Detección de objetos I

Visión por computador

Juan Carlos Arbeláez
jarbel16@eafit.edu.co

https://modelcards.withgoogle.com/object-detection

Contenido

1. Introducción
2. Algoritmos
3. Métricas de evaluación
4. Aplicaciones
5. Retos
6. Taller

Parte 1

Parte 2

UNIVERSIDAD EAFIT

flaticon.com

Contenido

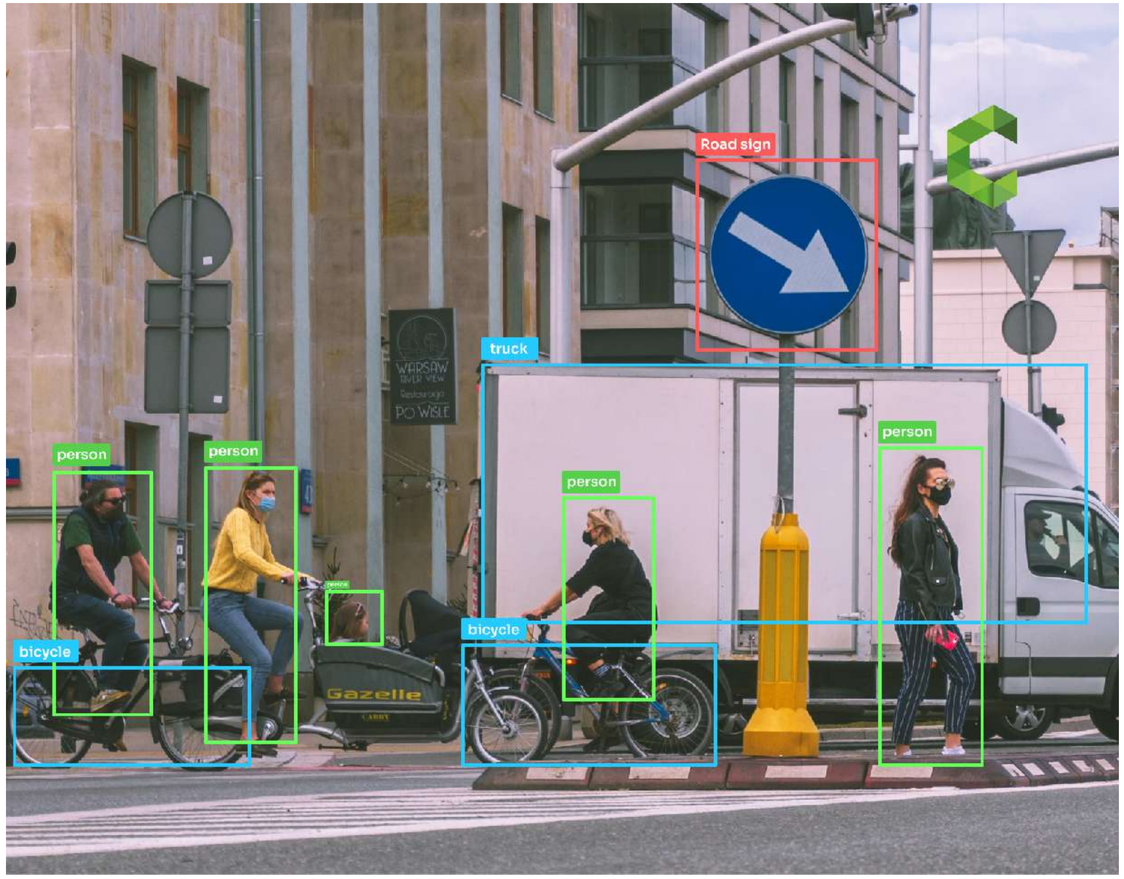
1. **Introducción**
2. Algoritmos
3. Métricas de evaluación
4. Aplicaciones
5. Retos
6. Taller

UNIVERSIDAD EAFIT

flaticon.com

¿Qué es Detección de Objetos?

- La detección de objetos es una tarea de visión artificial
- Se utiliza para detectar instancias de objetos visuales de determinadas clases
- El objetivo de la detección de objetos es desarrollar modelos que respondan a la pregunta: **¿Qué objetos y dónde?**



https://www.cameralyze.co/blog/object-detection-and-person-detection-in-computer-vision

UNIVERSIDAD EAFIT®

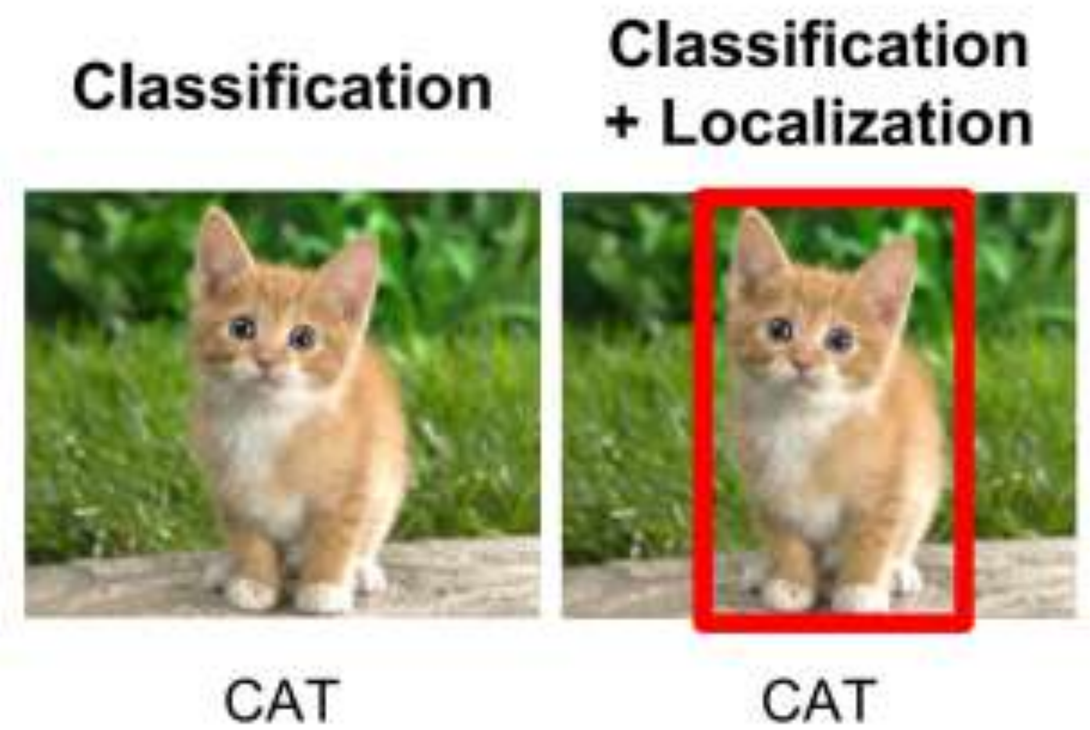
¿Qué es Detección de Objetos?

La clasificación de imágenes predecir la clase de un objeto en una imagen

La localización de objetos determinar el área (normalmente rectangular) que ocupa un objeto(s) en la imagen

La detección de objetos combina estas dos tareas y localiza y clasifica uno o más objetos en una imagen

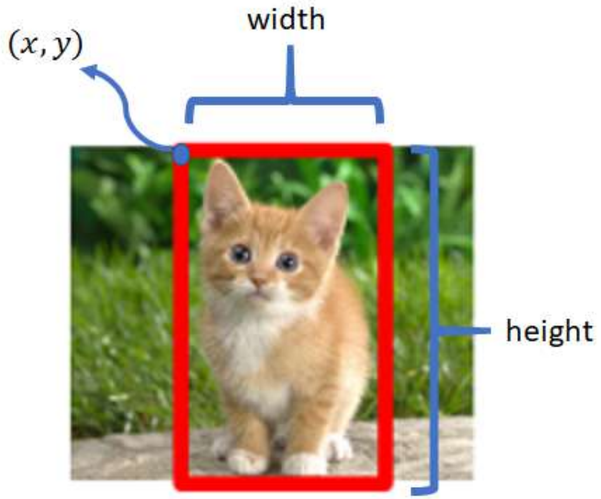
**Dos términos que se usan para el reconocimiento de objetos son: "Object detection" y "Object recognition"*



https://medium.com/analytics-vidhya/object-localization-using-keras-d78d6810d0be

¿Qué es Detección de Objetos?

¿Cuántas variables por objeto debe predecir un modelo de detección?



Localization (4) + Clases (N)

UNIVERSIDAD EAFIT

https://medium.com/analytics-vidhya/object-localization-using-keras-d78d6810d0be

¿Cómo son entrenados los modelos?

- El proceso de entrenamiento de un modelo para la *detección de objetos* es similar al *clasificación*
- Un conjunto de datos de detección de objetos vinculan una imagen con una lista de objetos que contiene y su ubicación
- El modelo acepta una imagen como entrada y devuelve una lista de predicciones por cada objeto con: ubicación (Coordenadas) y la clase

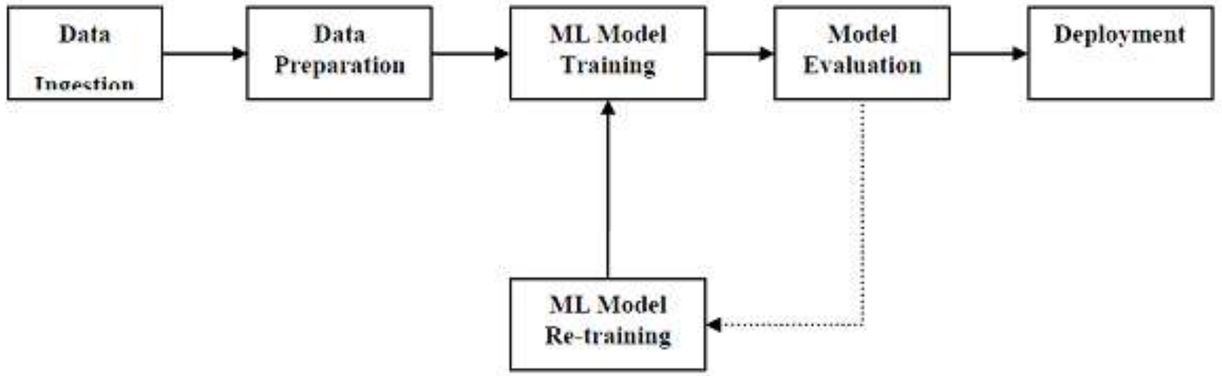


Diagrama de flujo de entrenamiento de modelos de ML

https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_pipelines_automatic_workflows.htm

Contenido

1. Introducción
2. **Algoritmos**
3. Métricas de evaluación
4. Aplicaciones
5. Retos
6. Taller

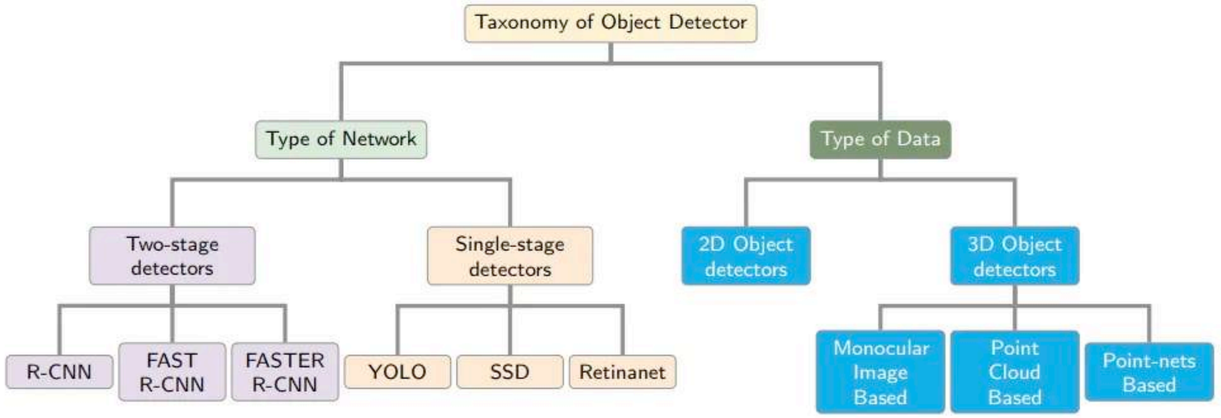
UNIVERSIDAD EAFIT

flaticon.com

Algoritmos

Hay muchos algoritmos para la detección de objetos, cada uno tiene sus pros y contras. Principales que se utilizan en la industria:

1. Faster R-CNN
2. SSD
3. YOLO
4. DETR



2022 - Object Detection in Autonomous Vehicles: Status and Open Challenges

Sliding Window

- Solución convencional: ventana deslizante para buscar en cada posición de la imagen
- Diferentes objetos o incluso el mismo tipo pueden tener diferentes relaciones de aspecto y tamaños dependiendo del tamaño del objeto y la distancia desde la cámara.
- Extremadamente lento si utilizamos CNN para la clasificación de imágenes en cada ubicación

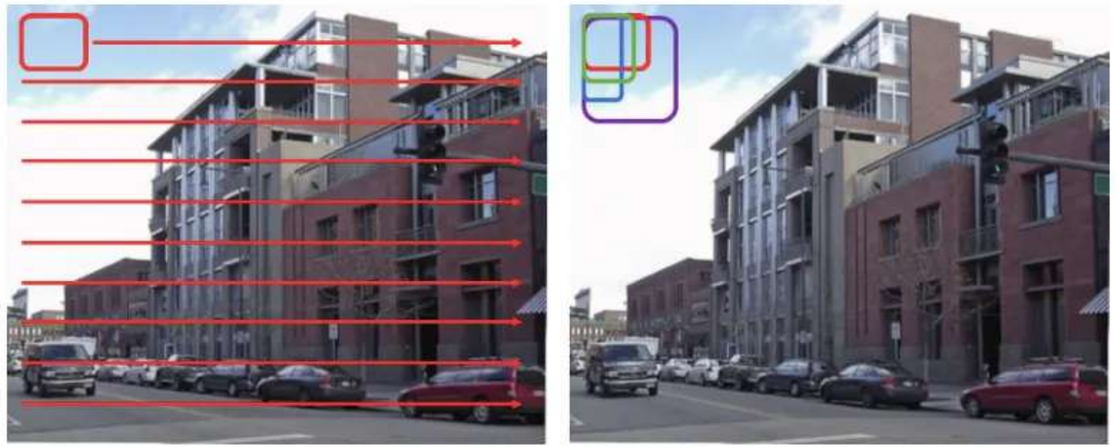


Illustration of Sliding Window (Left) with Different Aspect Ratios and Sizes (Right)

https://medium.com/analytics-vidhya/beginners-guide-to-object-detection-algorithms-6620fb31c375

Contenido

1. Introducción
2. Algoritmos
    1. **R-CNN**
    2. Faster R-CNN
    3. Single Shot Detector (SSD)
    4. You Only Look Once (Yolo)
    5. DETR
3. Métricas de evaluación
4. Aplicaciones
5. Retos
6. Taller

UNIVERSIDAD EAFIT

flaticon.com

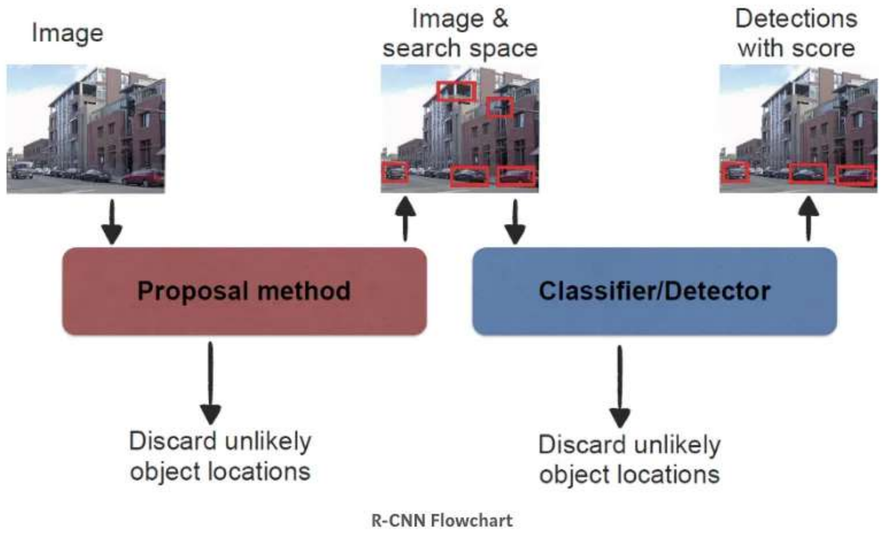
R-CNN

Region-based Convolutional Neural Network

Es un detector de dos etapas

1. Localizar regiones de interés
2. Clasificación de regiones

Dos implementaciones comúnmente usadas: Fast RCNN y Faster RCNN

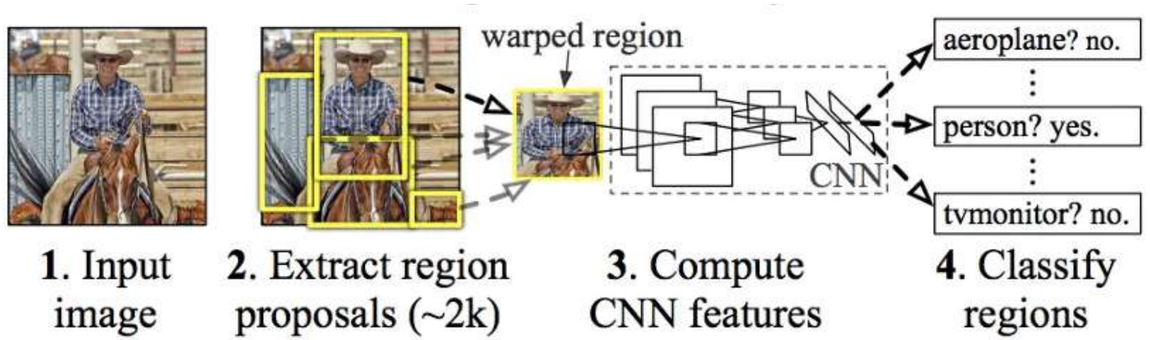


UNIVERSIDAD EAFIT

https://medium.com/analytics-vidhya/beginners-guide-to-object-detection-algorithms-6620fb31c375

R-CNN

A. Proponer regiones de interés: Extraemos las regiones con un algoritmo (búsqueda selectiva)
B. Clasificación: Redimensionar todos los regiones y pasarlas por una CNN



2014 - Rich feature hierarchies for accurate object detection and semantic segmentation

Selective Search

Es un algoritmo que se usa para la propuesta de regiones que agrupa regiones de forma jerárquica basada su similitud

UNIVERSIDAD EAFIT®

https://pyimagesearch.com/2020/06/29/opencv-selective-search-for-object-detection/

Selective Search

1. Sobre-segmentar la imagen usando el algoritmo *Felzenszwalb & Huttenlocher*
2. Añade los recuadros de las partes segmentadas a la lista de propuestas de regiones
3. Agrupa los segmentos adyacentes en función de la similitud
.

Imagen original

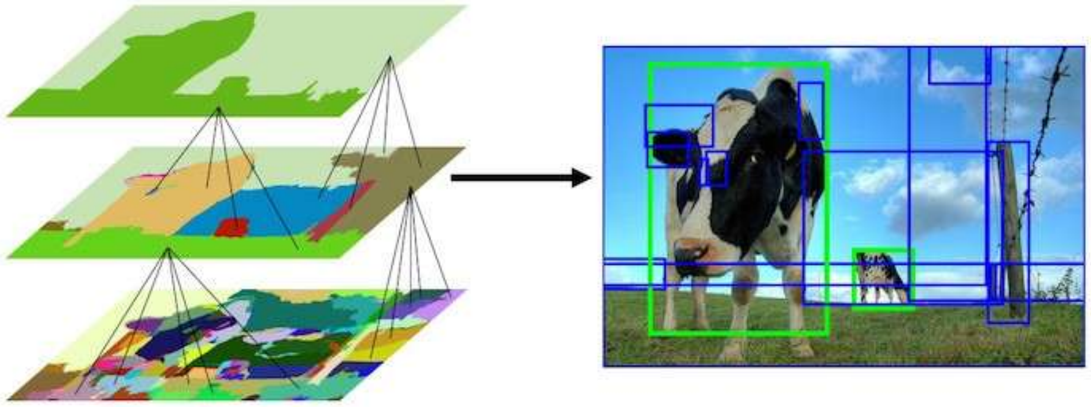
Over-segmented

UNIVERSIDAD EAFIT

2004-Efficient Graph-Based Image Segmentation

https://learnopencv.com/selective-search-for-object-detection-cpp-python/

Selective Search



- En cada iteración se forman segmentos más grandes y se añaden a la lista de propuestas de regiones
- Se crean propuestas de regiones de segmentos más pequeños a segmentos más grandes

UNIVERSIDAD EAFIT

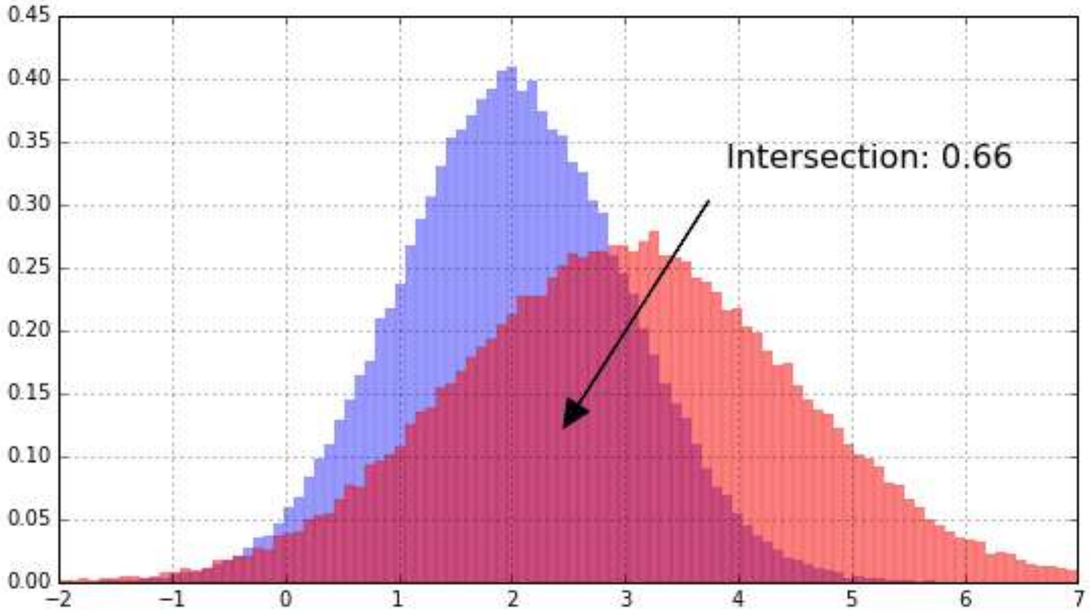
2013 - Selective Search for Object Recognition

Similitud entre recuadros

1. Intersección de histogramas de:
   - Color
   - Textura (derivados de color)
2. Tamaño: anima a las regiones más pequeñas a fusionarse antes
3. Compatibilidad de forma: lo bien que encajan dos regiones entre sí

Similitud final:

$$s(r_i, r_j) = \alpha_1 S_{color} + \alpha_2 S_{texture} + \alpha_3 S_{size} + \alpha_4 S_{shape}$$

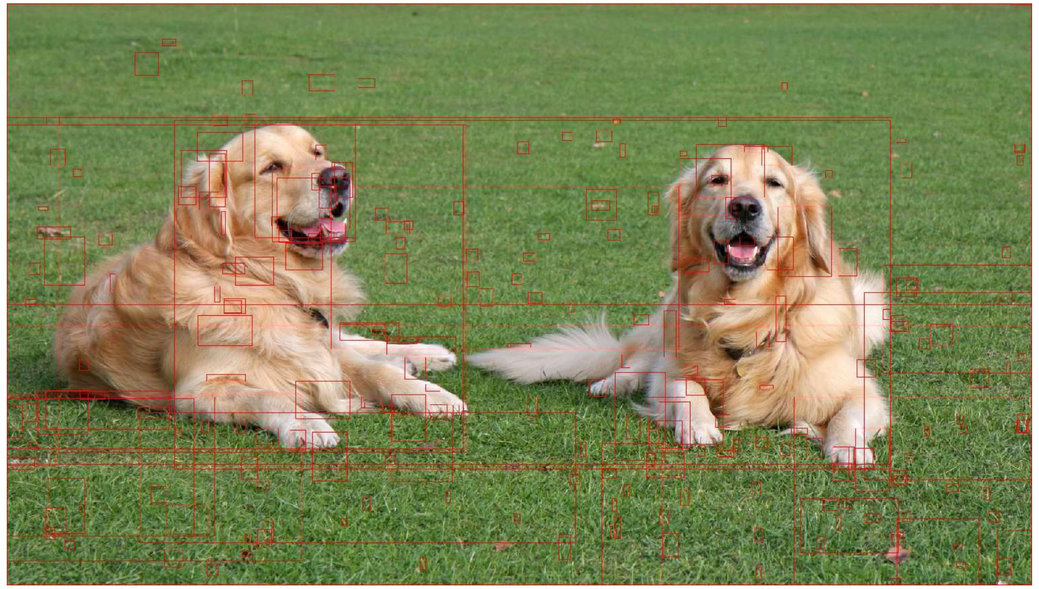


Intersección de histogramas

2013 - Selective Search for Object Recognition
https://blog.datadive.net/histogram-intersection-for-change-detection/

Selective Search

- Es necesario etiquetar cada una de las regiones propuestas para entrenar y medir el modelo de clasificación
- Asignar a qué clase pertenece dependiendo de su intersección con la etiqueta manual (IOU)



Dogs: top 250 region proposals

https://learnopencv.com/selective-search-for-object-detection-cpp-python/

UNIVERSIDAD EAFIT®

IOU

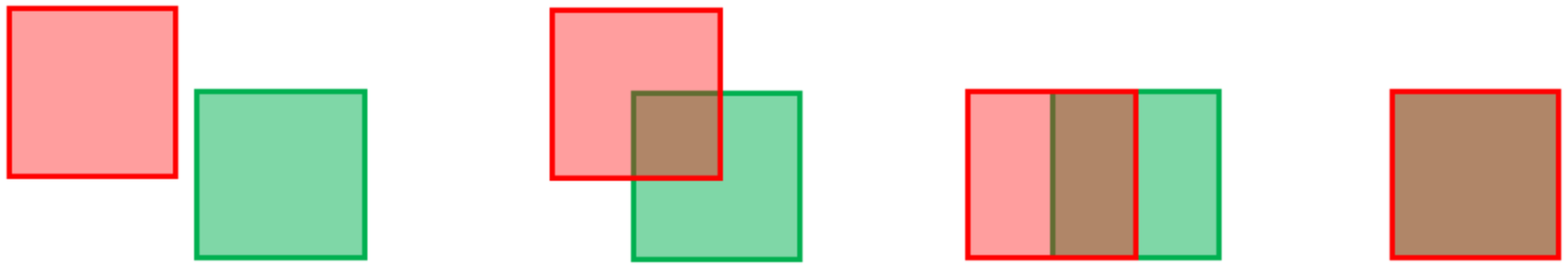
Intersection over Union

- Describe el grado de superposición de rectángulos
- Cuanto mayor sea la región de superposición: mayor será el IOU
- Se utiliza principalmente en aplicaciones relacionadas con la detección de objetos

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

UNIVERSIDAD EAFIT



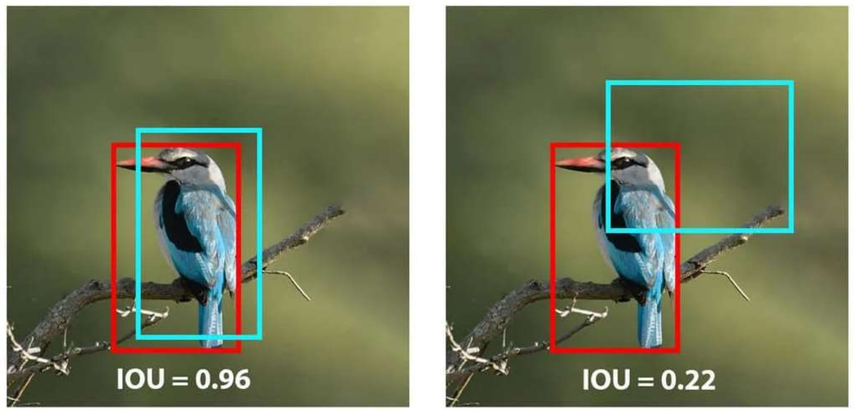
IOU (Intersection over Union)

UNIVERSIDAD EAFIT

https://www.baeldung.com/cs/object-detection-intersection-vs-union

Etiquetado de regiones

- Es necesario definir el valor del threshold sobre el cual considerar un recuadro como perteneciente a una clase
- En el artículo original usan “grid search” para definir este valor (*IOU* inferior a 0.3 se etiquetan como fondo)



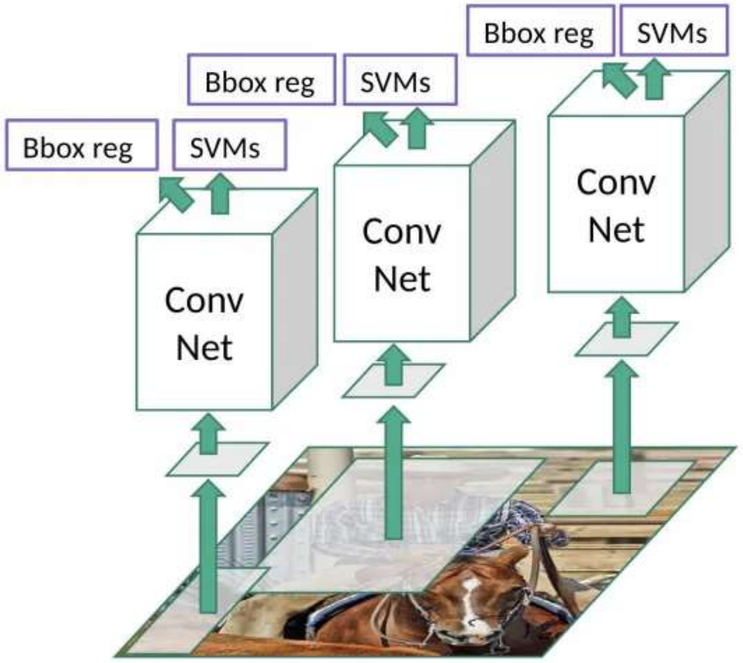
UNIVERSIDAD EAFIT

2014 - Rich feature hierarchies for accurate object detection and semantic segmentation
https://wikidocs.net/177706

Arquitectura

Clasificación de regiones

- Las regiones candidatas se deforman en un cuadrado y se introducen en una CNN que produce un vector de características como salida
- Las características extraídas se introducen en una *SVM* para clasificar la presencia del objeto
- También se predicen 4 valores de desplazamiento de las coordenadas del rectángulo para aumentar la precisión

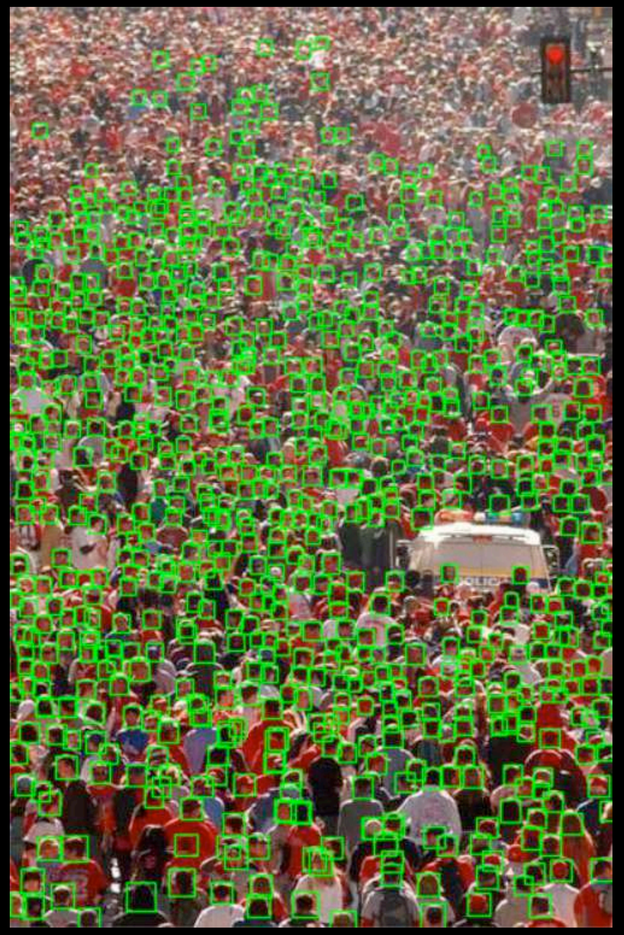


UNIVERSIDAD EAFIT

https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Problemas con R-CNN

- Alto tiempo de entrenamiento: clasificar cada región (~2000)
- No puede aplicarse en tiempo real (~47 segundos por cada imagen)
- No es posible mejorar el algoritmo de proposición de regiones, desacoplado del clasificador y sin ML



UNIVERSIDAD EAFIT®

https://github.com/ultralytics/yolov5/issues/1779

Contenido

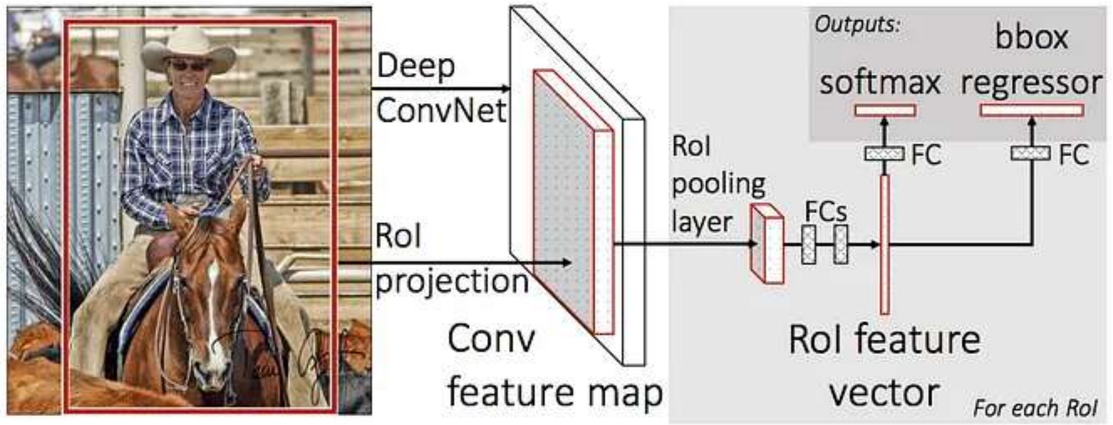
1. Introducción
2. Algoritmos
    1. R-CNN
    2. **Faster R-CNN**
    3. Single Shot Detector (SSD)
    4. You Only Look Once (Yolo)
    5. DETR
3. Métricas de evaluación
4. Aplicaciones
5. Retos
6. Taller

UNIVERSIDAD EAFIT

flaticon.com

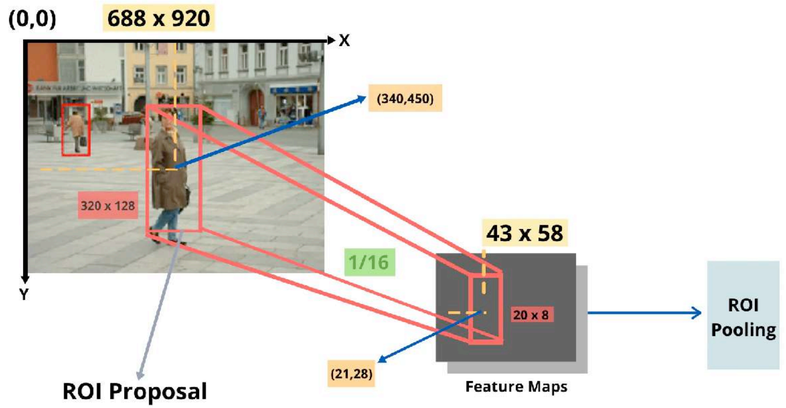
Fast-RCNN

- El mismo autor resolvió algunos de los problemas para construir un algoritmo de detección de objetos más rápido y lo llamó *Fast R-CNN*
- Similar a R-CNN pero se usa un mapa de características de toda la imagen generado por una CNN



Proyección de la región de interés

- Proyectar las regiones propuestas sobre el mapa de características
- Una imagen de *688 x 920* se alimenta a una CNN con “subsampling” de *1/16,* el mapa de características resultante tiene un tamaño de 43 x 58
- Coordenadas son escaladas usando la transformación geométrica de escalado

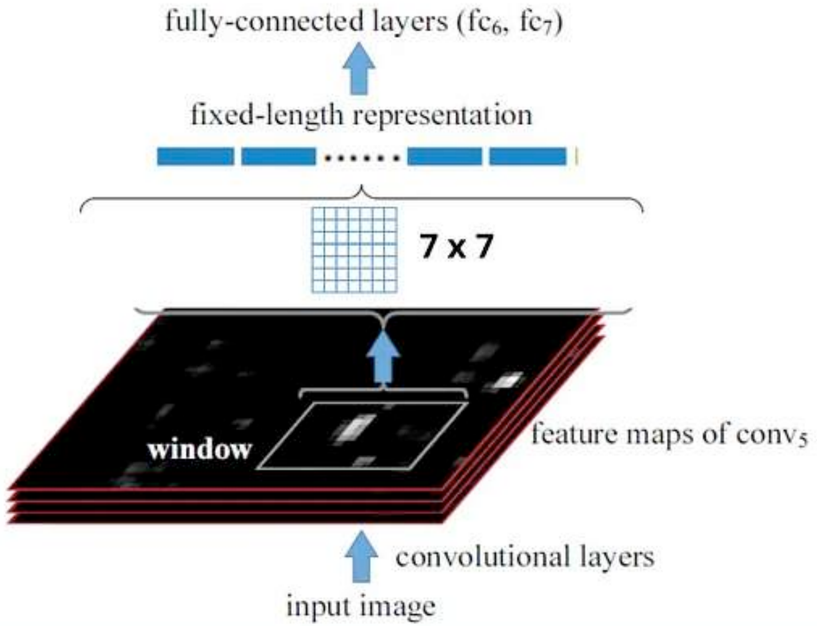


UNIVERSIDAD EAFIT

https://towardsdatascience.com/understanding-fast-r-cnn-and-faster-r-cnn-for-object-detection-adbb55653d97

Prediction Head

- Las capas FC de la cabeza de predicción esperan vectores de tamaño fijo pero las proyecciones ROI son de tamaños variables
- Para resolver este problema, los autores proponen “ROI Pooling”
- ROI Pooling: transforma en dimensiones fijas de las proyección de ROI del mapa de características

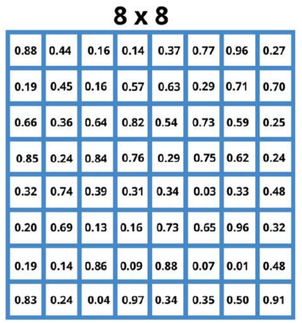


https://towardsdatascience.com/understanding-fast-r-cnn-and-faster-r-cnn-for-object-detection-adbb55653d97

ROI Pooling

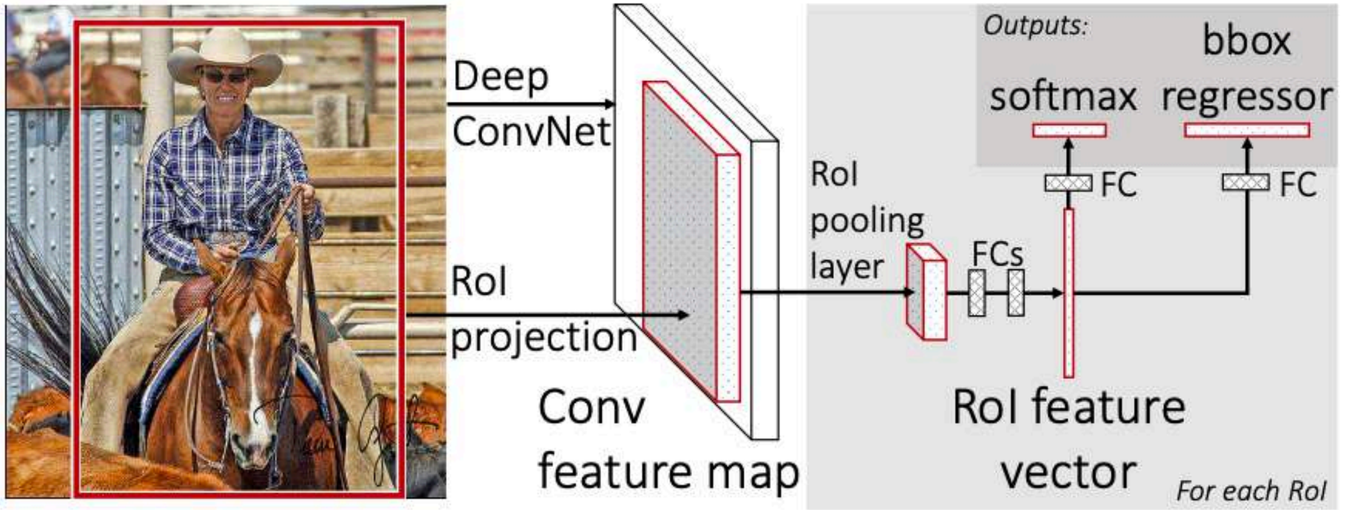
Region of Interest Pooling

- Produce mapas de características de tamaño fijo a partir de entradas no uniformes
- El mapa de características es dividido en dimensiones fijas y se aplica *max-pooling*



UNIVERSIDAD EAFIT

https://towardsdatascience.com/understanding-fast-r-cnn-and-faster-r-cnn-for-object-detection-adbb55653d97



Arquitectura Fast RCNN

UNIVERSIDAD EAFIT

2015 - Fast RCNN

Contenido

1. Introducción
2. Algoritmos
    1. R-CNN
    2. Faster R-CNN
    3. **Single Shot Detector (SSD)**
    4. You Only Look Once (Yolo)
    5. DETR
3. Métricas de evaluación
4. Aplicaciones
5. Retos
6. Taller

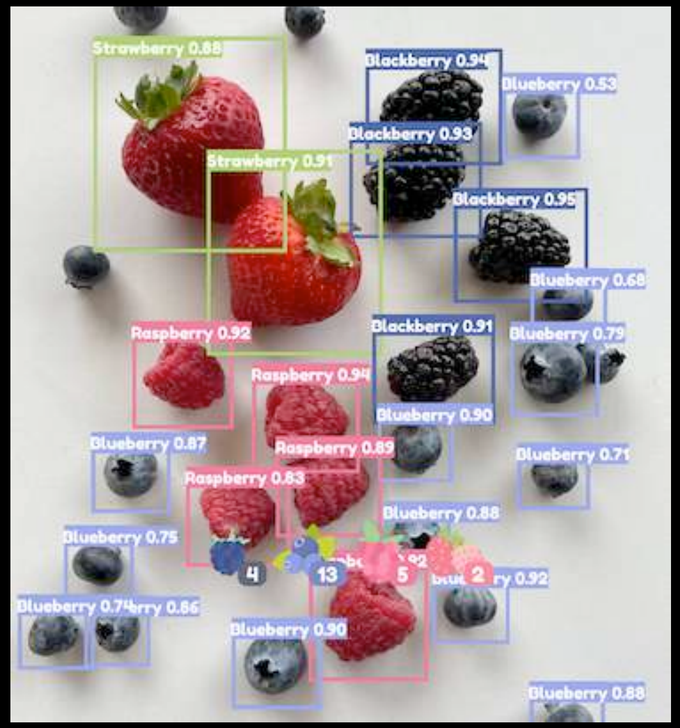
UNIVERSIDAD EAFIT

flaticon.com

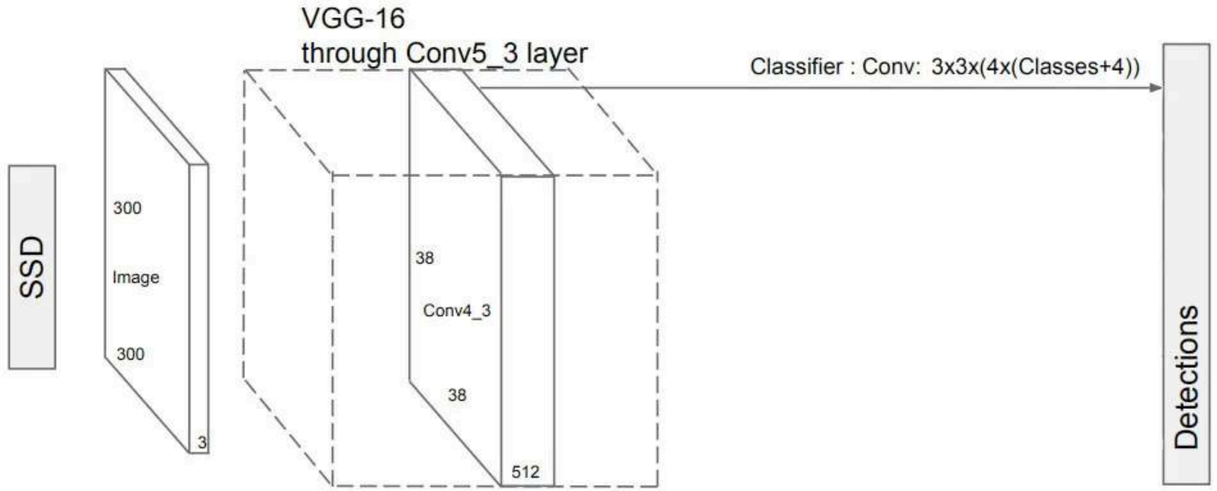
SSD

Single Shot Detector

- SSD está diseñado para la detección de objetos en tiempo real
- Es un detector de una sola etapa. (Fast R-CNN utiliza un *“region proposal”* y un clasificador CNN)
- SSD acelera el proceso eliminando el *“region proposal”*



https://docs.snap.com/lens-studio/references/templates/ml/multi-object-detection



SSD

Single Shot Detector

Componentes del modelo SSD:

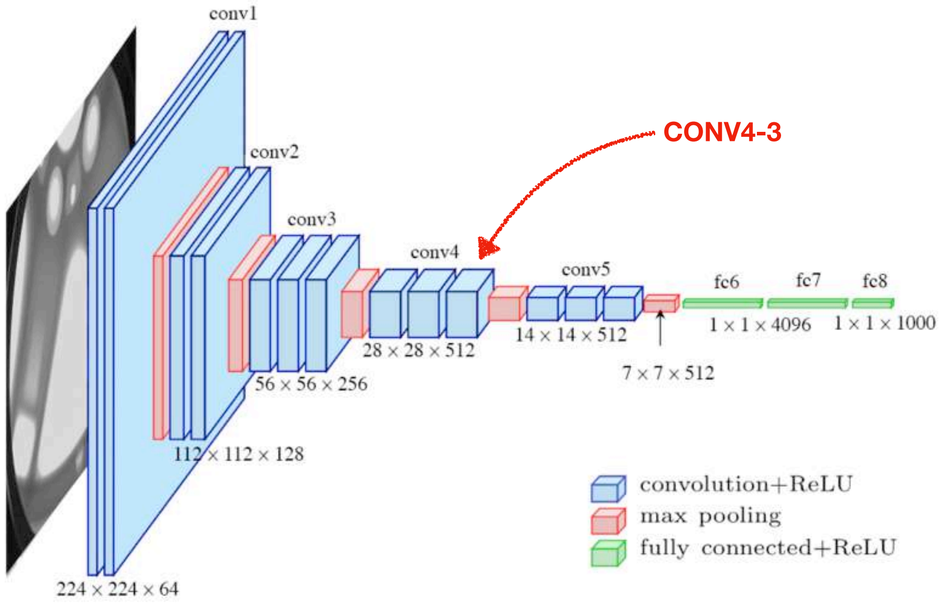
1. Extractor de características
2. CNN de detección

UNIVERSIDAD EAFIT

2016 - SSD: Single Shot MultiBox Detector

Extractor de Características

SSD Usa VGG16 para generar el mapa de características. Luego identifica los objetos usando los mapas desde CONV4-3



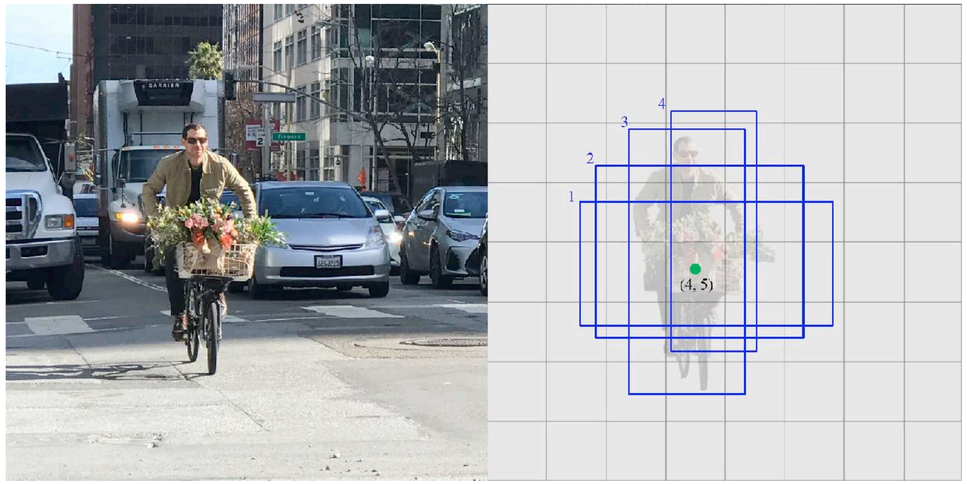
UNIVERSIDAD EAFIT

2017 - Automatic localization of casting defects with convolutional neural networks

Predicción

Single Shot Detector

- Por cada celda del mapa de características **predice *k* objetos**
- Cada predicción se compone de:
  A. Rectángulo (Bounding Box)
  B. El vector correspondiente a la clase del BB + una clase de no objeto



Predicción de detección por cada celda del mapa de características

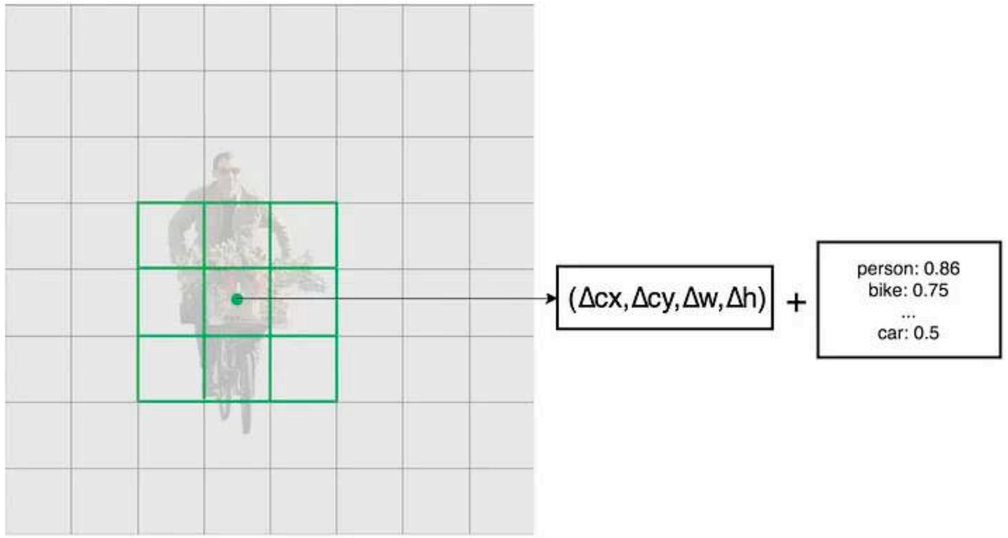
UNIVERSIDAD EAFIT

https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

Predicción

Single Shot Detector

- Por cada “celda” del mapa predice k objetos (rectángulos más la clase)
- Cada k corresponden a rectángulos de distintos ratios y tamaños
- Un rectángulo vertical es más adecuado para una persona y uno horizontal para un vehículo
- Aplica una capa de filtros convolucionales de 3x3 con stride de 1 sobre el mapa de características



Predicción de detección por cada celda del mapa de características

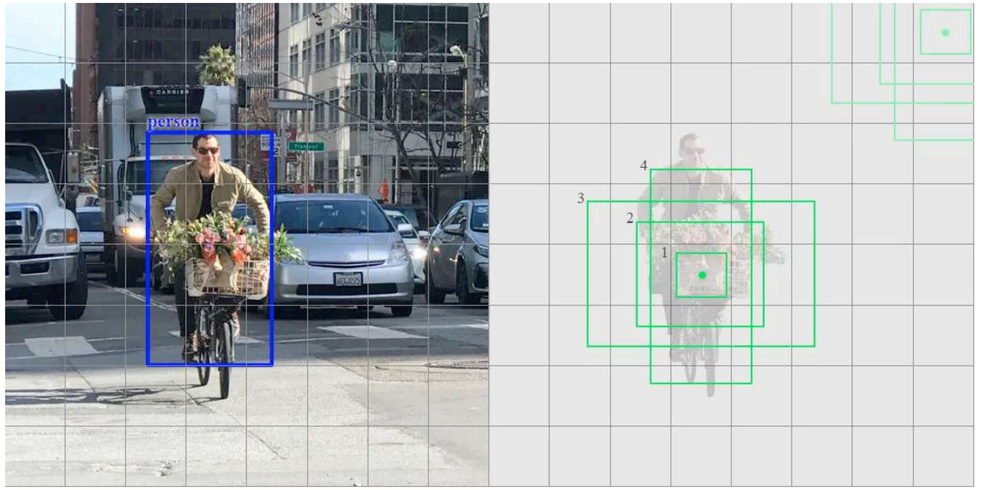
https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

UNIVERSIDAD EAFIT®

Predicción

Single Shot Detector

- El mapa de características es de $m \times n$
- Por cada celda predice k número de objetos (aspect ratios)
- Cada k está compuesto de:
  - c puntajes de la clase
  - 4 offsets relativos al rectángulo base



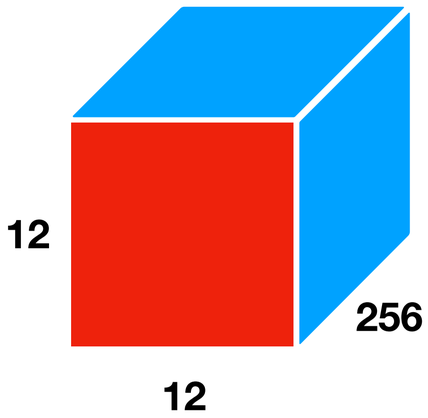
UNIVERSIDAD EAFIT

https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

Predicción

Single Shot Detector

- Si tenemos un mapa de características de 12x12
- Vamos a predecir 10 clases
- Y tenemos 2 rectángulos base (2 aspect ratio)
- ¿Cuántos filtros debe tener la capa convolucional de detección?



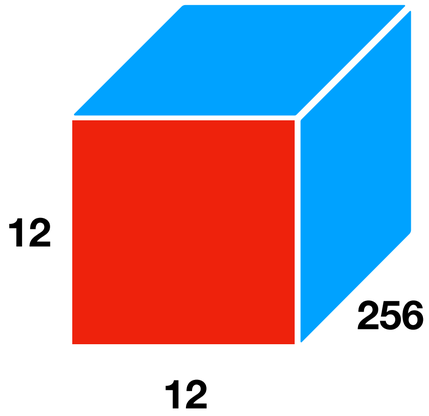
Predicción

Single Shot Detector

Número de filtros de la capa de predicción

$$(c + 5) \times k$$

$$(10 + 5) \times 2 = 30$$

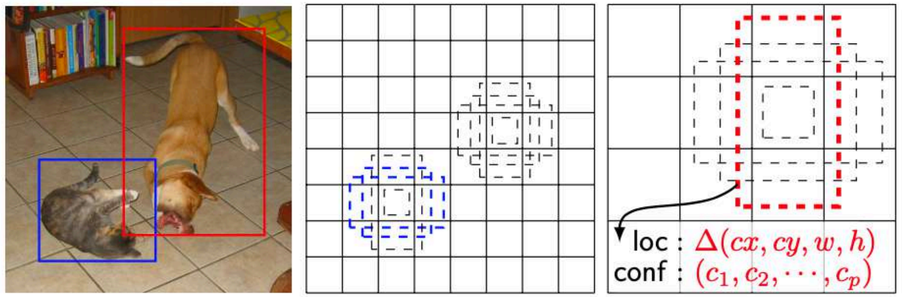


UNIVERSIDAD EAFIT

Múltiples escalas

Single Shot Detector

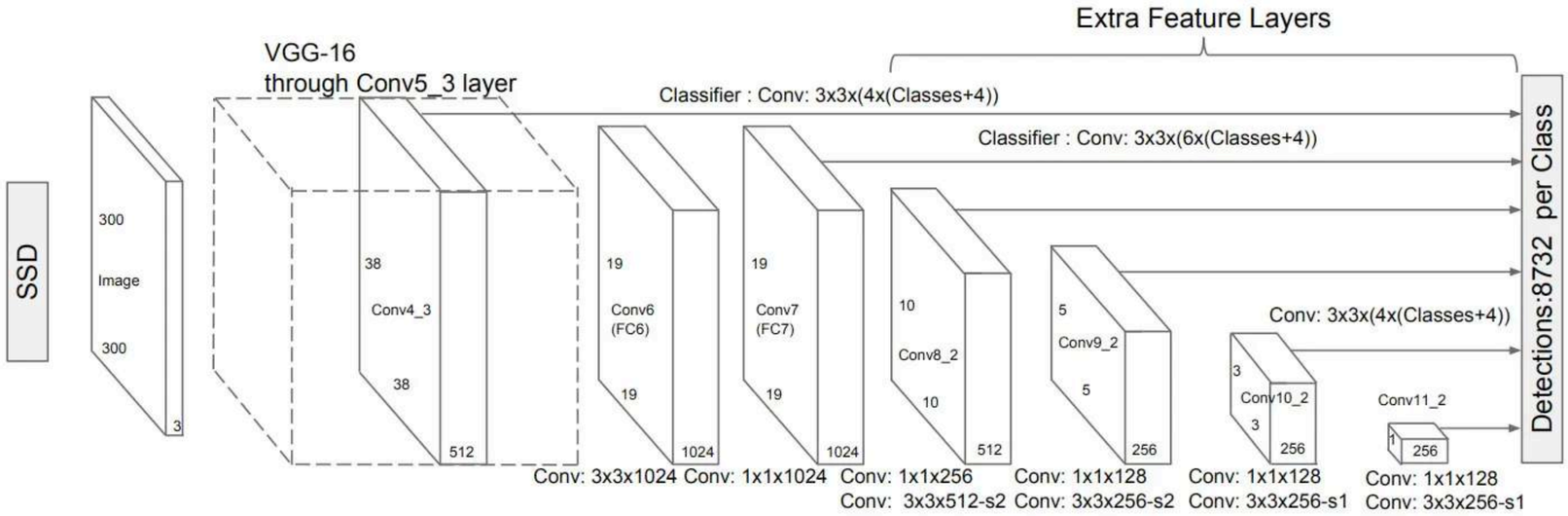
- VGG16 reduce gradualmente la dimensión espacial, la resolución de los mapas de características también disminuye
- SSD utiliza capas de menor resolución para detectar objetos a mayor escala



Los mapas de características de menor resolución (derecha) detectan objetos de mayor escala.

SSD

Arquitectura



SSD añade 6 capas más a la VGG16 que produce
k predicciones de objetos por capa

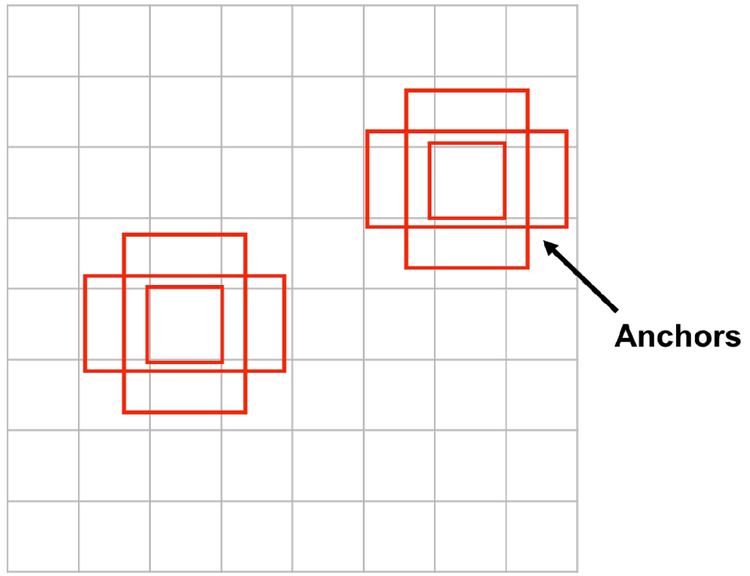
UNIVERSIDAD EAFIT

2016 - SSD- Single Shot MultiBox Detector

Default boxes & aspect ratios

Single Shot Detector

- Durante el entrenamiento el modelo puede luchar entre sí para determinar qué formas (personas o vehículos) deben optimizarse
- El entrenamiento inicial puede ser muy inestable. Las predicciones de los rectángulos que funcionan bien para una categoría para otras no
- Se quiere que las predicciones iniciales sean diversas y no se parezcan

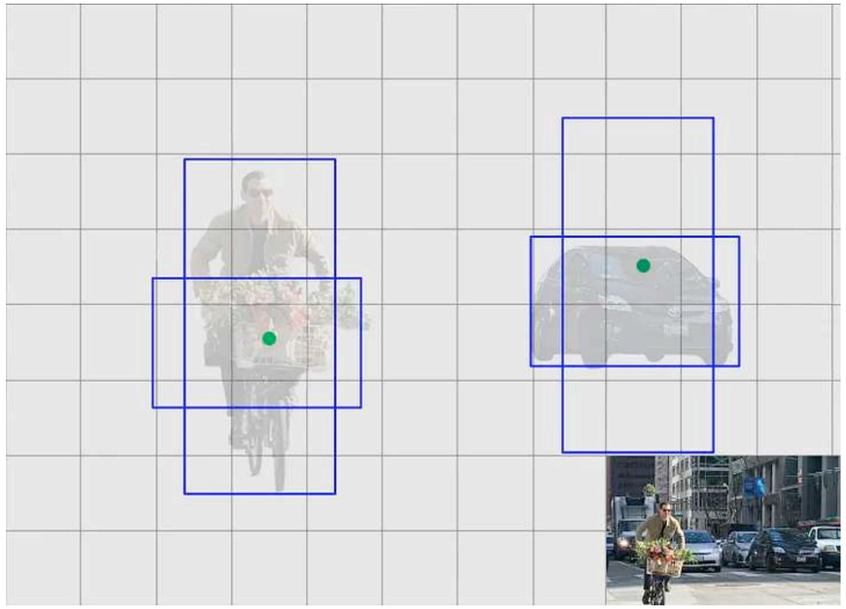


UNIVERSIDAD EAFIT

http://kimberlyfessel.com/algorithms/literature%20reviews/object-detection-challenges/

Default boxes and aspect ratios

- Se usan recuadros predeterminados para cubrir el espectro de los objetos a predecir y reducir la complejidad computacional
- SSD sugiere 4 o 6 rectángulos por defecto
- Las diferentes capas utilizan diferentes rectángulos por defecto para detecciones de diferentes tamaños



UNIVERSIDAD EAFIT

https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

Default boxes and aspect ratios

- SSD define un valor de escala para cada capa del mapa de características
- Se calcula el tamaño del rectángulo usando la escala y los aspect ratios pre definidos
- Las escalas van desde 0.2 a 0.9 (última capa)
- SSD propone 5 aspect ratios: 1, 2, 3, 1/2, y 1/3

Cálculo del ancho

$$w = s\sqrt{a}$$

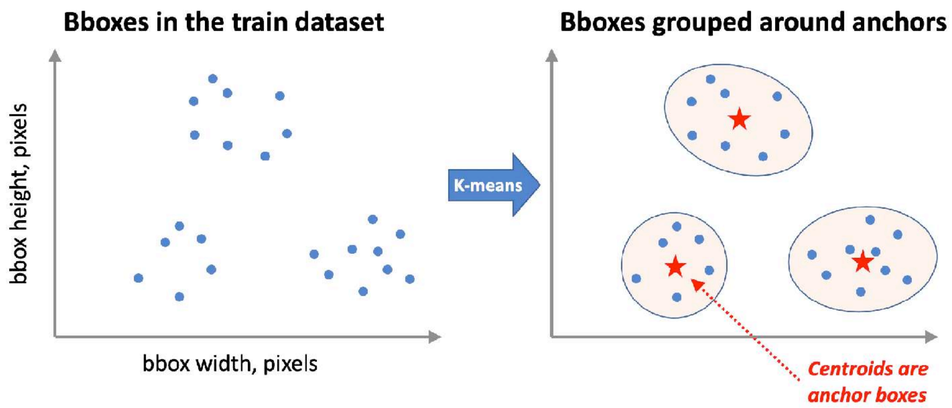
Calculo del alto

$$h = \frac{s}{\sqrt{a}}$$

En donde s es la escala y a el *“aspect ratio”*

Default boxes and aspect ratios

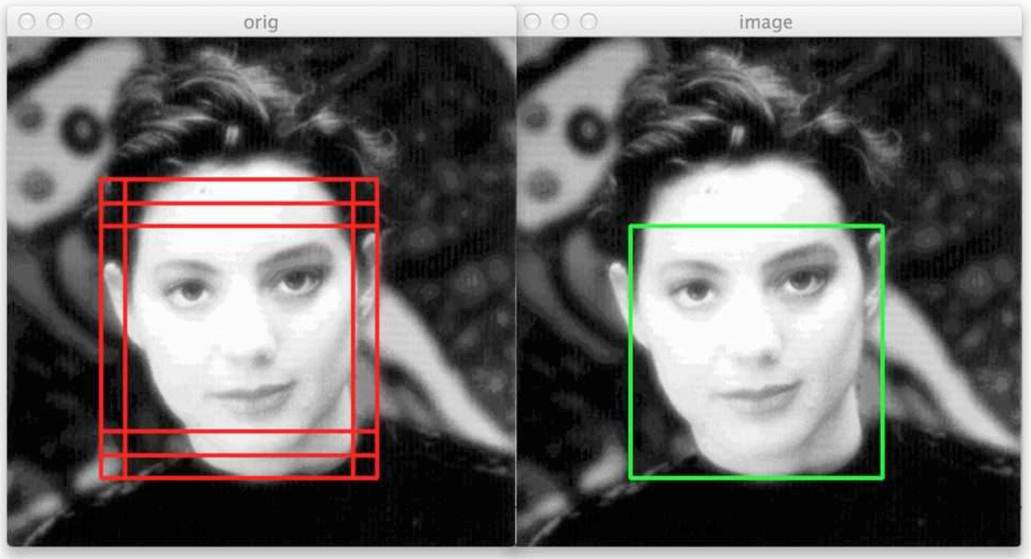
- Los *Bounding Box* también pueden ser definidos por agrupamiento (k-means)
- Se usan las etiquetas de entrenamiento para el calculo de los BB
- Como calcular los BB con K-means



https://towardsdatascience.com/training-yolo-select-anchor-boxes-like-this-3226cb8d7f0b

Predicción final

- Luego de la predicción se tienen múltiples rectángulos que se solapan
- Se realiza un paso de ***non-maximum suppression*** para producir el resultado final

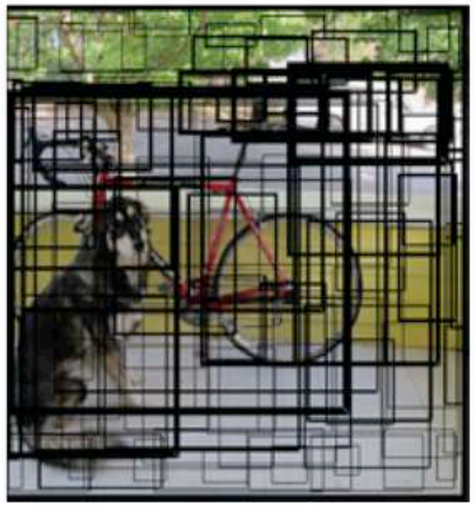


https://pyimagesearch.com/2015/02/16/faster-non-maximum-suppression-python/

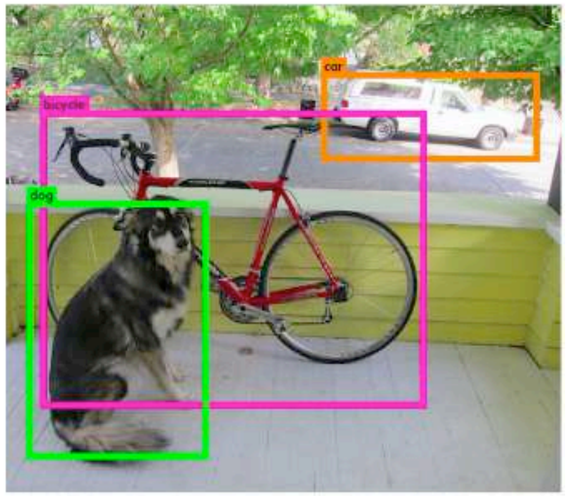
UNIVERSIDAD EAFIT

Non-maximum Suppression

- La primera etapa tiene una alta sensibilidad para predecir todas las posibles regiones, esto da paso a tener cientos de regiones similares
- Non-maximum Suppression es una técnica para “fusionar” rectángulos del mismo objeto



Multiple Bounding Boxes

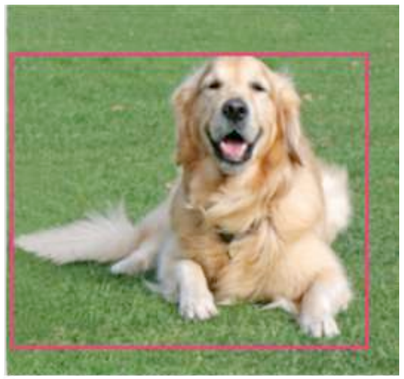


Final Bounding Boxes

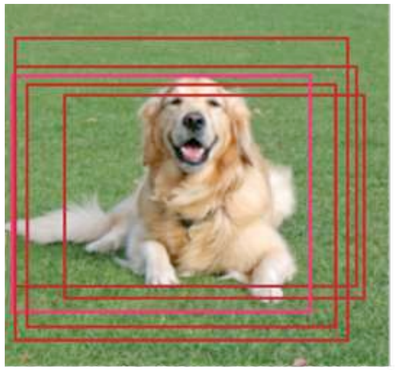
https://www.analyticsvidhya.com/blog/2020/08/selecting-the-right-bounding-box-using-non-max-suppression-with-implementation/

Non-maximum Suppression

1. Seleccione la propuesta con mayor *score*, elimínela de B y añádala a la lista final de propuestas D (Inicialmente D está vacía)
2. Calcula el IOU de la propuesta con los otros rectánculos en B. Si el IOU es mayor que el umbral N (*Y es la misma clase) se elimina
3. Este proceso se repite hasta que no queden más propuestas en B



After applying non-maximum suppression



Predictions before NMS

UNIVERSIDAD EAFIT

https://livebook.manning.com/concept/deep-learning/object

Contenido

1. Introducción
2. Algoritmos
    1. R-CNN
    2. Faster R-CNN
    3. Single Shot Detector (SSD)
    4. **You Only Look Once (Yolo)**
    5. DETR
3. Métricas de evaluación
4. Aplicaciones
5. Retos
6. Taller

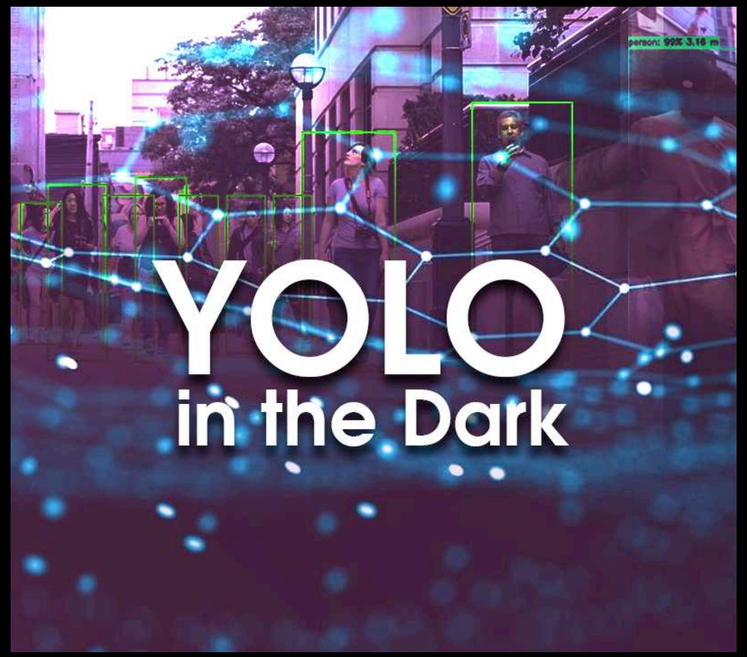
UNIVERSIDAD EAFIT

flaticon.com

Yolo

You Only Look Once

- Es uno de los algoritmos de detección más conocidos y usados
- Es un detector de una sola etapa
- Arquitectura basada en los clasificadores CNN



https://socionextus.com/pressreleases/deep-learning-method-for-object-detection-in-low-light-conditions/

Introducción

- YOLOv1 (2016): la primera de las muchas iteraciones por las que ha pasado esta arquitectura
- La idea básica de la arquitectura sigue siendo la misma. YOLOv1 se denomina simplemente YOLO (Actual v12)
- La clave es velocidad, predice a 45 frames por segundo: es una buena opción para aplicaciones en tiempo real

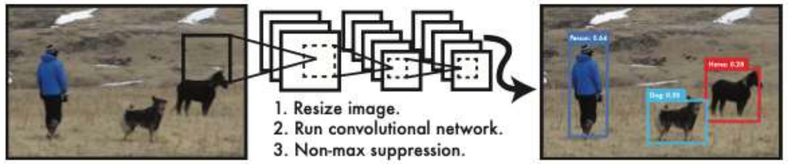


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448×448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

UNIVERSIDAD EAFIT

2016 - You Only Look Once Unified Real-Time Object Detection

Introducción

Yolo

- YOLO se basa en la idea de segmentar una imagen en partes más pequeñas
- La imagen se divide en una cuadrícula de dimensiones **S×S**
- La celda en la que se encuentra el centro de un objeto es la responsable de detectar ese objeto (centro del perro)

S

S

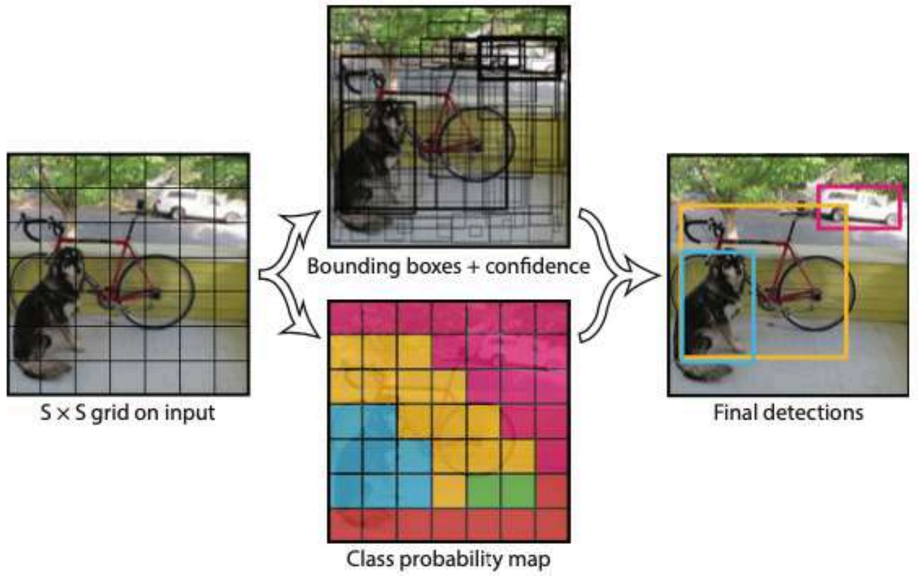
UNIVERSIDAD EAFIT

https://medium.com/analytics-vidhya/yolo-explained-5b6f4564f31

Predicciones

Yolo

- Cada celda predecirá ***K*** rectángulos y una puntuación de confianza para cada uno
- Estos niveles de confianza reflejan la certeza del modelo de que existe un objeto en esa celda
- La confianza representa la diferencia entre el IOU de la predicción y el rectángulo verdadero



UNIVERSIDAD EAFIT

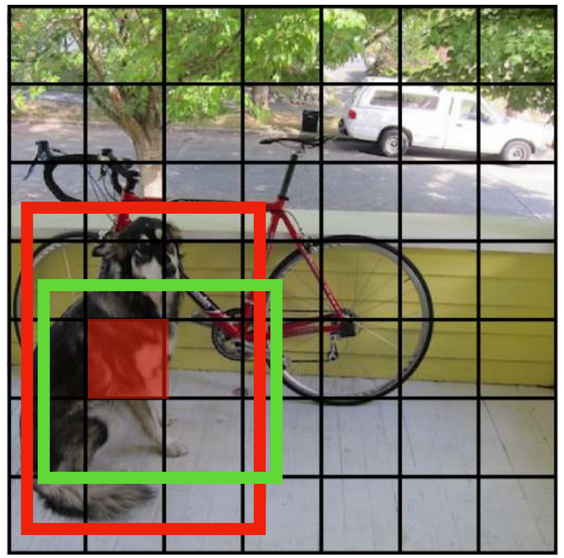
2016 - You Only Look Once Unified Real-Time Object Detection

Predicciones

Yolo

Además de rectángulos y la puntuación de confianza, cada celda predice la clase del objeto (on-hot)

¿Cuántas predicciones hace un modelo si tiene 5 clases y una cuadrícula de 7x7 y predice 2 rectángulos por cuadro?



7

7

UNIVERSIDAD EAFIT

2016 - You Only Look Once Unified Real-Time Object Detection

Predicciones

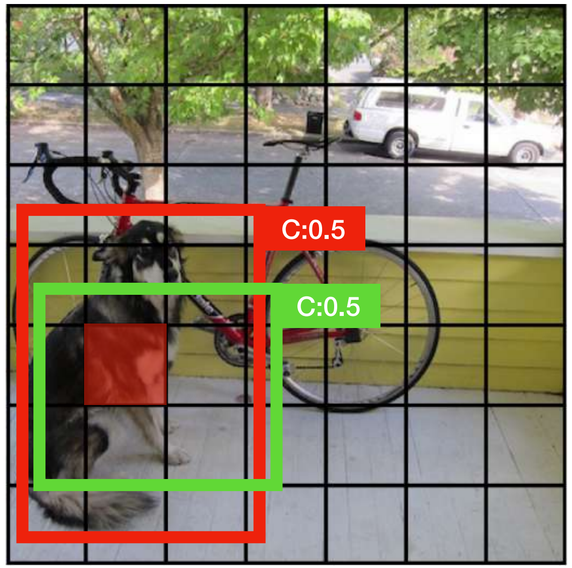
Yolo

7x7 x (1+4+5) x 2

CUADRÍCULA | BOUNDING BOX + CONFIABILIDAD + CLASES | PREDICCIONES

= 980

** El modelo predice el centro del recuadro con el ancho y alto en lugar de las posiciones de las esquinas superior izquierda e inferior derecha.



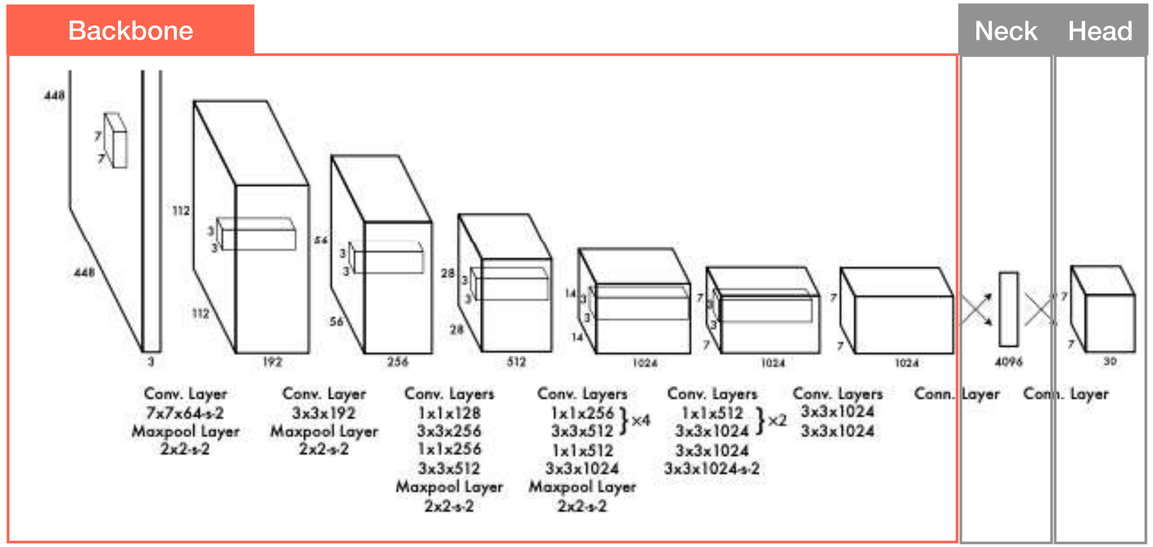
UNIVERSIDAD EAFIT

2016 - You Only Look Once Unified Real-Time Object Detection

Arquitectura

Backbone

- Arquitectura de YOLO tiene tres componentes (“head”, “neck”, y “backbone”) básicamente un modelo de clasificación
- El **“backbone”** encarga de la extracción de características y se entrena primero en un conjunto de datos de clasificación (ImageNet)
- Los “parches” corresponden a la reducción espacial del extractor de características

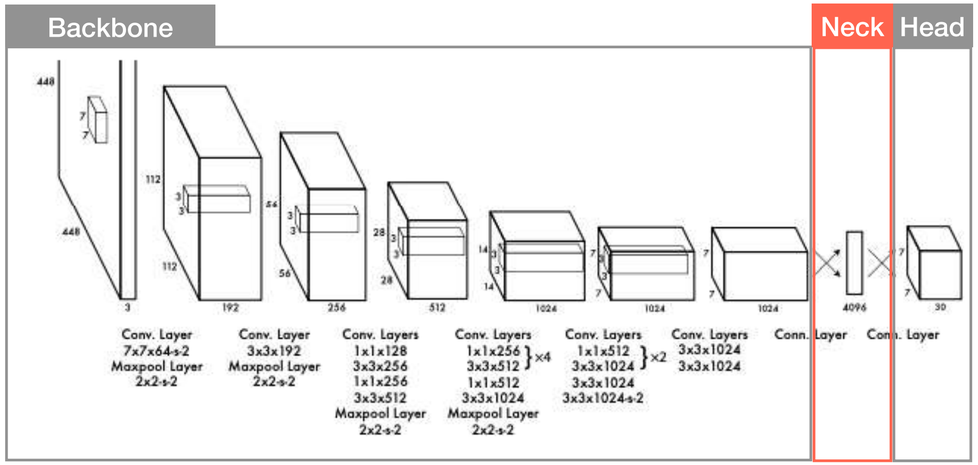


2016 - You Only Look Once Unified Real-Time Object Detection

Arquitectura

Neck

- El mapa de características se aplana y comprime en una representación latente mediante una capa totalmente conectada (FC)
- Produce un vector que contiene la información necesaria para las predicciones que posteriormente es utilizado por la cabeza



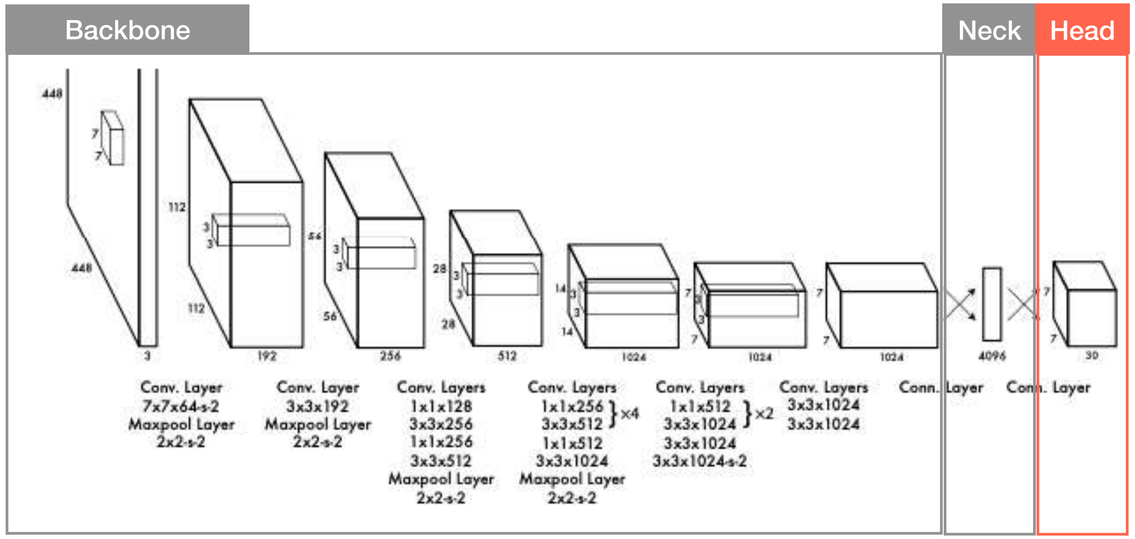
UNIVERSIDAD EAFIT

2016 - You Only Look Once Unified Real-Time Object Detection

Arquitectura

Head

- La capa de salida final de la red que realiza las predicciones
- Una segunda capa (FC) toma este vector latente y lo expande en una salida estructurada correspondiente a la malla $S \times S$



Función de costo

Yolo

- Es la suma del error cuadrático ponderado de todas sus predicciones
- λ_{coord} y λ_{noobj} balancean el costo dandole mas fuerza a la localización y disminuyendo el impacto de confidence de no objeto
- $\mathbf{1}_{ij}^{obj}$ Sólo penaliza el error si un objeto está presente en esa celda “responsable” de su predicción

Localización

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

Confidence

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

Clasificación

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

UNIVERSIDAD EAFIT

2016 - You Only Look Once Unified Real-Time Object Detection

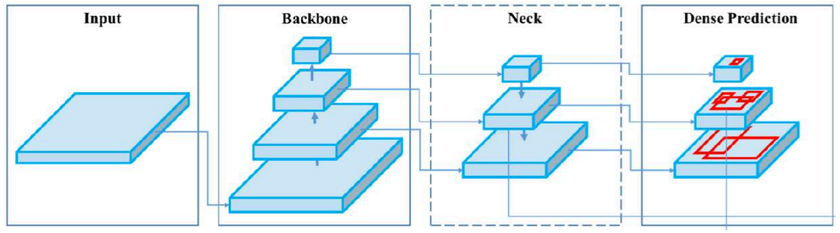
Actualizaciones

Yolo

Anchor Boxes (YOLOv2): Adoptado de Faster R-CNN para mejorar la localización

Multi-Scale Detection (YOLOv3): inspirado en Feature Pyramid Networks, mejora la detección de objetos a diferentes escalas

Data Augmentation (YOLOv4): Uso de “Mosaic augmentation” mejorar la generalización



Feature Pyramid

Mosaic augmentation

UNIVERSIDAD EAFIT

2017 - YOLO9000 Better, Faster, Stronger
2018 - YOLOv3 An Incremental Improvement
2020 - YOLOv4 Optimal Speed and Accuracy of Object Detection

Referencias

2014 - Rich feature hierarchies for accurate object detection and semantic segmentation

2004 - Efficient Graph-Based Image Segmentation

2013 - Selective Search for Object Recognition

2015 - Fast R-CNN

2016 - SSD: Single Shot MultiBox Detector

2016 - You Only Look Once Unified Real-Time Object Detection

2017 - YOLO9000 Better, Faster, Stronger

2018 - YOLOv3 An Incremental Improvement

2020 - YOLOv4 Optimal Speed and Accuracy of Object Detection

2020 - End-to-End Object Detection with Transformers

2019 - Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression