Laboratorio No. 2: Diseño de implementación de reconocedor básico de paisajes

Joshua Gamboa Calvo, *Estudiante, ITCR*

***Resumen*—En el presente reporte técnico se ofrece el recopilatorio de experiencias a la hora de desarrollar el laboratorio No.2 del curso de Introducción al Reconocimiento de Patrones. En el cual se debe crear un reconocedor de patrones que implemente una etapa de pre-procesamiento, extracción de características y clasificación de imágenes.**

***Índice de Términos*—Patrones, reconocimiento, *opencv*, paisajes**

I. Introducción

Desde 1950 se ha experimentado con la visión por computadora, la cual es una rama de la inteligencia artificial (*IA*) la cual tiene como objetivo proveer la capacidad a las computadoras de interpretar y comprender el mundo visual, utilizando imágenes digitales y modelos de *deep learning* [1]. Este campo de la *IA* es crucial ya que abre un amplio abanico de posibilidades en la interacción humano-máquina y en la automatización de tareas anteriormente exclusivas de los humanos.

En este reporte se documenta el proceso de diseño, desarrollo y ejecución de un reconocedor de patrones básico el cual únicamente utiliza los valores *RGB* de una imagen. Este reconocedor de patrones fue diseñado para reconocer únicamente paisajes con una jerarquía particular. Además, se brinda a detalle el proceso de recolección de un modelo reconocedor y su uso a través de la herramienta documentada en el presente reporte. A través de este análisis se pretende cuantificar la eficacia de un modelo de visión por computadora la cual tome en cuenta únicamente tres características de las imágenes, las cuales deben basarse en color o iluminación.

Los apartados que componen este reporte se encuentran organización de la siguiente manera: *I Introducción* le brinda a lector contexto respecto al objetivo del proyecto y las herramientas a utilizar; en la segunda sección *II Desarrollo del reconocedor de paisajes* se describe a detalle la selección de características, módulos utilizados y modelo obtenido; en la tercera sección *III Resultados obtenidos* se exponen las observaciones obtenidas al utilizar el modelo generado; en la cuarta sección *IV Análisis de los resultados* se presentan diagramas y análisis estadístico de los resultados obtenidos; y por último, en la quinta sección *V Conclusiones* se brinda una síntesis de la información expuesta a lo largo del reporte. Al final del reporte se pueden encontrar las referencias a las fuentes consultadas, así como una breve biografía del autor del presente documento.

II. Desarrollo del reconocedor de paisajes

Para el desarrollo del reconocedor de paisajes se tomaron diferentes decisiones de diseño y suposiciones, las mismas se van a detallar a continuación:

- **Selección de características**: Para la selección de características se tenía la restricción de no poder utilizar bordicidad, por lo que aspectos como figuras, lineas divisorias entre otros estaban fuera del alcance. Por ende se decidió trabajar con las siguientes características:
  1) **Porcentaje de cielo** (SP): Esta característica representa el porcentaje de cielo en la parte superior de la imagen. Para esto se calculó una gradiente de **50** franjas en la imagen de arriba a abajo, con la cual se buscaba un delta entre franjas de más de 65 puntos en la escala *RGB*.
  2) **Verdocidad** (G): Este corresponde al porcentaje de píxeles verdes. Para efectos de este documento la verdocidad se encuentra entre el límite inferior $G_L = (25, 25, 20)$ y el límite superior $G_U = (230, 255, 86)$.
  3) **Cielocidad** (S): Este corresponde al porcentaje de píxeles celestes o color "cielo". Para efectos de este documento la cielocidad se encuentra entre el límite inferior $S_L = (170, 45, 100)$ y el límite superior $S_U = (255, 255, 255)$.

  Creando así el vector de características $\vec{F} = (SP, G, S)$ tal que $\vec{F} \in \mathbb{R}^3$.
- **Biblioteca de manipulación de imagenes utilizada**: Por recomendación del profesor y por investigación propia, la biblioteca escogida fue *OpenCV* para *Python*, ya que posee los métodos necesarios para completar el presente proyecto y está muy bien documentado.
- **Datos de entreno**: Para los datos de entreno se preseleccionaron 12 imágenes de las que los estudiantes compartieron vía *Discord*, de las cuales se tomaron 8 imágenes para entreno y el resto se utilizó para las pruebas con otro *dataset* que se explicará en el siguiente punto.
- **Datos de prueba**: Para los datos de prueba se tiene 4 imágenes seleccionadas por estudiantes del curso las cuales corresponden a paisajes urbano/rurales, se selecciono una imagen adicional de paisaje urbano/rural para un total de 5 imágenes de paisajes. Luego se buscaron otras 5 imágenes las cuales no fueran paisajes pero que sean un reto para el reconocedor de paisajes, las imágenes seleccionadas fueron: un mono, una tortuga en el agua, una foto de un vecindario, una sala y un gato. Teniendo así un *dataset* 50/50, 50% paisajes, 50% imágenes aleatorias.
- **Modelo de reconocimiento**: Para el modelo reconocedor de paisajes se va a utilizar el *dataset* de pruebas para extraer los vectores de características, posteriormente

se calcula el vector promedio ($E(\vec{F})$) y el vector de desviación estándar ($\sigma_{\vec{F}}$). Para que una imagen sea clasificada como paisaje, su vector de características debe estar entre el vector promedio más menos el vector de desviación estándar: $E(\vec{F}) + \sigma_{\vec{F}} \geq \vec{F} \geq E(\vec{F}) - \sigma_{\vec{F}}$.

A. Arquitectura del reconocedor

1) Fase 1: En la figura 1, se presenta la primera fase donde se muestran los grandes módulos que forman parte del reconocedor de paisajes. Los módulos que componen esta fase del diagrama son:

- **Imágenes de paisajes urbano/rural**: Este set de imágenes es utilizado para el entrenamiento del modelo, donde se presentan imágenes con una jerarquía específica, sea el cielo en la parte más alta de la imagen, un delta elevado en el horizonte, niveles relativamente altos de verde en la parte media-inferior y niveles moderados de celeste (cielo).
- **Pre-procesador de imágenes**: Este módulo se encarga de generar imágenes con información filtrada para poder extraer características más fácilmente, para el caso del reconocedor de paisajes genera 3 imágenes, una gradiente de color de arriba a abajo, una mascara del color verde y una mascara del color celeste o cielo.
- **Extractor de características**: El presente módulo se encarga de extraer las características de las imágenes generadas por el pre-procesador y colocarlos en un vector donde $\vec{F} \in \mathbb{R}^3$. Además calcula los vectores necesarios para el modelo de reconocimiento.
- **Clasificador de imágenes**: Este utiliza el modelo generado por el extractor y lo compara con los vectores de características de cada imagen para clasificarlas como paisaje o no paisaje.
- **Utilitarios**: En este módulo se tienen funciones utilitarias para todos los módulos para reutilizarlas, como el leer datos de un archivo, escribir en un archivo, entre otros.

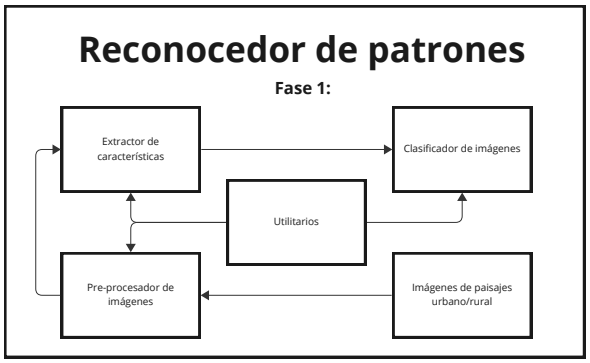

Fig. 1. Fase 1 del reconocedor básico de paisajes. Figura de elaboración propia.

2) Fase 2: En las figuras 2, 3, 4, se presentan el desglose de los bloques que componen los tres módulos principales del reconocedor de paisajes, un diagrama de flujo o explicación de cada uno de estos módulos puede ser encontrado en la fase 3.

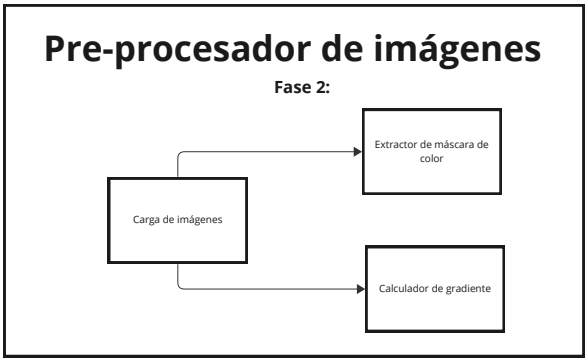

Fig. 2. Fase 2 del reconocedor básico de paisajes: Pre-procesador de imágenes. Figura de elaboración propia.

En la figura 2 se presenta el diagrama de bloques para el módulo de pre-procesamiento de imágenes, el cual se alimenta de las imágenes de paisajes y del módulo de utilitarios.

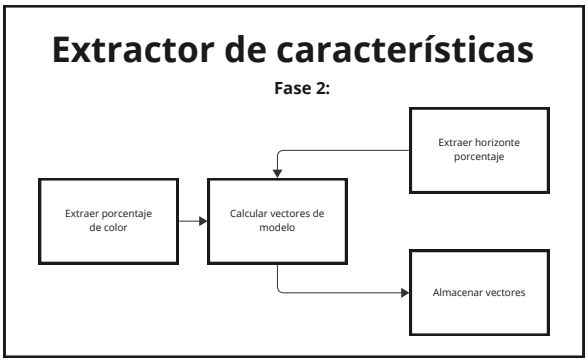

Fig. 3. Fase 2 del reconocedor básico de paisajes: Extractor de características. Figura de elaboración propia.

En la figura 3 se presenta el diagrama de bloques para el módulo de extracción de características, el cual se alimenta del módulo de pre-procesamiento de imágenes y del módulo de utilitarios.

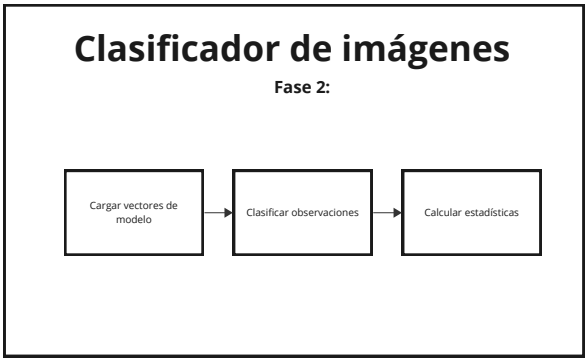

Fig. 4. Fase 2 del reconocedor básico de paisajes: Clasificador de imágenes. Figura de elaboración propia.

En la figura 4 se presenta el diagrama de bloques para el módulo de clasificación de imágenes, el cual se alimenta

del módulo de extracción de características y del módulo de utilitarios.

3) Fase 3: En la presente y última fase se va a profundizar en el funcionamiento de cada uno de los bloques en la tercera fase de la arquitectura del programa:

- **Pre-procesador de imágenes**:
  - **Carga de imágenes**: Esta función utiliza el método de *OpenCV imread* para cargar una imagen en memoria en formato la escala de colores *RGB*.
  - **Extractor de máscara de color**: Esta función toma una imagen, un limite inferior y un limite superior en escala *RGB* y los aísla de la imagen. Así se logra que únicamente los pixeles que se encuentren entre los límites sean observados.
  - **Calculador de gradiente**: El funcionamiento de la presente función se describe mediante un diagrama de flujo presente en la figura 5.

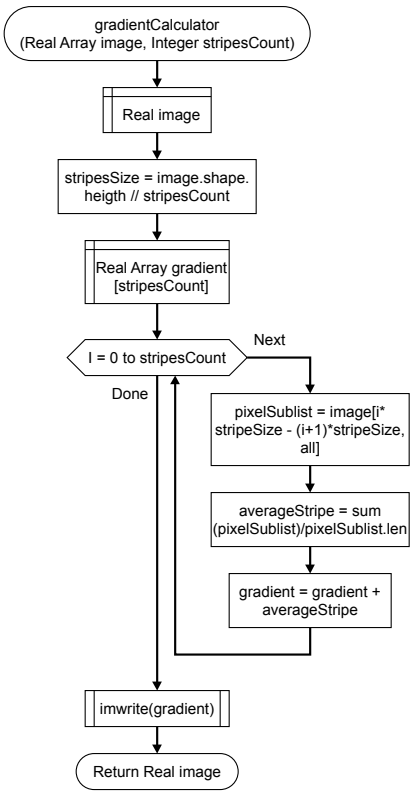

Fig. 5. Diagrama de flujo para la función "Calculador de gradiente". Figura de elaboración propia.

- **Extractor de características**:
  - **Extraer porcentaje de color**: Esta función calcula la cantidad de píxeles que no son cero o $(0, 0, 0)$ y los divide entre la cantidad de píxeles totales. Esto ya que tiene como entrada imágenes con las máscaras de color ya calculadas.
  - **Extraer porcentaje de horizonte**: El funcionamiento de la presente función se describe mediante un diagrama de flujo presente en la figura 6.

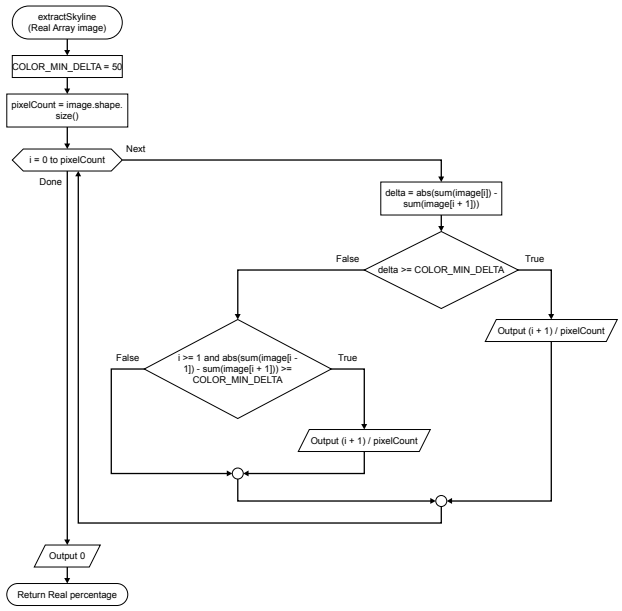

Fig. 6. Diagrama de flujo para la función "Extraer porcentaje de horizonte". Figura de elaboración propia.

  - **Calcular vectores de modelo**: En esta función se utilizan dos fórmulas para el cálculo del vector promedio y el vector de varianza las cuales son:

$$E(\vec{F}) = \frac{\sum_{n=0}^{obs} \vec{F_n}}{obs}$$

Para el calculo de el vector promedio de características, donde obs corresponde a la cantidad de observaciones del *dataset* y $\vec{F_n}$ es el vector de características para la observación n.

$$\sigma_{\vec{F}} = \sqrt{\frac{\sum_{n=0}^{obs} (\vec{F_n} - E(\vec{F}))^2}{obs}}$$

Y para el calculo de el vector de desviación estándar, donde obs corresponde a la cantidad de observaciones del *dataset* y $\vec{F_n}$ es el vector de características para la observación n.

  - **Almacenar vectores**: Esta función almacena en archivos de texto plano tanto los vectores de características individuales en un archivo llamado "Individuales.txt" y el cálculo del modelo de reconocimiento en un archivo llamado "Modelo_reconocedor_de_paisajes.txt".

- **Clasificador de imágenes**:
  - **Cargar vectores de modelo**: Esta función abre el archivo de texto "Modelo_reconocedor_de_paisajes.txt" y carga los dos vectores almacenados ($E(\vec{F})$ y $\sigma_{\vec{F}}$) para su uso en la clasificación.
  - **Clasificador de observaciones**: Esta función clasifica como paisaje a una imagen si su vector de características se encuentre entre el vector el límite inferior y el límite superior del modelo para todas

sus dimensiones. Con que solo una dimensión del vector de características ya no se clasificaría como paisaje.

– **Calcular estadísticas**: Para esta función se calcula la cantidad de verdaderos positivos (ambos el modelo y el objetivo son positivos), verdaderos negativos (ambos el modelo y el objetivo son negativos), falsos positivos (el modelo dio positivo pero el objetivo era negativo) y falsos negativos (el modelo dio negativo pero el objetivo era positivo).

B. Modelo calculado

Mediante el uso de los módulos anteriormente explicados a detalle se calculó el módulo a utilizar para el cálculo de la precisión del modelo para reconocer paisajes. El modelo a utilizar, específicamente el vector promedio y el vector de desviación estándar se presenta a continuación:

$$E(\vec{F}) = (0.32, 0.3315, 0.2601)$$

$$\sigma_{\vec{F}} = (0.1841, 0.0540, 0.1428)$$

Estos serán utilizados en la siguiente sección para la recolección y presentación de resultados.

III. Resultados obtenidos

En la presente sección se resumen los resultados encontrados en la ejecución de las pruebas para determinar si el modelo obtenido mediante el presente programa es suficientemente preciso a la hora de clasificar paisajes. Cabe aclarar que la presente y siguientes secciones muestran valores redondeados a 4 decimales para una mejor comprensión y legibilidad del documento.

CUADRO I
MATRIZ DE VECTORES DE CARACTERÍSTICAS UTILIZADOS PARA CALCULAR EL MODELO DE RECONOCIMIENTO.

Imagen	Vector de características
01.jpg	(0.04, 0.4194, 0.1480)
02.jpg	(0.4, 0.3040, 0.3784)
03.jpg	(0.44, 0.3485, 0.3784)
04.jpg	(0.18, 0.2273, 0.1387)
05.jpg	(0.32, 0.2980, 0.0317)
06.jpg	(0.42, 0.3774, 0.2777)
07.jpg	(0.64, 0.3251, 0.3762)
08.jpg	(0.12, 0.3517, 0.2354)

En el cuadro I se muestran los ocho vectores de características utilizados para el cálculo del modelo de reconocimiento, estos datos se encuentran en el archivo "Individuales.txt".

Posterior al entrenamiento del modelo, se procedieron con las pruebas las cuales consistieron en aplicar el pre-procesamiento del *dataset*, la extracción de características y la clasificación de imágenes. Las imágenes resultantes tanto para el pre-procesamiento de las imágenes de entrenamiento como de pruebas se encuentran en las carpetas $data > test$ y $data > training$. Los vectores de características de las imágenes de prueba se pueden observar el cuadro II junto con el resultado de la clasificación del modelo y la clasificación objetivo. En la columna "Veredicto modelo" se muestra si el modelo reconoció la imagen como un paisaje o no y en la columna "Objetivo" se muestra si la imagen representa un paisaje o no.

CUADRO II
MATRIZ DE RESULTADOS OBTENIDOS UTILIZANDO EL MODELO PREVIAMENTE CALCULADO PARA EL RECONOCIMIENTO DE IMÁGENES.

Imagen	Vector de características	Veredicto Modelo	Objetivo
09.jpg	(0.34, 0.3431, 0.1419)	True	True
10.jpg	(0.4, 0.6390, 0.2551)	False	True
11.jpg	(0.32, 0.5555, 0.0)	False	True
12.jpg	(0.22, 0.3751, 0.2716)	True	True
13.jpg	(0.42, 0.3449, 0.2739)	True	True
cat.jpg	(0, 0.3490, 0.0056)	False	False
livingroom.jpg	(0.2, 0.1987, 0.1078)	False	False
monkey.jpg	(0.12, 0.4922, 0.0)	False	False
street.jpg	(0.2, 0.3073, 0.1211)	True	False
tortouise.jpg	(0.3, 0.3598, 0.0055)	False	False

En el cuadro II se muestra la matriz de estadísticas calculadas, en el cual se presenta la frecuencia absoluta y relativa porcentual para cada tipo de resultado. Estos datos son relevante para el cálculo del nivel de precisión del modelo.

CUADRO III
MATRIZ DE CLASIFICACIÓN DE RESULTADOS.

Tipo de resultado	Frec. absoluta	Frec. relativa porcentual
Verdadero positivo	3	30%
Verdadero negativo	4	40%
Falso positivo	1	10%
Falso negativo	2	20%

IV. Análisis de los resultados

En la presente sección, se va a proceder a analizar los resultados expuestos en la anterior sección mediante el uso de gráficos y el cálculo de datos estadísticos con el fin de visualizar la información de mejor manera.

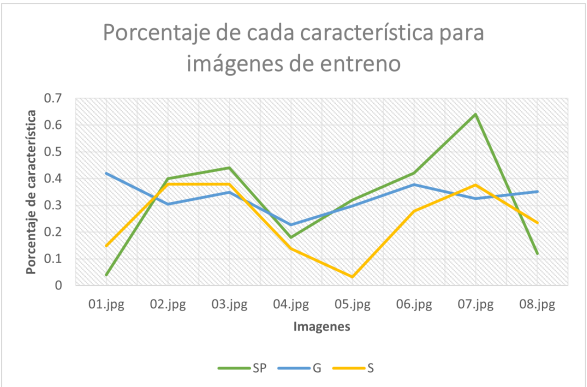

Fig. 7. Gráfico de líneas sobre las características de las imágenes utilizas para calcular el modelo de reconocimiento. Figura de elaboración propia.

En la figura 7 se presenta un gráfico de líneas con base al cuadro I, el cual incluye los valores del vector de características para las imágenes utilizadas para el cálculo del modelo de reconocimiento de paisajes. En base a este gráfico

podemos observar que la característica de verdocidad muestra una variación menor y se mantiene más constante en comparación con las demás. Las características de porcentaje de cielo y cielocidad presentan mínimos y máximo más drásticos en comparación.

En la figura 8 se presenta otro gráfico de líneas con base al cuadro II, el cual muestra los valores de los vectores de características para la imágenes a utilizar en pruebas. En base al gráfico se puede obsevar que las imágenes *10.jpg, 11.jpg, cat.jpg, monkey.jpg, tortoise.jpg* alcanzan mínimos y máximos drásticos en almenos una dimensión de su vector de características. Y contrastando esta informacón con las columnas de "Veredicto modelo" y "Objetivo" se puede asociar el alcanzar estos mínimos y máximos drásticos con dar un resultado falso.

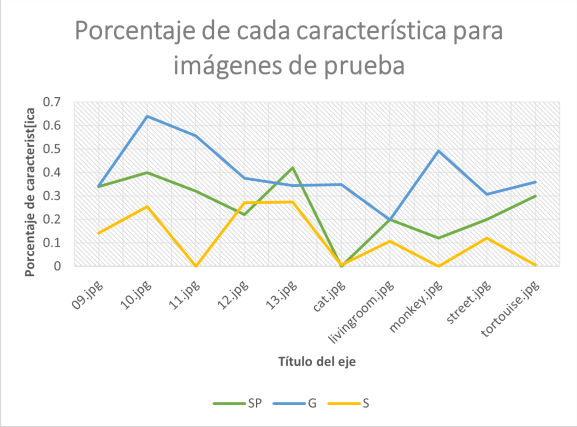

Fig. 8. Gráfico de líneas sobre las características de las imágenes utilizas para las pruebas. Figura de elaboración propia.

Por último, con respecto al cuadro III se puede observar que el modelo tiene una precisión del **70%**. También se puede observar que el modelo tiende a clasificar una imagen como no paisaje la mayoría del tiempo (60%), y que el modelo es mejor para identificar imágenes que no son paisajes que imágenes que si son paisajes (40% contra 30% respectivamente).

V. Conclusiones

A continuación se expone el corolario resultante de analizar los resultados obtenidos y la implementación lograda:

- La etapa de pre-procesamiento es la más importante debido a que la selección de las características y procesamiento correcto de los datos propicia una mejor extracción de características, lo cual representa la base del modelo. Una selección errónea o al azar puede ocasionar un modelo torpe a la hora de clasificar imágenes.
- Con respecto al modelo generado, se concluye que este tiene una precisión aproximadamente del 70% al clasificar imágenes como paisajes o no paisajes.
- El modelo generado es mejor al identificar imágenes que no son paísajes con respecto a imágenes que sí son paisajes. Por lo que se debería recalcular el modelo para que pueda enfocarse en identificar paisajes más que identificar no paisajes.
- El no tomar en cuenta la ubicación de los niveles de verdocidad puede haber provocado falsos positivos ya que en la mayoría de imágenes esta característica se encontraba en los límites establecidos por el modelo, pero no en el lugar correcto en la imagen.

Referencias

[1] SAS, "Computer vision: Qué es y por qué es importante," www.sas.com, 2023. [Online]. Available: https://www.sas.com/es_ar/insights/analytics/computer-vision.html

Joshua Gamboa Calvo Estudiante de la carrera de Ingeniería en Computación en el Instituto Tecnológico de Costa Rica. Carné 2020037148. Correo electrónico joshgc.19@estudiantec.cr.