

Tarea N° #1: Aprendiendo a Ver

Cristopher Acevedo Guajardo

ELE4001 – Robótica

Escuela de Ingeniería, Universidad de O'Higgins

30, abril, 2024

Abstract— En este informe sobre la tarea 1 del curso se comparan y analizan distintos filtros en imágenes extraídas del simulador Gym-Duckietown, como filtros pasa-bajos y pasa-altos, encontrando que la diferencia principal en la detección de formas, bordes y detalles está en la frecuencia y en la intensidad de los píxeles que detecta cada tipo de filtro. Utilizando también las imágenes de Gym-Duckietown, se diseña e implementa un sistema de detección de duckies y control de movimiento para el vehículo autónomo Duckiebot, generando bounding boxes alrededor de los duckies, una alarma visual y un frenado de emergencia para el Duckiebot.

Keywords— Procesamiento de imágenes, visión computacional, Gym-Duckietown, filtros pasa bajos, filtros pasa altos, espacios de color, operaciones morfológicas.

I. INTRODUCCIÓN

El procesamiento de imágenes es un área multidisciplinaria que se enfoca en la manipulación y análisis de imágenes para extraer información útil para la identificación de objetos mediante la segmentación de objetos y la mejora de la calidad visual utilizando herramientas para reducir el ruido de la imagen. Tiene un papel fundamental en una amplia variedad de aplicaciones, incluyendo la fotografía, la medicina, la agricultura, la seguridad, entre otras.

El objetivo de esta tarea es procesar imágenes utilizando distintos filtros pasa bajos y pasa altos para comparar y analizar el funcionamiento de los distintos tipos de filtros según la forma en la que interpretan objetos y formas en una imagen, y utilizar filtros y operaciones morfológicas para diseñar e implementar un sistema que detecte objetos (los duckies) en el entorno Gym-Duckietown y evitar que los duckies sean atropellados.

El entorno Gym-Duckietown es un simulador de ambientes para vehículos automatizados de Duckietown, desarrollado en el Massachusetts Institute of Technology (MIT) orientado en la

educación, investigación y desarrollo en robótica autónoma.

II. MARCO TEÓRICO

En el ámbito del procesamiento de imágenes, un filtro o máscara es una matriz numérica que varía en tamaño, según el tipo, para realizar operaciones de convolución. Esta operación implica deslizar la máscara sobre la imagen y calcular una operación matemática en cada conjunto de píxeles cercanos, usualmente conocidos como vecindarios. La máscara se utiliza para resaltar ciertas características de la imagen o suavizarla al eliminar detalles que no sean deseados.

Cuando aplicamos un filtro o máscara a una imagen, cada píxel de la imagen se multiplica por los valores correspondientes en la máscara, esto es, por los valores de la matriz numérica, y se suma el resultado para obtener el nuevo valor del píxel en la imagen filtrada. Este proceso se repite para cada píxel en la imagen, lo que produce una imagen filtrada que puede tener características diferentes según el filtro o máscara utilizada.

A. Filtro Pasa-bajos

Un filtro pasa bajos es un tipo de filtro que se utiliza para atenuar o eliminar las frecuencias altas de una señal, conservando principalmente las frecuencias bajas.

El filtro promedia los valores de intensidad de los píxeles en un vecindario alrededor de cada uno de los píxeles de la imagen, permitiendo que las componentes de baja frecuencia pasen a través del filtro mientras atenúa las frecuencias más altas o los cambios lentos en la intensidad.

En esta tarea se utilizarán tres filtros pasa bajos distintos:

1) *Filtro pasa bajos recto: se utiliza una máscara (o kernel) que consiste en una matriz de valores idénticos, todos iguales a 1.*

2) *Filtro pasa bajos unidimensional aplicado por filas y columnas: se utilizan máscaras o kernels unidimensionales, esto es, que consisten de una matriz de una fila y tres columnas, y se aplica el suavizado a lo largo de las filas y a lo largo de las columnas por separado.*

3) *Filtro Gaussiano pasa bajos dimensional: se utiliza una máscara o kernel bidimensional que se deriva de la función gaussiana, para luego calcular los nuevos valores de los píxeles de la imagen filtrada. Al utilizar una función gaussiana, se utiliza la distribución de probabilidad normal con un foco central y valores que decaen suavemente hacia los píxeles de los lados.*

B. Filtro Pasa-altos

Un filtro pasa altos es un tipo de filtro utilizado para resaltar detalles finos y bordes en una imagen, atenuando o eliminando las componentes de baja frecuencia, y conservando principalmente las frecuencias altas.

Similar al filtro pasa bajos, un filtro pasa altos permite que las frecuencias altas o los cambios rápidos en la intensidad de los píxeles de la imagen pasen a través del filtro, mientras que atenúan o eliminan las frecuencias bajas o los cambios lentos en la intensidad.

En esta tarea se utilizarán cuatro filtros pasa altos distintos:

1) *Filtro pasa altos Prewitt horizontal: se utiliza una máscara o kernel con tres columnas, la primera con valores de -1, la segunda con valores de 0 y la tercera con valores de 1. Se utiliza para resaltar los bordes verticales de la imagen, es decir, cambios abruptos o rápidos de intensidad en la dirección vertical.*

2) *Filtro pasa altos Prewitt vertical: En este filtro, la primera fila tiene valores de -1, la segunda fila valores de 0 y la tercera fila valores de 1. Tiene el mismo funcionamiento que el filtro pasa altos Prewitt horizontal, pero al cambiar de posición los valores de la máscara o kernel, se resaltan los bordes verticales de la imagen, es decir, cambios abruptos o rápidos de intensidad en la dirección vertical.*

3) *Filtro laplaciano: se utiliza para resaltar los bordes y detalles finos en la imagen, produciendo una imagen filtrada donde los bordes y las regiones de alta frecuencia se vuelven más prominentes.*

4) *Filtro laplaciano gaussiano: es una combinación del filtro laplaciano y el filtro pasa altos gaussiano. Se utiliza para*

resaltar los bordes y detalles finos, mientras que a la misma vez se suaviza la imagen (mediante la función gaussiana) para reducir el ruido.

Para el procesamiento de imágenes y la aplicación de filtros, es importante y también influye en el resultado el espacio de color que estemos utilizando.

Un espacio de color es un modelo matemático que describe y organiza los colores de manera abstracta. Se representa mediante un espacio multidimensional, donde cada punto en este espacio corresponde a un color único y se define por un conjunto de coordenadas numéricas. Esto permite representar los colores mediante valores numéricos, facilitando el almacenamiento, transmisión y procesamiento computacional de la imagen, así como también la comparación de colores de forma precisa y objetiva, realizar acciones como ajustar el brillo o el contraste, y representar los colores de manera más cercana a como los percibe el ojo humano.

Los espacios de color que utilizaremos serán los espacios RGB, HSV y CIELAB.

El espacio de color RGB es un modelo aditivo que representa los colores mediante la combinación de luz roja, verde y azul. Se representan mediante tres coordenadas (red, green, blue), donde cada valor varía entre 0 y 255, siendo 0 la ausencia de dicho color y 255 la intensidad máxima del color.

El espacio de color HSV es un modelo perceptual que representa los colores en términos de matiz (H), saturación (S) y valor (V). El matiz define el todo del color, la saturación indica la intensidad o pureza del color y el valor representa la luminosidad o brillo del color. El espacio HSV describe de forma más intuitiva y perceptual la percepción de los colores del ojo humano.

El espacio de color CIELAB es un espacio perceptualmente uniforme, diseñado también para ser más consistente con la percepción humana del color. Representa los colores mediante tres componentes: L (luminosidad), a (componente rojo-verde) y b (componente amarillo-azul). Se utiliza más comúnmente en aplicaciones donde se requiere una representación precisa y uniforme del color, por ejemplo, como en la impresión e industria textil.

III. METODOLOGÍA

Esta sección describe en detalle la metodología, el proceso y la implementación que se utilizó para comparar y analizar cada tipo de filtro pasa bajos y pasa altos utilizando las imágenes de Gym-Duckietown provistas por el equipo docente y también utilizando imágenes propias capturadas del simulador que fueran lo suficientemente representativas para la evaluación de los filtros, con las mismas imágenes, se describe en detalle el diseño y la implementación de un sistema para la detección de duckies en las imágenes junto con un freno de emergencia y una alarma visual.

Para el diseño e implementación en código de los filtros y del sistema de detección, se utilizó el lenguaje de programación Python junto con la librería cv2 de OpenCV.

A. Función para aplicar la convolución (filtrar la imagen)

Se implementó una función (Anexo 1) que recibe una imagen en escala de grises junto a una máscara y que calcula la convolución entre ambas, para generar una imagen de salida que será la imagen filtrada, utilizando funciones propias de la librería de OpenCV.

B. Filtros Pasa-bajos

Los tres filtros otorgados por el equipo docente para la realización de esta tarea se implementaron en código (Anexo 2) para aplicar la máscara o kernel a la imagen.

Para el filtro pasa bajos recto, la máscara que se utilizó estaba definida por la siguiente matriz:

FIGURA I
MÁSCARA PARA FILTRO PASA BAJOS RECTO

$$\frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Para el filtro pasa bajos unidimensional aplicado por filas y columnas, la máscara que se utilizó fue la siguiente:

FIGURA II
MÁSCARA PARA FILTRO PASA BAJOS UNIDIMENSIONAL APLICADO POR FILAS Y COLUMNAS

$$\frac{1}{3} \cdot \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

Y el filtro pasa bajos Gaussiano bidimensional utilizado fue de tamaño 5x5, con $\sigma = 1$.

C. Filtros Pasa-altos

Se utilizaron e implementaron en código cuatro filtros pasa altos (Anexo 3), también otorgados por el equipo docente para la realización de esta tarea.

Para el filtro pasa altos Prewitt vertical, se utilizó la siguiente máscara:

FIGURA III
MÁSCARA PARA FILTRO PASA ALTOS PREWITT VERTICAL

$$\frac{1}{9} \cdot \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Para el filtro pasa altos Prewitt horizontal, se utilizó una máscara similar, pero utilizando la matriz transpuesta del filtro vertical:

FIGURA IV
MÁSCARA PARA FILTRO PASA ALTOS PREWITT HORIZONTAL

$$\frac{1}{9} \cdot \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Para el filtro laplaciano, que para esta tarea se proporcionó un filtro laplaciano de 3x3, se utilizó la siguiente máscara:

FIGURA V
MÁSCARA PARA FILTRO PASA LAPLACIANO DE 3X3

$$\frac{1}{9} \cdot \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Y para el filtro laplaciano gaussiano, se utilizó un filtro laplaciano de gaussiana de 5x5, también con $\sigma = 1$.

Cada uno de los filtros se aplicó a una de las imágenes provistas por el equipo docente y a una imagen propia capturada desde el simulador Gym-Duckietown.

Para la comparación y el análisis de los filtros pasa bajos, tanto de las imágenes provistas por el equipo docente como para la imagen propia, se implementó en código el cálculo de la FFT (Transformada rápida de Fourier), y se graficó el logaritmo de la amplitud del espectro, en este caso, en escala de grises.

En cuanto a la comparación y análisis de los filtros pasa altos, se realizó un análisis visual de las imágenes filtradas, utilizando la interpretación

propia de la función y diferencias entre los filtros según la apreciación visual propia de la imagen.

D. Espacios de color

Para probar y analizar los espacios de color, se implementó en código una función que toma como valor de entrada una imagen en color, y genera como salida una imagen los tres espacios de colores: RGB, HSV y CIELAB.

La función se probó utilizando las imágenes provistas por el equipo docente y con imágenes propias capturadas desde el simulador Gym-Duckietown.

E. Sistema de detección de duckies

Para el sistema de detección de los duckies en el entorno del simulador Gym-Duckietown, se implementó una función que toma como valor de entrada una imagen o un frame del simulador y genera como salida una imagen con la detección de los duckies mediante bounding boxes, un freno de emergencia que no permite al Duckiebot seguir avanzando en la dirección del duckie si este está muy cerca y una alarma visual en el caso de que el freno de emergencia se haya activado y que el duckie esté muy cerca.

Lo único que recibe la función es la imagen o el frame actual del Duckiebot, y se realizaron las recomendaciones entregadas por el equipo docente para diseñar e implementar la función.

Posterior a que se recibe la imagen, se filtran los colores de la imagen utilizando un espacio de color arbitrario para binarizar la imagen.

Luego, sobre la imagen binarizada, se aplican dos operaciones morfológicas. Primero, se utiliza la operación morfológica de erosión para eliminar pequeños detalles y reducir el tamaño de los objetos, eliminando el ruido y separando los objetos que están muy cerca entre sí. A la imagen erosionada, se aplica la operación morfológica de dilatación, que expande los bordes de los objetos para unir regiones de primer plano que están cerca entre sí y rellenar pequeños huecos dentro de los objetos (como, por ejemplo, la parte entre el cuerpo y la cabeza del duckie, que se separan primeramente por la diferencia de color de la boca del duckie).

Con estas operaciones realizadas, se generan bounding boxes (cuadrados delimitadores) alrededor de los objetos detectados, que, en este caso, son los duckies.

Finalmente, sobre los duckies detectados, se aplica un freno de emergencia en caso de que los duckies estén muy cerca. Para esto se estableció una diferencia de 120 píxeles entre el borde inferior del bounding box del duckie y el borde inferior de la imagen, y además, que la relación de aspecto sea menor a 0.8 o mayor a 8, así, se solucionó el problema de que se detectaran figuras que no son los duckies, como por ejemplo, algunas líneas amarillas del medio de la calle del simulador cuando estas son muy notorias y coinciden con el color de los duckies. En caso de que la diferencia sea de 120 o menor, se aplica el freno de emergencia, evitando que el Duckiebot avance, y se genera una alerta visual en la parte superior del bounding box con la frase "Freno de emergencia". El freno se debe implementar en el código original del entorno y simulador Gym-Duckietown, utilizando las mismas variables que no permiten avanzar presentes en la sección (d) de la función.

TABLA IV
FUNCIÓN PARA LA DETECCIÓN DE LOS DUCKIES

```
def detector_de_duckies(imagen):
    # (a) Filtrado de colores
    imagen_original = cv2.imread(imagen)
    hsv = cv2.cvtColor(imagen_original,
                       cv2.COLOR_BGR2HSV)
    print("Imagen/frame original:")
    cv2_imshow(imagen_original)

    # Rango de colores de los duckies
    rango_inf = np.array([20, 100, 100])
    rango_sup = np.array([30, 255, 255])
    imagen_filtrada = cv2.inRange(hsv, rango_inf,
                                  rango_sup)
    print("(a) Filtrado de colores:")
    cv2_imshow(imagen_filtrada)

    # (b) Operaciones morfológicas
    kernel = np.ones((5, 5), np.uint8)
    erosion = cv2.erode(imagen_filtrada, kernel,
                        iterations=1)
    print("(b) Operaciones morfológicas:")
    print(" Erosión")
    cv2_imshow(erosion)
    dilatacion = cv2.dilate(erosion, kernel,
                           iterations=1)
    print("Dilatación")
    cv2_imshow(dilatacion)

    # (c) Bounding boxes
    imagen_op_morf = imagen_original.copy()
    contours, _ = cv2.findContours(dilatacion,
                                   cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```



```
for contour in contours:
    x, y, w, h = cv2.boundingRect(contour)
    relacion_aspecto = h / w
    if relacion_aspecto < 0.8 or
    relacion_aspecto > 8: # Para solucionar el
    problema de que se generen bounding boxes para las
    imágenes donde las líneas amarillas de la calle
    son muy notorias
        continue
    cv2.rectangle(imagen_original, (x, y), (x +
    w, y + h), (0, 0, 255), 2)
    print("(c) Bounding boxes:")
    print("Bounding boxes utilizando erosión")
    cv2.imshow(imagen_original)

# (d) Freno de emergencia, bloqueo de
movimiento: Esta parte del código hay que
habilitarla en el código original de Gym-
Duckietown
    #if y + h >= imagen_original.shape[0] - 120:
    #if key_handler[key.UP]:
    #action = np.array([0.0, 0.0])

# (e) Alarma visual
    if y + h >= imagen_original.shape[0] - 120: #
    120 pixeles de diferencia entre el borde inferior
    de la imagen y el del bounding box
        cv2.putText(imagen_original, "Freno de
        emergencia", (x, y - 10),
        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)

    print("Imagen final, con freno de emergencia y
    alarma visual si el duckie está muy cerca:")

    return cv2.imshow(imagen_original)
```

IV. RESULTADOS

A. Filtro Pasa-bajos recto, imagen "calle4.png"

FIGURA VI
IMAGEN ORIGINAL "CALLE4.PNG"

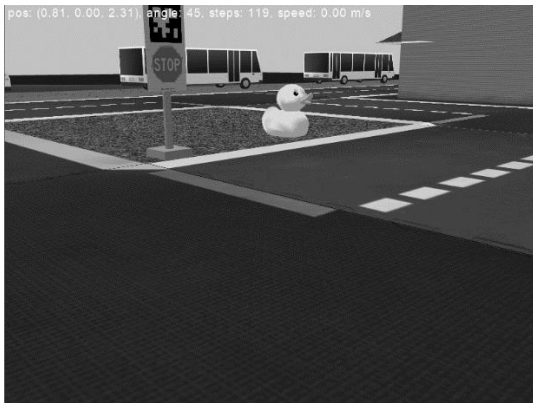


FIGURA VII
GRÁFICA DEL LOGARITMO DE LA AMPLITUD DEL ESPECTRO DE LA FFT DE LA IMAGEN ORIGINAL

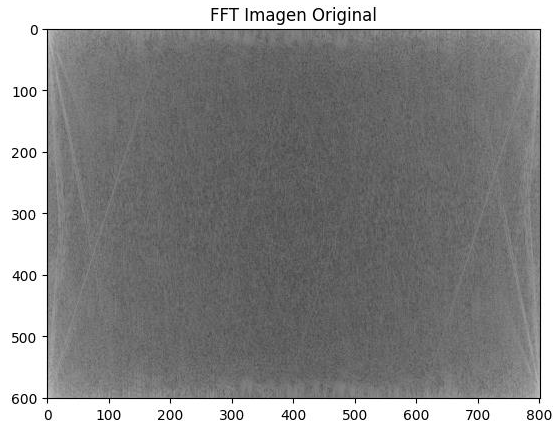
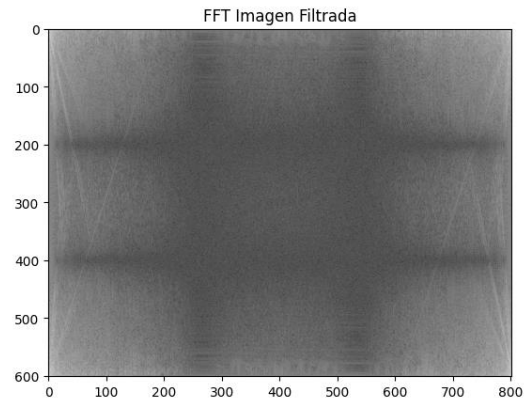
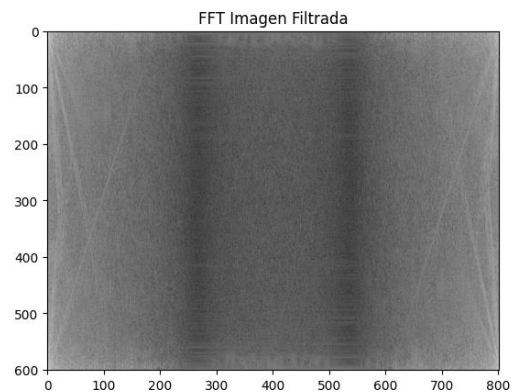


FIGURA VIII
GRÁFICA DEL LOGARITMO DE LA AMPLITUD DEL ESPECTRO DE LA FFT DE LA IMAGEN FILTRADA



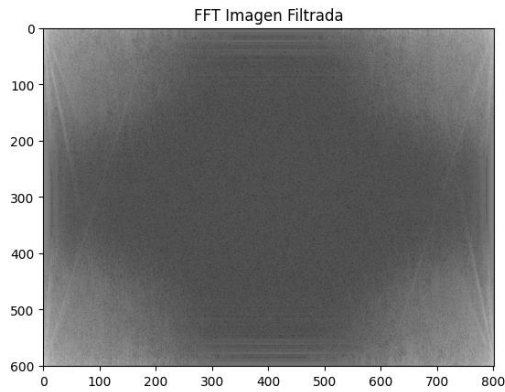
B. Filtro Pasa-bajos unidimensional, imagen "calle4.png"

FIGURA IV
GRÁFICA DEL LOGARITMO DE LA AMPLITUD DEL ESPECTRO DE LA FFT DE LA IMAGEN FILTRADA



C. Filtro Pasa-bajos Gaussiano, imagen "calle4.png"

FIGURA X
GRÁFICA DEL LOGARITMO DE LA AMPLITUD DEL ESPECTRO DE LA FFT DE LA IMAGEN FILTRADA



D. Filtro Pasa-bajos recto, imagen "propia1.png"

FIGURA XI
IMAGEN ORIGINAL "PROPIA1.PNG"

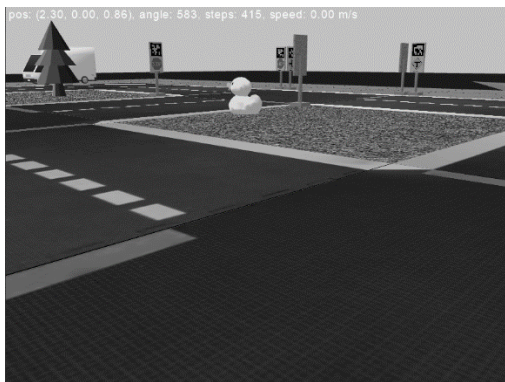


FIGURA XII
GRÁFICA DEL LOGARITMO DE LA AMPLITUD DEL ESPECTRO DE LA FFT DE LA IMAGEN ORIGINAL

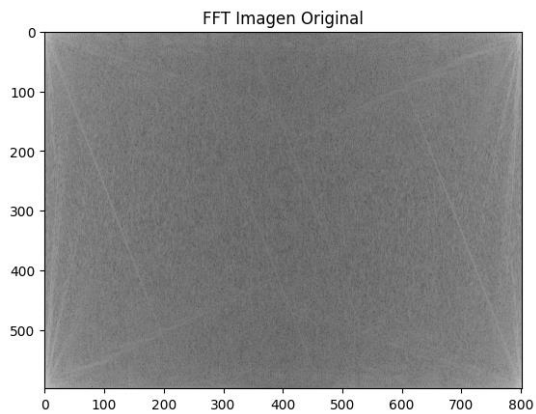
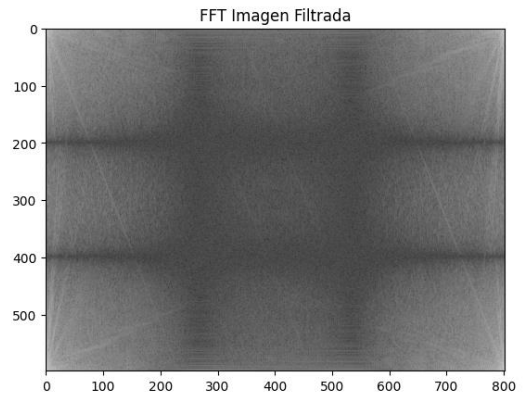
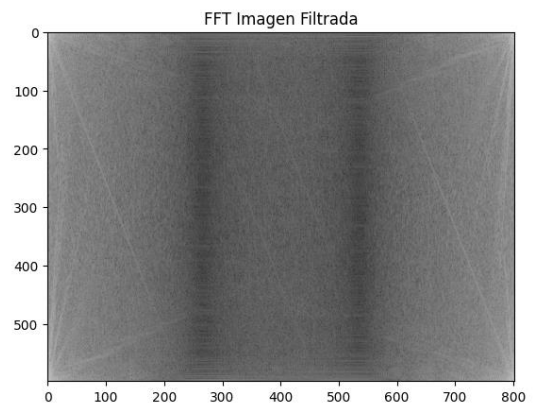


FIGURA XIII
GRÁFICA DEL LOGARITMO DE LA AMPLITUD DEL ESPECTRO DE LA FFT DE LA IMAGEN FILTRADA



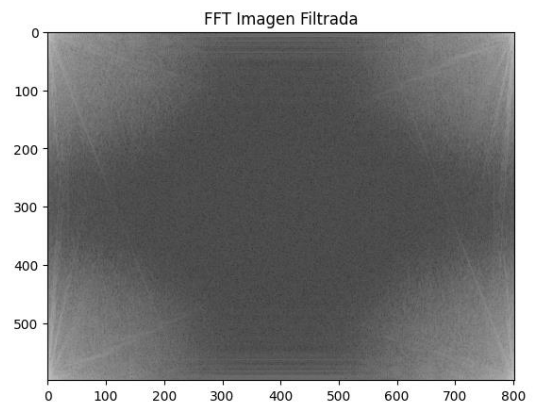
E. Filtro Pasa-bajos unidimensional, imagen "propia1.png"

FIGURA XIV
GRÁFICA DEL LOGARITMO DE LA AMPLITUD DEL ESPECTRO DE LA FFT DE LA IMAGEN FILTRADA



F. Filtro Pasa-bajos Gaussiano, imagen "propia1.png"

FIGURA XV
GRÁFICA DEL LOGARITMO DE LA AMPLITUD DEL ESPECTRO DE LA FFT DE LA IMAGEN FILTRADA



G. Filtro Pasa-altos Prewitt vertical, imagen "calle3.png"

FIGURA XVI
IMAGEN ORIGINAL "CALLE3.PNG"

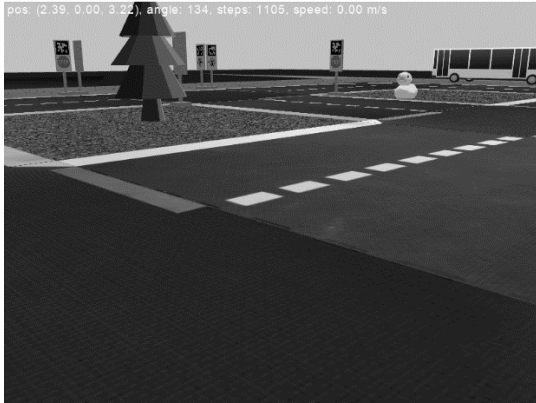
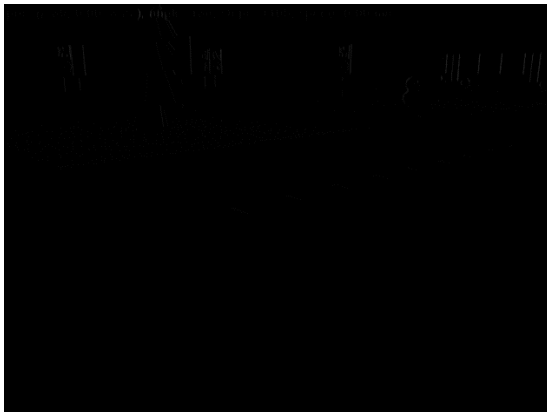


FIGURA XVII
IMAGEN FILTRADA "CALLE3.PNG"



H. Filtro Pasa-altos Prewitt horizontal, imagen "calle3.png"

FIGURA XVIII
IMAGEN FILTRADA "CALLE3.PNG"



I. Filtro Pasa-altos Laplaciano, imagen "calle3.png"

FIGURA XIX
IMAGEN FILTRADA "CALLE3.PNG"



J. Filtro Pasa-altos Laplaciano Gaussiano, imagen "calle3.png"

FIGURA XX
IMAGEN FILTRADA "CALLE3.PNG"



K. Filtro Pasa-altos Prewitt vertical, imagen "propia3.png"

FIGURA XXI
IMAGEN ORIGINAL "PROPIA3.PNG"

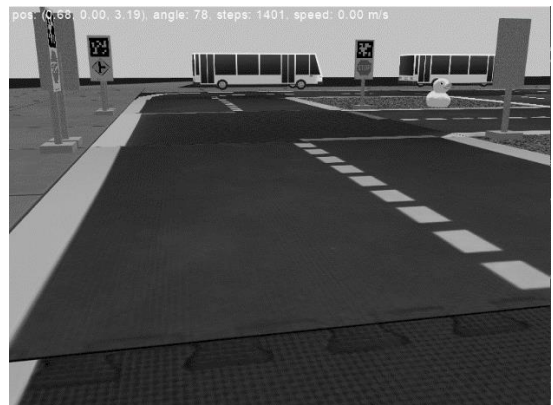
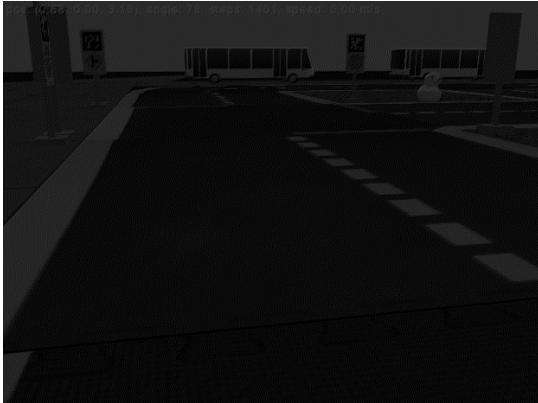
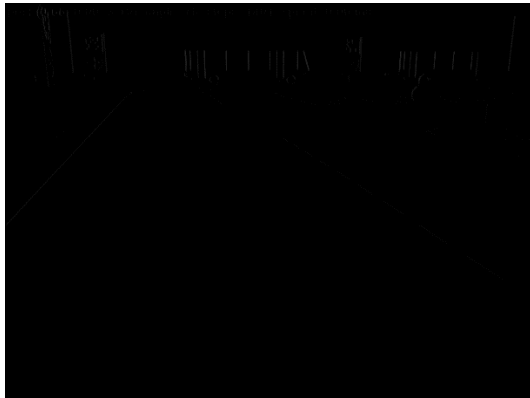


FIGURA XXII
IMAGEN FILTRADA "PROPIA3.PNG"



L. Filtro Pasa-altos Prewitt horizontal, imagen "propia3.png"

FIGURA XXIII
IMAGEN FILTRADA "PROPIA3.PNG"



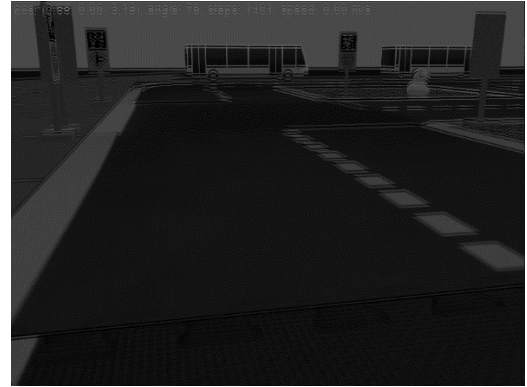
M. Filtro Pasa-altos Laplaciano, imagen "propia3.png"

FIGURA XXIV
IMAGEN FILTRADA "PROPIA3.PNG"



N. Filtro Pasa-altos Laplaciano Gaussiano, imagen "propia3.png"

FIGURA XXV
IMAGEN FILTRADA "PROPIA3.PNG"



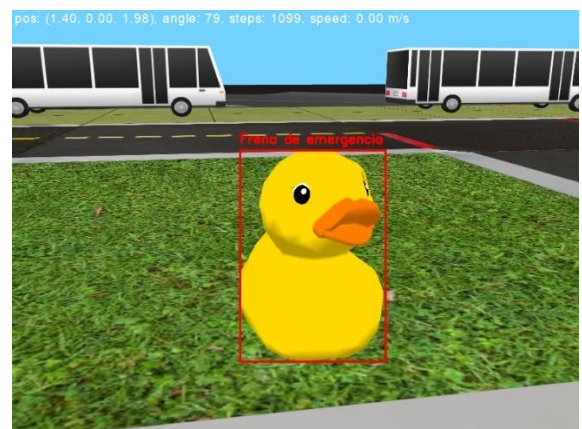
O. Detección de los duckies, bounding box y freno de emergencia

Se aplicó un filtro a las imágenes originales (Anexo 4, 5, 6 y 7), luego las operaciones morfológicas (Anexo 7, 8, 9 y 10), y se implementó el freno de emergencia y la alarma visual, todo en la misma función, cuyo resultado es el siguiente:

FIGURA XXVI
IMAGEN "CALLE3.PNG" CON DETECCIÓN DE DUCKIES



FIGURA XXVII
IMAGEN "PROPIA5.PNG" CON DETECCIÓN DE DUCKIES



V. ANÁLISIS DE LOS RESULTADOS

A. Filtros pasa bajos

Al aplicar un filtro pasa bajos recto en las imágenes, se pudo notar que las imágenes tienen una componente de alta frecuencia o intensidad en cuatro partes o direcciones principales: diagonal superior derecha, diagonal superior izquierda, diagonal inferior derecha y diagonal inferior izquierda, o, dicho de otro modo, en cada una de las esquinas. El filtro atenúa estas frecuencias en menor medida que en otras direcciones, lo que permite conservar cierta definición de bordes diagonales.

En cuanto a la aplicación de un filtro pasa bajos unidimensional, se pudo notar que las imágenes tienen una componente de alta frecuencia o intensidad sólo en la dirección vertical.

Con el filtro pasa bajos gaussiano se pudo notar un efecto diferente a los filtros anteriores, teniendo componentes de alta frecuencia atenuados en todas las direcciones. Se pudo observar una sombra amplia en la gráfica de la FFT, “dividiendo” el plano en cuatro cuadrantes, lo cual puede estar relacionado con la forma circular del “núcleo” gaussiano, que atenúa las frecuencias o intensidades altas en gran medida desde el centro y reduciendo gradualmente hacia los bordes de la imagen.

B. Filtros pasa altos

Para el caso de los filtros pasa altos, se analizó solamente la diferencia visual entre las imágenes originales y las imágenes filtradas.

Al aplicar un filtro pasa altos Prewitt vertical, se pudo notar una alta atenuación de las frecuencias o intensidades bajas, manteniendo los detalles de aquellos objetos o formas cuya intensidad o frecuencia es alta. Es decir, las intensidades bajas se atenúan y se destacan las intensidades altas.

En el caso del filtro pasa altos Prewitt horizontal, se pudo observar que las frecuencias o intensidades bajas se atenúan completamente, pero sucediendo lo mismo con las frecuencias o intensidades altas. Solo algunos detalles de objetos o formas que tenían muy alta frecuencia o intensidad son visibles, como algunos bordes del duckie o de vehículos.

Al aplicar el pasa altos filtro laplaciano, se pudo notar una alta atenuación de las frecuencias o intensidades bajas, pero, a diferencia del filtro Prewitt vertical, ahora las intensidades o frecuencias altas son mucho más visibles o destacadas, pudiendo identificar de mucha mejor forma los bordes de formas u objetos.

En el caso del filtro pasa altos laplaciano gaussiano, se pudo notar una menor atenuación de las frecuencias o intensidades bajas en comparación con el filtro laplaciano, pero manteniendo la característica de destacar en gran medida los bordes de formas u objetos de alta frecuencia o intensidad.

C. Detección de duckies

En el diseño e implementación del detector de duckies, la utilización de operaciones morfológicas tuvo un gran impacto en la detección de la forma y bordes de los duckies. Se observó que, en conjunto con aplicar un filtro para aumentar o destacar la intensidad o frecuencia del color del duckie, la utilización de la operación morfológica de erosión funcionó como una buena forma para resaltar la intensidad de los duckies y reducir las intensidades o frecuencias de los demás objetos. Luego, la aplicación de la operación morfológica de dilatación hizo que sea posible “expandir” aquellos espacios que en el proceso de erosión se separan, como, por ejemplo, en la boca del duckie, y se pudo notar que de esta forma se reduce el problema de que se identifique el cuerpo y la cabeza del duckie como objetos por separado.

Esto hizo posible que los duckies estén contenidos en sólo un bounding box, y no se generen dos bounding boxes para el mismo objeto.

VI. CONCLUSIONES GENERALES

Se pudo observar que la utilización de diferentes tipos de filtros pasa bajos y pasa altos afectan la apariencia visual de las imágenes y la forma en que se atenúan o resaltan diferentes frecuencias o intensidades. Mientras que los filtros pasa bajos tienen la misma finalidad, que es reducir o atenuar las frecuencias altas, la forma en que atenúan las intensidades o frecuencias altas varía dependiendo de la máscara o kernel (es decir, la matriz) que se utilice, pudiendo reducir las intensidades o

frecuencias altas de forma horizontal, vertical, o en varias direcciones.

Lo mismo sucedió con los filtros pasa altos. La finalidad o el objetivo es la reducción de las frecuencias o intensidades bajas, y mantener aquellas que sean altas, pero la forma en que atenúan las frecuencias bajas varía también dependiendo de la máscara que se utilice, pudiendo atenuar notoriamente las intensidades bajas y manteniendo las altas, o atenuando demasiado las frecuencias o intensidades bajas y casi de la misma forma las altas, dificultando la identificación de bordes u objetos en la imagen, o bien sin atenuar demasiado las frecuencias bajas pero identificando bien los bordes de aquellas intensidades o frecuencias altas.

En cuanto a las operaciones morfológicas, como la erosión y la dilatación, son herramientas poderosas para procesar imágenes y mejorar la detección de objetos. Con la combinación de estas dos operaciones morfológicas se pudo lograr una correcta identificación de los duckies, complementándose para ayudar a unificar las regiones de interés en un solo bounding box.

La combinación de la aplicación de filtros y operaciones morfológicas ayudó a la creación e implementación de un sistema efectivo de detección de duckies, reduciendo la generación de múltiples bounding boxes para el mismo objeto o para objetos con un mismo color que el del duckie, e identificando solamente a los duckies en la imagen, independiente de la distancia a la que se encuentren.

Por otro lado, una aplicación correcta del bounding box contribuyó a poder generar correctamente el freno de emergencia utilizando el borde inferior del bounding box, y en consecuencia, la correcta implementación de la alerta visual.

VII. ANEXOS

ANEXO 1

FUNCIÓN PARA APLICAR LA CONVOLUCIÓN

```
def convolucion(imagen, mascara):
    # Pre-requisito: Escala de grises
    if len(imagen.shape) != 2:
        raise ValueError("La imagen no está en escala de grises.")

    # Convolución (desde la librería de OpenCV)
    imagen_filtrada = cv2.filter2D(imagen, -1, mascara)

    return imagen_filtrada
```

ANEXO 2

IMPLEMENTACIÓN DE LOS FILTROS PASA BAJOS

```
# Filtro pasa bajos recto
filtro_pb_recto = 1/9 * (np.ones((3, 3),
dtype=np.float32))

# Filtro pasa bajos unidimensional aplicado por
filas y por columnas
filtro_pb_unid = 1/3 * (np.ones((1,3),
dtype=np.float32))

# Filtro Gaussiano pasa bajos bidimensional
filtro_gaussiano_pb = cv2.getGaussianKernel(5, 1)
@ cv2.getGaussianKernel(5, 1).T
```

ANEXO 3

IMPLEMENTACIÓN DE LOS FILTROS PASA ALTOS

```
# Filtro Prewitt vertical
filtro_pa_prewitt_v = (1/9) * np.array([
    [-1, 0, 1],
    [0, 0, 0],
    [1, 0, 1]])

# Filtro Prewitt horizontal
filtro_pa_prewitt_h = (1/9) * np.array([
    [-1, 0, 1],
    [0, 0, 0],
    [-1, 0, 1]])

# Filtro laplaciano 3x3
filtro_pa_laplaciano = (1/9) * np.array([
    [1, 1, 1],
    [1, -8, 1],
    [1, 1, 1]])

# Filtro Laplaciano de Gaussiana 5x5 (sigma = 1)
filtro_pa_laplaciano_gaussiano =
cv2.Laplacian(cv2.getGaussianKernel(5, 1,
cv2.CV_32F), cv2.CV_32F)
```

ANEXO 4

DETECCIÓN DE DUCKIES, "CALLE3..PNG", IMAGEN ORIGINAL



ANEXO 5

DETECCIÓN DE DUCKIES, "CALLE3..PNG", IMAGEN FILTRADA



ANEXO 6

DETECCIÓN DE DUCKIES, "PROPIA5.PNG", IMAGEN ORIGINAL



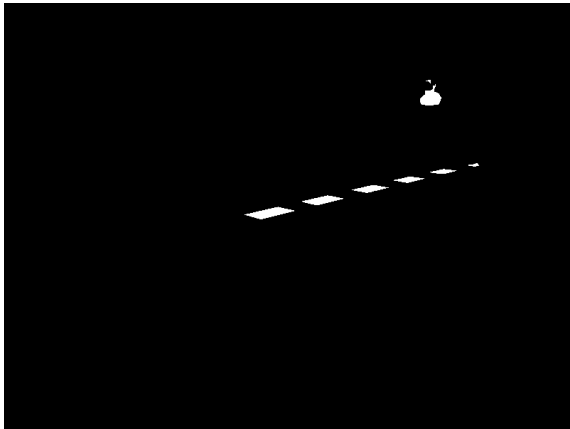
ANEXO 7

DETECCIÓN DE DUCKIES, "CALLE3..PNG", IMAGEN FILTRADA



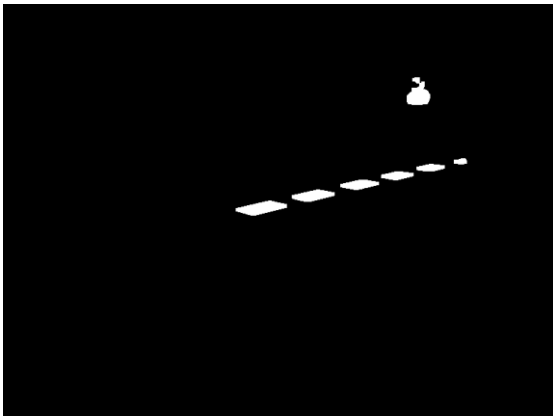
ANEXO 8

DETECCIÓN DE DUCKIES, "CALLE3..PNG", OPERACIÓN MORFOLÓGICA:
EROSIÓN



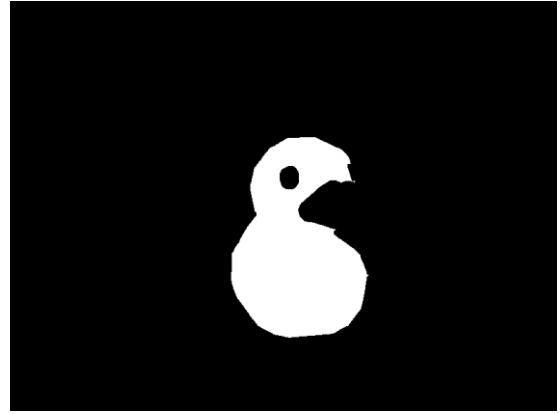
ANEXO 9

DETECCIÓN DE DUCKIES, "CALLE3..PNG", OPERACIÓN MORFOLÓGICA:
DILATACIÓN



ANEXO 9

DETECCIÓN DE DUCKIES, "PROPIA5.PNG", OPERACIÓN MORFOLÓGICA:
EROSIÓN



ANEXO 10

DETECCIÓN DE DUCKIES, "PROPIA5..PNG", OPERACIÓN MORFOLÓGICA:
DILATACIÓN

