

YOLO

Leonardo Valdivia, Luis Vilcapaza, Erick Gutierrez

Universidad Nacional de San Agustín

June 23, 2022

Overview

1 YOLO1

- Funcionamiento
- Arquitectura
- Loss Function
- Limitaciones
- Resultados

2 YOLO2

- Mejoras
- Mejoras Batch Normalization.
- Mejoras High Resolution Classifier.
- Mejoras Convolutional With Anchor Boxes.
- Dimension Clusters.
- Direct location prediction.
- Fine-Grained Features.
- Multi-Scale Training.
- Arquitectura Dark-net19.

Overview

1 YOLO1

- Funcionamiento
- Arquitectura
- Loss Function
- Limitaciones
- Resultados

2 YOLO2

3 YOLOv3

YOLO1

YOLO es una red neuronal convolucional que predice bounding boxes y probabilidad de clase para dichos bounding boxes. Utiliza características de toda la imagen para predecir cada cuadro delimitador. También predice todos los cuadros delimitadores en todas las clases para una imagen simultáneamente. Esto significa que nuestra red razona globalmente sobre la imagen completa y todos los objetos de la imagen. El diseño de YOLO permite velocidades en tiempo real mientras mantiene una alta precisión promedio

Funcionamiento

- ▶ Dividir la imagen en una grilla de tamaño $S \times S$.
- ▶ Si el centro de un objeto cae en una celda entonces esta celda sera responsable de detectar el objeto.

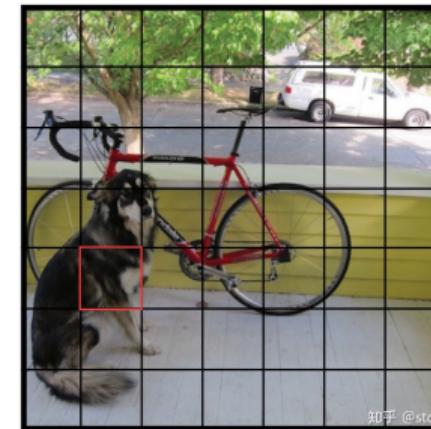


Figure 1: Grid. Joseph Redmon (2016b)

Funcionamiento

- ▶ Cada celda contiene B bounding boxes y C clases de probabilidad.
- ▶ Cada bounding box contiene 5 predicciones:
 - ▶ (x, y) que representa coordenadas relativas a la celda.
 - ▶ (W, h) que representan el ancho y el alto relativo al tamaño de toda la imagen.
 - ▶ Puntaje de confianza.

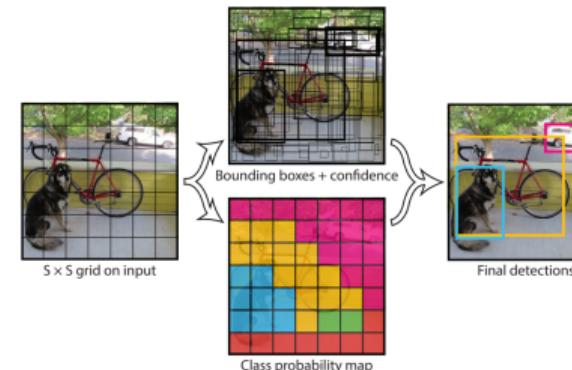


Figure 2: Bounding boxes Joseph Redmon (2016b)

Funcionamiento

- ▶ El puntaje de confianza refleja cuan confiado esta el modelo acerca de la precisión del bounding box y si este contiene algún objeto.
- ▶ El puntaje de confianza esta definido como: $Pr(\text{Object}) \times IOU_{truth}^{pred}$
- ▶ Si no existe ningún objeto en la celda entonces el puntaje de confianza es cero.
- ▶ Si existe un objeto en la celda entonces el puntaje de confianza es igual a IOU entre el bounding box predicho y el real.

Funcionamiento

Las dimensiones del tensor estan dadas por $S \times S \times (B \times 5 + C)$

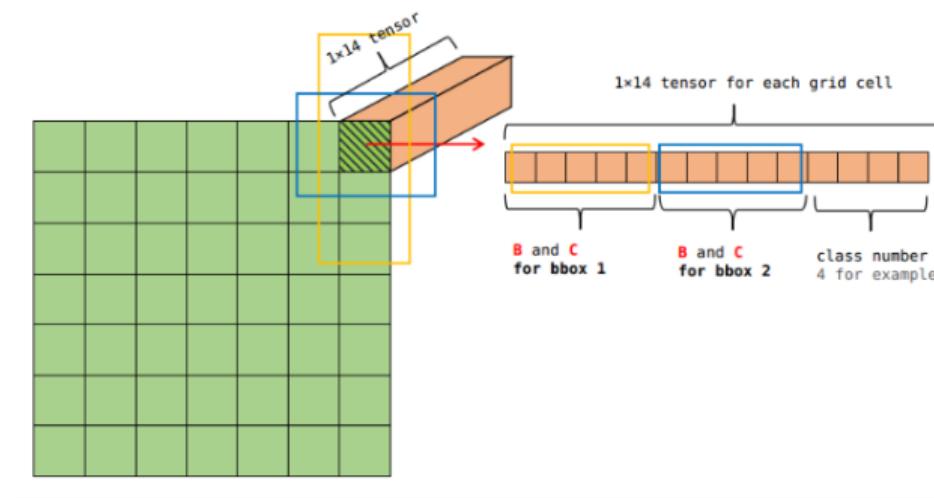


Figure 3: Yolo tensor example

Arquitectura

La red esta compuesta por 24 capas convolucionales seguida de 2 fully connected.

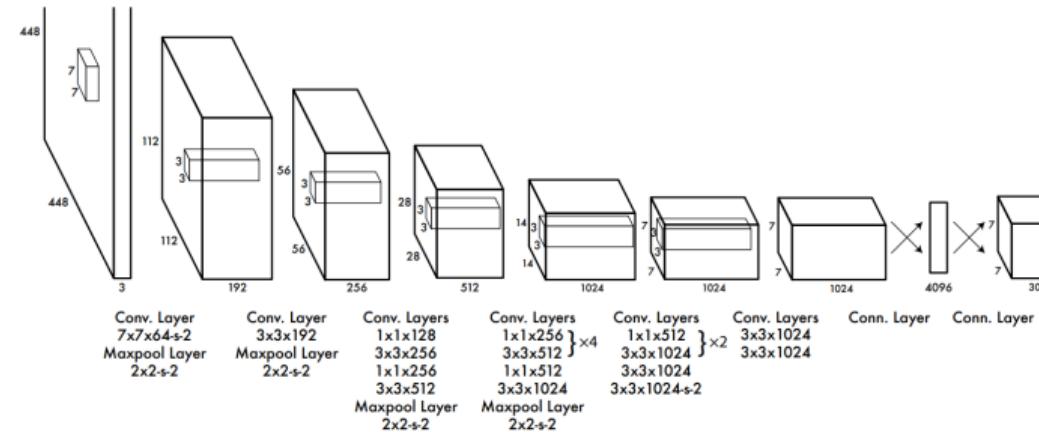


Figure 4: Arquitecture Joseph Redmon (2016b)

Loss Function

- ▶ Ponderar el error de localización por igual con el error de clasificación que puede no ser ideal.
- ▶ Muchas celdas de cuadrícula no contienen ningún objeto.
- ▶ Esto empuja los puntajes de "confianza" de esas celdas hacia cero, a menudo superando el gradiente de las celdas que contienen objetos. Esto puede conducir a la inestabilidad del modelo, lo que hace que el entrenamiento diverja desde el principio.

Loss Function

$$\begin{aligned}
 & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
 & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
 & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
 & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2 \quad (3)
 \end{aligned}$$

Figure 5: Loss Function Joseph Redmon (2016b)

Limitaciones

- ▶ YOLO impone fuertes restricciones espaciales ya que cada celda solo puede contener dos bounding boxes y una clase.
- ▶ Tiene dificultades para generalizar objetos en diferente relación aspecto.
- ▶ El error es el mismo en bounding boxes pequeños y grandes.

Resultados

Comparación con otros sistemas en tiempo real.

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	155
YOLO	2007+2012	63.4	45
<hr/>			
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Figure 6: Resultados. Joseph Redmon (2016b)

Resultados

Comparación con otros sistemas en tiempo real.

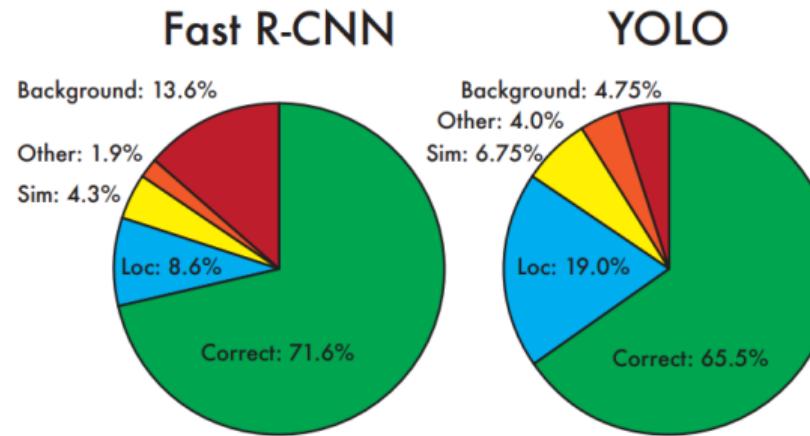


Figure 7: Resultados. Joseph Redmon (2016b)

Resultados

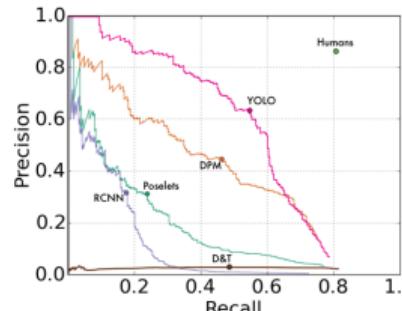
Generalización en datasets de arte.



Figure 8: Resultados. Joseph Redmon (2016b)

Resultados

Generalización en datasets de arte.



(a) Picasso Dataset precision-recall curves.

	VOC 2007	Picasso		People-Art
	AP	AP	Best F_1	AP
YOLO	59.2	53.3	0.590	45
R-CNN	54.2	10.4	0.226	26
DPM	43.2	37.8	0.458	32
Poselets [2]	36.5	17.8	0.271	
D&T [4]	-	1.9	0.051	

(b) Quantitative results on the VOC 2007, Picasso, and People-Art Datasets.
The Picasso Dataset evaluates on both AP and best F_1 score.

Figure 9: Resultados. Joseph Redmon (2016b)

Overview

1 YOLO1

2 YOLO2

- Mejoras
- Mejoras Batch Normalization.
- Mejoras High Resolution Classifier.
- Mejoras Convolutional With Anchor Boxes.
- Dimension Clusters.
- Direct location prediction.
- Fine-Grained Features.
- Multi-Scale Training.
- Arquitectura Dark-net19.
- STRONGER.

YOLO2

YOLO es una red neuronal convolucional que predice bounding boxes y probabilidad de clase para dichos bounding boxes.

Mejoras

- ▶ Mejorar recall.
- ▶ Mejorar la precisión pero mantener la velocidad
- ▶ Mejoran la red para que le sea más rápido aprender

Mejoras Batch Normalization.

- ▶ Mejoras significativas.
- ▶ Mejorar hasta un poco más del 2% en el mAP.

Mejoras High Resolution Classifier.

- ▶ clasificación a la resolución completa de 448×448 durante 10 épocas.
- ▶ Esta red de clasificación de alta resolución da un aumento de casi un 4% mAP.P.

Mejoras Convolutional With Anchor Boxes.

- ▶ Usa anchor boxes predecir los bounding boxes.
- ▶ Imagen con un factor de 32.
- ▶ IOU
- ▶ Usar anchor boxes reduce precisión pero aumenta el recall. (mAP)
- ▶ Aumento de recall indica que el modelo puede mejorar

Mejoras Convolutional With Anchor Boxes.

Cuadro azul = fondo verdadero

Cuadro naranja = la predicción

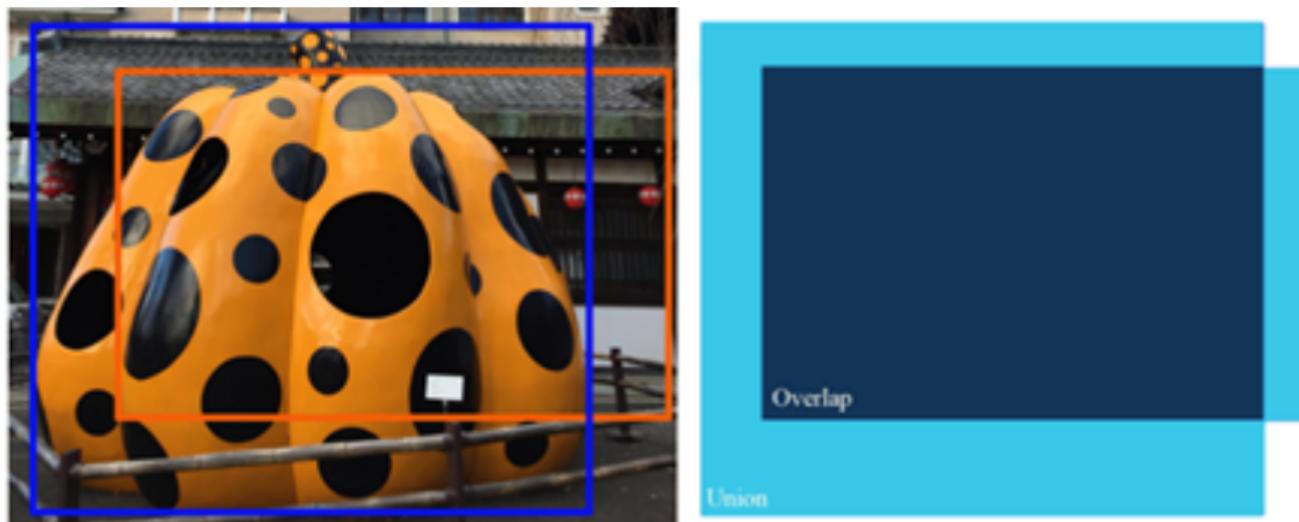


Figure 10: IOU

Dimension Clusters.

- ▶ 2 desventajas de YOLO con los anchor boxes.
- ▶ anchor box son hand picked(a mano).
- ▶ Puede variar el aprendizaje.
- ▶ En vez de *Prior* a mano se usa K-means.
- ▶ K-means para encontrar un buen *Prior*.

Dimension Clusters.

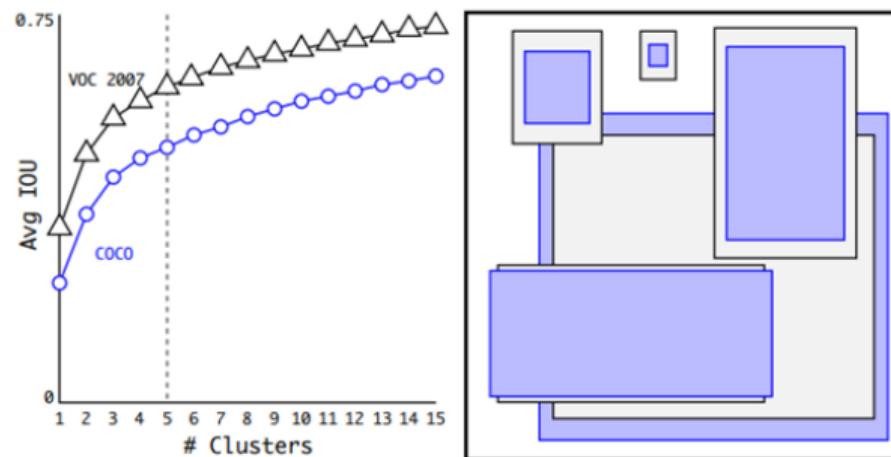


Figure 11: Cluster y anchor box Joseph Redmon (2016a)

Dimension Clusters.

Datos respecto a la figura 11.

- ▶ Usar K-means con $k = 5$.
- ▶ $K = 5$ buen equilibrio entre recall y la complejidad del modelo.
- ▶ Imagen de la derecha da centroides relativos.

Dimension Clusters.

- ▶ No usar la distancia euclidian en k-mean.
- ▶ Euclidiana genera errores.
- ▶ Usa lo siguiente.

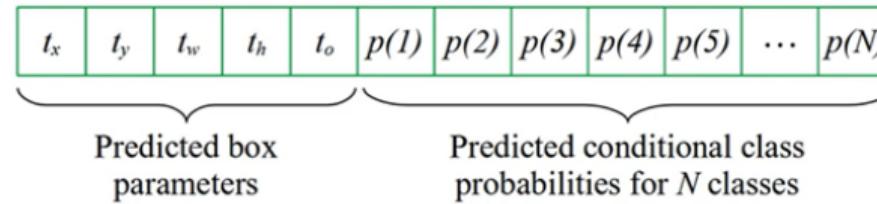
$$d(\text{box}, \text{centroid}) = 1 - \text{IOU}(\text{box}, \text{centroid}) \quad (1)$$

Direct location prediction.

- ▶ Aquí se encuentra el segundo problema con YOLO.
- ▶ Crea inestabilidad durante las primeras iteraciones.
- ▶ Errores al momento de predecir (x, y)

Direct location prediction.

Vector de una caja de predicción



This is the prediction vector for one box in YOLOv2

Figure 12: Vector box

Direct location prediction.

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w e^{t_w}$$

$$b_h = p_h e^{t_h}$$

$$Pr(\text{object}) * IOU(b, \text{object}) = \sigma(t_o)$$

Figure 13: Predicción

Fine-Grained Features.

- ▶ Predice mapas de características de 13x13.
- ▶ Bueno para detectar objetos largos.
- ▶ Detecta objetos más pequeños

Fine-Grained Features.

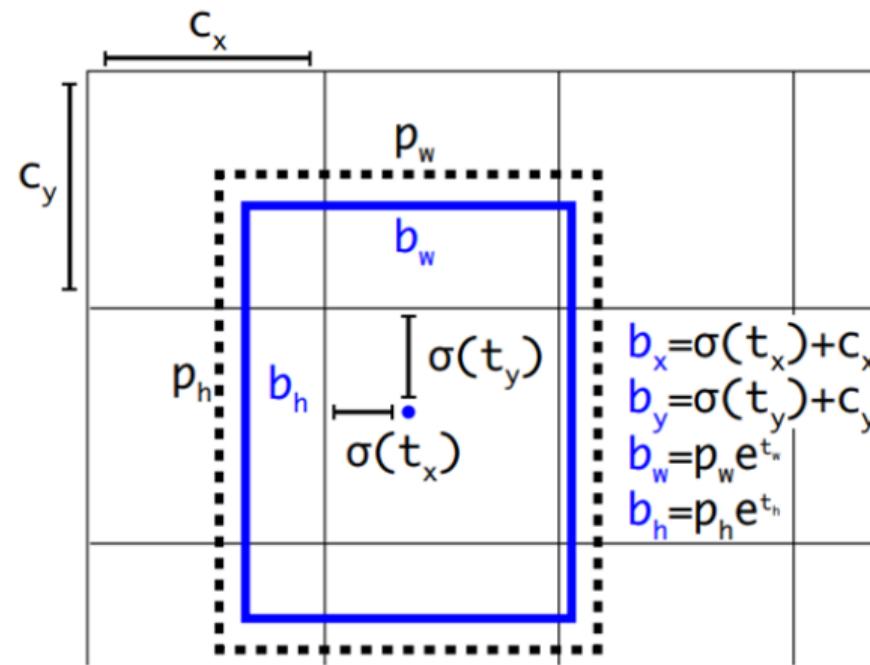


Figure 14: location prediction Joseph Redmon (2016a)

Multi-Scale Training.

- ▶ Se quiere que el modelo sea robusto
- ▶ Que el modelo pueda con imagenes de diferente tamaños.
- ▶ A bajas resoluciones, YOLOv2 funciona como un detector barato y bastante preciso.
- ▶ No una misma escala.

Multi-Scale Training.

Multi-scale training: +1.5% mAP

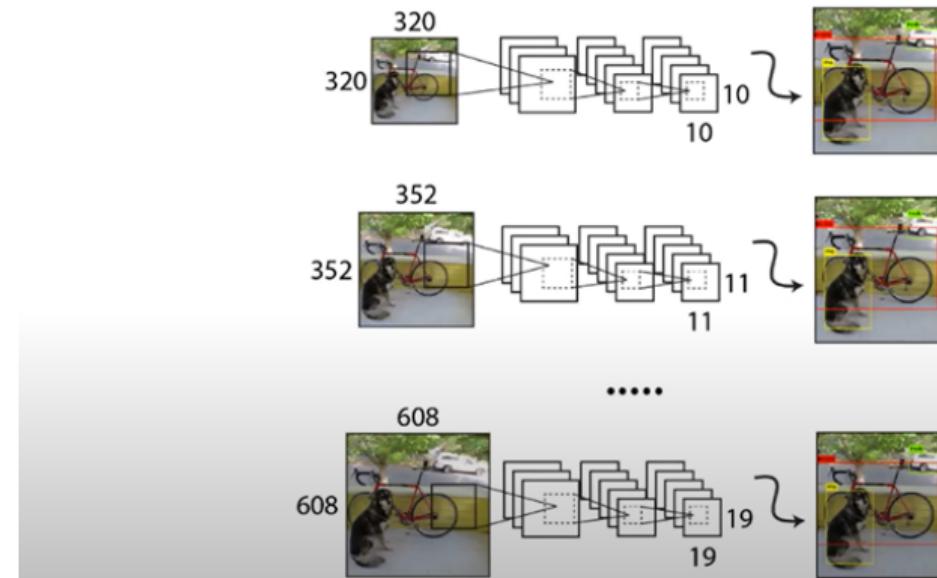


Figure 15: Cambio de tamaños

Multi-Scale Training.

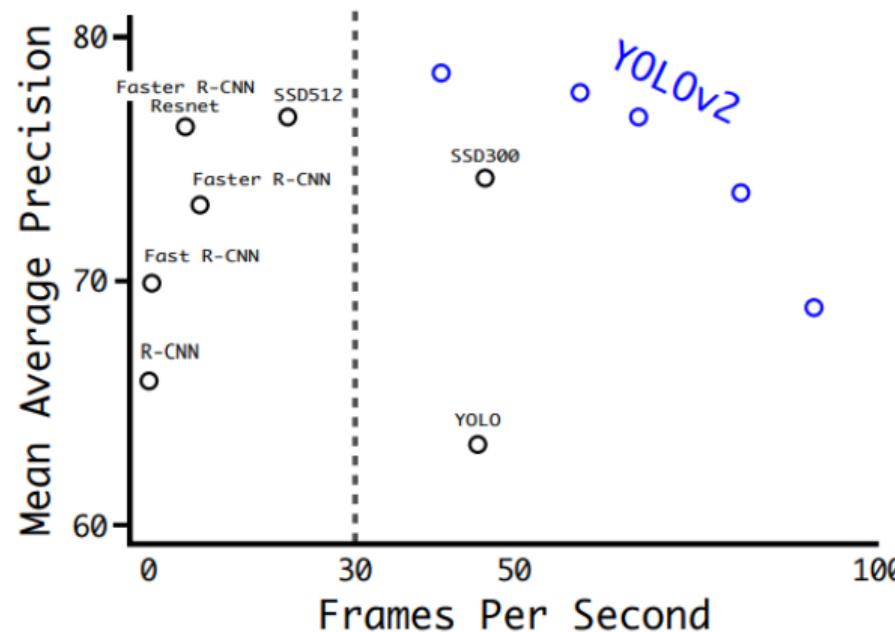


Figure 16: Comparaciones de modelos

Arquitectura Dark-net19.

- ▶ Se basa en VGG
- ▶ Usa filtros de 3x3.
- ▶ Se usa el tamaño 1x1 para comprimir las características.
- ▶ 19 capas de convoluciones y 5 capas de maxpooling

Arquitectura Dark-net19.

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

STRONGER.

- ▶ Detección y clasificación de datos
- ▶ Usar el loss-function y backpropagate
- ▶ Une los dataset COCO con ImageNet.

STRONGER.

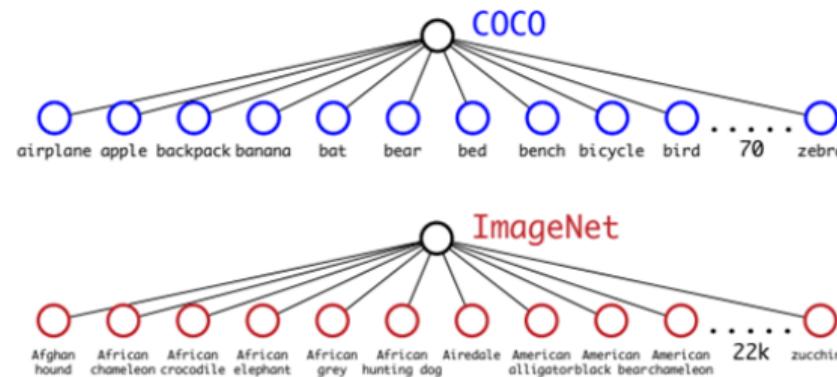


Figure 18: COCO - ImageNet Joseph Redmon (2016a)

STRONGER.

Combinarlo así no parece buena idea.

Can't just mash classes together...

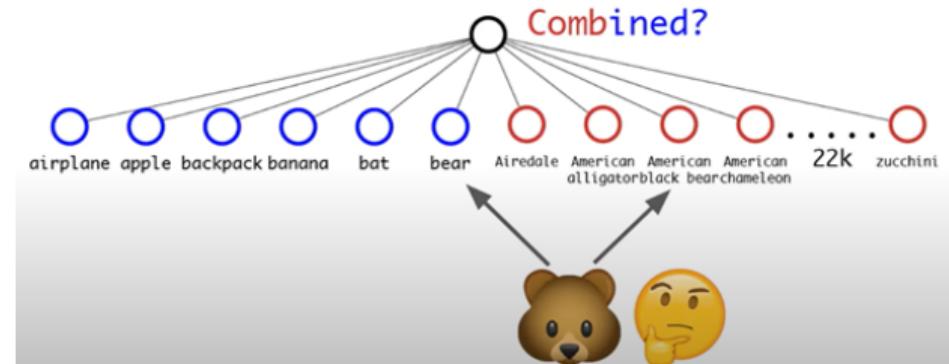


Figure 19: COCO-ImageNet mezclados

STRONGER.

Solución este problema surge el WordTree

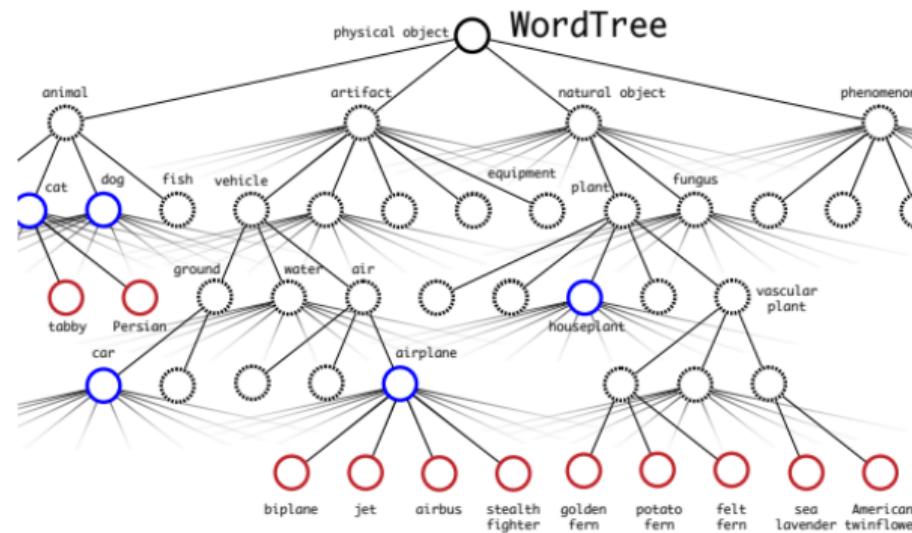


Figure 20: WordTree Joseph Redmon (2016a)

STRONGER.

El Softmax se agranda

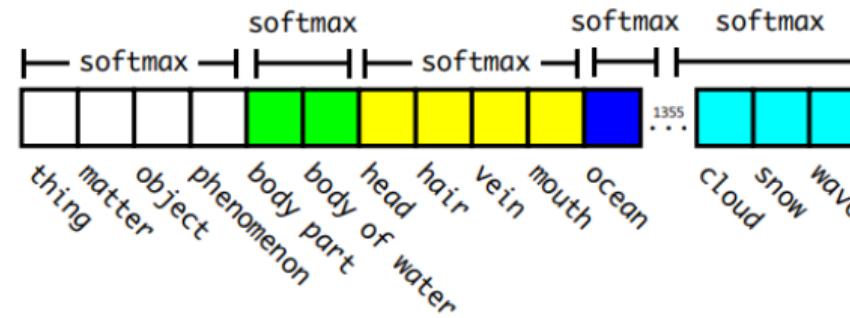


Figure 21: Softmax Joseph Redmon (2016a)

Overview

1 YOLO1

2 YOLO2

3 YOLOv3

- Darknet-53
- Detección a tres escalas
- Otros pequeños cambios
- Benchmarking

YOLOv3

En su momento, YOLO 9000 o YOLO2 fue el algoritmo más rápido y también uno de los más precisos. Sin embargo, un par de años después ya no es el más preciso con algoritmos como RetinaNet y SSD superándolo en términos de precisión. Todavía, sin embargo, fue uno de los más rápidos.

Pero esa velocidad se cambió por mejoras en la precisión en YOLO v3. Mientras que la variante anterior se ejecutaba a 45 FPS en un Titan X, YOLOv3 registra alrededor de 30 FPS. Esto tiene que ver con el aumento de la complejidad de la arquitectura subyacente llamada Darknet.

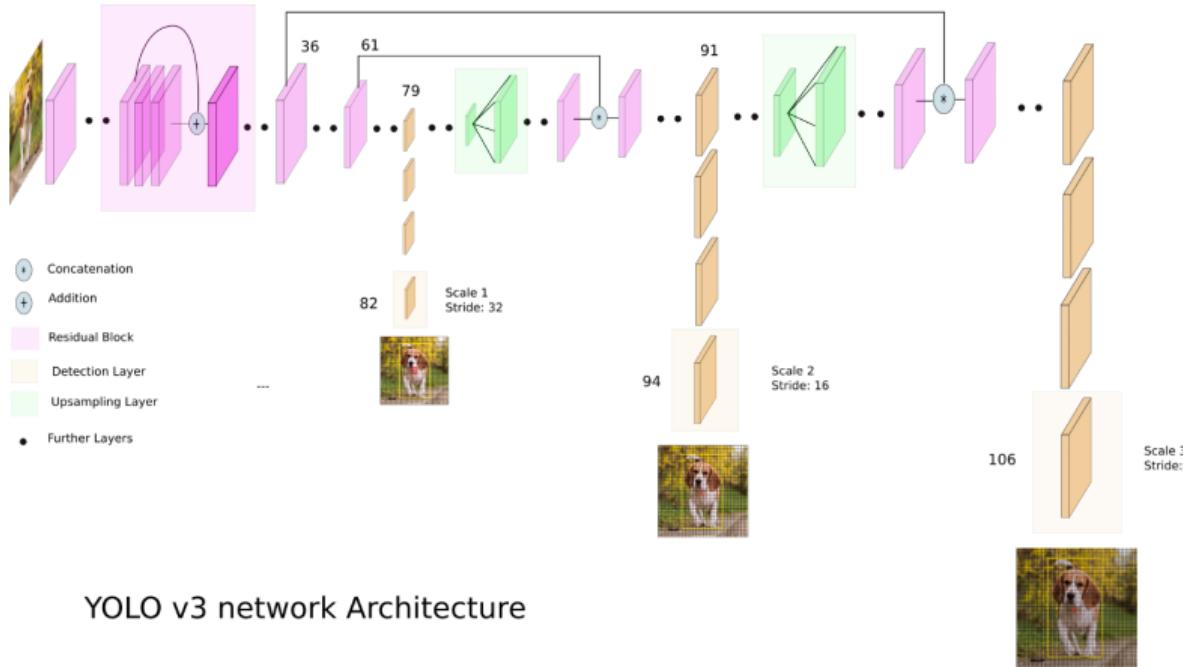
Darknet-53

YOLO v3 usa una variante de Darknet, que originalmente tiene una red de 53 capas entrenada en Imagenet. Para la tarea de detección, se apilan 53 capas más, lo que nos brinda una arquitectura subyacente totalmente convolucional de 106 capas para YOLO v3.

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

Figure 22: Darknet layers Redmon and Farhadi (2018)

Darknet-53



YOLO v3 network Architecture

Figure 23: Arquitectura darknet Kathuria

Detección a tres escalas

En YOLO v3, la detección se realiza mediante la aplicación de un kernel de detección 1×1 en mapas de características de tres tamaños diferentes en tres lugares diferentes de la red.

La forma del núcleo de detección es $1 \times 1 \times (B \times (5 + C))$. Aquí B es el número de bounding boxes que puede predecir una celda en el mapa de características, "5" es para los 4 atributos del bounding box y la confianza de un objeto, y C es el número de clases. En YOLO v3 entrenado en COCO, B = 3 y C = 80, por lo que el tamaño del kernel es $1 \times 1 \times 255$.

Detección a tres escalas

Image Grid. The Red Grid is responsible for detecting the dog

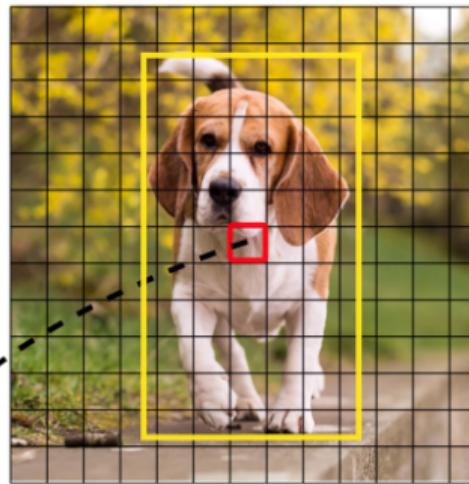


Figure 24: Deteccion en 3 escalas 01 Kathuria

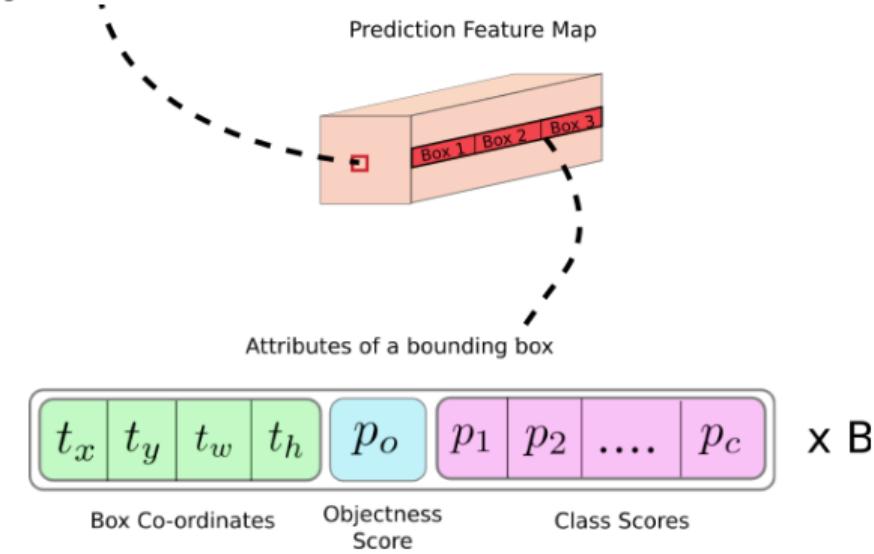
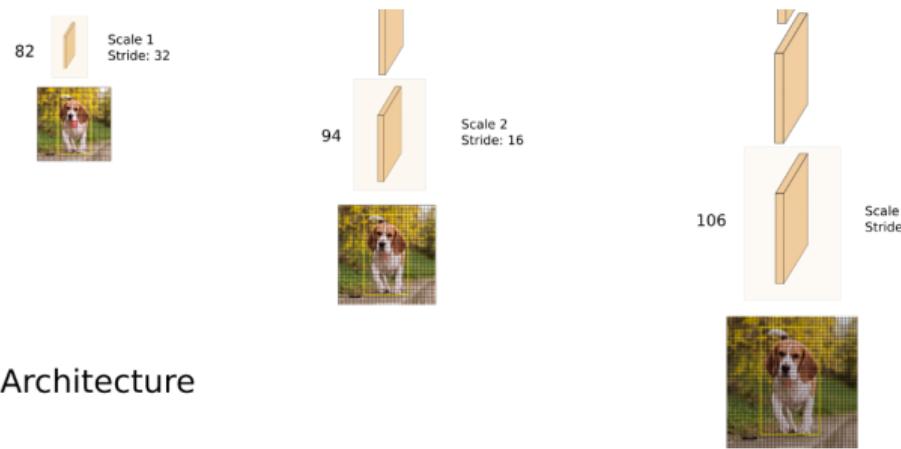


Figure 25: Deteccion en 3 escalas 02 Kathuria

Otros pequeños cambios

► Mejor en la detección de objetos más pequeños

- ▶ Las capas upsampling concatenadas con las capas anteriores ayudan a preservar las características de grano fino que ayudan a detectar objetos pequeños.
- ▶ La capa de 13×13 es responsable de detectar objetos grandes, mientras que la capa de 52×52 detecta los objetos más pequeños, y la capa de 26×26 detecta objetos medianos.



Otros pequeños cambios

► Elección de cajas de anclaje

- ▶ YOLO v3, en total utiliza 9 cajas de anclaje. Tres para cada escala. Si se está entrenando a YOLO para otro dataset, debe usar el agrupamiento de K-Means para generar 9 anclas.
- ▶ Luego, organice los anclajes en orden descendente de una dimensión. Asigne las tres anclas más grandes para la primera escala, las siguientes tres para la segunda escala y las últimas tres para la tercera.

Otros pequeños cambios

► Más bounding boxes por imagen

- ▶ Por ejemplo, con su resolución nativa de 416 x 416, YOLO v2 predijo $13 \times 13 \times 5 = 845$ cajas. En cada celda de la cuadrícula, se detectaron 5 cajas usando 5 anclas.
- ▶ Por otro lado, YOLO v3 predice cajas en 3 escalas diferentes. Y para 9 anchor boxes, es imaginar fácilmente por qué es más lento que YOLO v2.

Otros pequeños cambios

► Cambios en la función de pérdida

- ▶ Los errores al cuadrado en YOLO2, fueron reemplazados por términos de error de entropía cruzada. En otras palabras, las predicciones de clase y confianza de objetos en YOLO v3 se predican a través de la regresión logística.
- ▶ Mientras entrenamos al detector, para cada ground truth box, asignamos un bounding box, cuyo anchor box tiene la superposición máxima con el ground truth box.

Otros pequeños cambios

► No más softmaxing

- ▶ YOLO v3 ahora realiza una clasificación multietiqueta para los objetos detectados en las imágenes.
- ▶ Softmaxing classs se basa en la suposición de que las clases son mutuamente excluyentes, o en palabras simples, si un objeto pertenece a una clase, entonces no puede pertenecer a la otra.
- ▶ Cuando tenemos clases como Persona y Hombre en un conjunto de datos, la suposición anterior falla. Esta es la razón por la cual en YOLOv3 se han abstenido de suavizar las clases. En su lugar, cada puntaje de clase se predice mediante regresión logística y se usa un umbral para predecir múltiples etiquetas para un objeto.

Benchmarking

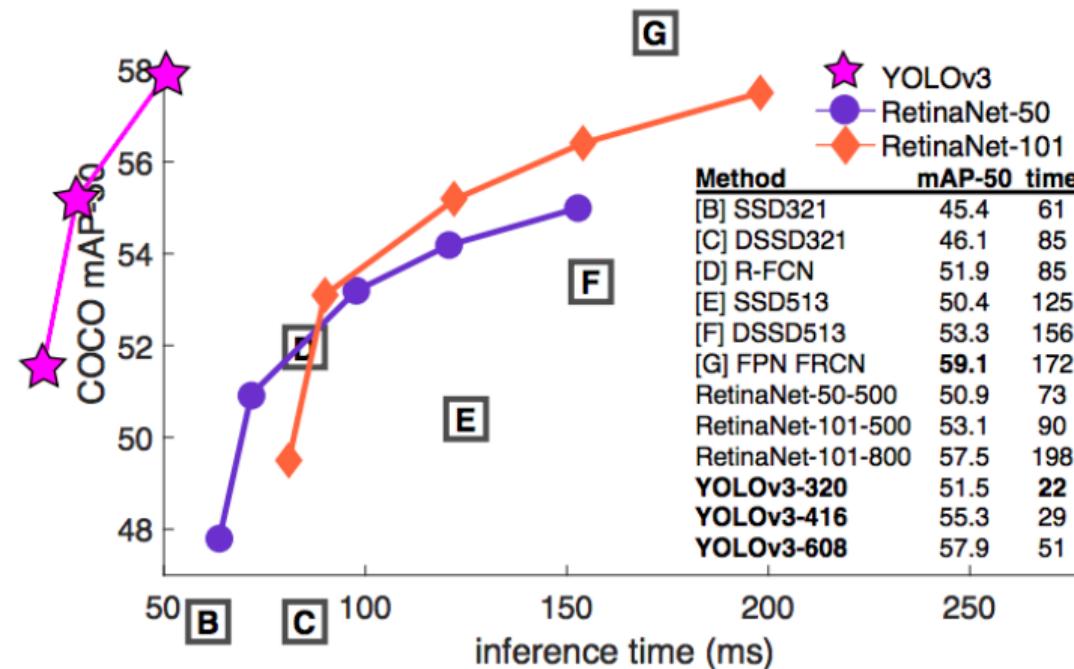


Figure 27: Graficas de resultados Redmon and Farhadi (2018)

Benchmarking

	backbone	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
<i>Two-stage methods</i>							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
<i>One-stage methods</i>							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	40.8	61.1	44.1	24.1	44.2	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

Figure 28: Tabla de resultados Redmon and Farhadi (2018)

References I

Ali Farhadi Joseph Redmon. Yolo9000: Better, faster, stronger. *Allen Institute for AI*, 2016a.

Ross Girshick Ali Farhadi Joseph Redmon, Santosh Divvala. You only look once: Unified, real-time object detection. 2016b. URL <https://arxiv.org/pdf/1506.02640v5.pdf>.

Ayoosh Kathuria. What's new in yolo v3? <https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b>.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. URL <https://arxiv.org/pdf/1804.02767.pdf>.