}]}	/api/searchimages	Resultado satisfactorio

BE18	Procesar imagen	Revisar el funcionamiento del servicio de predicción (procesamiento de imágenes)		filename: IMG_PER1_20190217152904_ORT_P_000659_subtile_15360-8192.tif	{is_located: ", image: { "bounding_box": {"top": top, "bottom": bottom, "left": left, "right": right}, 'masks': [{"url": '/static/filename: IMG_PER1_20190217152904_ ORT_P_000659_subtile_15360- 8192_bajo-o-medio.png', 'level': 'bajo-o-medio'}, {'url': '/static/filename: IMG_PER1_20190217152904_ ORT_P_000659_subtile_15360- 8192_medio-alto.png', 'level': 'medio-alto'}, {'url': '/static/filename: IMG_PER1_20190217152904_ ORT_P_000659_subtile_15360- 8192_alto.png', 'level': 'alto'}}}	/api/predict	Resultado satisfactorio
BE19	Revisar máscaras preexistentes (no existente)	Revisar el funcionamiento del servicio de revisión de máscaras preexistentes cuando una imagen no ha sido procesada anteriormente		filename: IMG_PER1_20190217152904_ORT_P_000659_subtile_15360-8192.tif	{is_located: False}	/api/checkFiles	Resultado satisfactorio

BE20	Revisar máscaras preexistentes (existente)	Revisar el funcionamiento del servicio de revisión de máscaras preexistentes cuando una imagen ha sido procesada anteriormente	La imagen solicitada ha sido procesada anteriormente (*)	filename: IMG_PER1_20190217152904_ORT_P_000659_subtile_15360-8192.tif	{is_located: True, image: { "bounding_box": {"top": top, "bottom": bottom, "left": left, "right": right}, 'masks': [{url': '/static/filename: IMG_PER1_20190217152904_ORT_P_000659_subtile_15360-8192_bajo-o-medio.png', 'level': 'bajo-o-medio'}, {'url': '/static/filename: IMG_PER1_20190217152904_ORT_P_000659_subtile_15360-8192_medio-alto.png', 'level': 'medio-alto'}, {'url': '/static/filename: IMG_PER1_20190217152904_ORT_P_000659_subtile_15360-8192_alto.png', 'level': 'alto'}}]}	/api/checkFiles	Resultado satisfactorio
(*) BE20 se realiza después de BE18 y BE19							

Frontend (pruebas unitarias)

Código	Nombre	Objetivo	Datos de entrada	Resultado esperado	Archivo	Estado
--------	--------	----------	------------------	--------------------	---------	--------

FE01	Área de Interés	Revisar si se renderiza los elementos del componente de área de interés	<pre>let bounds = { "north": 0, "south": 0, "west": 0, "east": 0 };</pre>	Se renderiza las coordenadas máximas e mínimas en el componente del área de interés, de acuerdo con la variable bounds de los datos de entrada, además de renderizarse los textos de encabezado y botones	aoi.test.js	Resultado satisfactorio
FE02	Lista de imágenes	Revisar si se renderiza los elementos de la lista de imágenes, además de que se genera una tabla con la lista de imágenes disponibles	<pre>images = [{"name": "foo"}, {"name": "bar"}]</pre>	Se renderiza una tabla de nombres de imágenes de acuerdo con los datos de entrada, además de renderizarse los textos de encabezado y botones	imagelist.test.js	Resultado satisfactorio
FE03	Infobox	Revisar si se renderiza los elementos del componente infobox	-	Se renderiza el encabezado	infobox.test.js	Resultado satisfactorio
FE04	Barra lateral	Revisar si el puerto designado en el archivo de configuración es leído	-	Se renderiza el encabezado y los componentes	sidebar.test.js	Resultado satisfactorio

				hijos		
--	--	--	--	-------	--	--

Plataforma

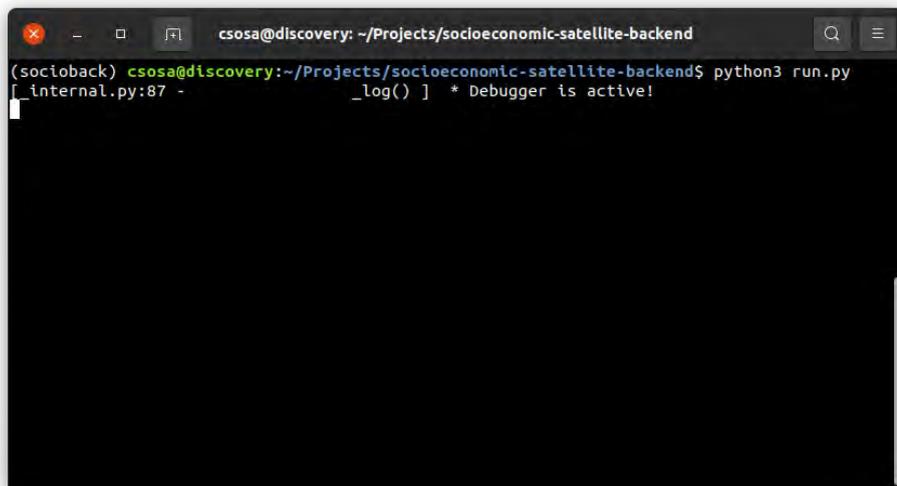
Código	Nombre	Objetivo	Precondiciones	Caso de prueba	Resultado esperado	Estado
PT01	Procesar imagen (no procesada anteriormente)	Revisar el correcto funcionamiento del flujo de la plataforma cuando la imagen no ha sido procesada	-	Ver detalle	Se obtiene una máscara dividida por capas que definen el nivel socioeconómico o de la imagen solicitada, además de la imagen original	Resultado satisfactorio
PT02	Procesar imagen (procesada anteriormente)	Revisar el correcto funcionamiento del flujo de la plataforma cuando la imagen ya ha sido procesada anteriormente	La imagen solicitada ha sido procesada anteriormente		Se obtiene una máscara dividida por capas que definen el nivel socioeconómico o de la imagen solicitada, además de la imagen	Resultado satisfactorio

					original. El tiempo de entrega se reducen en aproximadamente 5 segundos.	
--	--	--	--	--	--	--

Detalle	
Número	Pasos a ejecutar
1	Entrar a la plataforma de visualización de imágenes
2	Hacer clic en el centro del Estadio Miguel Grau
3	Hacer clic en el botón «seleccionar»
4	Se recibe una lista de nombres de imágenes
5	Seleccionar MG_PER1_20190217152904_OR_T_P_000659_subtile_17408-9728
6	Hacer clic en el botón «seleccionar»

Anexo J: Capturas de pantalla de las pruebas de ejecución de la plataforma

El presente anexo consiste en capturas de pantallas efectuadas durante la ejecución de pruebas del sistema web, tanto de los componentes cliente y servidor, así como de su integración.



```
csosa@discovery: ~/Projects/socioeconomic-satellite-backend
(socioback) csosa@discovery:~/Projects/socioeconomic-satellite-backend$ python3 run.py
[_internal.py:87 - _log() ] * Debugger is active!
```

Figura 27. Ejecución del servidor web en el servidor remoto, visto desde Bash

Fuente: Elaboración propia

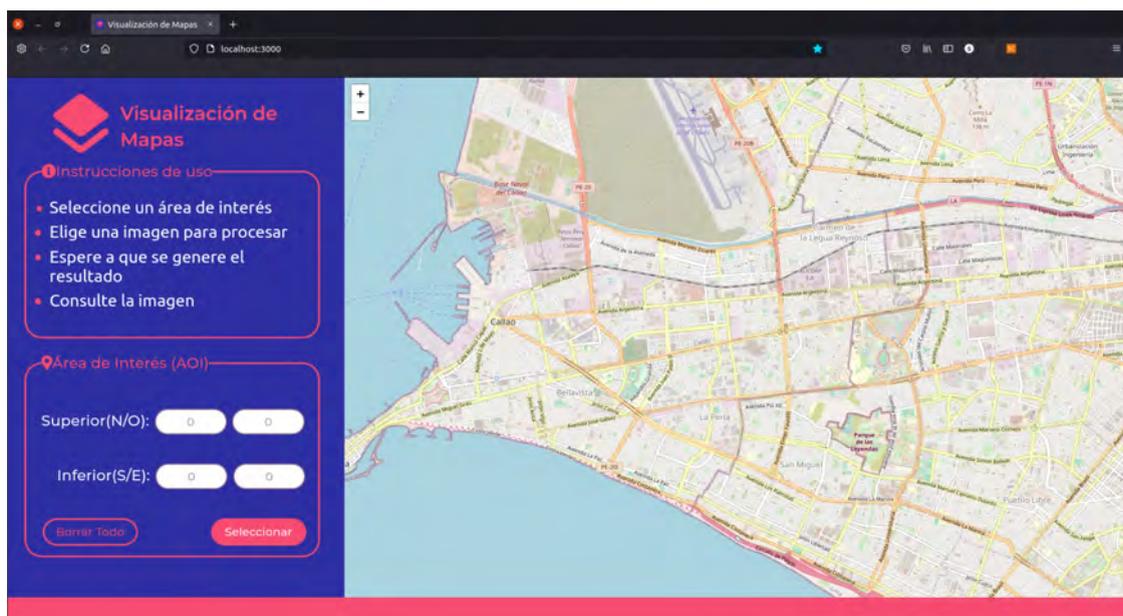


Figura 23. Ejecución de la plataforma web, vista inicial

Fuente: Elaboración propia

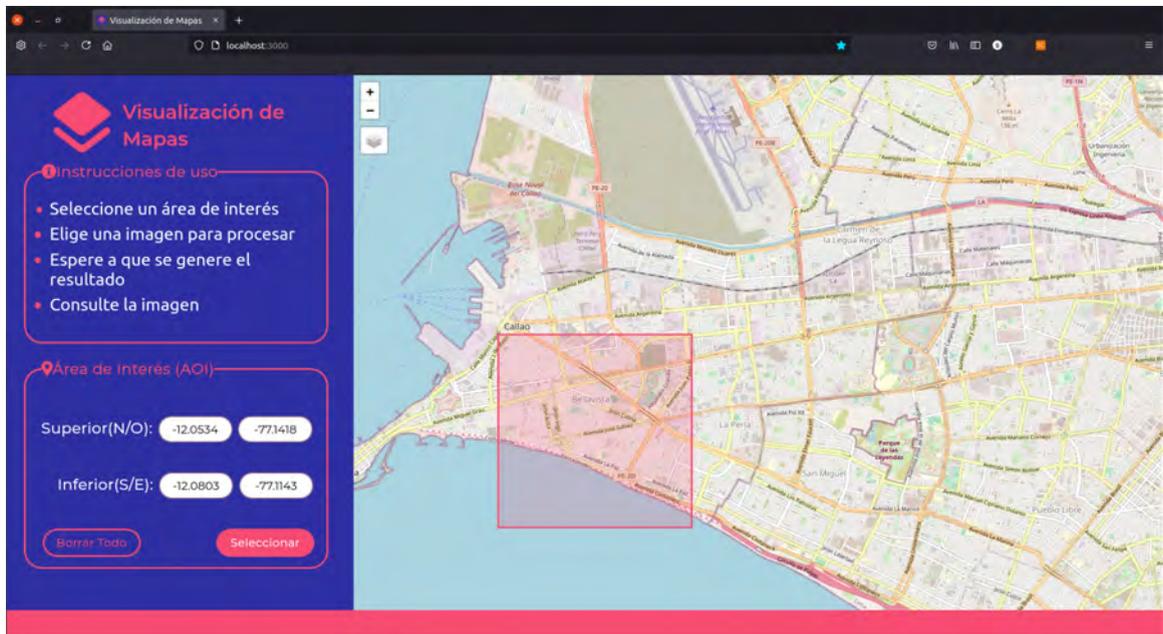


Figura 24. Ejecución de la plataforma web, selección del Área de Interés

Fuente: Elaboración propia

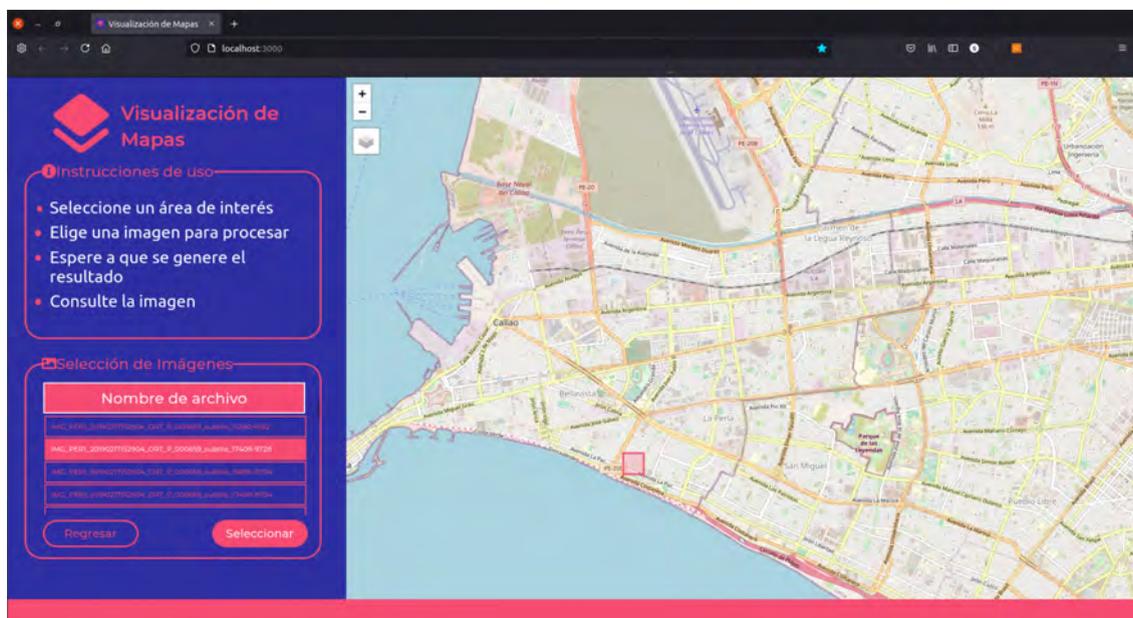


Figura 25. Ejecución de la plataforma web, selección de imagen a procesar

Fuente: Elaboración propia

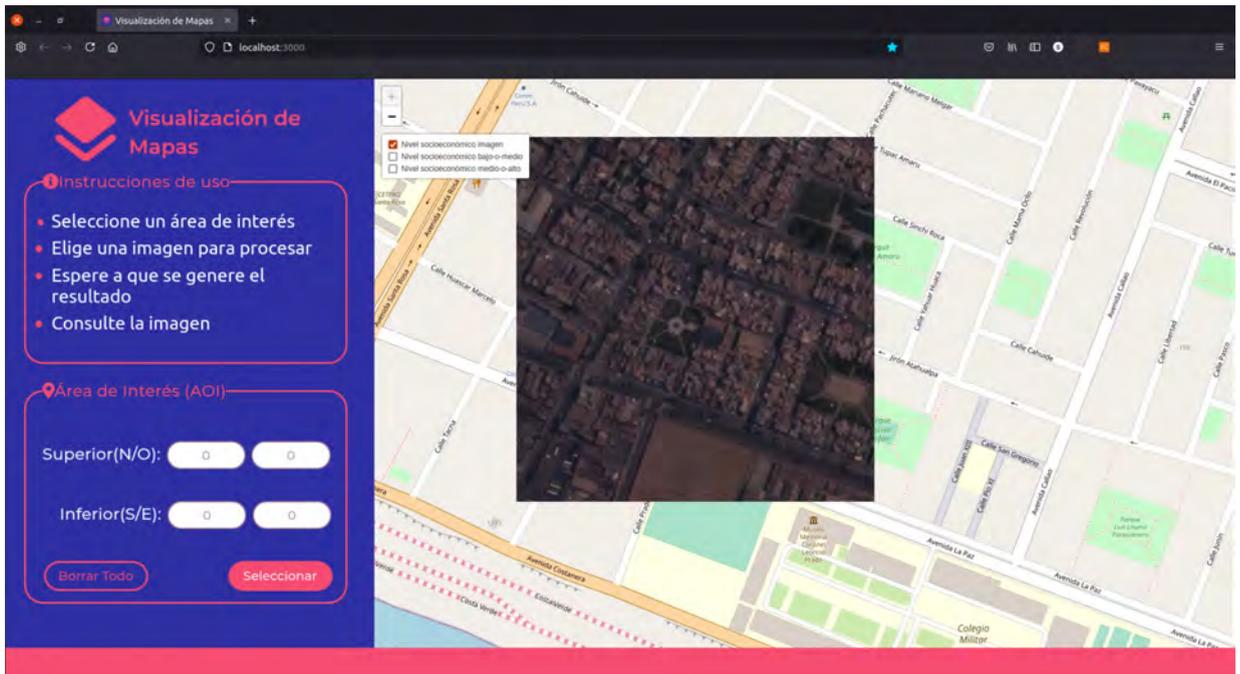


Figura 28. Ejecución de la plataforma web, ubicación de la imagen satelital en el mapa

Fuente: Elaboración propia

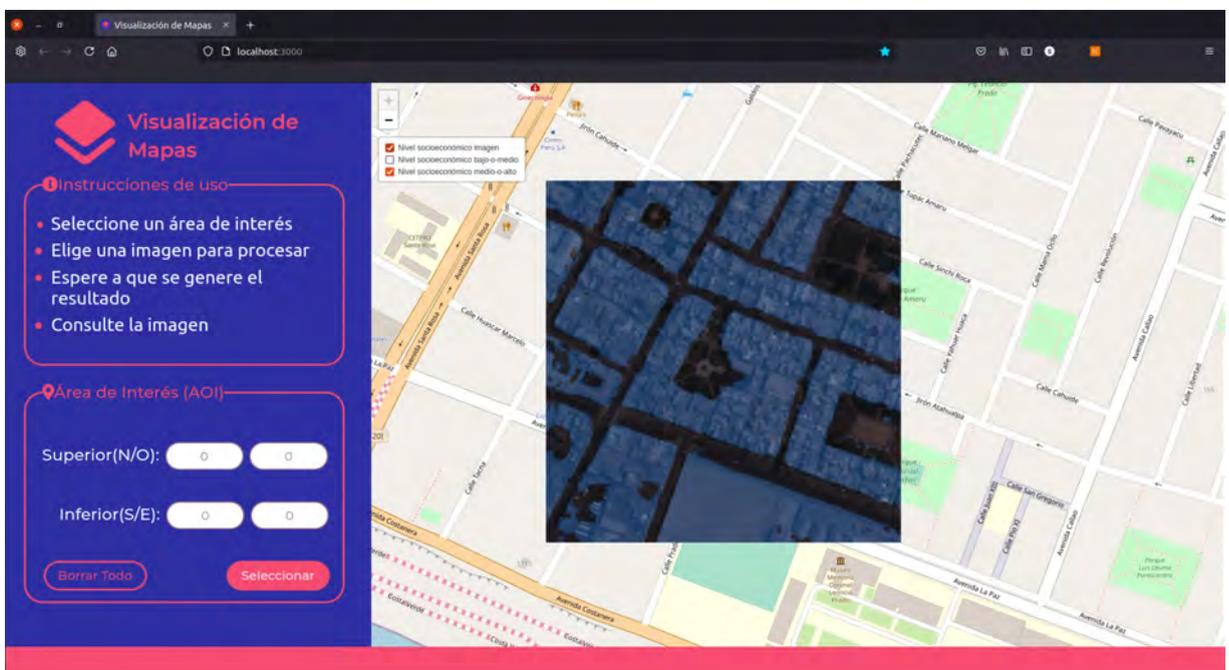


Figura 26. Ejecución de la plataforma web, ubicación de la imagen satelital y máscaras de predicción en el mapa

Fuente: Elaboración propia