

Detección de Rostros basada en la Red YOLOv3 Preentrenada

Proyecto Final - Visión por Computador

Alejandro Cárdenas Barranco, Álvaro Rodríguez Gallardo, Juan
Manuel Rodríguez Gómez

Universidad de Granada

Curso 2023/2024



UNIVERSIDAD
DE GRANADA

1 Detección de Rostros

2 YOLOv3

3 Datasets

4 Experimentación

5 Resultados y Discusión

6 Conclusiones

1 Detección de Rostros

2 YOLOv3

3 Datasets

4 Experimentación

5 Resultados y Discusión

6 Conclusiones

Definición y Aplicaciones en la Actualidad

- **Definición:** Técnica avanzada de visión por computador que se centra en **localizar e identificar rostros humanos dentro de imágenes digitales o secuencias de vídeo.**
- **Aplicaciones:** Sistemas de seguridad y vigilancia, desbloqueo de teléfonos móviles, etiquetado de fotos en redes sociales...

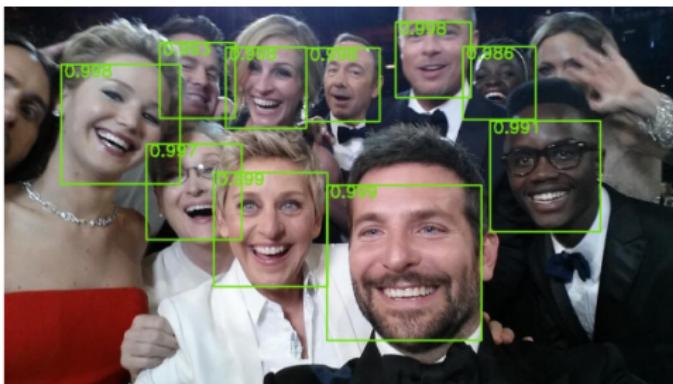


Imagen extraída de MIT Technology Review.

1 Detección de Rostros

2 YOLOv3

3 Datasets

4 Experimentación

5 Resultados y Discusión

6 Conclusiones

Descripción

- YOLOv3 (**You Only Look Once versión 3**). [1]
- Red neuronal con **arquitectura completamente convolucional**.
- Publicada en el año 2018. Dirigida a detección de objetos.
- **Característica Principal: Bajo tiempo de ejecución** frente a otros algoritmos (habilidad de detectar objetos en una imagen en **una sola pasada** de la red, de ahí su nombre).

[1] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement*. arXiv preprint arXiv:1804.02767, 2018.

Funcionamiento General

- **Detección Multiescala:** Implementa detección en **tres escalas diferentes**.

Funcionamiento General

- **Detección Multiescala:** Implementa detección en **tres escalas diferentes**.
- **División de la Imagen en Celdas para cada Escala.**

Funcionamiento General

- **Detección Multiescala:** Implementa detección en **tres escalas diferentes**.
- **División de la Imagen en Celdas para cada Escala.**
- **Tensor de Salida:** Devuelve un tensor 3D por cada celda.

Funcionamiento General

- **Detección Multiescala:** Implementa detección en **tres escalas diferentes**.
- **División de la Imagen en Celdas para cada Escala.**
- **Tensor de Salida:** Devuelve un tensor 3D por cada celda.
- En cada dimensión de cada tensor encontramos:
 - Coordenadas de la caja delimitadora (**bounding box**)
 - **Puntuación de objeto** (deseablemente 1 en el centro de la caja).
 - **Puntuación de clasificación para cada clase** de objeto.

Funcionamiento General

- **Detección Multiescala:** Implementa detección en **tres escalas diferentes**.
- **División de la Imagen en Celdas para cada Escala.**
- **Tensor de Salida:** Devuelve un tensor 3D por cada celda.
- En cada dimensión de cada tensor encontramos:
 - Coordenadas de la caja delimitadora (**bounding box**)
 - **Puntuación de objeto** (deseablemente 1 en el centro de la caja).
 - **Puntuación de clasificación para cada clase** de objeto.
- **Predicción de 3 Cajas Delimitadoras en cada Celda:** Se ajustan las desviaciones respecto a las **anchor boxes**.

Funcionamiento General

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

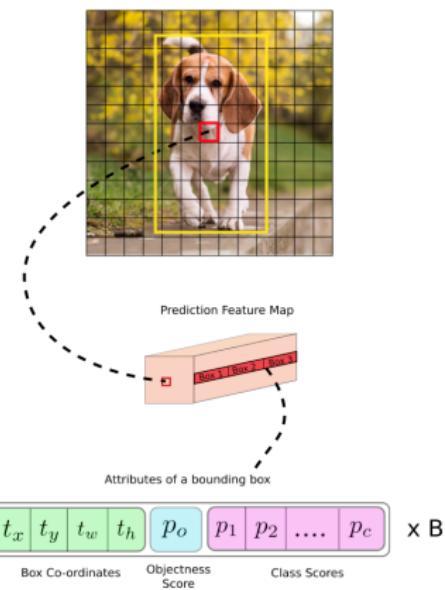


Imagen extraída de Towards Data Science.

Detección

- Manejo de Múltiples Bounding Boxes.

Detección

- **Manejo de Múltiples Bounding Boxes.**
- **Ordenamiento de Cajas por Puntuación de Objeto.**

Detección

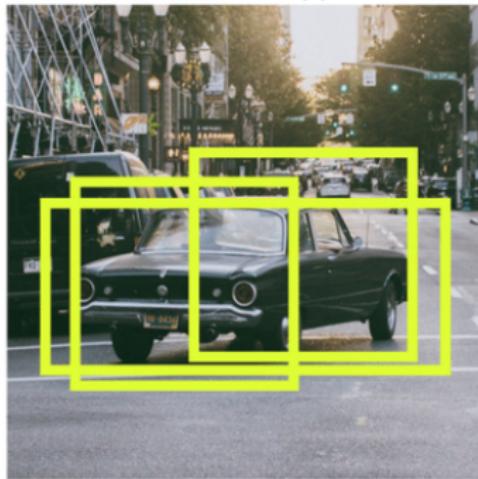
- **Manejo de Múltiples Bounding Boxes.**
- **Ordenamiento de Cajas por Puntuación de Objeto.**
- **Filtrado por Umbral de Puntuación** → Reducción del número de bounding boxes.

Detección

- **Manejo de Múltiples Bounding Boxes.**
- **Ordenamiento de Cajas por Puntuación de Objeto.**
- **Filtrado por Umbral de Puntuación** → Reducción del número de bounding boxes.
- **Supresión de No-Máximos** → Se deja solo la caja con la puntuación más alta en cada región.

Detección

Before non-max suppression



Non-Max
Suppression



After non-max suppression

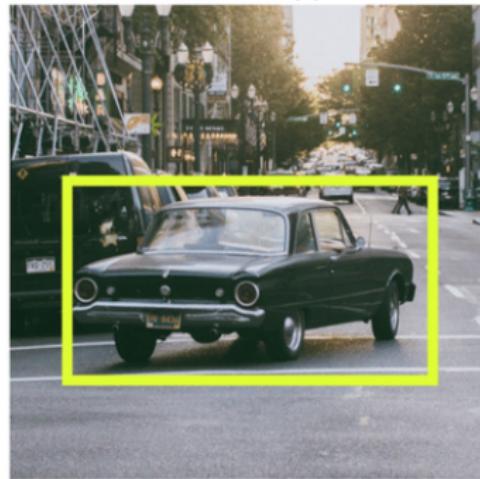


Imagen extraída de ResearchGate.

Arquitectura

- Darknet-53 + Feature Pyramid Network.

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

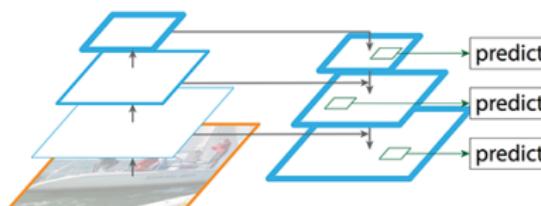


Imagen extraída de ResearchGate. / Imagen extraída de Papers With Code.

Arquitectura

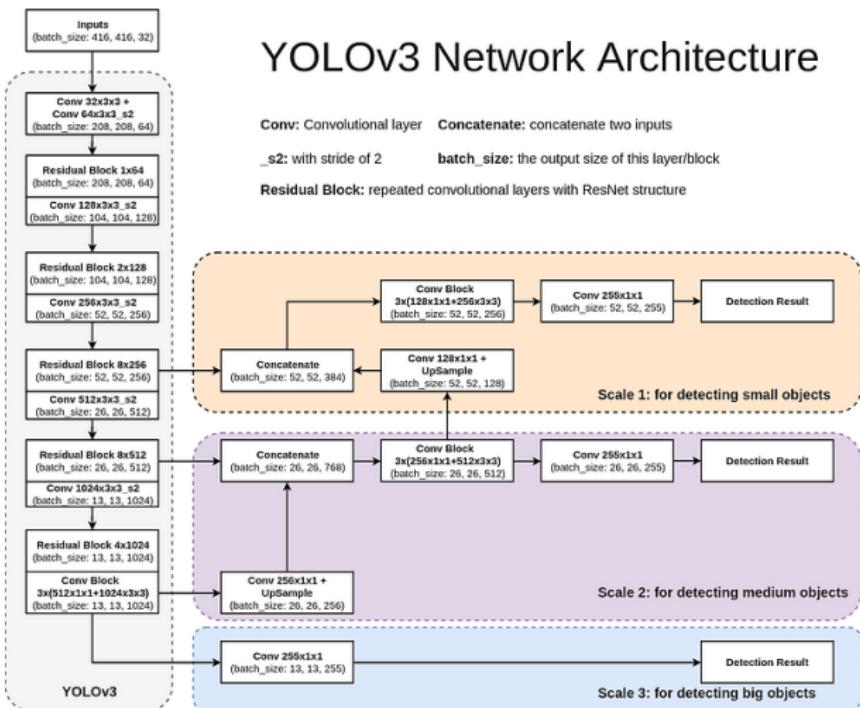


Imagen extraída de Sopra Steria Analytics.



1 Detección de Rostros

2 YOLOv3

3 Datasets

4 Experimentación

5 Resultados y Discusión

6 Conclusiones

COCO

- **COCO (Common Objects in COntext)**. [1]
- Dataset extenso para detección y segmentación de objetos.
- Contiene **más de 200,000 imágenes** con **1.5 millones de anotaciones** de objetos.
- Incluye **80 categorías** de objetos.

[1] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. *Microsoft COCO: Common Objects in Context*. arXiv preprint arXiv:1405.0312, 2015.

COCO

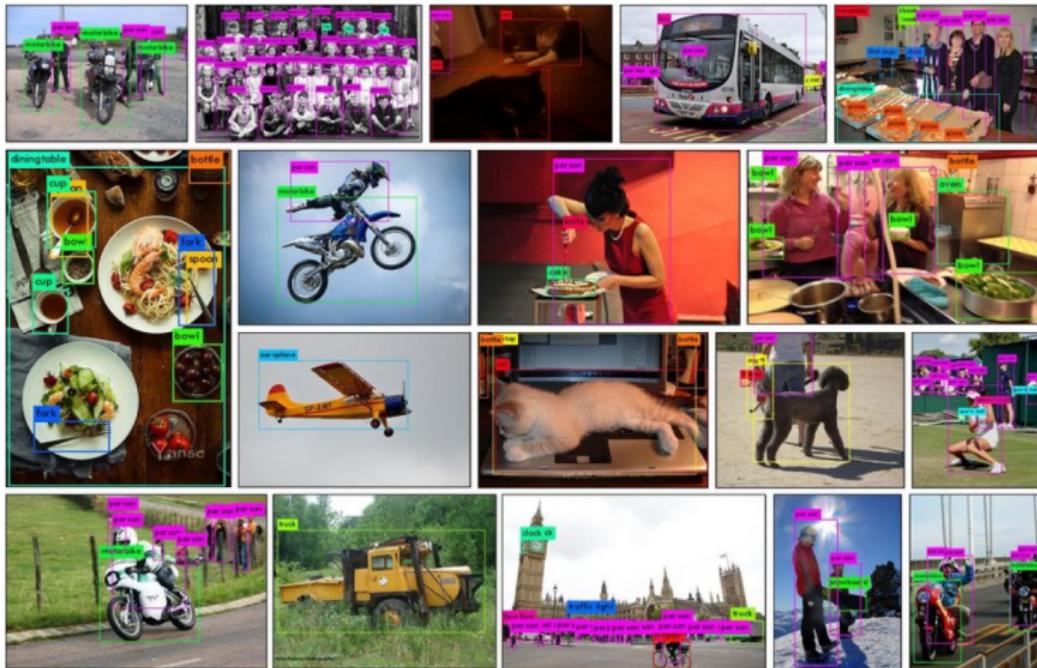


Imagen extraída de V7 Labs.



WIDER FACE

- **WIDER FACE:** Dataset público **específico para detección de rostros [1]**.
- Contiene **más de 32,000 imágenes** con **393,703 caras anotadas (ground truth)**.
- Amplia variabilidad en términos de escala, pose y ocultación.

[1] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. *WIDER FACE: A Face Detection Benchmark*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

WIDER FACE

Scale



Pose



Occlusion



Expression



Makeup



Illumination



Imagen extraída de ResearchGate.

1 Detección de Rostros

2 YOLOv3

3 Datasets

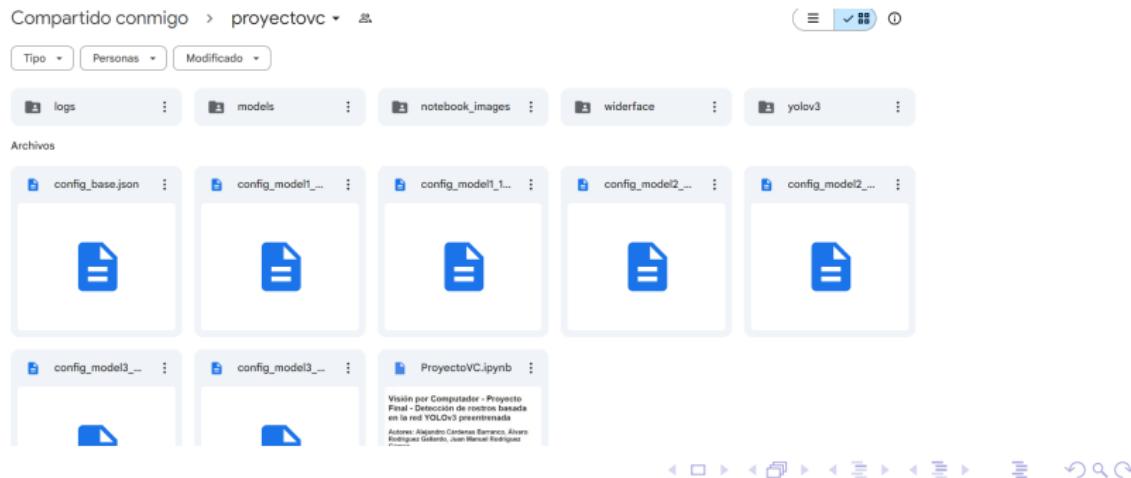
4 Experimentación

5 Resultados y Discusión

6 Conclusiones

Consideraciones Previas

- **Implementación en Keras de YOLOv3** obtenida de un repositorio de GitHub (la hemos adaptado a **Tensorflow 2** y le hemos añadido el código necesario para realizar **fine-tuning**).
- Entorno de trabajo: **Google Drive** (organización de archivos) + **Google Colab** (código y ejecuciones).



Consideraciones Previas

- Generación de las correspondientes **anchor boxes** para el dataset WIDER FACE.
[[2, 4, 4, 8, 7, 14], [12, 23, 20, 36, 35, 56], [56, 95, 101, 149, 177, 234]]

Consideraciones Previas

- Generación de las correspondientes **anchor boxes** para el dataset WIDER FACE.
[[2, 4, 4, 8, 7, 14], [12, 23, 20, 36, 35, 56], [56, 95, 101, 149, 177, 234]]
- **Anotaciones de los rostros** del dataset WIDER FACE convertidas a formato **VOC** (es el formato que maneja la implementación utilizada de YOLOv3).

Consideraciones Previas

- Generación de las correspondientes **anchor boxes** para el dataset WIDER FACE.
[[2, 4, 4, 8, 7, 14], [12, 23, 20, 36, 35, 56], [56, 95, 101, 149, 177, 234]]
- **Anotaciones de los rostros** del dataset WIDER FACE convertidas a formato **VOC** (es el formato que maneja la implementación utilizada de YOLOv3).
- Descarga de los **pesos de la red** preentrenada en el dataset **COCO**.

Consideraciones Previas

- Generación de las correspondientes **anchor boxes** para el dataset WIDER FACE.
[[2, 4, 4, 8, 7, 14], [12, 23, 20, 36, 35, 56], [56, 95, 101, 149, 177, 234]]
- **Anotaciones de los rostros** del dataset WIDER FACE convertidas a formato **VOC** (es el formato que maneja la implementación utilizada de YOLOv3).
- Descarga de los **pesos de la red** preentrenada en el dataset **COCO**.
- Uso de **archivos de configuración JSON** para los parámetros cada modelo.

Consideraciones Previas

- **mAP (mean Average Precision)**: Métrica que evalúa la **precisión** (porcentaje de predicciones correctas) y el **recall** (proporción de objetos detectados frente al total de objetos a detectar) de un modelo en la detección de objetos para todas las clases (en nuestro caso solo hay una, la **clase face**) en un conjunto de datos.
- **Métricas** para evaluar los modelos:
 - ***mAP@0.5***: Calcula el mAP con un umbral de IoU (Intersection over Union) de 0.5.
 - ***mAP@[.5:.95]***: Calcula la media de los mAP para varios umbrales de IoU, desde 0.5 hasta 0.95, en pasos de 0.05.

Consideraciones Previas

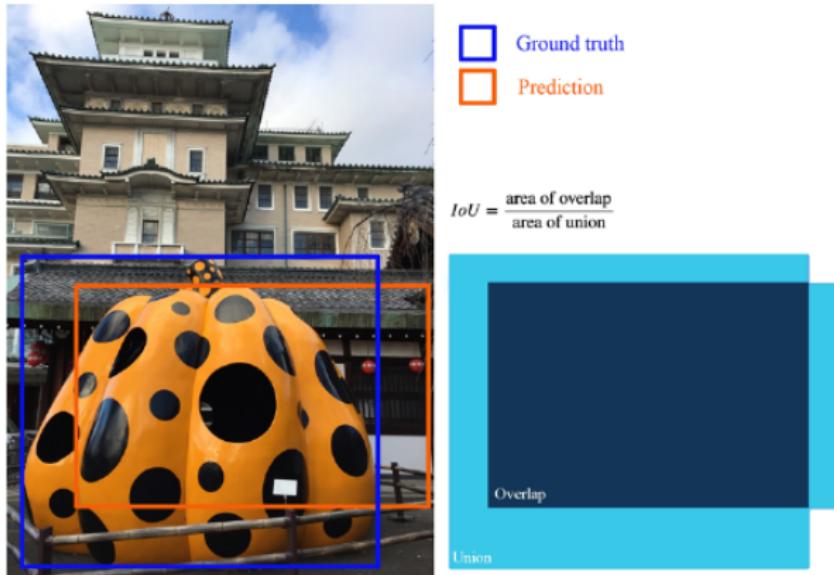
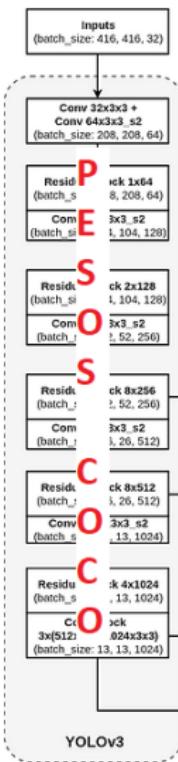


Imagen extraída de Medium.

Modelo Base



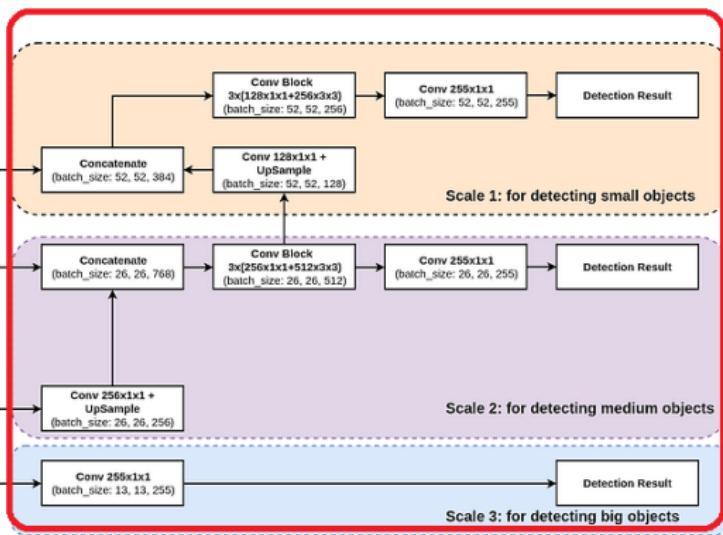
YOLOv3 Network Architecture

Conv: Convolutional layer **Concatenate:** concatenate two inputs

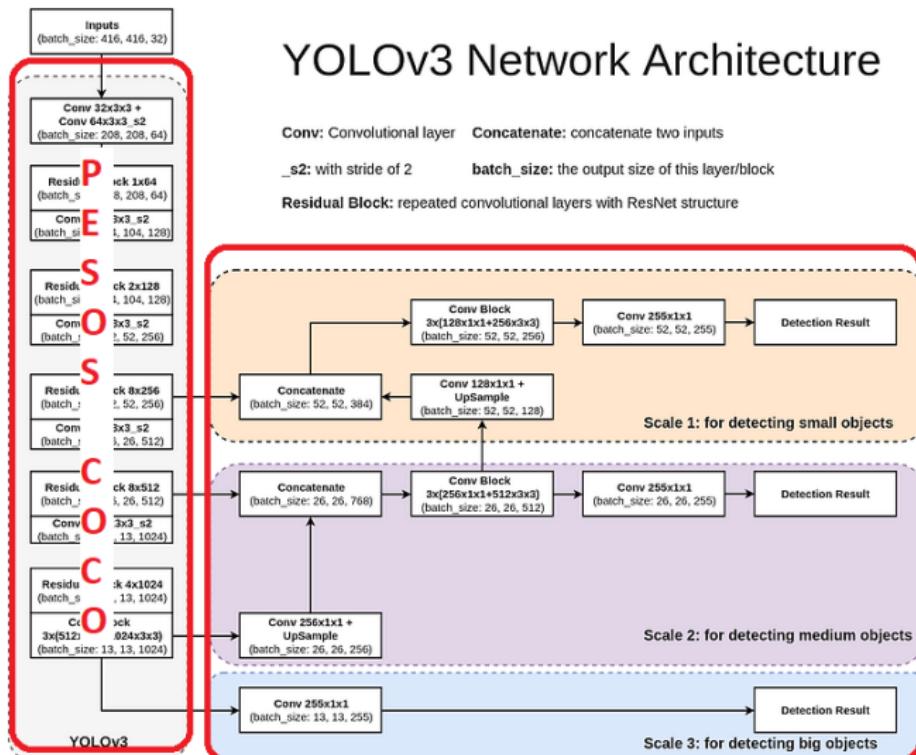
_s2: with stride of 2

batch_size: the output size of this layer/block

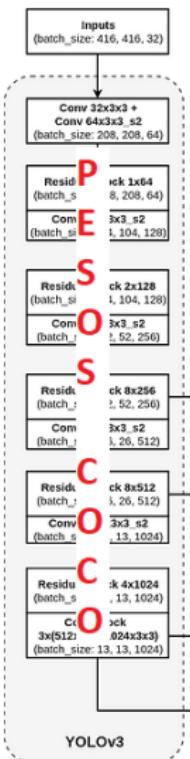
Residual Block: repeated convolutional layers with ResNet structure



Modelo 1: Entrenamiento Completo



Modelo 2: Fine-tuning en los Bloques de Detección

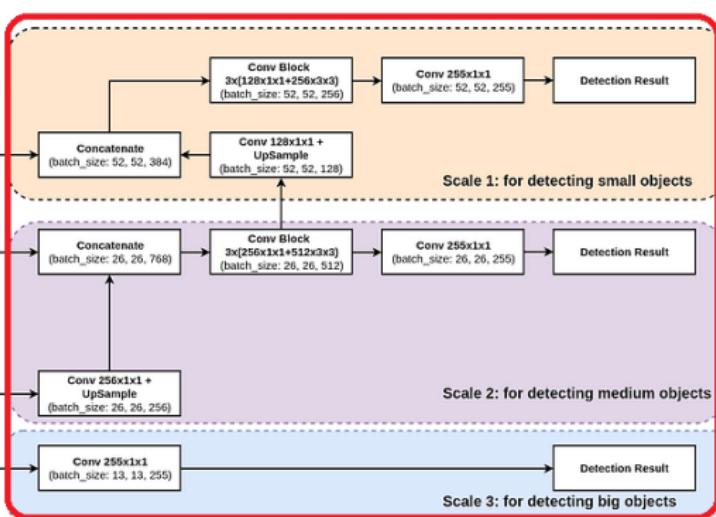


YOLOv3 Network Architecture

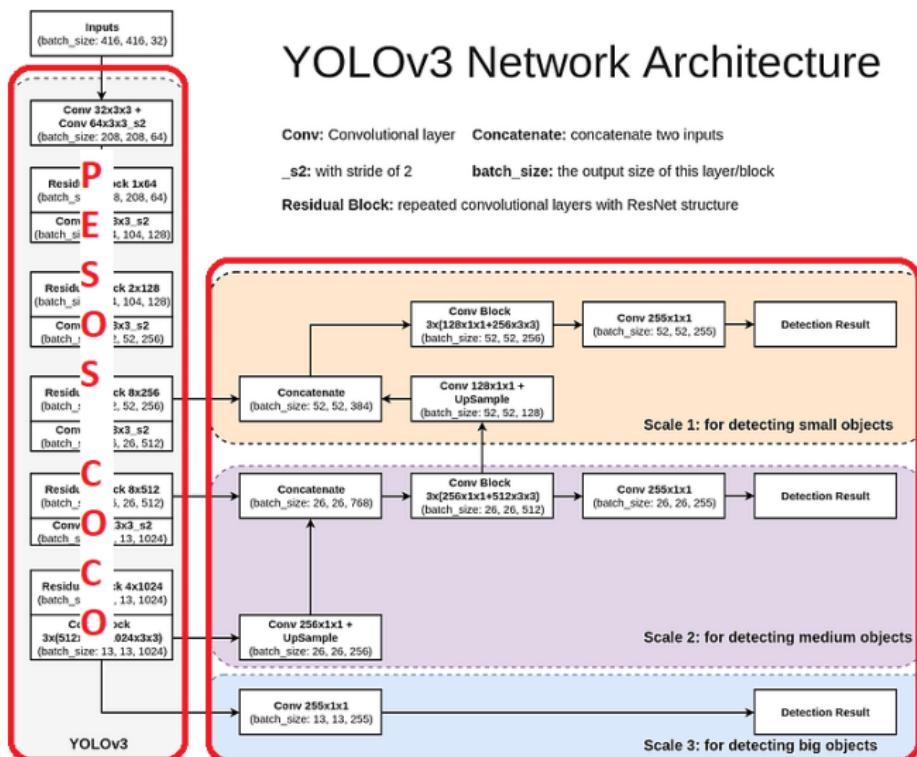
Conv: Convolutional layer **Concatenate:** concatenate two inputs

_s2: with stride of 2 **batch_size:** the output size of this layer/block

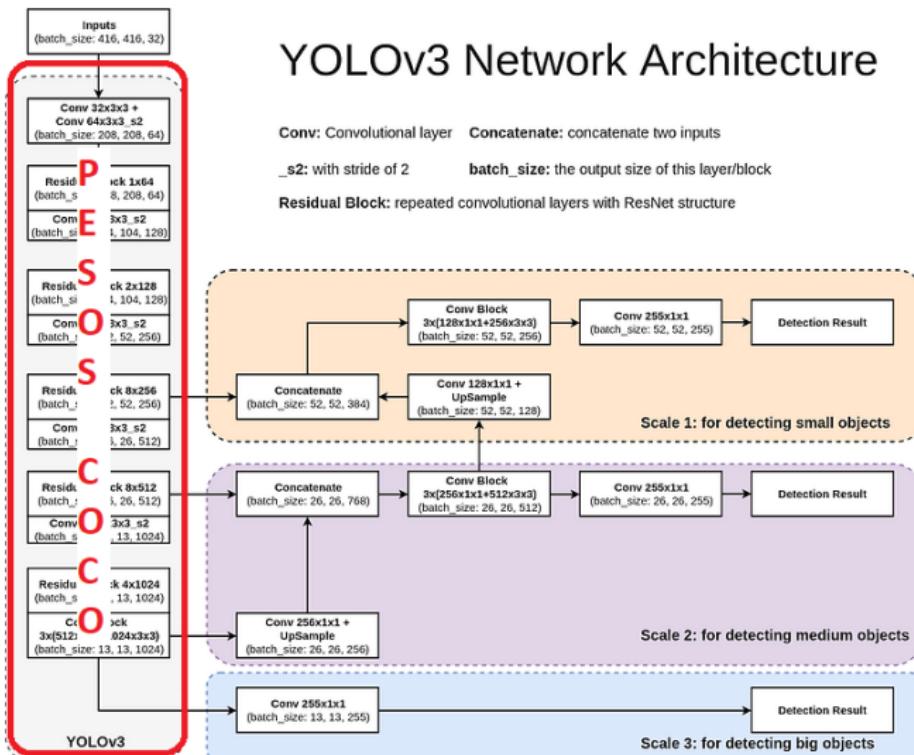
Residual Block: repeated convolutional layers with ResNet structure



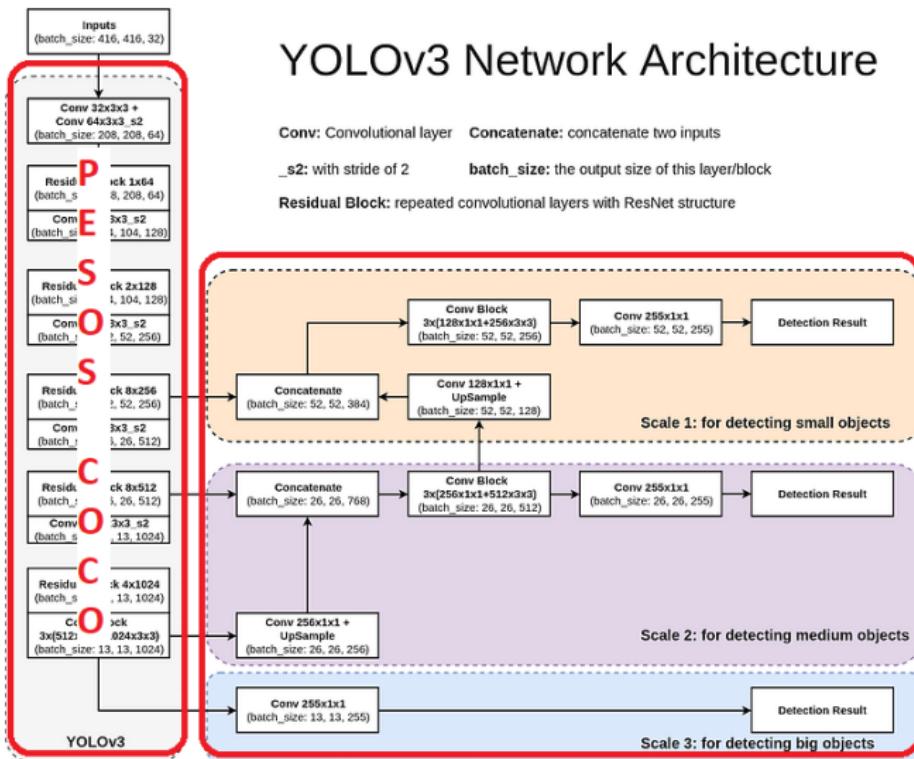
Modelo 2: Fine-tuning en los Bloques de Detección



Modelo 3: Fine-Tuning en el Extractor de Características



Modelo 3: Fine-Tuning en el Extractor de Características



1 Detección de Rostros

2 YOLOv3

3 Datasets

4 Experimentación

5 Resultados y Discusión

6 Conclusiones

Resultados

| Modelo | Épocas de entrenamiento | Tamaño de entrada | $mAP@[.5:.95]$ | $mAP@0.5$ |
|---------------|-------------------------|-------------------|----------------|--------------|
| Modelo Base | 10 | 416 x 416 | 0.025 | 0.083 |
| Modelo 1-416 | 16 | 416 x 416 | 0.257 | 0.468 |
| Modelo 1-1024 | 15 | 1024 x 1024 | 0.376 | 0.666 |
| Modelo 2 | 20 | 1024 x 1024 | 0.405 | 0.726 |
| Modelo 3 | 25 | 1024 x 1024 | 0.392 | 0.708 |

Table 1. Resultados de la evaluación de los modelos.

Discusión

- **Modelo Base:** Puntuaciones de $mAP@[.5:.95]$ y $mAP@0.5$ bajas.

Discusión

- **Modelo Base:** Puntuaciones de $mAP@[.5:.95]$ y $mAP@0.5$ bajas.
- **Modelo 1:** Mejora significativa; el aumento del tamaño de entrada de las imágenes de entrenamiento de 416x416 a 1024x1024 resalta la **importancia de la resolución de entrada en la precisión de la detección**.

Discusión

- **Modelo Base:** Puntuaciones de $mAP@[.5:.95]$ y $mAP@0.5$ bajas.
- **Modelo 1:** Mejora significativa; el aumento del tamaño de entrada de las imágenes de entrenamiento de 416x416 a 1024x1024 resalta la **importancia de la resolución de entrada en la precisión de la detección**.
- **Modelo 2:** Alcanza las **puntuaciones más altas**.

Discusión

- **Modelo Base:** Puntuaciones de $mAP@[.5:.95]$ y $mAP@0.5$ bajas.
- **Modelo 1:** Mejora significativa; el aumento del tamaño de entrada de las imágenes de entrenamiento de 416x416 a 1024x1024 resalta la **importancia de la resolución de entrada en la precisión de la detección**.
- **Modelo 2:** Alcanza las **puntuaciones más altas**.
- **Modelo 3:** No supera al Modelo 2; esto sugiere que **la capacidad de adaptarse a características de bajo nivel es importante para la detección de rostros y debe mantenerse flexible durante el entrenamiento**.

Ejemplos de Detección con el Modelo 2



Ejemplos de Detección con el Modelo 2



1 Detección de Rostros

2 YOLOv3

3 Datasets

4 Experimentación

5 Resultados y Discusión

6 Conclusiones

Conclusiones

- **Objetivo Alcanzado:** Adaptación exitosa de YOLOv3, preentrenada en COCO, para la detección eficaz de rostros con alta precisión, sin conocimiento específico previo sobre objetos tipo rostro.
- Importancia de **equilibrar precisión y eficiencia** según las necesidades de la aplicación.
- **Oportunidades y Consideraciones Futuras:**
 - Experimentación con diferentes optimizadores y el fine-tuning congelando bloques específicos.
 - Comparación con alternativas como Faster R-CNN o RetinaNet.

Fin de la Presentación

Gracias por su atención.

¿Preguntas?