Detección de Objetos con Redes Neuronales Profundas para un Robot de Servicio

Edgar Roberto Silva Guzmán

Tutor: Dr. Jesús Savage Carmona
Posgrado en Ingeniería Eléctrica, área de Procesamiento Digital de Señales, UNAM

16 de enero de 2020

Agenda

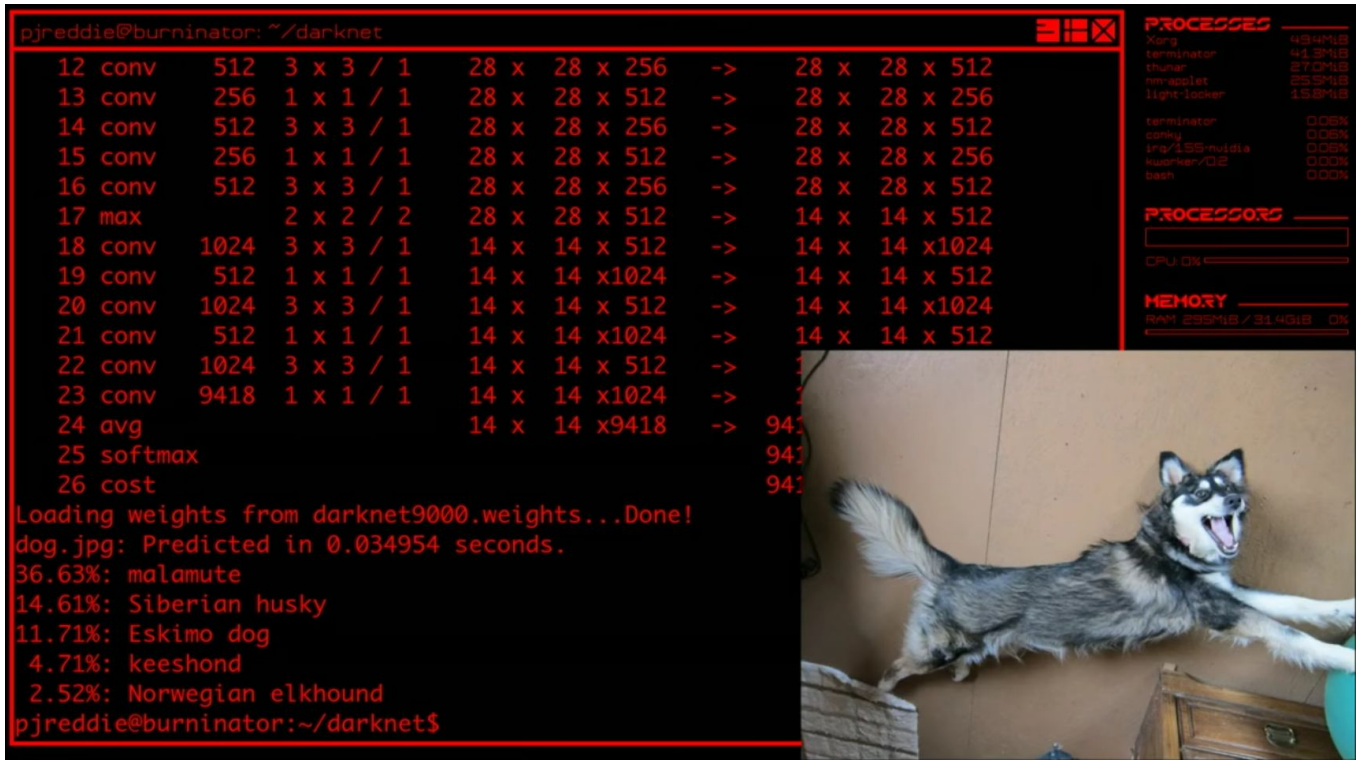
1. Introducción
2. Objetivos
3. Redes Neuronales Convolucionales
4. Desarrollo
5. Pruebas y resultados
6. Conclusiones y trabajo futuro

Introducción

Hace 15 años...

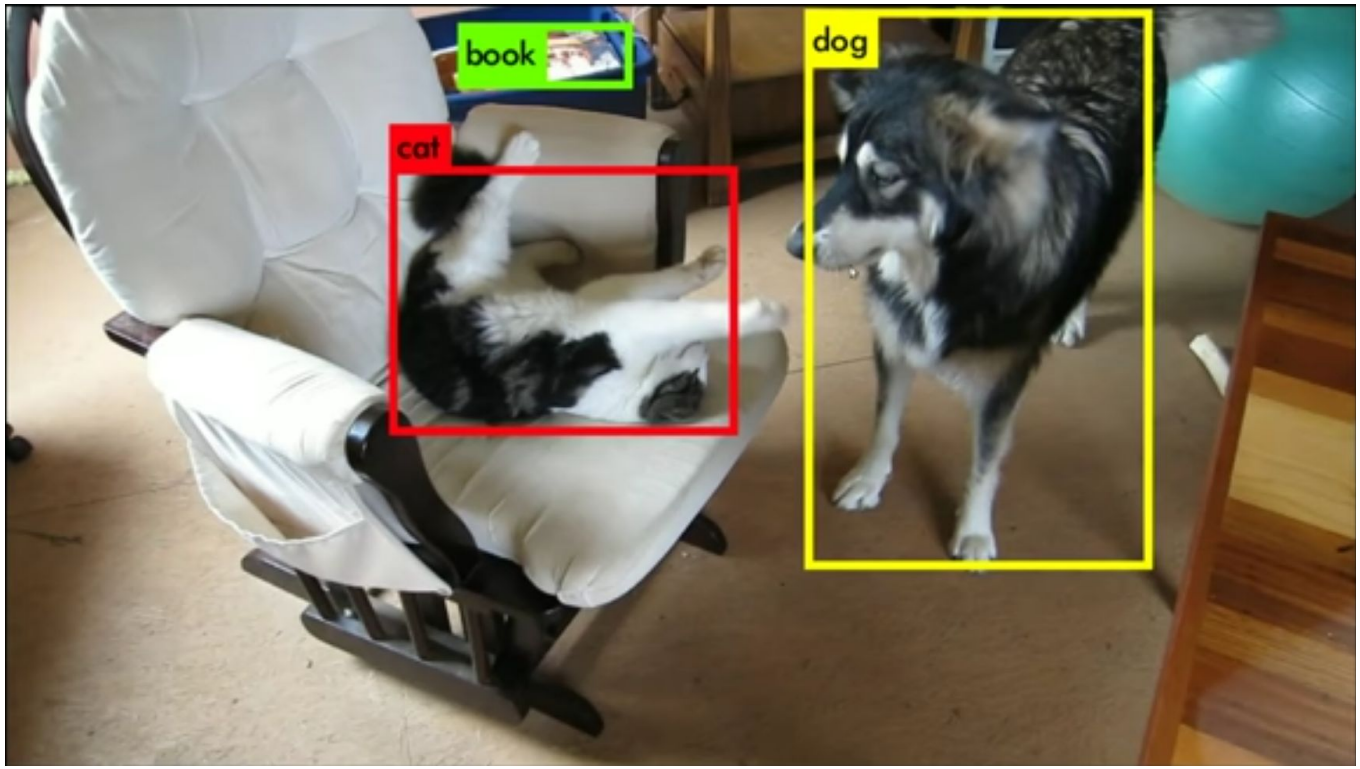
[1] Joseph Redmon. YOLOv3

Sistemas de clasificación.



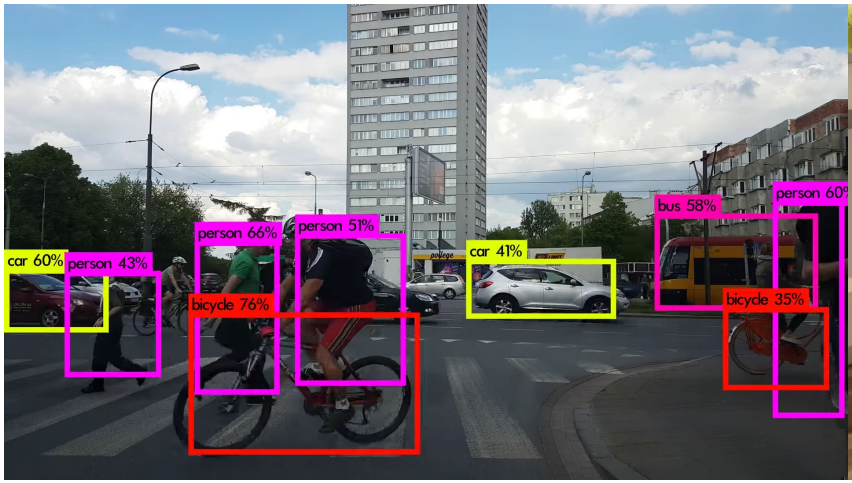
[1] Joseph Redmon. YOLOv3

Sistemas de detección.



[1] Joseph Redmon. YOLOv3

¿Para qué se utiliza un sistema de detección de objetos?



(a) Carros autónomos

(b) Robots de Servicio

¿Cómo evaluar los sistemas desarrollados en los robots de servicio?

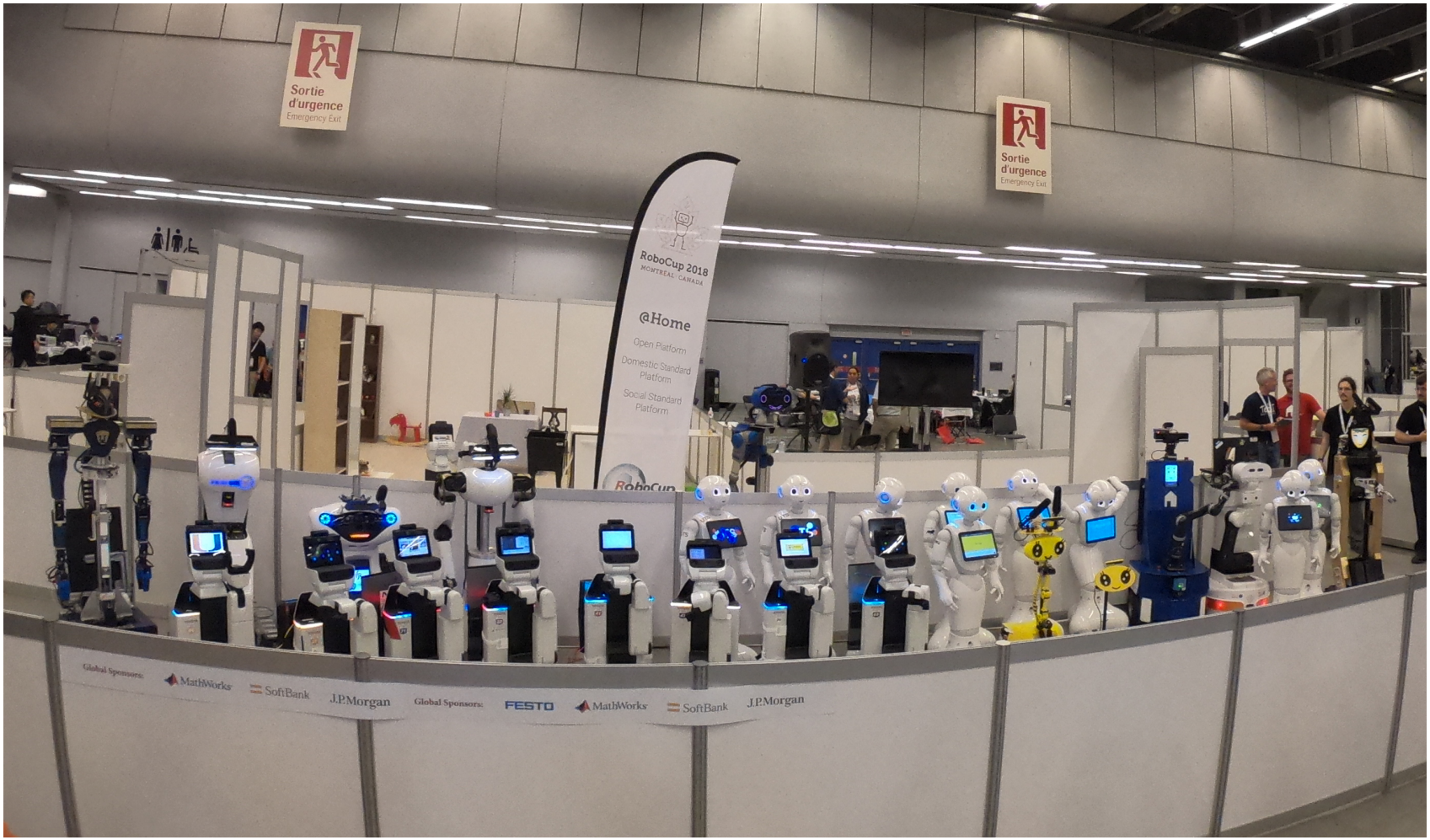


Figura: Robots de servicio. RoboCup@home 2018 Montreal, Canadá

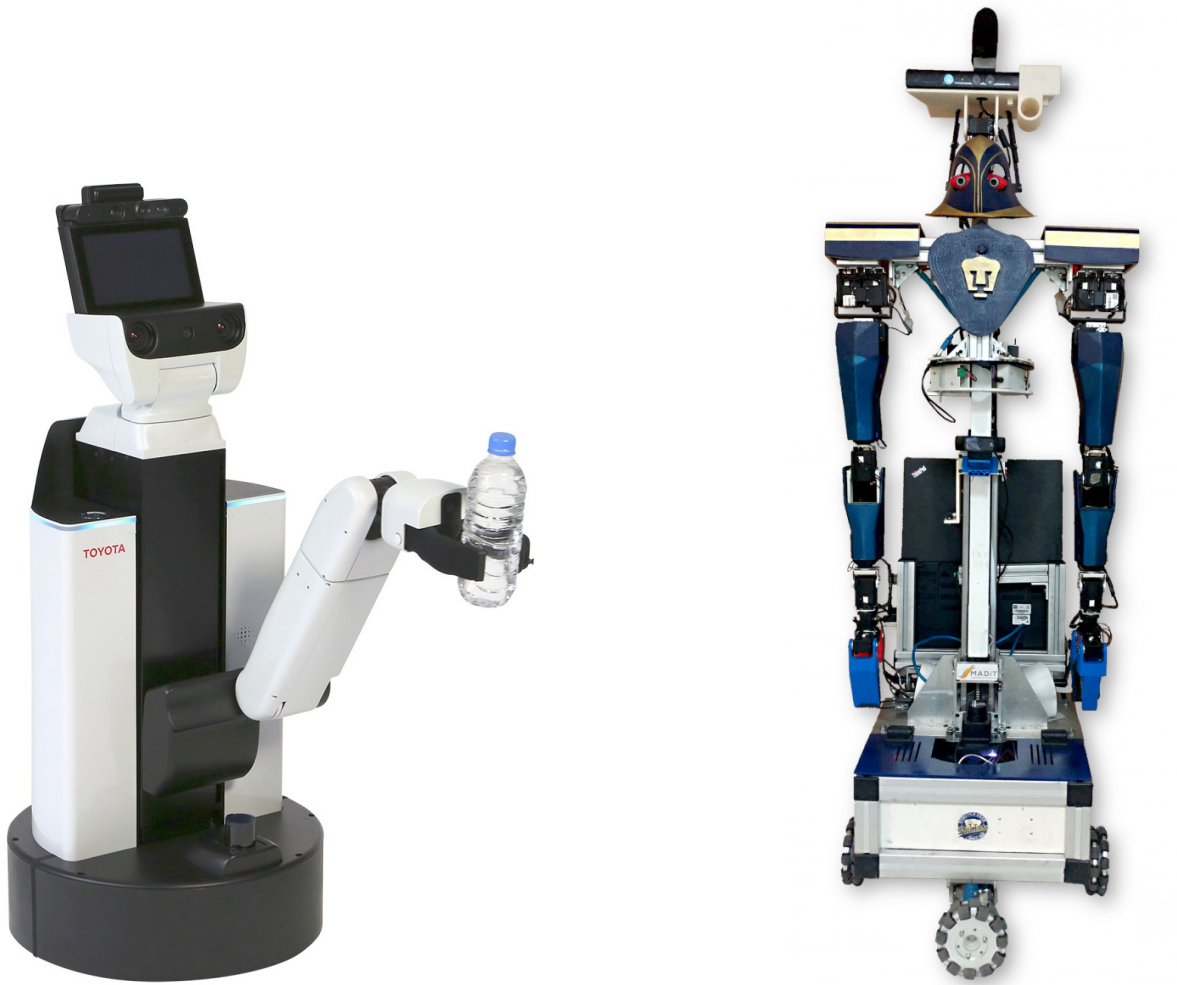
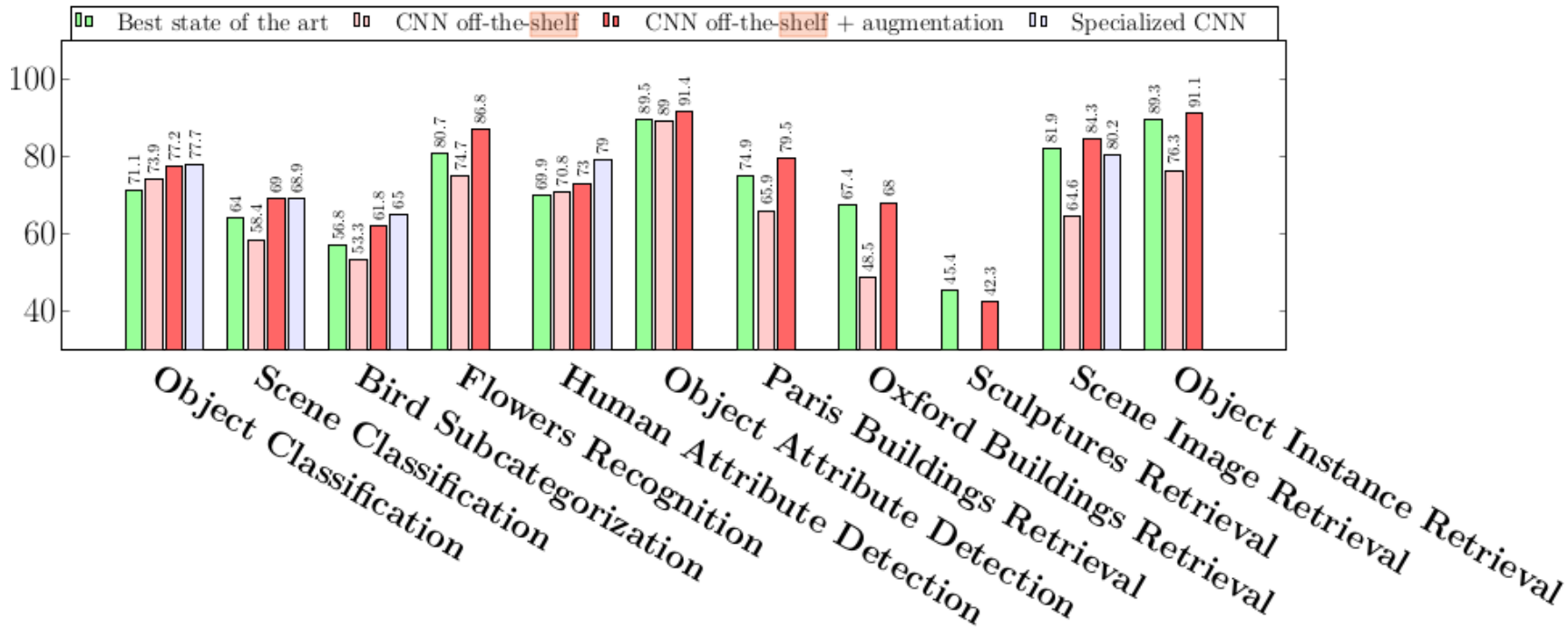


Figura: Robots de Servicio del laboratorio de Bio-Robótica

Justificación

2014. Ali Razavian. CNN Features off-the-shelf.

- Redes neuronales convolucionales frente a los métodos estado del arte en tareas de reconocimiento.



Justificación

Objetos de interés.

Problemática

Las Redes Neuronales Convolucionales (RNC) son el estado del arte en tareas de reconocimiento de objetos, pero se tienen los siguientes inconvenientes:

- Requieren de grandes conjuntos de datos de imágenes etiquetados.
  - COCO: 330,000 imágenes en 80 clases. Aproximadamente 200,000 imágenes etiquetadas.
  - ImageNet: 14,000,000 imágenes. Aproximadamente 1,000,000 imágenes etiquetadas.
- Se ven limitadas en el entrenamiento de objetos propios, debido a que las imágenes deben ser etiquetadas por un software especializado.
- En la mayoría de las RNC, el entrenamiento es lento y costoso computacionalmente.
- En las RNC, la etapa de reconocimiento o la etapa de detección se requiere de mucho poder de cómputo.

Objetivo Principal

El objetivo principal para el presente proyecto de tesis es el siguiente:

- Crear un conjunto de escenas sintéticas con objetos de interés etiquetados para re-entrenar una Red Neuronal Convolucional. El modelo neuronal obtenido será utilizado como sistema de detección de objetos en un robot de servicio.

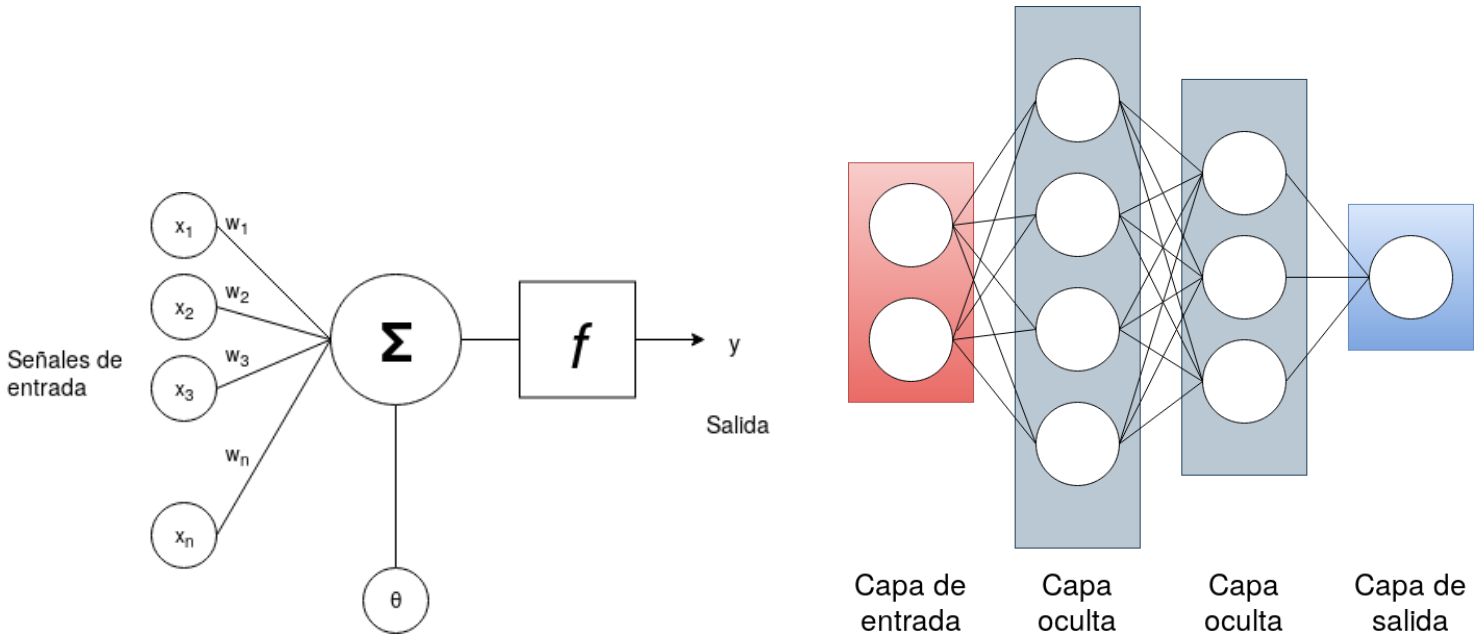
Objetivos Específicos

Por lo cual, se plantean los siguientes objetivos específicos:

- Desarrollar un sistema de visión computacional que permita segmentar objetos utilizando archivos de video adquiridos por una cámara digital compacta.
- Desarrollar un sistema de visión computacional que cree un conjunto de imágenes sintéticas, es decir, escenas artificiales que contengan los objetos segmentados del sistema anterior.
- Realizar el re-entrenamiento del modelo neuronal YOLOv3 con el conjunto de imágenes sintéticas generado.
- Desarrollar un sistema que adapte el modelo neuronal re-entrenado como un sistema de detección de objetos en un robot de servicio.

Hipótesis

Hipótesis:
"El entrenamiento de un modelo neuronal con imágenes sintéticas generadas de manera automática, tendrá altos índices de precisión en la detección de objetos en una escena real. La implementación de este modelo mejorará el sistema de detección de objetos en un robot de servicio, aumentando los índices de confidencia en la etapa de reconocimiento y mejorando la detección de objetos, para que puedan ser manipulados en una escena tridimensional."



(a) Modelo de una Neurona Artificial (b) Redes Neuronales Artificiales

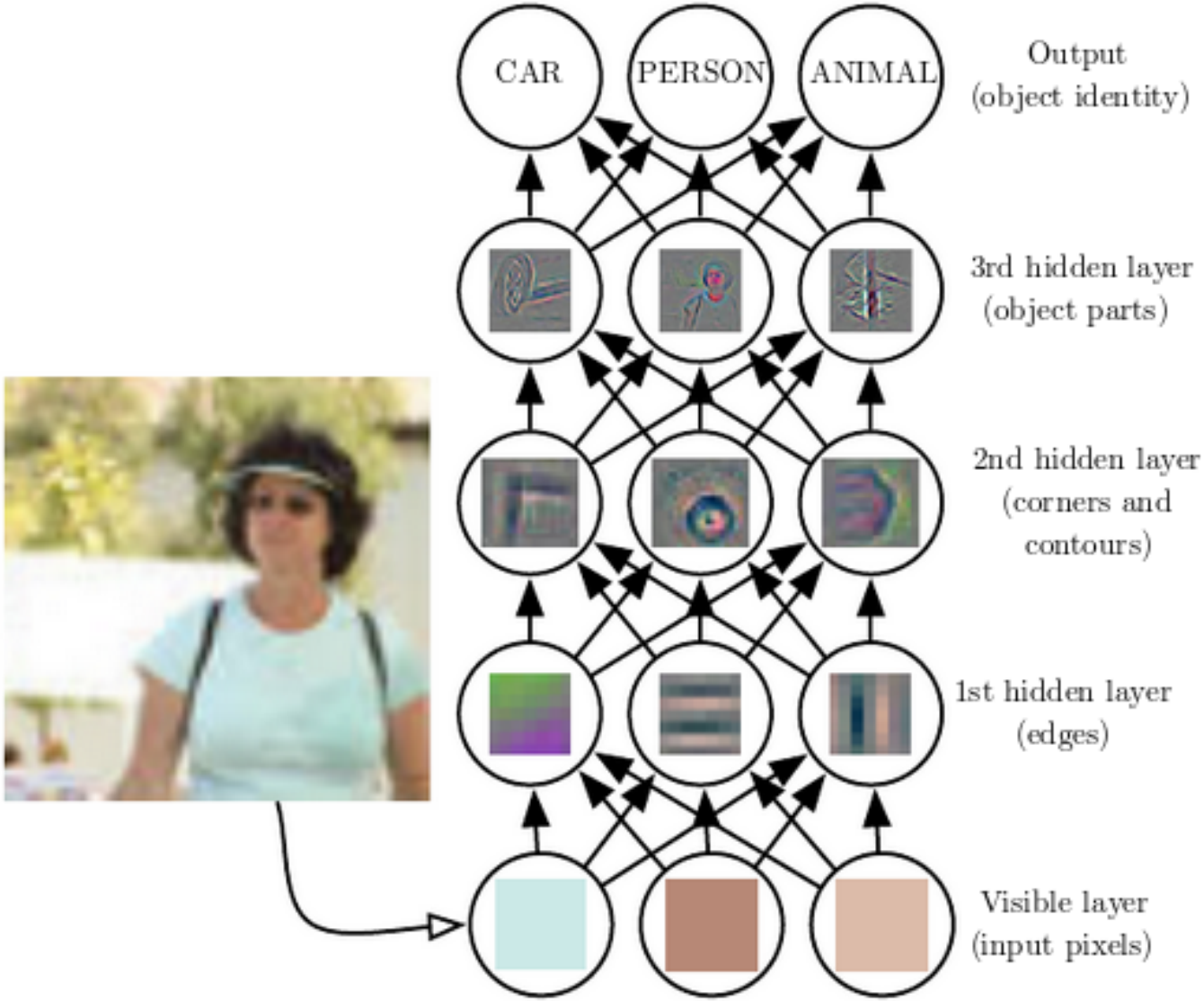
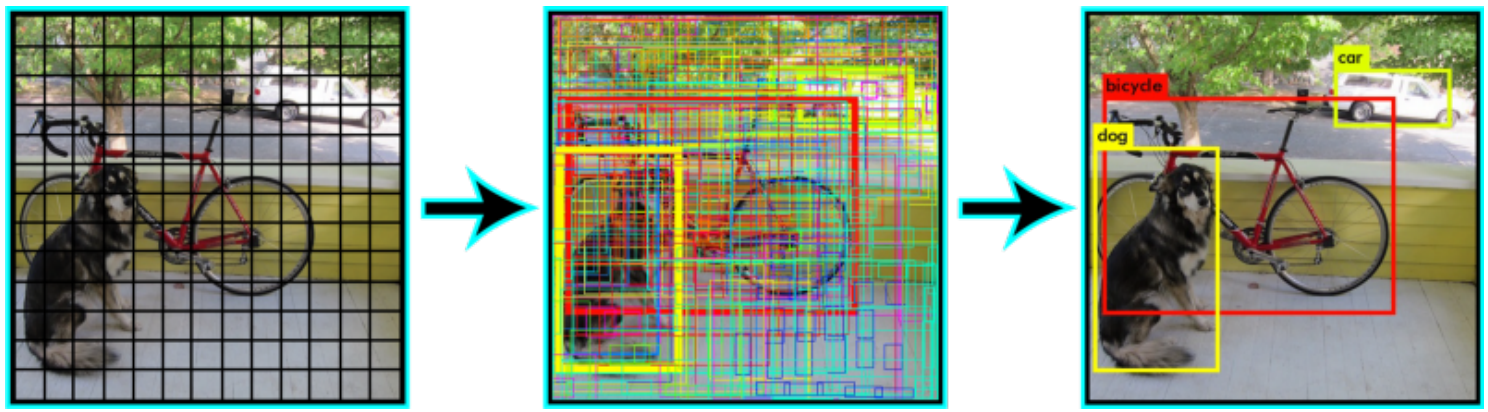


Figura: Visualización de una RNC

YOLOv3

YOLO (You Only Look Once) es un sistema estado del arte, que utiliza una red neuronal convolucional para la detección de objetos en tiempo real.



[1] Joseph Redmon. YOLOv3

Arquitectura Neuronal

Darknet-53.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

YOLOv3-tiny.

Layer	Type	Filters	Size/Stride	Input	Output
0	Convolutional	16	3 × 3/1	416 × 416 × 3	416 × 416 × 16
1	Maxpool		2 × 2/2	416 × 416 × 16	208 × 208 × 16
2	Convolutional	32	3 × 3/1	208 × 208 × 16	208 × 208 × 32
3	Maxpool		2 × 2/2	208 × 208 × 32	104 × 104 × 32
4	Convolutional	64	3 × 3/1	104 × 104 × 32	104 × 104 × 64
5	Maxpool		2 × 2/2	104 × 104 × 64	52 × 52 × 64
6	Convolutional	128	3 × 3/1	52 × 52 × 64	52 × 52 × 128
7	Maxpool		2 × 2/2	52 × 52 × 128	26 × 26 × 128
8	Convolutional	256	3 × 3/1	26 × 26 × 128	26 × 26 × 256
9	Maxpool		2 × 2/2	26 × 26 × 256	13 × 13 × 256
10	Convolutional	512	3 × 3/1	13 × 13 × 256	13 × 13 × 512
11	Maxpool		2 × 2/1	13 × 13 × 512	13 × 13 × 512
12	Convolutional	1024	3 × 3/1	13 × 13 × 512	13 × 13 × 1024
13	Convolutional	256	1 × 1/1	13 × 13 × 1024	13 × 13 × 256
14	Convolutional	512	3 × 3/1	13 × 13 × 256	13 × 13 × 512
15	Convolutional	255	1 × 1/1	13 × 13 × 512	13 × 13 × 255
16	YOLO				
17	**Route 13**				
18	Convolutional	128	1 × 1/1	13 × 13 × 256	13 × 13 × 128
19	Up-sampling		2 × 2/1	13 × 13 × 128	26 × 26 × 128
20	**Route 19 8**				
21	Convolutional	256	3 × 3/1	13 × 13 × 384	13 × 13 × 256
22	Convolutional	255	1 × 1/1	13 × 13 × 256	13 × 13 × 256
23	YOLO				

[1] Joseph Redmon. YOLOv3

Entrada

La entrada viene dada por:

$$Entrada(m, h, w, d)$$

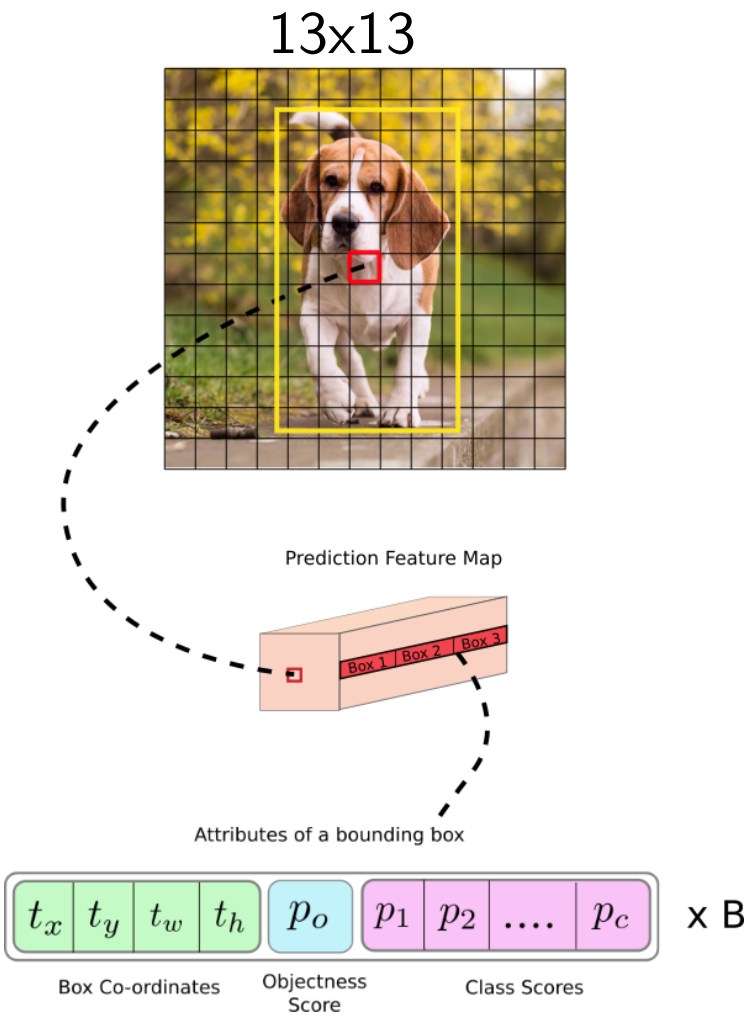
Donde:

- m es el número de imágenes en el lote de entrenamiento (en inglés, *batch*).
- h es la altura de la imagen en pixeles (en inglés, *height*).
- w es el ancho de la imagen en pixeles (en inglés, *width*).
- d es el número de canales de la imagen de entrada (en inglés, *depth*).

Si se tiene un lote de tamaño 64 y se utiliza una imagen de 3 canales (RGB) de 416x416, a la entrada de YOLOv3 tendremos:

$$Entrada(64, 416, 416, 3)$$

Procesamiento



- Proceso de detección en tres diferentes escalas: 32, 16 y 8.
- En el mapa de características, cada celda predice un número fijo de cajas delimitadoras (BB).
- En un mapa de características se tiene: BB*(5+C).
- Para una imagen de 416x416 = ((52x52) + (26x26) + (13x13))x3 = 10,647 BB.
- Por ejemplo: COCO 80 clases = 10,647 BB = 904,995 parámetros.

[1] YOLOv3 theory explained.
https://medium.com/@pythonlessons0/yolo-v3-theory-explained

Salida

Para solucionar este problema se utilizan dos filtros.

- Se utiliza un umbral mínimo para la probabilidad de detección de un objeto en la caja delimitadora, si no se supera dicho umbral, la caja delimitadora será ignorada.
- Se utiliza la supresión de no máximos para las demás cajas delimitadoras detectadas. Para esto se utiliza la métrica IoU.

Entrenamiento

Figura: Objetos etiquetados

Sistema de creación de un conjunto de escenas sintéticas

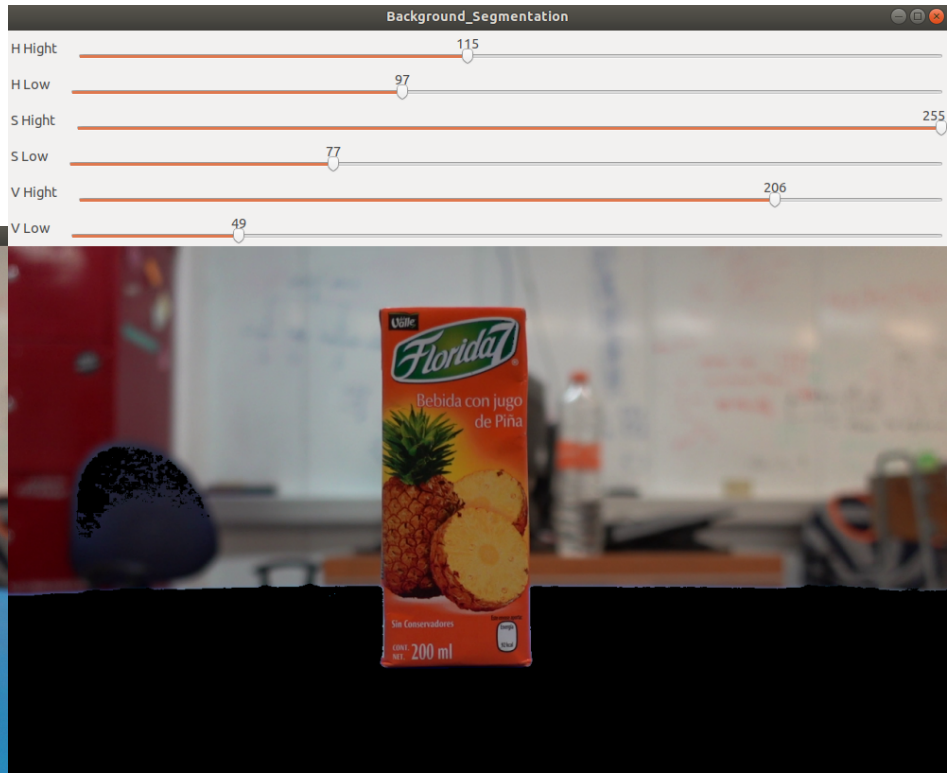
- Módulo: Módulo de segmentación de objetos.
- Módulo: Creación de escenas sintéticas.

Figura: Objetos de interés

Segmentación del fondo controlado

- Creación de un conjunto de objetos segmentados.
- Transformación de RGB a HSV
- Segmentación por umbralización.

(a) Imagen RGB

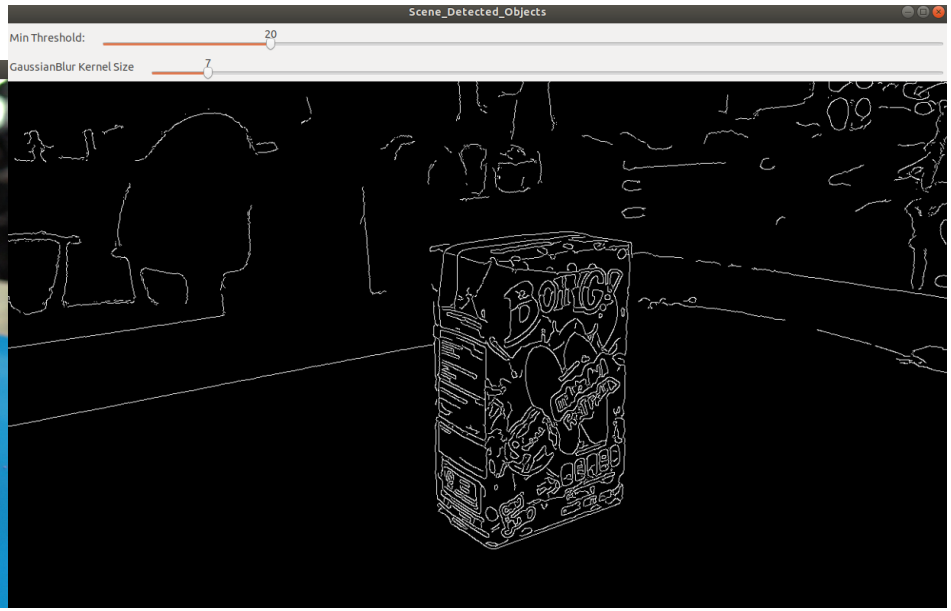


(b) Fondo controlado segmentado

Detección de bordes

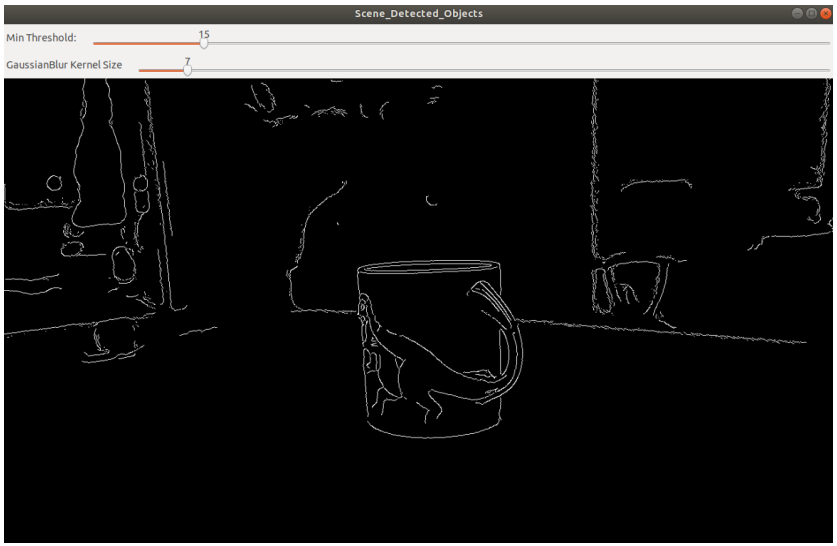
- Transformación de RGB a GRAY.
- Filtro Gaussiano.
- Canny.

(a) Imagen RGB

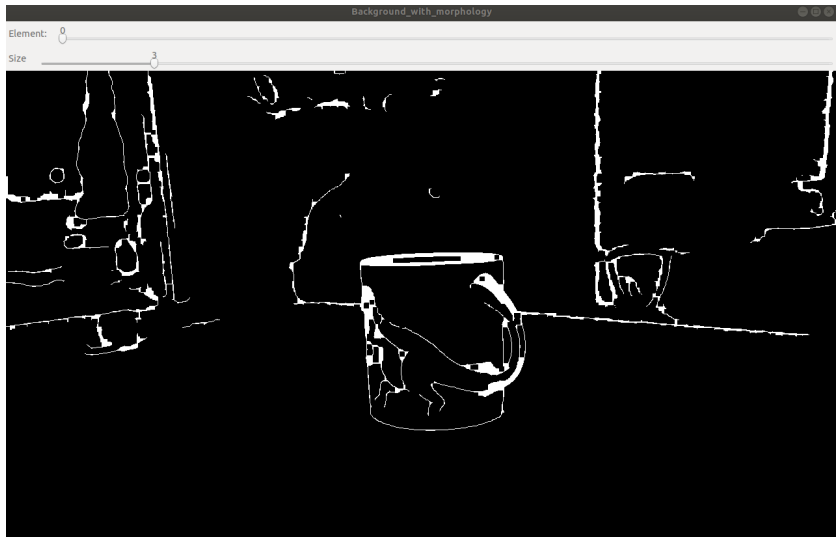


(b) Detección de bordes

Operaciones morfológicas



(a) Detección de bordes



(b) Operación Cerradura

Crecimiento de regiones

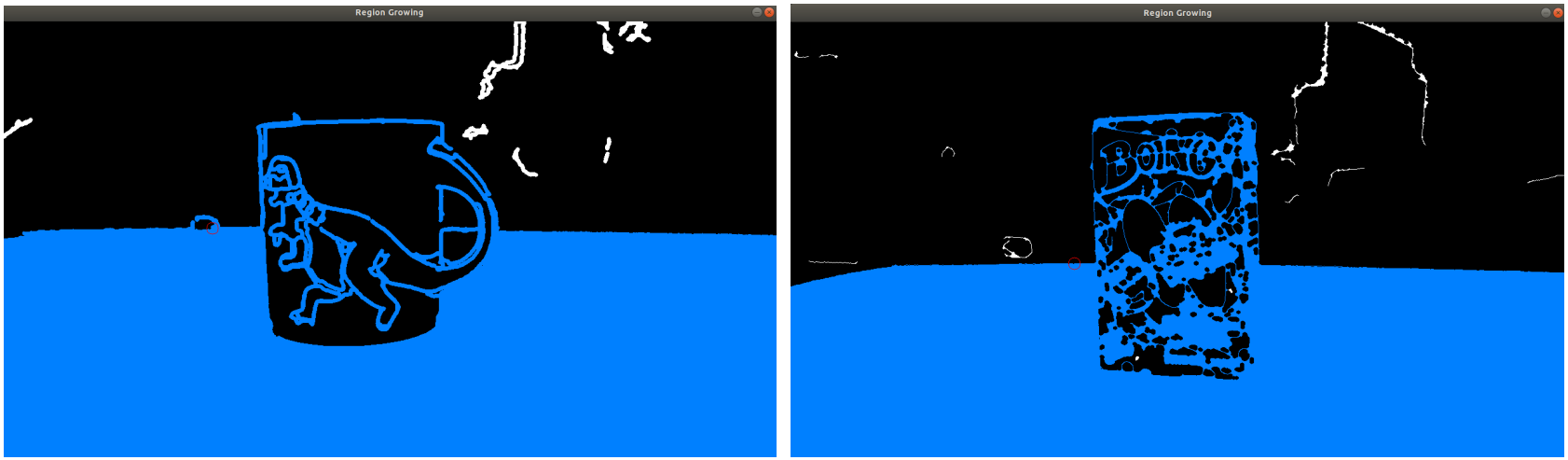
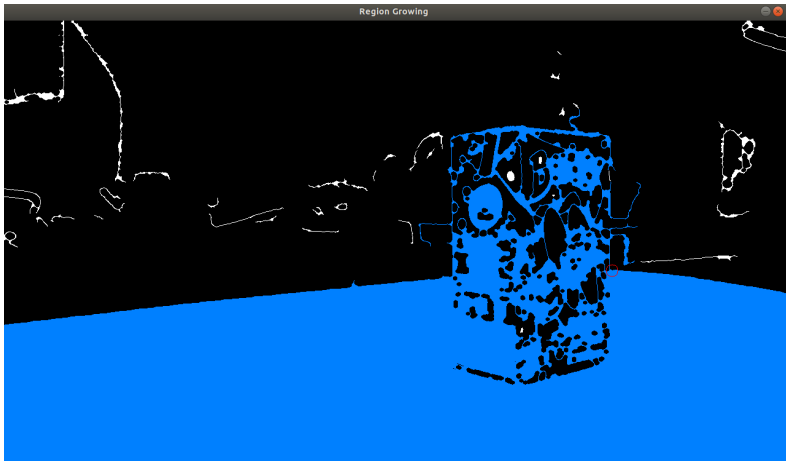
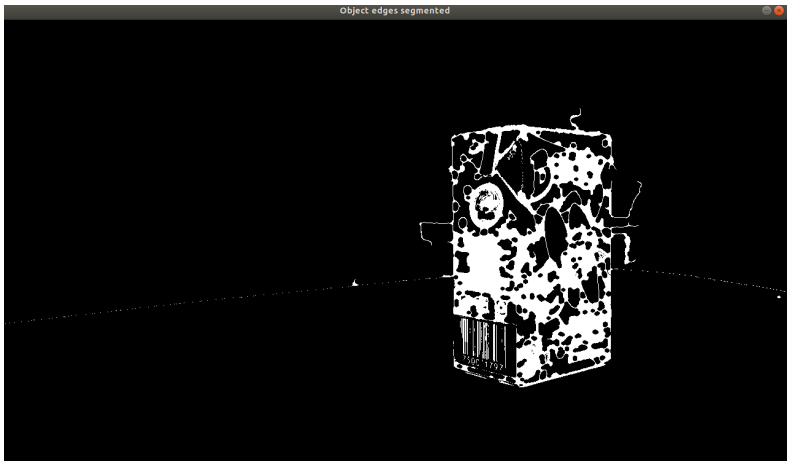


Figura: Segmentación por crecimiento de regiones.

Segmentación de los bordes de interés



(a) Crecimiento de Regiones



(b) Segmentación de bordes

(c) Segmentación del objeto de interés

Segmentación del objeto de interés

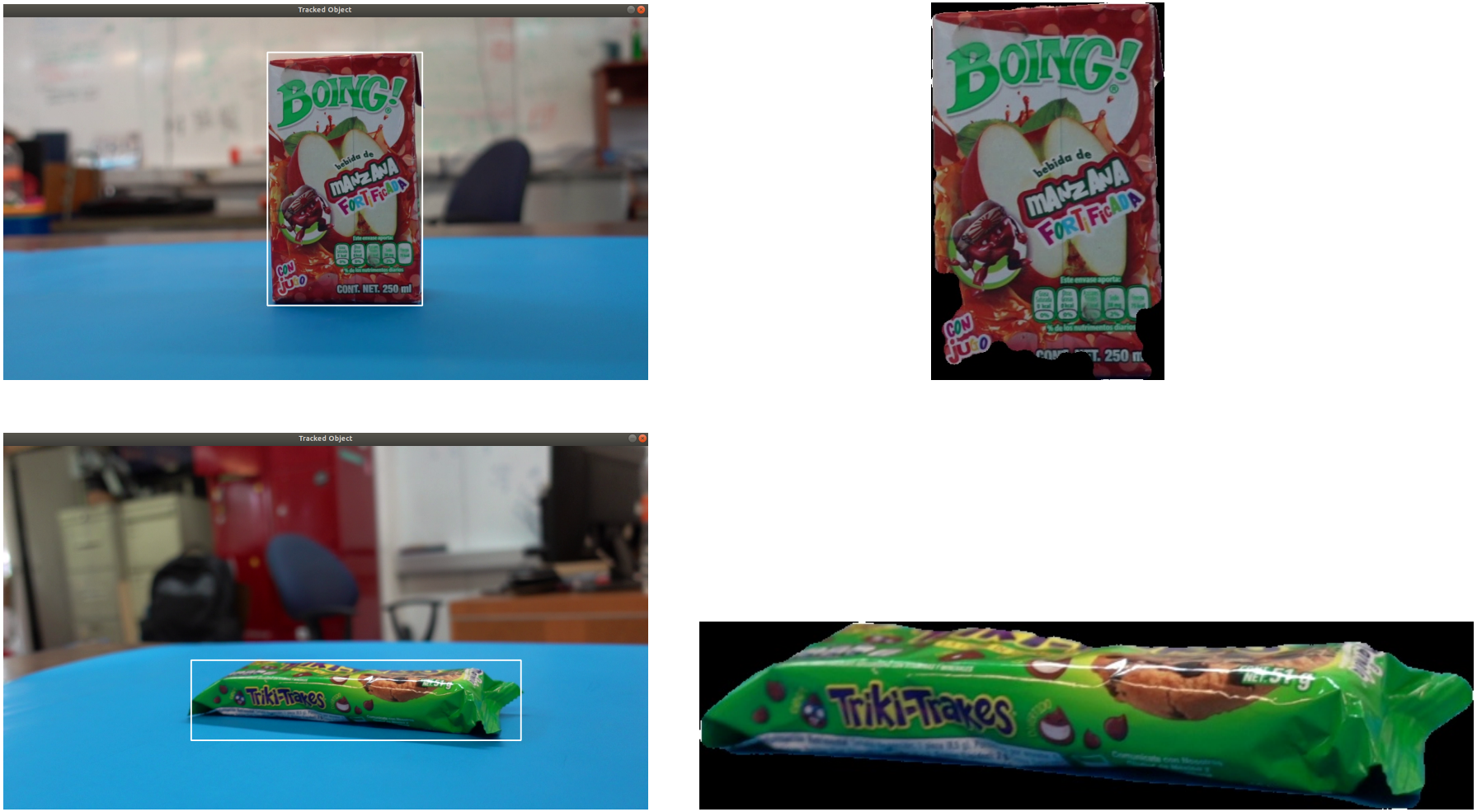


Figura: Segmentación del objeto de interés

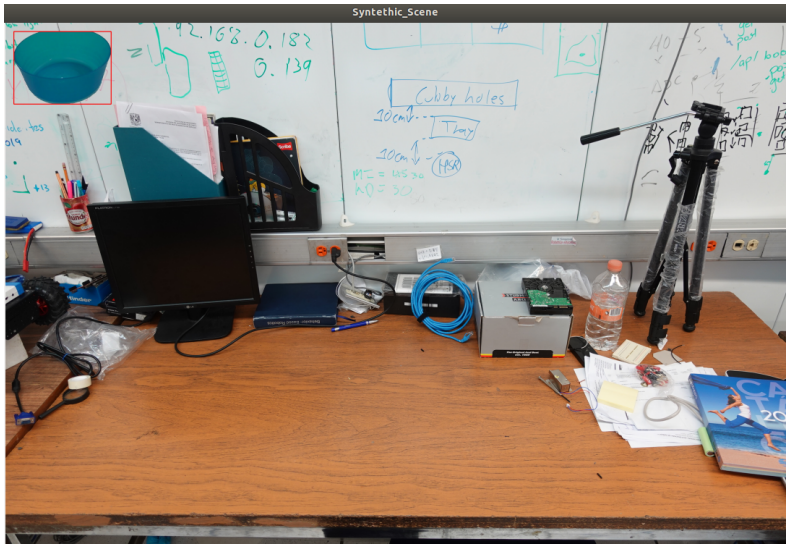
Módulo de creación de escenas sintéticas

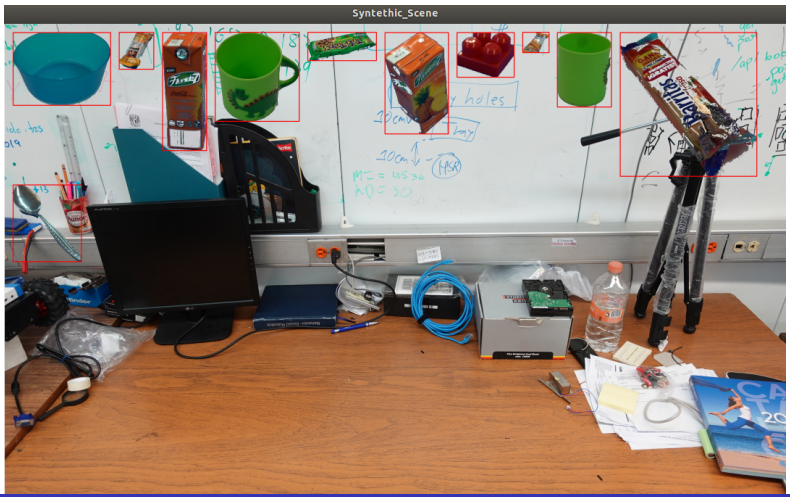
- Conjunto de objetos segmentados.
- Escenas reales.

Aumento de datos

- Desplazamientos.
- Cambio de tamaño.

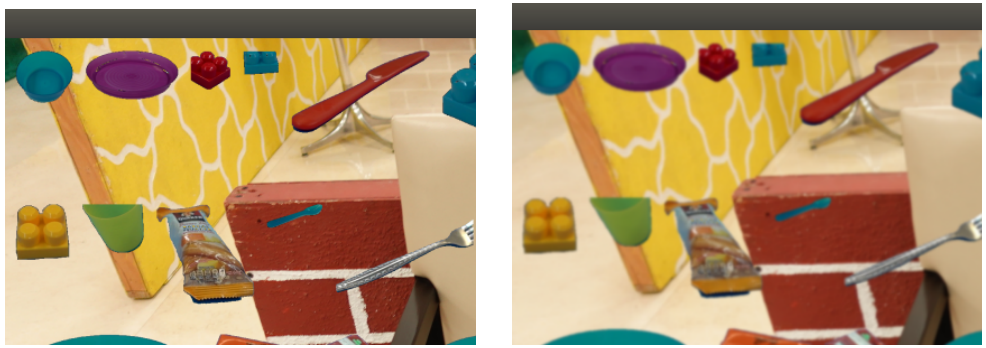
Proceso de colocación de los objectos segmentados en la escena real





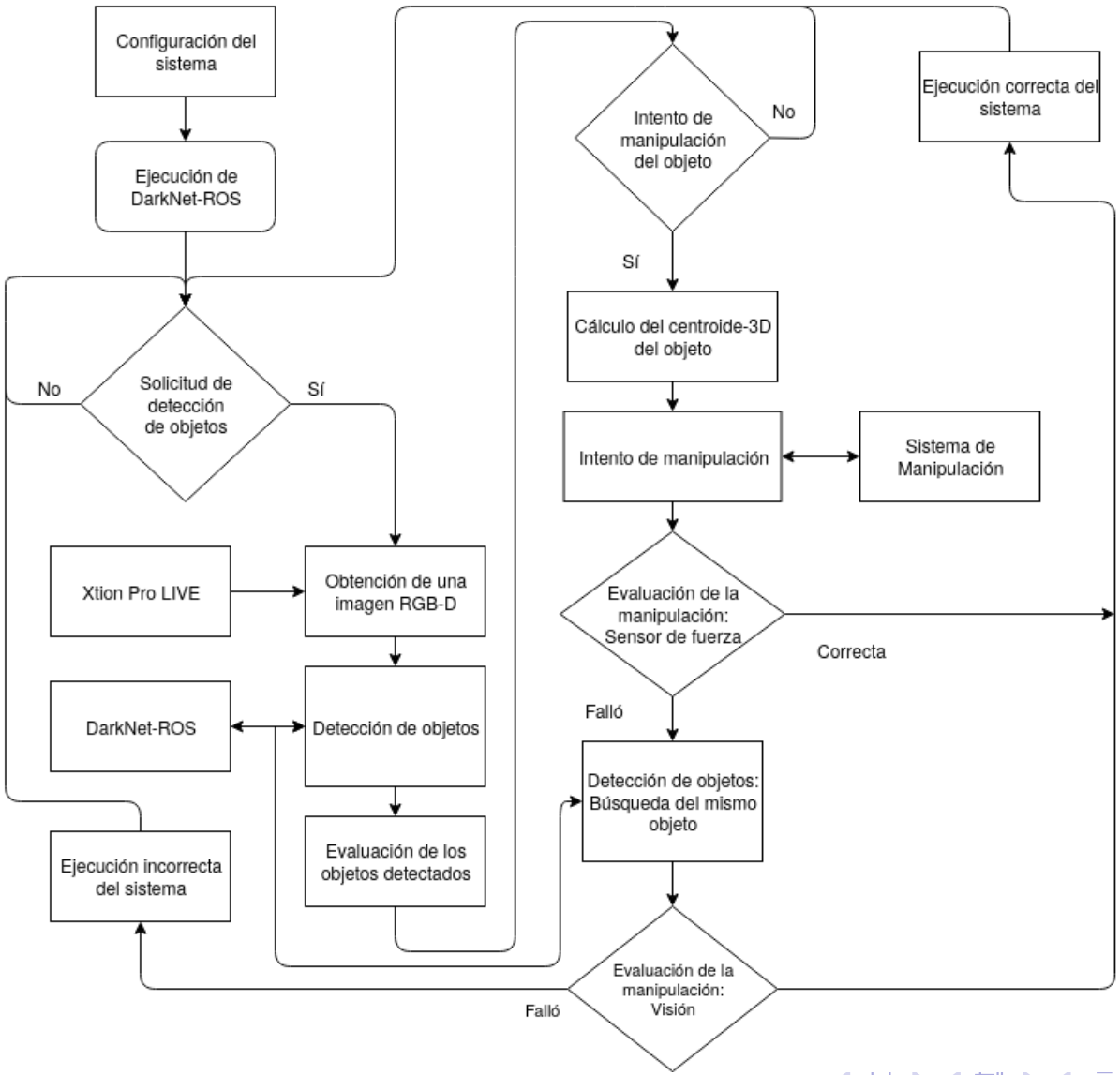
Aumento de datos en escenas

- Cambio de brillo.
- Cambio de contraste.
- Difuminación de la escena.



Generación de Escenas Sintéticas

Nodo de Detección de Objetos usando ROS



Herramientas de hardware

Computadoras de desarrollo:

- Computadora de escritorio. Procesador Intel Xeon(R) CPU E5-2650 2.20 GHz. Tarjeta gráfica Nvidia Quadro P4000 8 GB. RAM: 16 GB. Disco Duro: 1 TB.
- Alienware 15R3. Intel Core i7 7820HK. NVIDIA GTX 1080 8GB. 16 GB RAM. 128 SSD + 1TB HDD.

Objetos

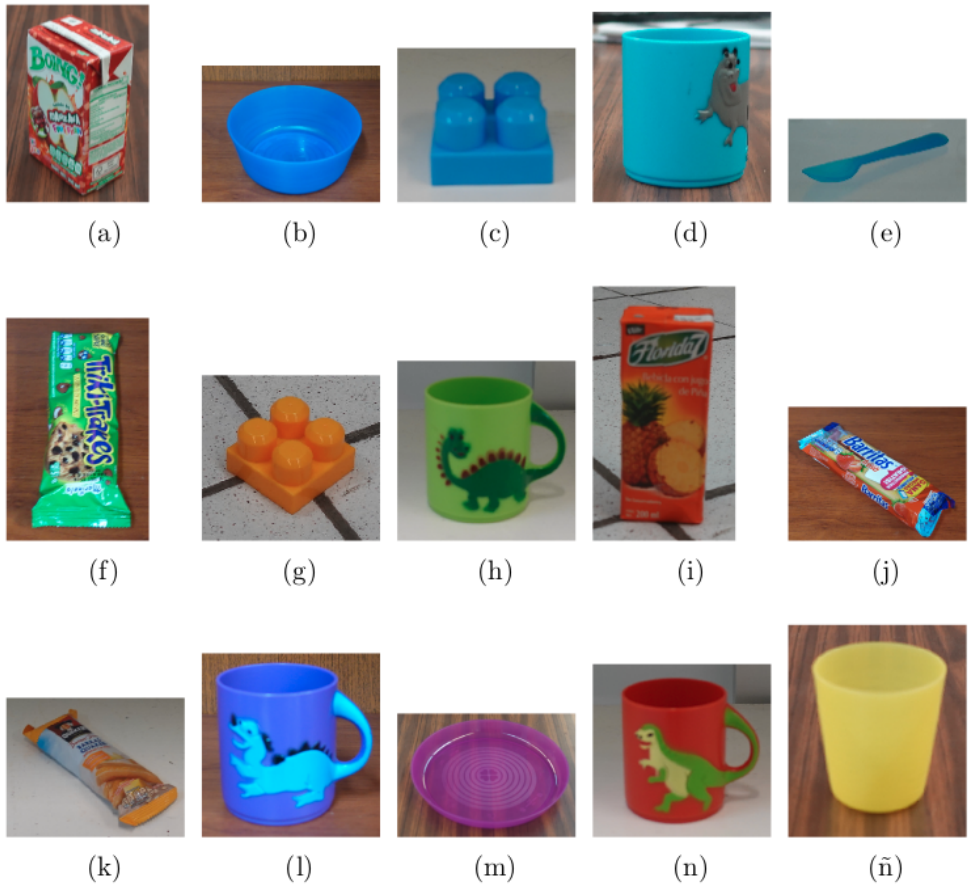


Figura 6.1: Objetos elegidos para las pruebas del sistema. 6.1a Jugo de manzana. 6.1b Tazón azul. 6.1c Lego azul. 6.1d Taza azul. 6.1e Cuchara azul. 6.1f Galletas de chocolate. 6.1g Lego amarillo. 6.1h Taza verde. 6.1i Jugo de piña. 6.1k Galletas de piña. 6.1l Taza morada. 6.1m Plato morado. 6.1n Taza Roja. 6.1ñ Vaso amarillo

Nombre del objeto contenido en el archivo de video	Duración (s.)	Número de cuadros	Número de objetos correctamente segmentados	Porcentaje de objetos correctamente segmentados
1.Jugo de manzana	47	2,820	2,779	98.54
2.Galletas de chocolate	45	2,700	2,700	100.00
3.Jugo de piña	48	2,910	2,729	93.78
4.Galletas de piña	50	3,000	2,917	97.23
5.Taza roja	47	2,820	2,374	84.18
6.Vaso amarillo	42	2,520	2,473	98.13
7.Lego amarillo	41	2,490	2,367	95.06
8.Taza morada	43	2,610	2,172	83.21
9.Tazón azul	37	2,250	2,190	97.33
10.Lego azul	44	2,640	2,604	98.63
11.Taza azul	47	2,820	2,735	96.98
12.Cuchara azul	41	2,490	2,476	99.43
13.Plato morado	38	2,280	2,007	88.02
14.Taza verde	48	2,880	2,539	88.15
15.Galletas de fresa	49	2,970	2,967	99.89
Total	**667**	**40,200**	**38,029**	**94.59**

Tabla 6.1: Resultados del módulo de segmentación de objetos para cada archivo de video.

Resultados del módulo de segmentación de objetos

Como resultado se tienen los siguientes datos:

- El sistema tuvo un tiempo de ejecución de **5 min. 57 s.** para el procesamiento de los **15** archivos de video.
- El sistema segmentó **38,029** objetos de un total de **40,200** cuadros de video (**94 %**).

Se tomaron los siguientes criterios para evaluar el resultado de la segmentación de cada objeto.

- El objeto segmentado debe tener más del **80 %** del área total del objeto real.
- El objeto segmentado no debe contener partes de la escena superiores al **10 %** de su tamaño real.

Resultados del módulo de creación de imágenes sintéticas

Conjuntos de escenas sintéticas:

- **2,500** imágenes sintéticas. 2,250 para el conjunto de entrenamiento y 250 para el conjunto de validación. Tiempo de ejecución del sistema: **0 min 40 s**.
- **5,000** imágenes sintéticas. 4,500 para el conjunto de entrenamiento y 500 para el conjunto de validación. Tiempo de ejecución del sistema: **1 min 18 s**.
- **10,000** imágenes sintéticas. 9,000 para el conjunto de entrenamiento y 1,000 para el conjunto de validación. Tiempo de ejecución del sistema: **2 min 26 s**.
- **20,000** imágenes sintéticas. 18,000 para el conjunto de entrenamiento y 2,000 para el conjunto de validación. Tiempo de ejecución del sistema: **4 min 54 s**.

Resultados del módulo de creación de imágenes sintéticas

Tipo de Aumento de datos	C.E.S. de 2,500 imágenes	C.E.S. de 5,000 imágenes	C.E.S. de 10,000 imágenes	C.E.S. de 20,000 imágenes
Número de objetos con desplazamientos	62,382	124,685	249,404	499,899
Número de objetos con cambios de tamaño	62,273	124,811	249,987	499,608
Número de escenas con cambios de brillo	480	949	1,796	3,706
Número de escenas con cambios de contraste	475	902	1,838	3,673
Número de escenas con difuminación	481	889	1,889	3,602

Tabla 6.3: Número de escenas reales y número de objetos segmentados a los cuales se les aplicó técnicas de aumento de datos.

Resultados del módulo de creación de imágenes sintéticas

Nombre del Objeto	Número de objetos en el C.E.S. de 2,500 imágenes	Número de objetos en el C.E.S. de 5,000 imágenes	Número de objetos en el C.E.S. de 10,000 imágenes	Número de objetos en el C.E.S de 20,000 imágenes
1.Jugo de manzana	8,439	16,758	33,433	66,651
2.Galletas de chocolate	8,252	16,572	33,272	66,356
3.Jugo de piña	8,384	16,612	33,416	66,689
4.Galletas de piña	8,351	16,640	33,090	66,991
5.Taza roja	8,262	16,578	33,171	66,791
6.Vaso amarillo	8,368	16,480	33,255	66,726
7.Lego amarillo	8,418	16,673	33,184	66,382
8.Taza morada	8,457	16,449	33,071	66,786
9.Tazón azul	8,283	16,666	33,440	67,040
10.Lego azul	8,313	16,765	33,644	66,739
11.Taza azul	8,129	16,879	33,485	67,128
12.Cuchara azul	8,302	16,727	33,499	66,922
13.Plato morado	8,371	16,648	33,567	67,048
14.Taza verde	8,400	16,784	33,571	66,731
15.Galletas de fresa	8,242	16,862	33,030	66,426
Total de Objetos	**124,971**	**250,093**	**500,128**	**1,001,406**

Tabla 6.2: Número de objetos segmentados que se colocaron en cada conjunto de escenas sintéticas (C.E.S.).

Entrenamiento YOLOv3

Parámetros del entrenamiento de la red neuronal convolucional:

- La arquitectura *YOLOv3_tiny*.
- *batch*=64.
- *subdivisions*=4.
- *width*=416.
- *height*=416.
- *channels*=3.
- *max_batches*=500200.
- 8 horas de entrenamiento.

Métrica IoU

Intersection over union (IoU, IU)

Toma el conjunto A de pixeles predichos del objeto y el conjunto de pixeles B del objeto verdadero (ground truth) y calcula:

$$IoU(A, B) = \frac{A \cap B}{A \cup B}$$

$IoU > 0.5$ Detección exitosa.

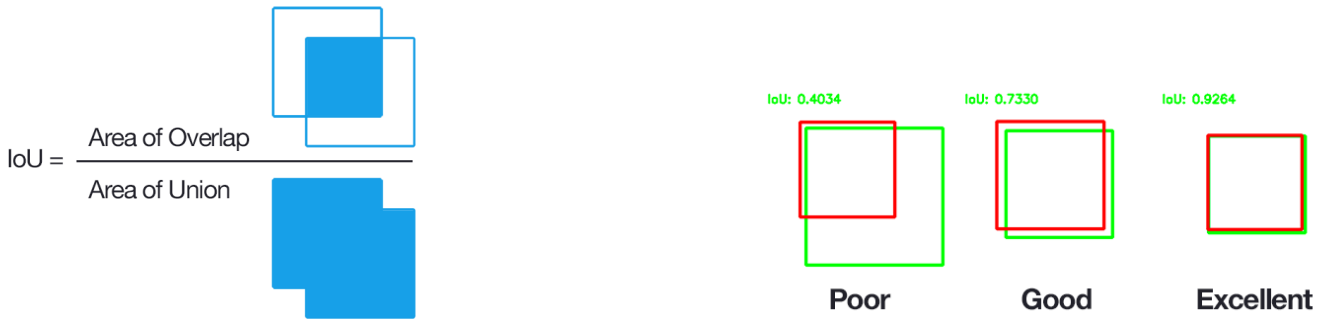


Figura: IoU

Métrica mAP

Para cada clase (C), se pueden calcular:

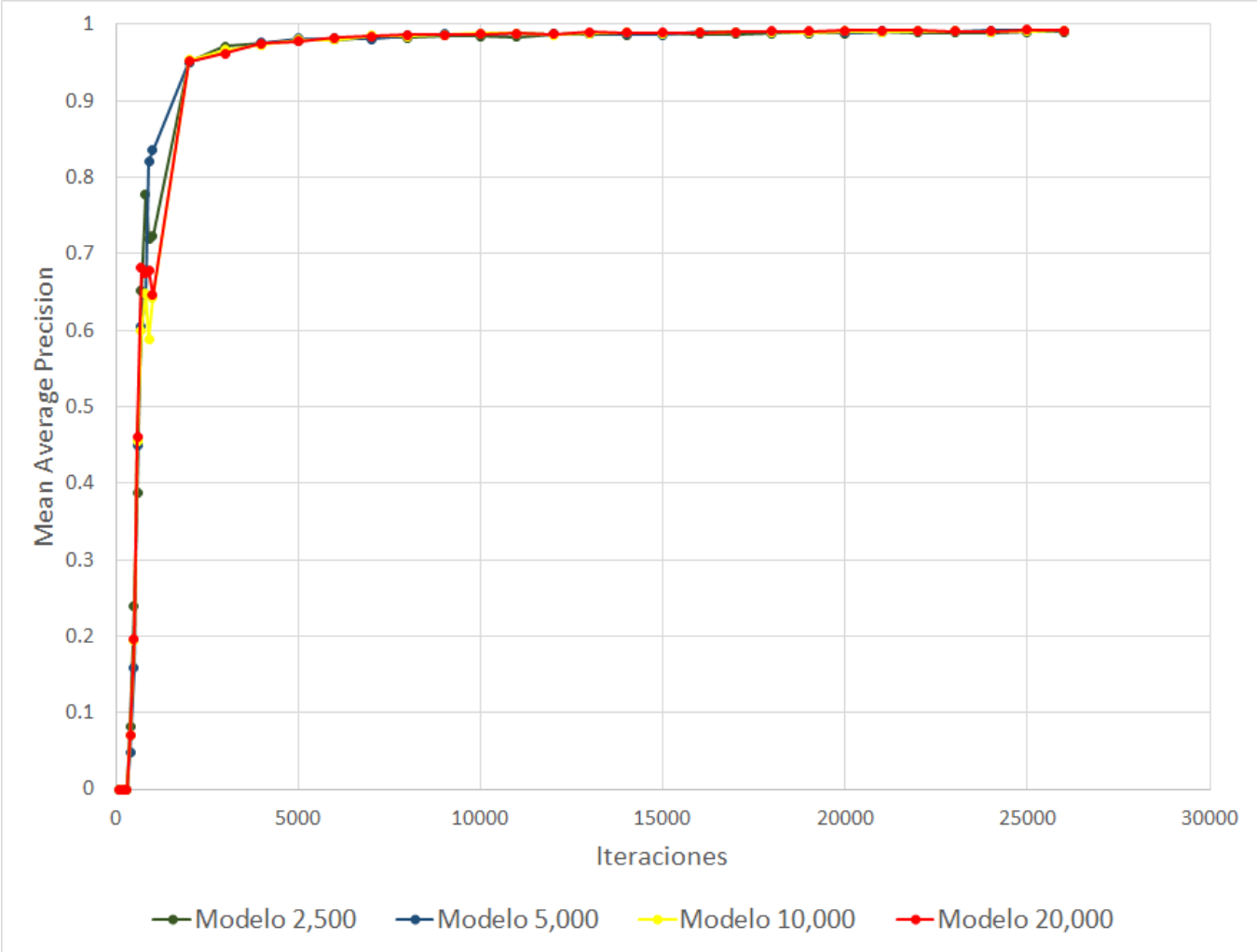
- **Número de Verdaderos positivos (TP)**: se detectó un objeto de la clase C y realmente había un objeto de clase C.
- **Número de Falsos positivos (FP)**: se detectó un objeto de la clase C, pero no hay ningún objeto de la clase C.
- **Precisión media (aP) para una clase C**:

$$aP = \frac{TP}{TP + FP}$$

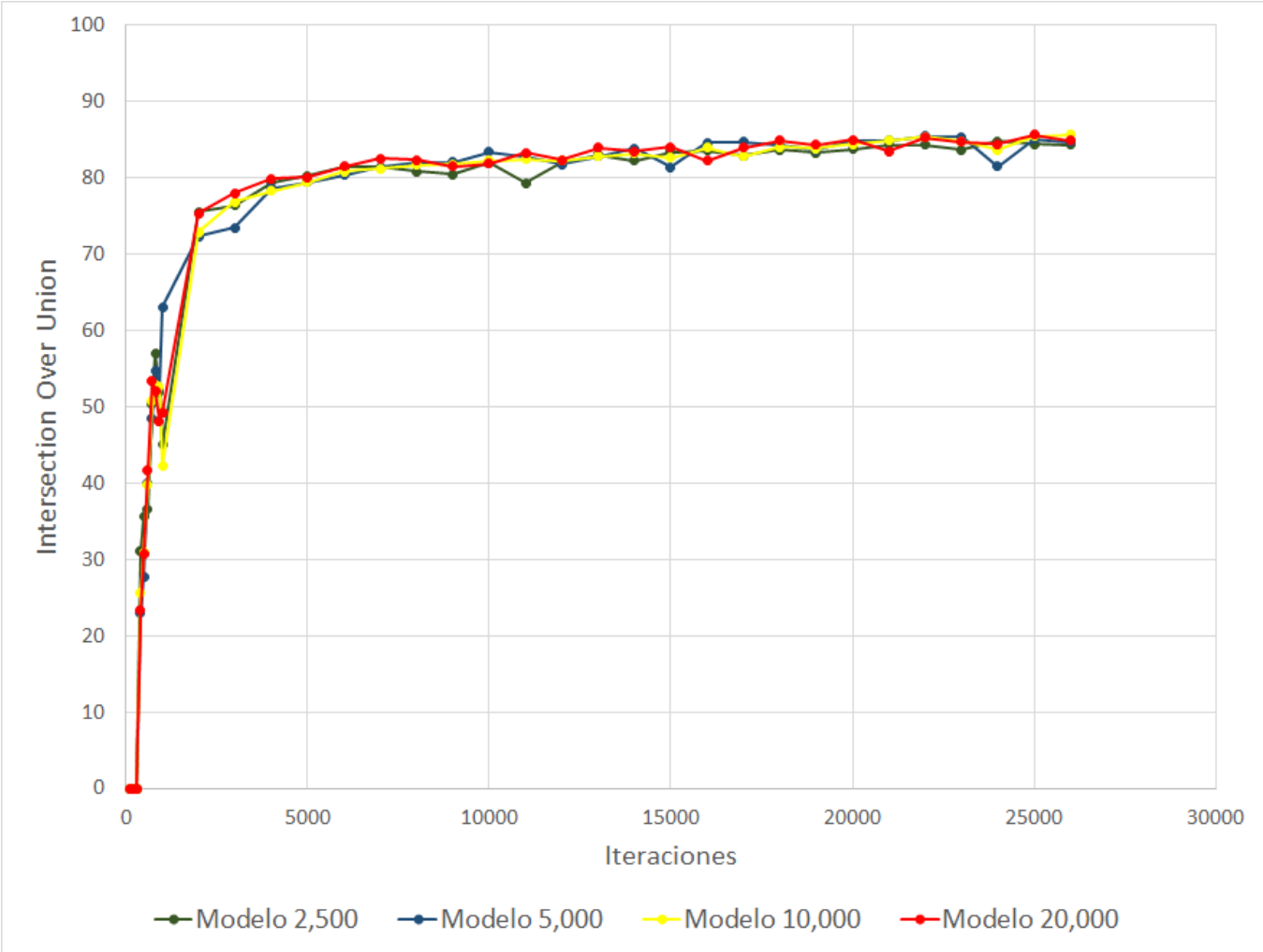
El **mAP** (*mean average precision*) resulta entonces en:

$$mAP = \frac{1}{clases} \sum_{C \in clases} \frac{TP(C)}{TP(C) + FP(C)}$$

Resultados del re-entrenamiento. Conjunto de validación



Resultados del re-entrenamiento. Conjunto de validación

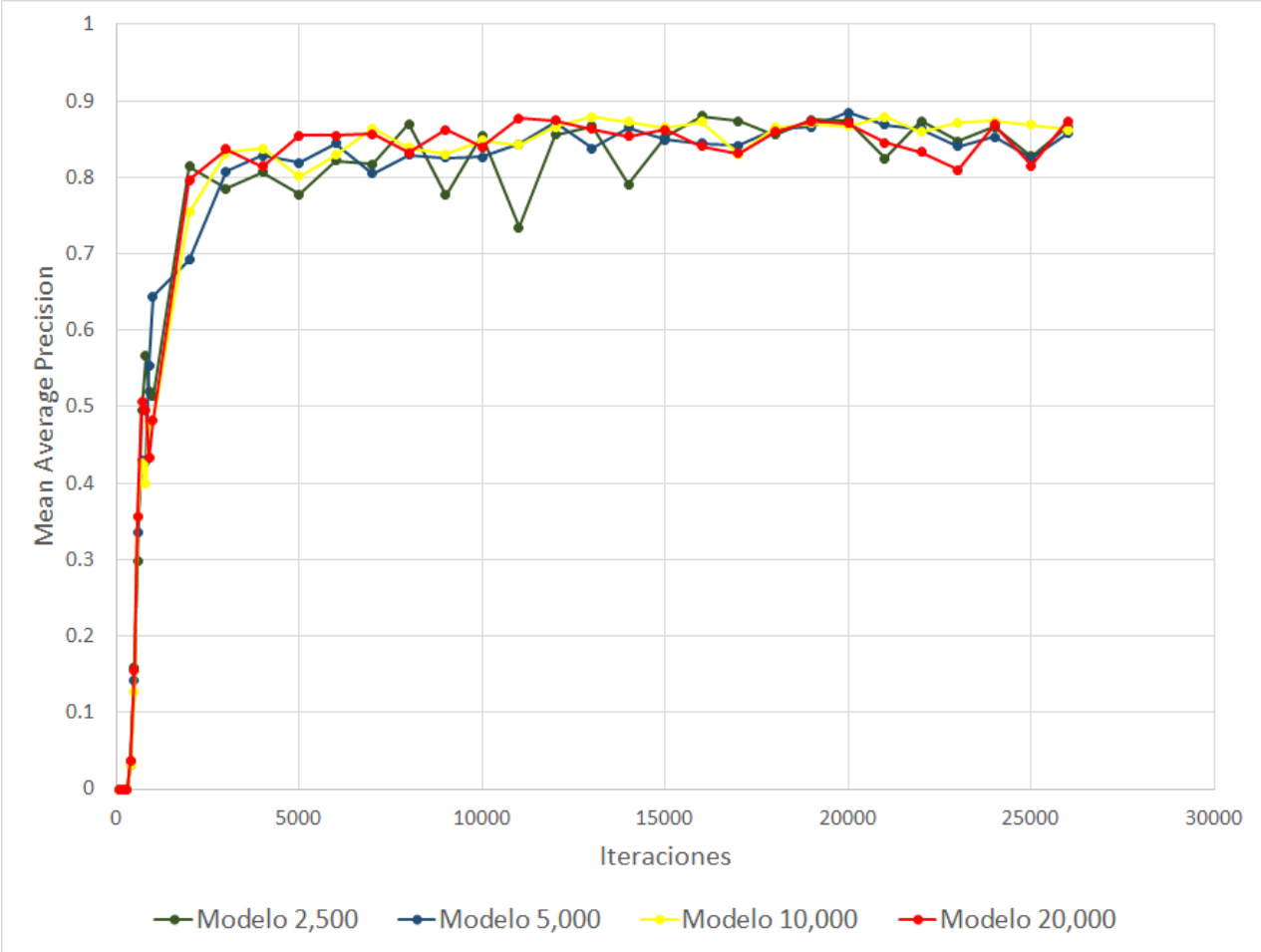


Conjunto de prueba

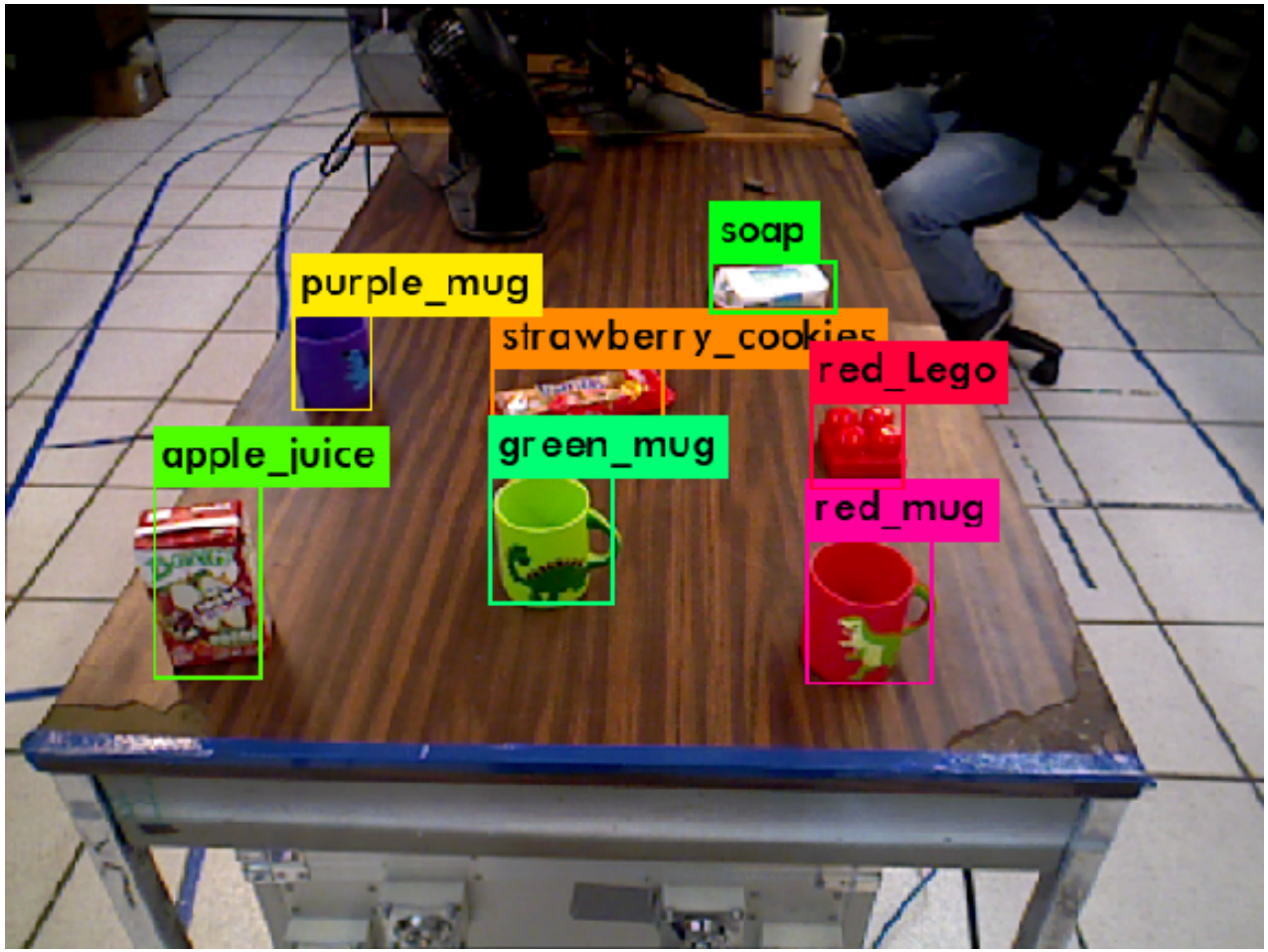
Distancia del objeto al foco de la cámara (m.)	Inclinación de 0° respecto al plano horizontal	Inclinación de 30° respecto al plano horizontal	Inclinación de 45° respecto al plano horizontal	Inclinación de 60° respecto al plano horizontal
0.3	✓	✓	✓	✓
0.5	✓	✓	✓	✓
1.0	✓	✓	✓	✓
1.5	✓	✓	✓	✓

Tabla 6.4: Procedimiento para la adquisición del conjunto de imágenes de prueba para cada objeto.

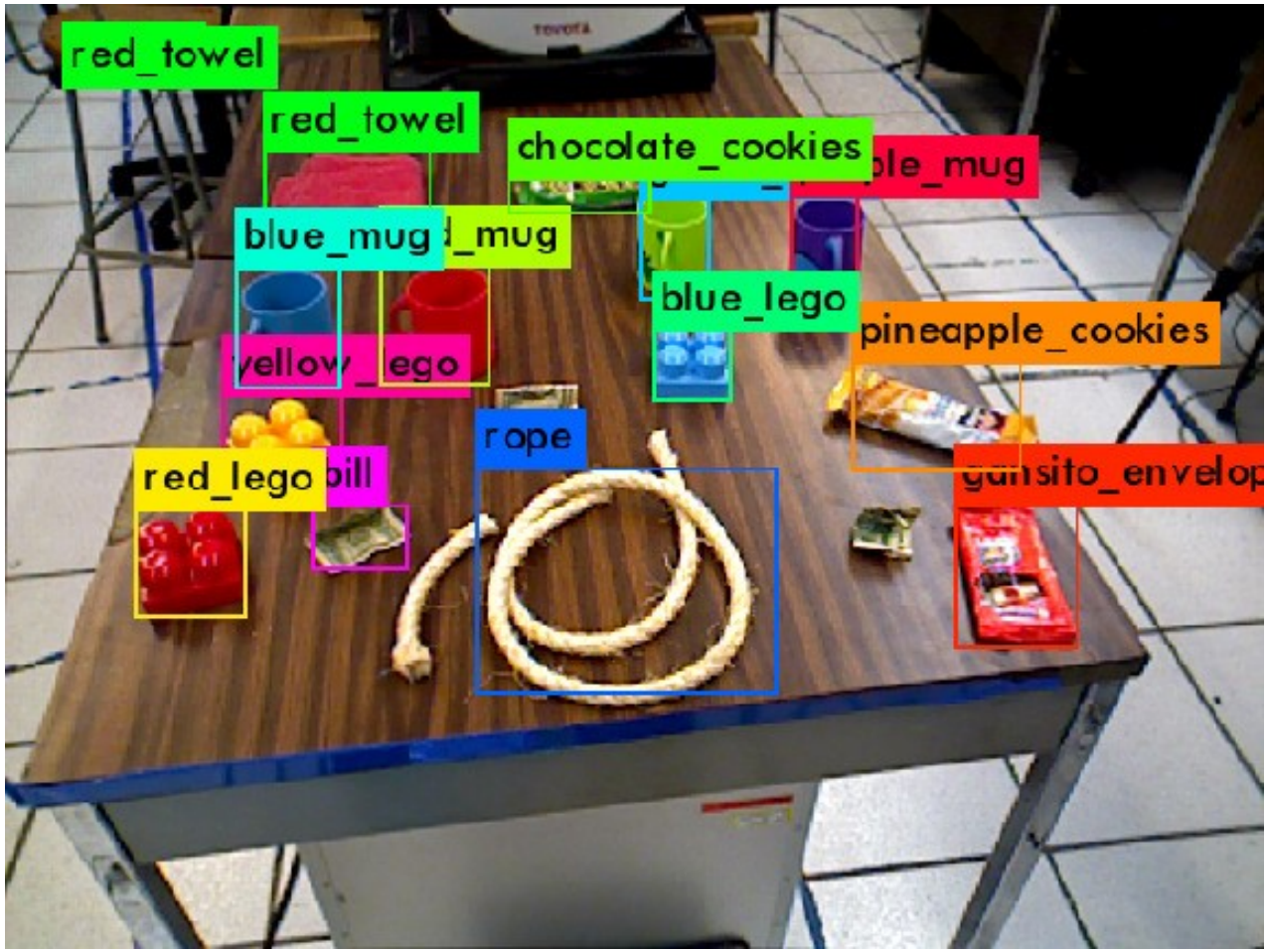
Evaluación de los modelos neuronales re-entrenados



Detecciones



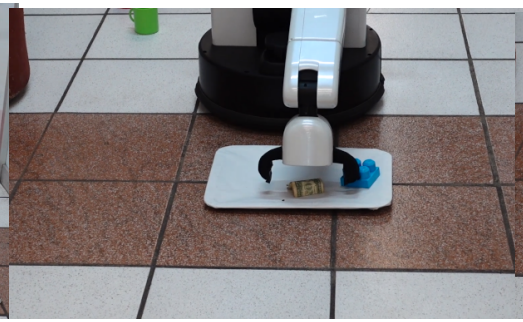
Detecciones



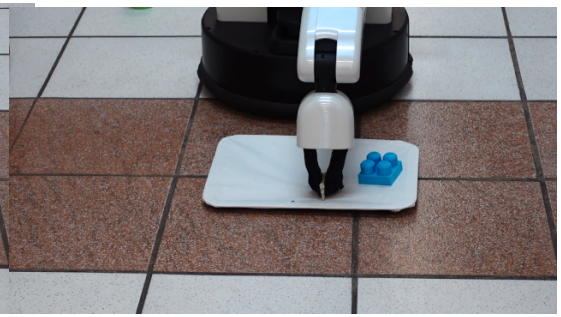
Pruebas

HSR-Challenge

(a)



(b)



(c)

(d)

(e)

Demo

Video

Conclusiones

Módulo de Segmentación de Objetos.

- 15 archivos de video (1920x1080 pixeles) procesados.
- Tiempo de ejecución: 5 minutos 57 segundos.
- Objetos segmentados: 94.59 %

Se concluye que el módulo de segmentación cumple con el objetivo: *desarrollar un sistema de visión computacional que permita segmentar objetos utilizando archivos de video adquiridos por una cámara digital compacta*.

Conclusiones

Módulo de Creación de Imágenes Sintéticas

- Conjunto de escenas sintéticas de 5,000 imágenes.
- Tiempo de ejecución: 1 minuto 58 segundos.
- En promedio se tienen 16,000 objetos de cada clase para el conjunto de entrenamiento y el conjunto de validación.

Se concluye que el módulo de creación de imágenes sintéticas cumple con el objetivo: *desarrollar un sistema de visión computacional que cree un conjunto de imágenes sintéticas, es decir, escenas artificiales que contengan los objetos segmentados del sistema anterior*.

Conclusiones

Nodo de Detección de Objetos

Conjunto de prueba:

- mAP: 85 % para 15 clases.

Competencia HSR-Challenge-3:

- 100 % en la clasificación de 12 objetos.

Se concluye que el nodo de detección de objetos propuesto cumple con el objetivo: *desarrollar un sistema que adapte el modelo neuronal re-entrenado como un sistema de detección de objetos en un robot de servicio*.

Conclusiones

Hipótesis

Se comprobó la hipótesis planteada al inicio del proyecto:
El entrenamiento de un modelo neuronal con imágenes sintéticas generadas de manera automática, tendrá altos índices de precisión en la detección de objetos en una escena real. La implementación de este modelo mejorará el sistema de detección de objetos en un robot de servicio, aumentando los índices de confidencia en la etapa de reconocimiento y mejorando la detección de objetos, para que puedan ser manipulados en una escena tridimensional.

Trabajo futuro

Se plantean mejoras a los sistemas propuestos:

- Módulo de segmentación de objetos. Mejorar el procesamiento de bajo nivel de las escenas capturadas con la cámara compacta.
- Módulo de creación de imágenes sintéticas. Desarrollar un sistema que permita detectar con precisión los lugares específicos donde pueden aparecer los objetos.
- Aumentar el tamaño de la imagen de entrada del modelo neuronal YOLOv3.
- Arquitectura de la red neuronal. Explorar otras arquitecturas y evaluar el desempeño de los conjuntos de entrenamiento y validación.

¡Gracias por su atención!