



# Sistema de detección automática de baches en el asfalto a partir de imágenes

Autor: Diego Castro

# Objetivos

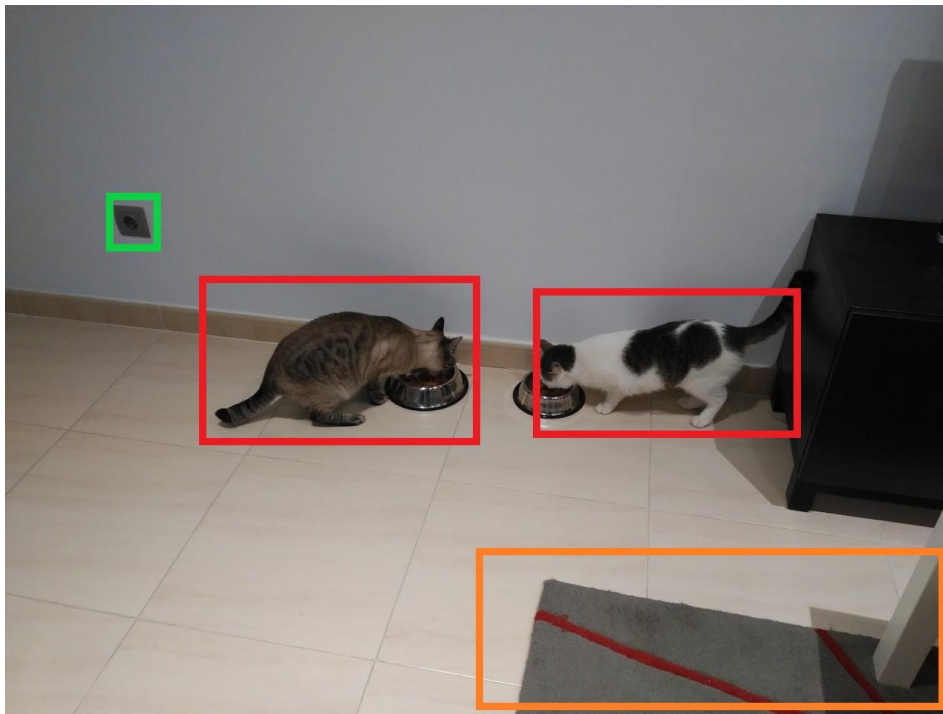
Con este trabajo se pretende desarrollar un sistema de **detección de baches en tiempo real** para un **dispositivo móvil**.

Se busca que tenga una **utilidad social**, y que al mismo tiempo permita **ampliar los conocimientos** adquiridos durante el máster de Big Data & Data Science, en el ámbito de la **visión artificial**, concretamente en la **detección de objetos**.

# Agenda

- Estado del arte: detección de objetos
- Solución
  - Arquitectura
  - Datos
  - Implementación
- Evaluación
- Resultados
- Conclusiones
- Mejoras

# Estado del arte: detección de objetos



Gato



Gato



Alfombra



Enchufe

# Estado del arte: detección de objetos

Faster R-CNN

Fast R-CNN

R-FCN

YOLO

R-CNN

Mark R-CNN

Viola-Jones

SSD

# Estado del arte: detección de objetos

Viola-Jones

R-CNN

Fast R-CNN

Faster R-CNN

YOLO

2001

AdaBoost

Tiempo Real

# Estado del arte: detección de objetos

Viola-Jones

R-CNN

Fast R-CNN

Faster R-CNN

YOLO

2014

Deep Learning

Búsqueda Selectiva

CNN

SVM + Regresión Lineal

# Estado del arte: detección de objetos

Viola-Jones

R-CNN

Fast R-CNN

Faster R-CNN

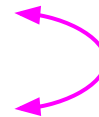
YOLO

2014

Búsqueda Selectiva

CNN

SVM >> Softmax





# Estado del arte: detección de objetos

Viola-Jones

R-CNN

Fast R-CNN

Faster R-CNN

YOLO

2016

Búsqueda selectiva >> RPN

# Estado del arte: detección de objetos

Viola-Jones

R-CNN

Fast R-CNN

Faster R-CNN

YOLO

2015

Una única CNN

Tiempo real

# Estado del arte: detección de objetos

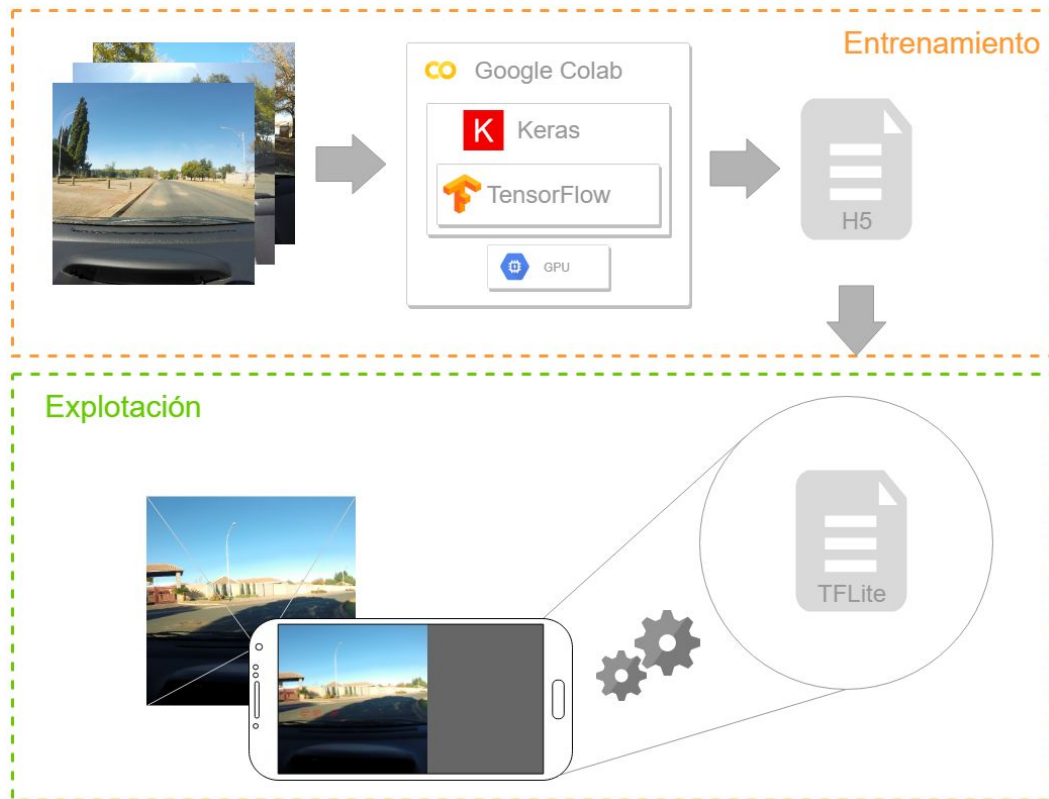
SSD

Mark R-CNN

R-FCN

...

# Solución - Arquitectura



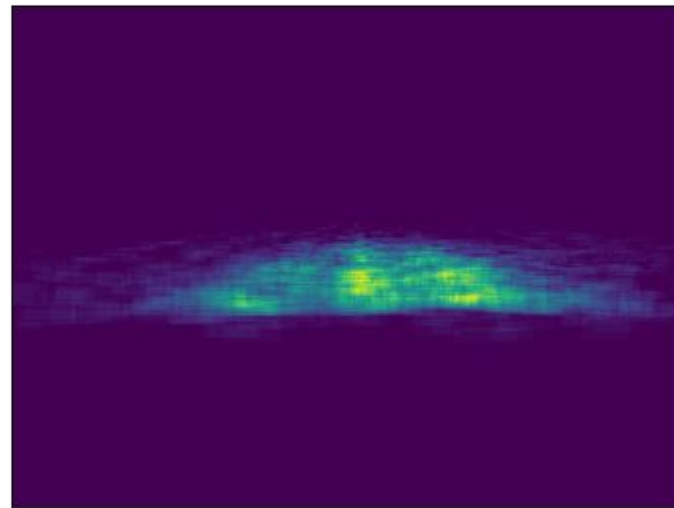
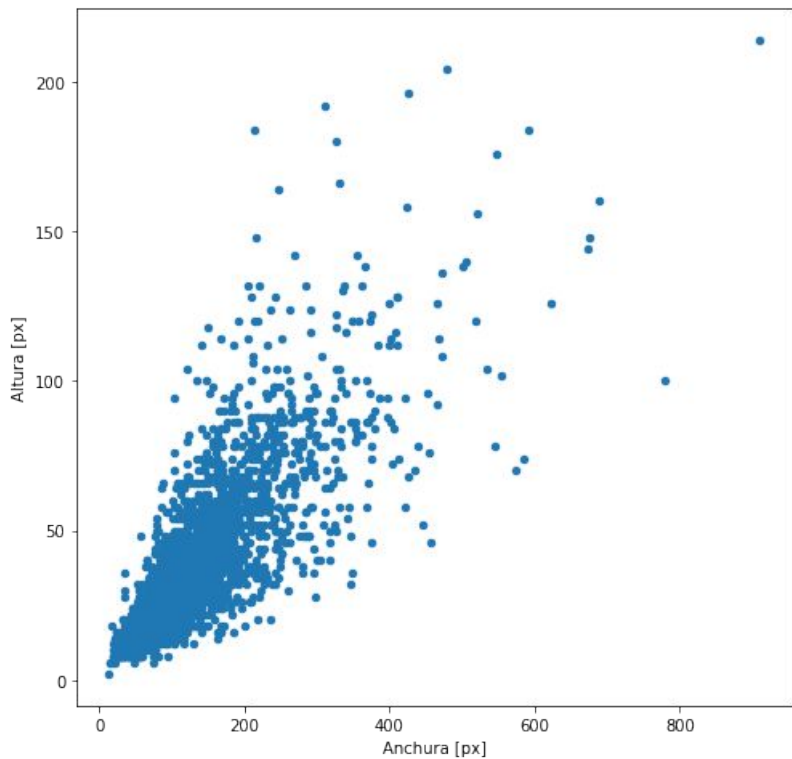
# Solución - Datos I

- 1900 imágenes
- Tamaño: 3680x2760 píxeles (formato 4:3)
- Imágenes de entrenamiento: 1297
- Imágenes de test: 603

Cada línea del fichero contiene las etiquetas de una imagen:

```
<RUTA_IMG> <NUMERO_DE_ETIQUETAS>( <X0> <Y0> <ANCHO> <ALTO>)+
```

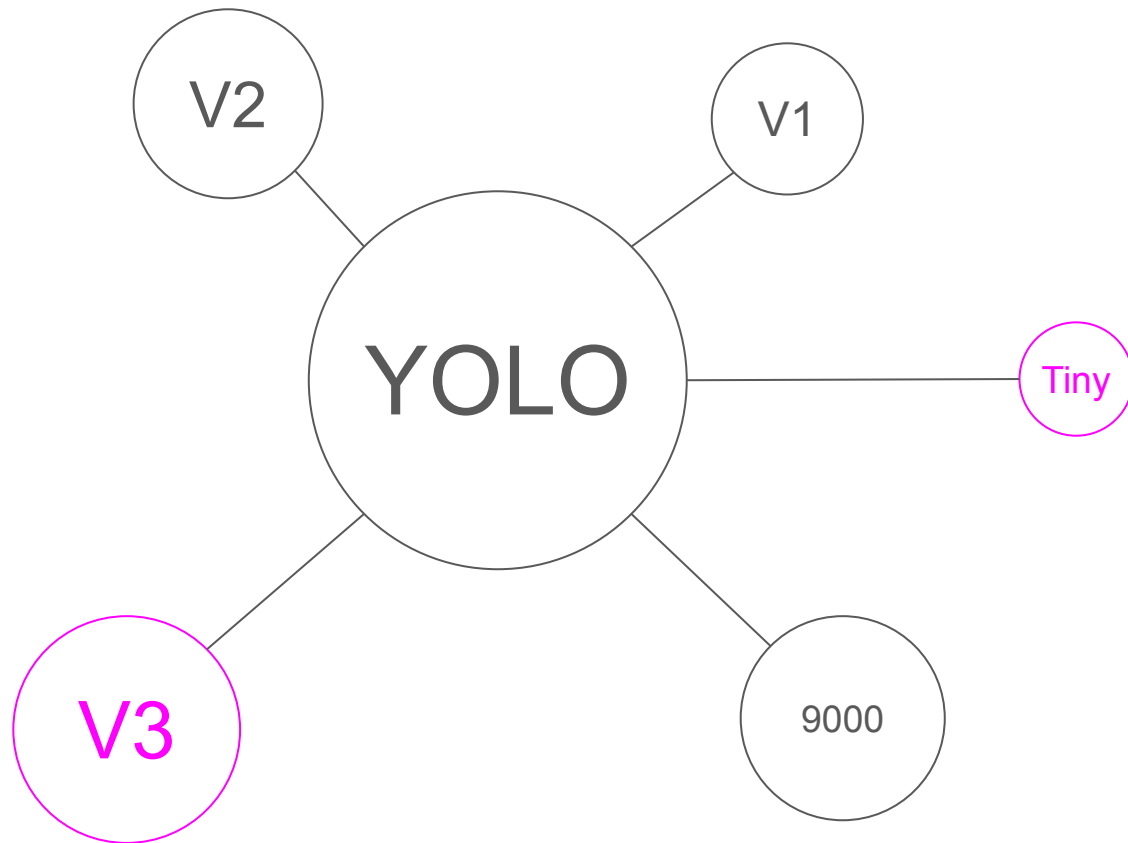
# Solución - Datos II



# Solución - Datos III



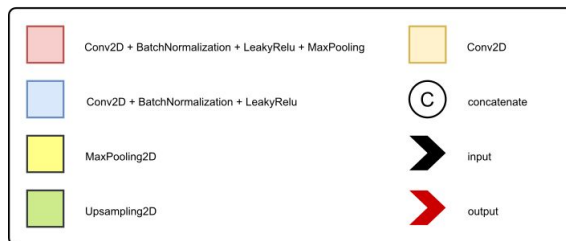
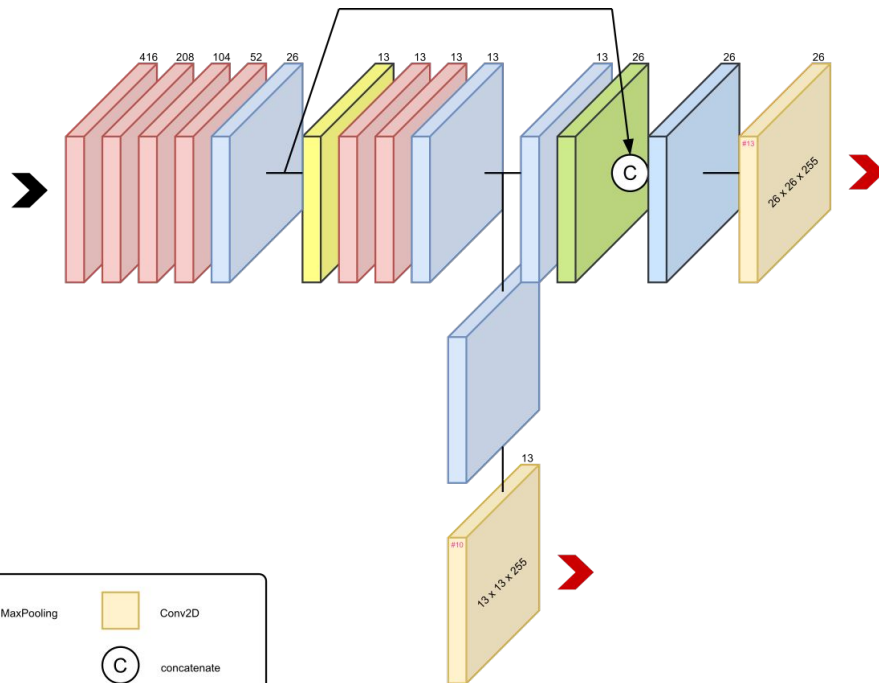
# Solución - Implementación I





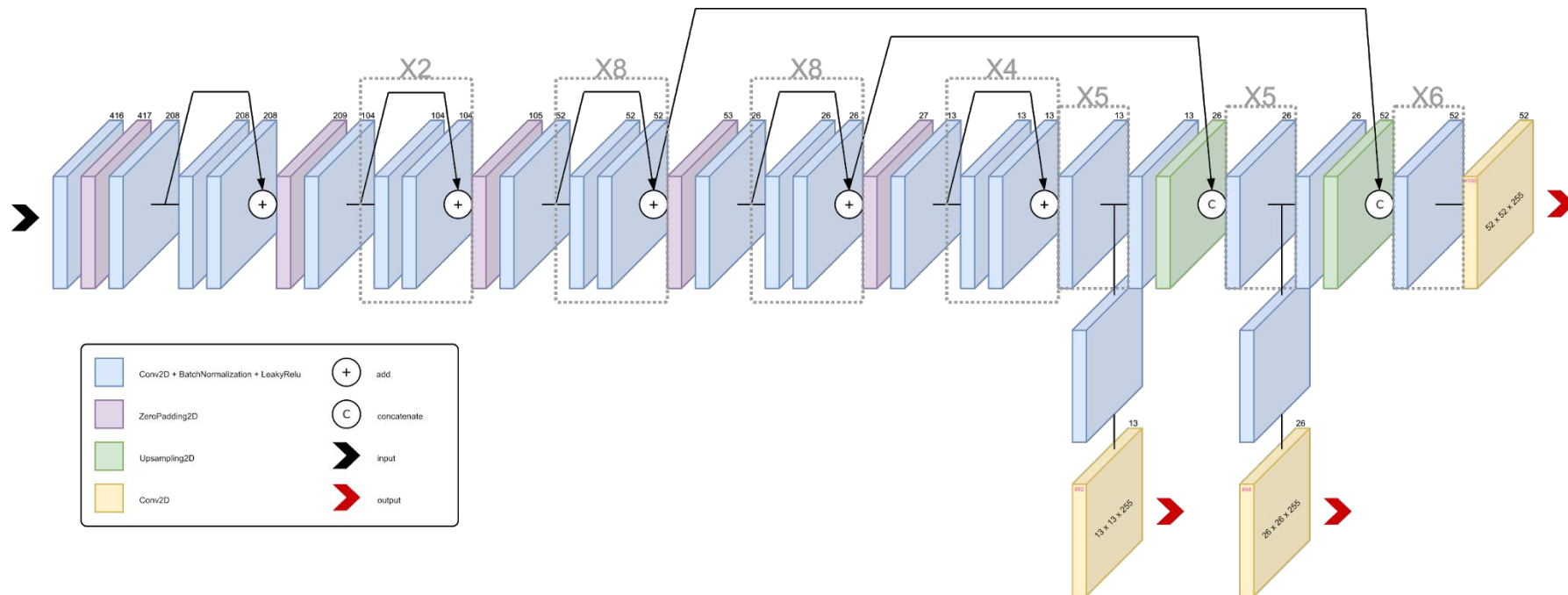
# Solución - Implementación II

YOLO V3 Tiny



## Solución - Implementación III

# YOLO V3



# Evaluación I

La métrica de evaluación utilizada es **AP** (*Average Precision*)

Se basa en 3 conceptos: **Precisión** (*Precision*), **Sensibilidad** (*Recall*) e **IoU** (*Intersection over Union*)

## Precisión

*Capacidad del modelo para detectar únicamente los objetos relevantes.*

TP

---

Todas las predicciones positivas

## Sensibilidad

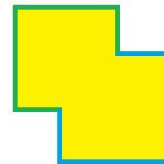
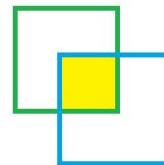
*Capacidad del modelo para detectar todos los objetos relevantes.*

TP

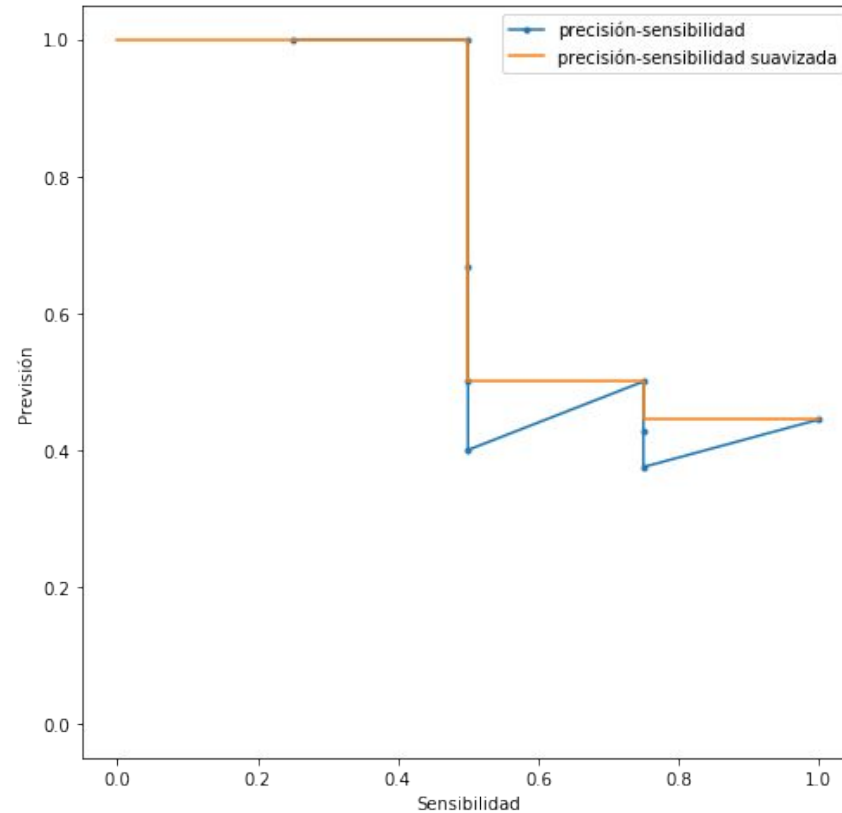
---

Todas las regiones a detectar

## IoU

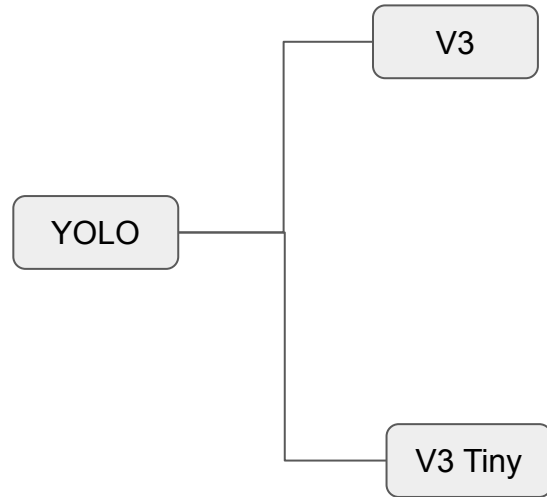


# Evaluación II



# Evaluación III

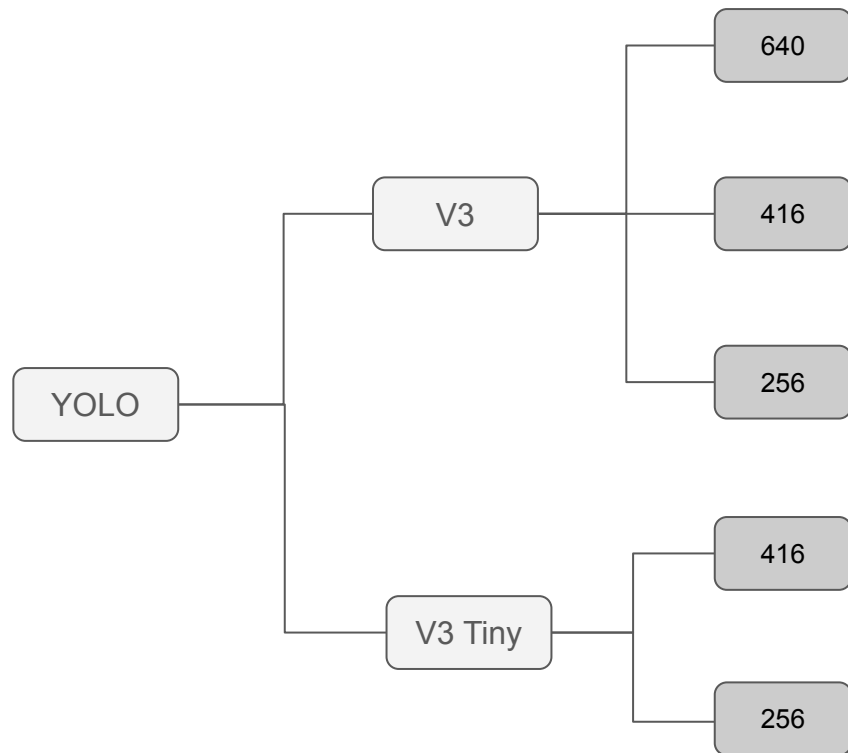
*Modelo*



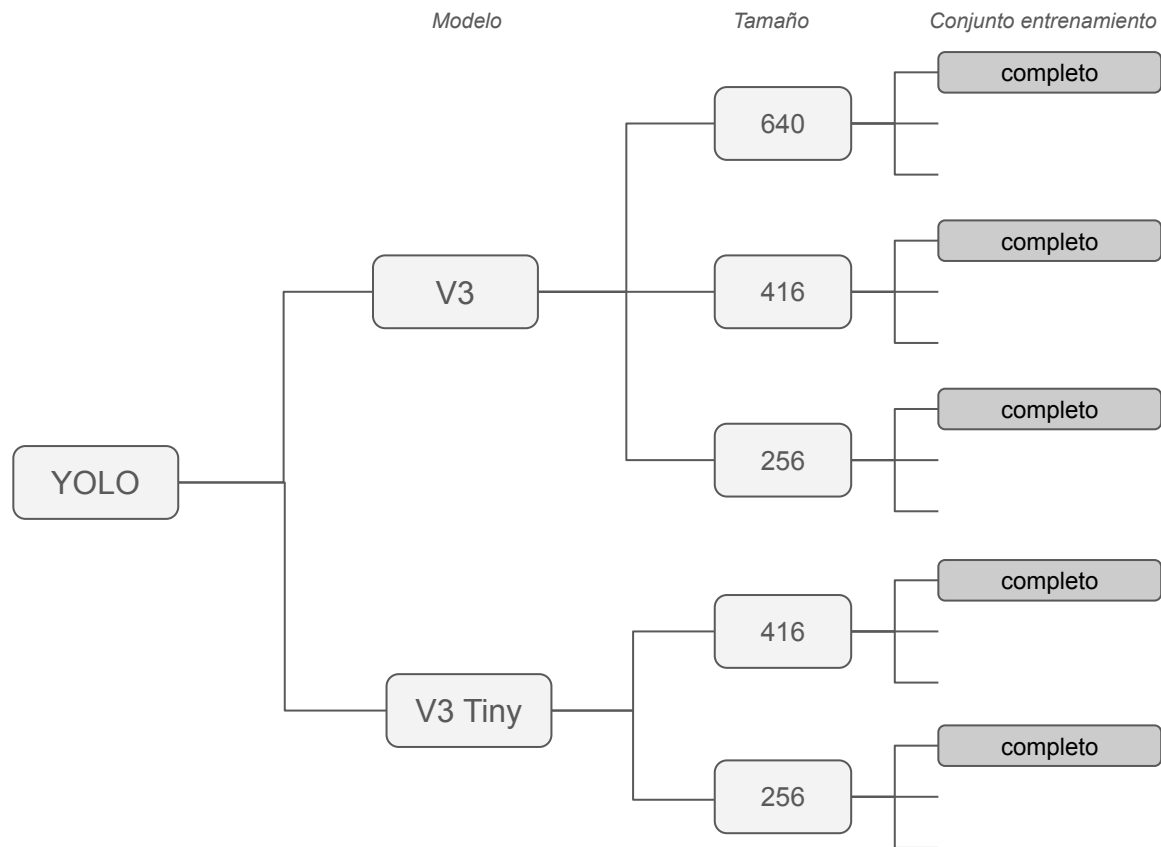
# Evaluación III

*Modelo*

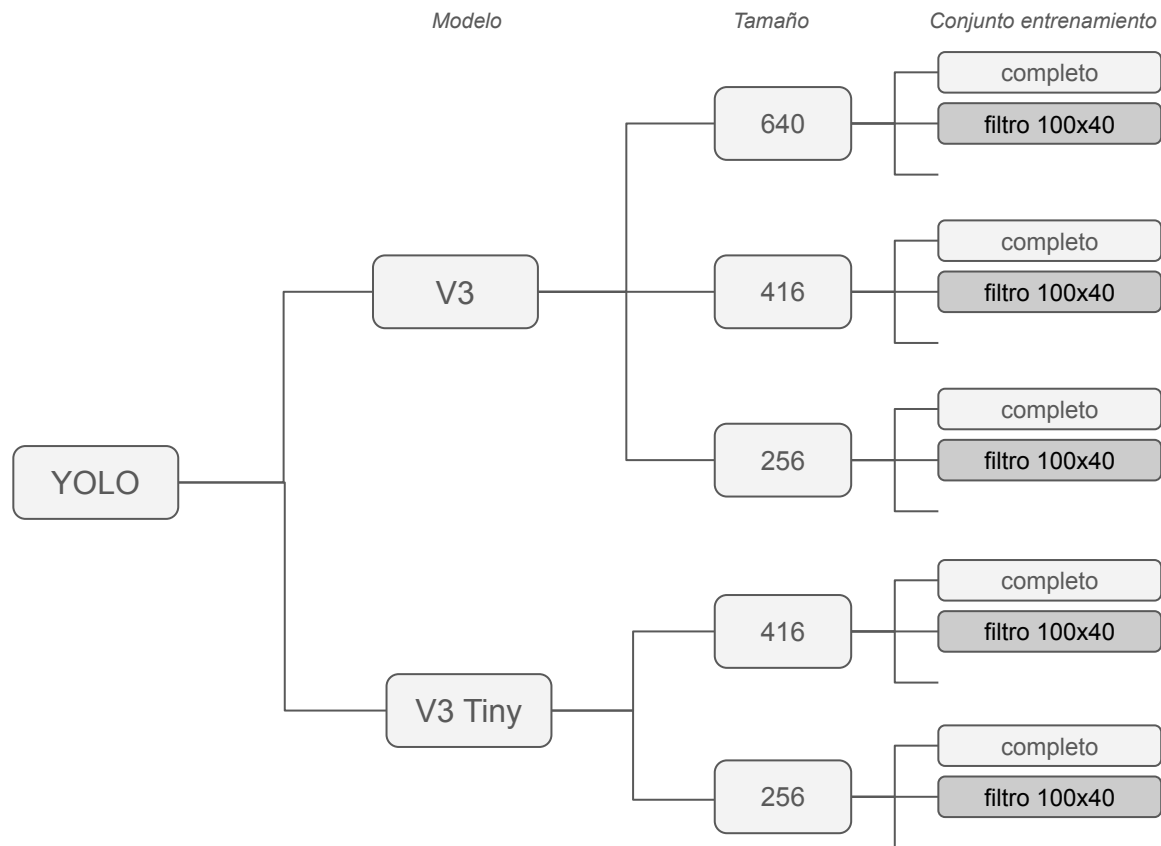
*Tamaño*



# Evaluación III

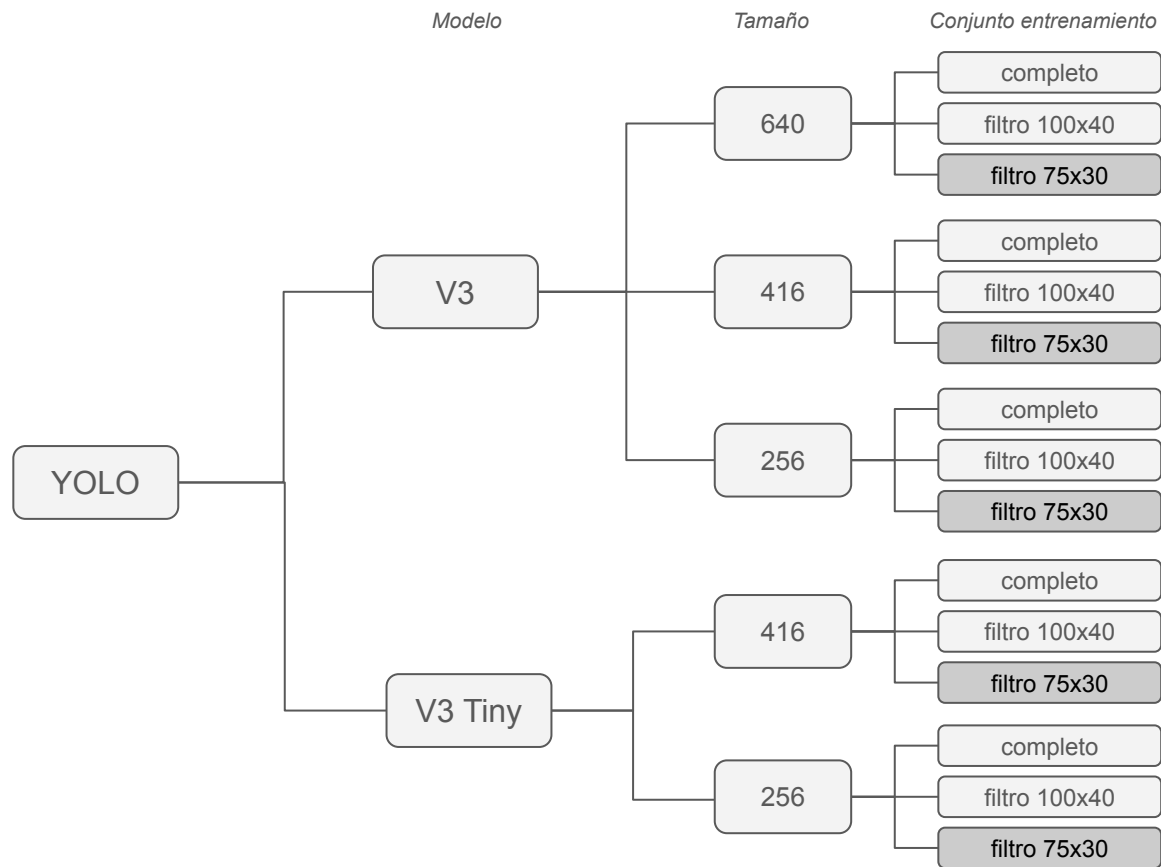


# Evaluación III

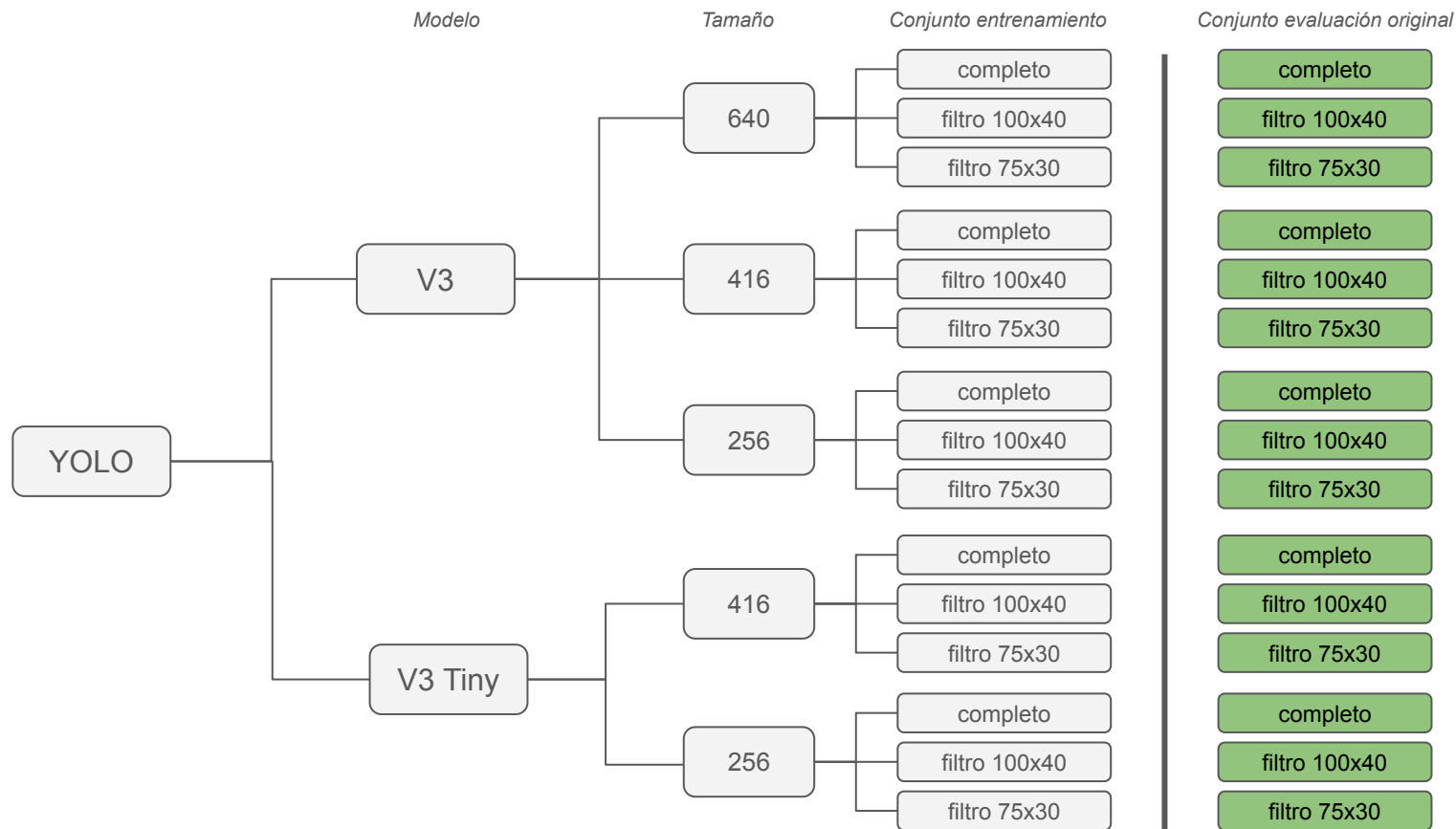




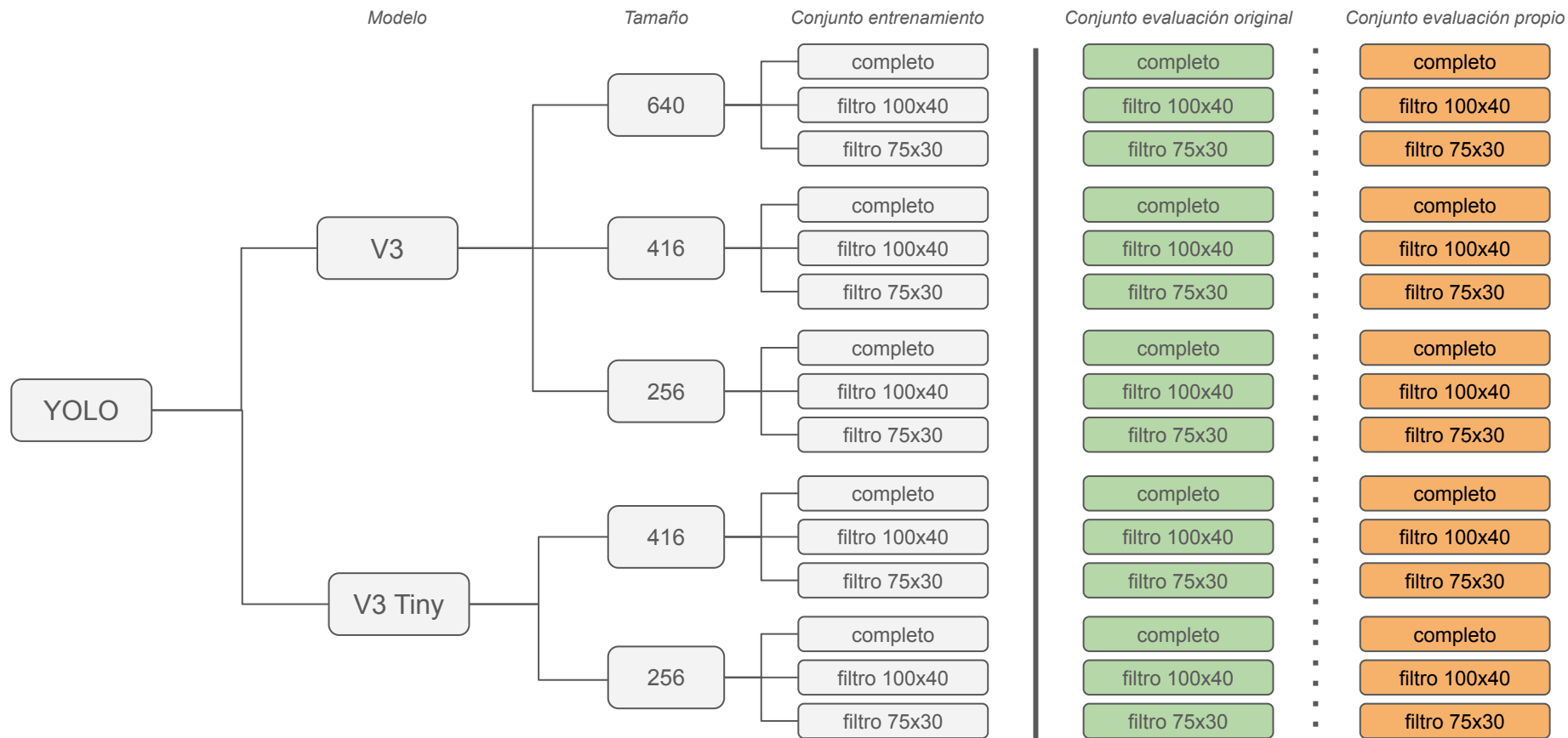
# Evaluación III



# Evaluación III



# Evaluación III

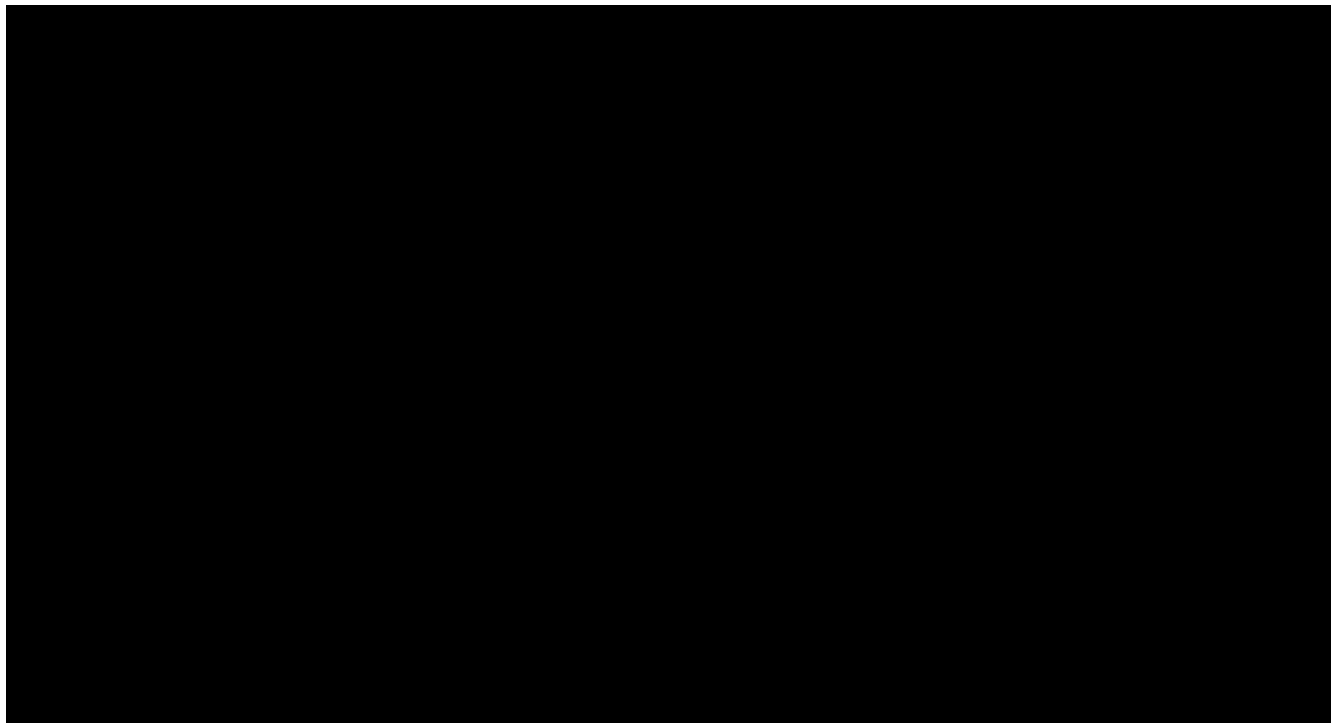


# Evaluación IV

Versión	Tamaño	Juego Datos	Épocas	AP
V3	256	completo	43	0.0747
V3	256	filtro 100x40	93	0.3077
V3	256	filtro 75x30	88	0.2513
V3	416	completo	18	0.1467
V3	416	filtro 100x40	93	0.4161
V3	416	filtro 75x30	93	0.3611
V3	640	completo	13	0.0186
<b>V3</b>	<b>640</b>	<b>filtro 100x40</b>	<b>63</b>	<b>0.5475</b>
V3	640	filtro 75x30	53	0.4106
V3 Tiny	256	completo	144	0.0046
V3 Tiny	256	filtro 100x40	136	0.0510
V3 Tiny	256	filtro 75x30	153	0.0392
V3 Tiny	416	completo	153	0.0145
<b>V3 Tiny</b>	<b>416</b>	<b>filtro 100x40</b>	<b>153</b>	<b>0.1307</b>
V3 Tiny	416	filtro 75x30	146	0.0869

Versión	Tamaño	Juego Datos	Épocas	AP
V3	256	completo	43	0.0289
<b>V3</b>	<b>256</b>	<b>filtro 100x40</b>	<b>93</b>	<b>0.1018</b>
V3	256	filtro 75x30	88	0.0179
V3	416	completo	18	0.0354
V3	416	filtro 100x40	93	0.0089
V3	416	filtro 75x30	93	0.0294
V3	640	completo	13	0.0017
V3	640	filtro 100x40	63	0.0342
V3	640	filtro 75x30	53	0.0961
V3 Tiny	256	completo	144	0.0086
V3 Tiny	256	filtro 100x40	136	0.0232
<b>V3 Tiny</b>	<b>256</b>	<b>filtro 75x30</b>	<b>153</b>	<b>0.0371</b>
V3 Tiny	416	completo	153	0.0000
V3 Tiny	416	filtro 100x40	153	0.0000
V3 Tiny	416	filtro 75x30	146	0.0006

# Resultados



YOLO V3 Tiny 256

# Conclusiones

- Para obtener unos buenos resultados es clave tener un **juego de datos** con **abundantes** imágenes y **correctamente etiquetadas**
- **Es primordial conocer los datos.** Gracias al estudio del aspecto y tamaño de los baches se ha realizado tratamiento inicial de las imágenes y se han creado distintos subconjuntos de entrenamiento que han mejorado los resultados del modelo
- Si la detección de objetos es un problema complejo a resolver, **la detección de objetos pequeños es mucho más complejo**

# Mejoras

- Predicciones

- Redistribución de conjunto entrenamiento/test
- Revisión de etiquetado
- Complementar juego de datos con imágenes de carreteras españolas
- Complementar juego de datos con imágenes con otro encuadre

- Rendimiento

- Transformación del modelo a uno *quantized*
- Aprovechar la GPU del dispositivo móvil

Muchas gracias

Q&A