



Universidad
de Huelva

TRABAJO FIN DE GRADO

DEEP LEARNING PARA CLASIFICACIÓN Y DETECCIÓN DE PERSONAS

Autor: José María Baeza-Herrazti Vázquez

Tutor: Diego Marín Santos

Cotutor: Manuel Emilio Gegúndez Arias



Índice

1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. Experimentación
 - Base de datos
 - Arquitectura
 - Entrenamiento y resultados
 - Análisis
4. Conclusiones

Propuesta del proyecto

1. **Propuesta del proyecto**
2. Introducción al Deep Learning y CNN
3. Experimentación
 - Base de datos
 - Arquitectura
 - Entrenamiento y resultados
 - Análisis
4. Conclusiones



Propuesta del proyecto

- ◎ Aprendizaje de los fundamentos de DL y CNN.
- ◎ Implementación de CNN para clasificación y detección de personas en imágenes.

Introducción al DL y CNN

1. Propuesta del proyecto
2. **Introducción al Deep Learning y CNN**
3. Experimentación
 - Base de datos
 - Arquitectura
 - Entrenamiento y resultados
 - Análisis
4. Conclusiones



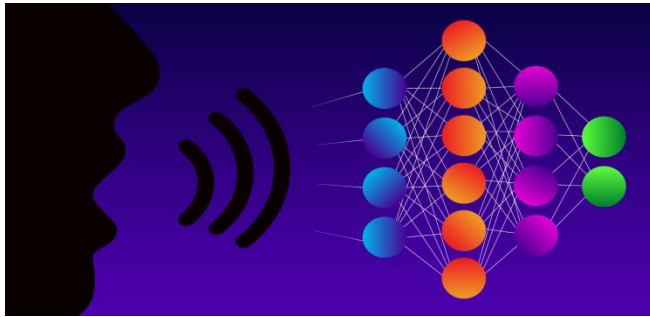
Introducción al DL y CNN

Interés por el término «Deep Learning» en base a las búsquedas realizadas en Google.

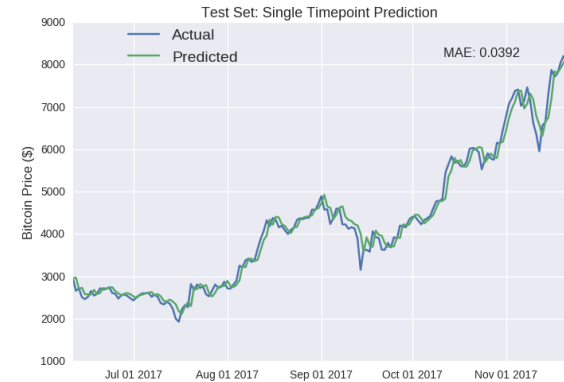


Introducción al DL y CNN

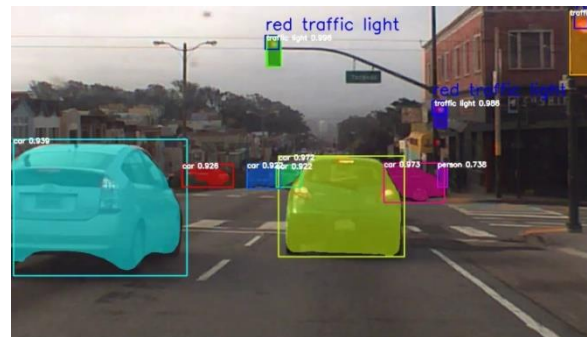
Aplicaciones más exitosas



Procesamiento del habla



Predicción de precios



Sistema de visión artificial
para coches autónomos

Experimentación

1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. **Experimentación**
 - Base de datos
 - Arquitectura
 - Entrenamiento y resultados
 - Análisis
4. Conclusiones



Experimentación

Clasificación

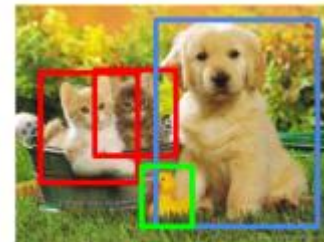
- Base de datos
- Arquitectura
- Entrenamiento y resultados
- Análisis



CAT

Detección

- Base de datos
- Arquitectura
- Entrenamiento y resultados
- Análisis



CAT, DOG, DUCK

Experimentación Clasificación

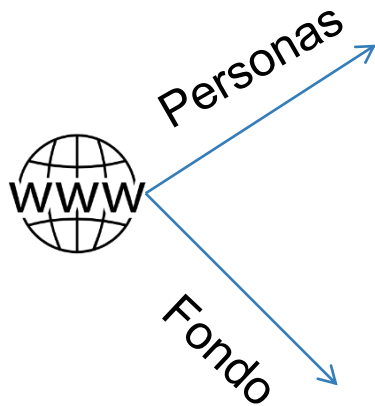
1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. **Experimentación (clasificación)**
 - **Base de datos**
 - Arquitectura
 - Entrenamiento y resultados
 - Análisis
4. Conclusiones



Experimentación Clasificación

◎ Base de datos (imágenes)

- Conjunto de entrenamiento:



Persona_1.bmp



Persona_2.bmp

...



Persona_702.bmp



Fondo_1.bmp



Fondo_2.bmp

...



Fondo_702.bmp

Experimentación Clasificación

◎ Base de datos (imágenes)

- Transformaciones:



Rotación -15°



Volteo



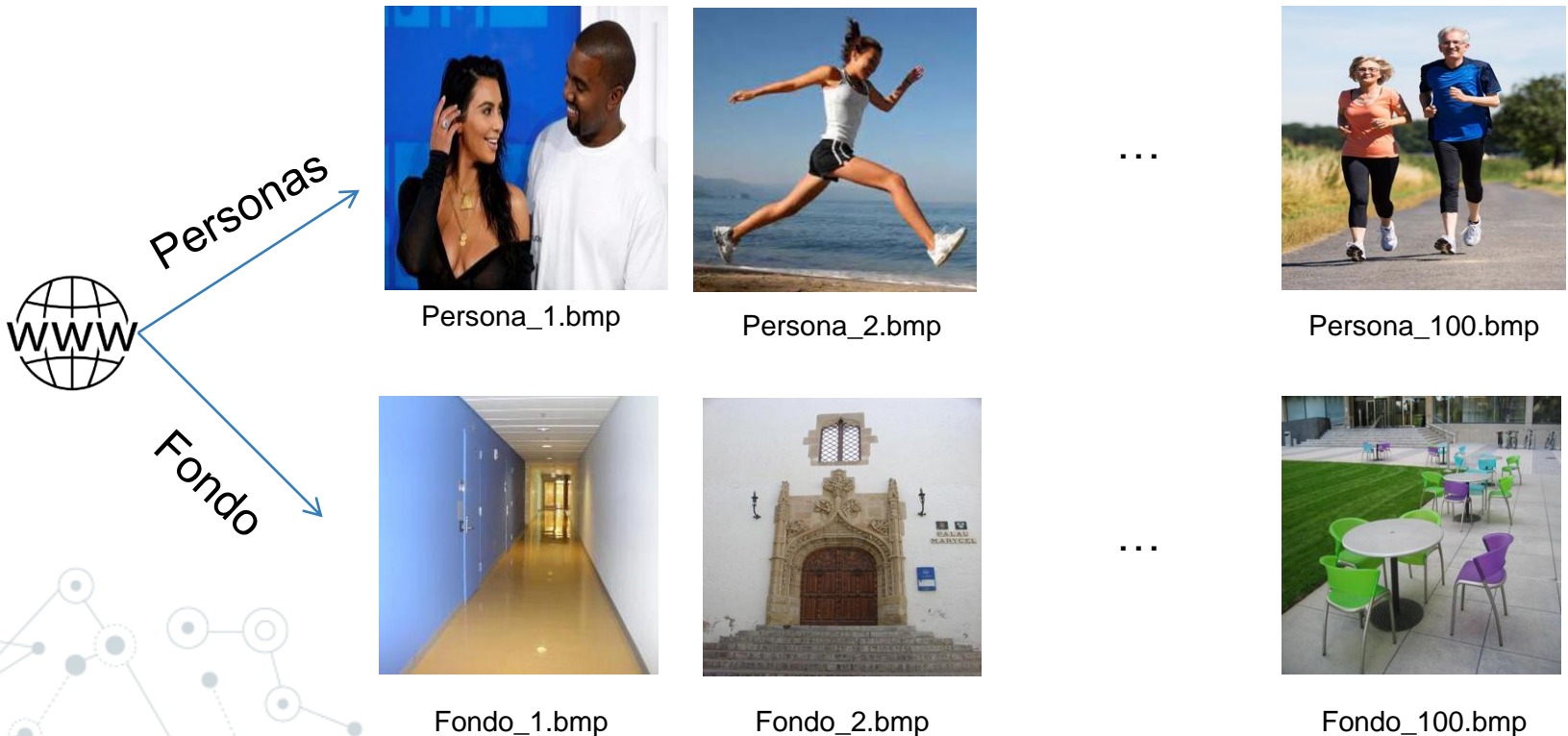
Rotación 15°



Experimentación Clasificación

◎ Base de datos (imágenes)

- Conjunto de test:



Experimentación Clasificación

◎ Base de datos (imágenes)

○ Batch:

- Formado por 14 imágenes de ambas clases
- Misma probabilidad de elegir una imagen de una clase u otra
- En cada iteración formamos un nuevo batch
- Entrenamiento más rápido
- Límite de cantidad de imágenes en el batch por la memoria de la GPU

Experimentación Clasificación

1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. **Experimentación (clasificación)**
 - Base de datos
 - **Arquitectura (Fundamentos teóricos)**
 - Entrenamiento y resultados
 - Análisis
4. Conclusiones



Experimentación Clasificación

◎ Capa Convolutiva

- Tamaño y número de filtros
- Stride
- Padding

Convolutional Neural Network

□ Conv 3x3 with **stride=2, padding=1**

0	0	0	0	0	0	0
0	1	2	1	1	1	0
0	1	1	5	3	9	0
0	2	4	4	7	5	0
0	3	6	7	5	6	0
0	1	6	5	3	1	0
0	0	0	0	0	0	0

5x5 Image



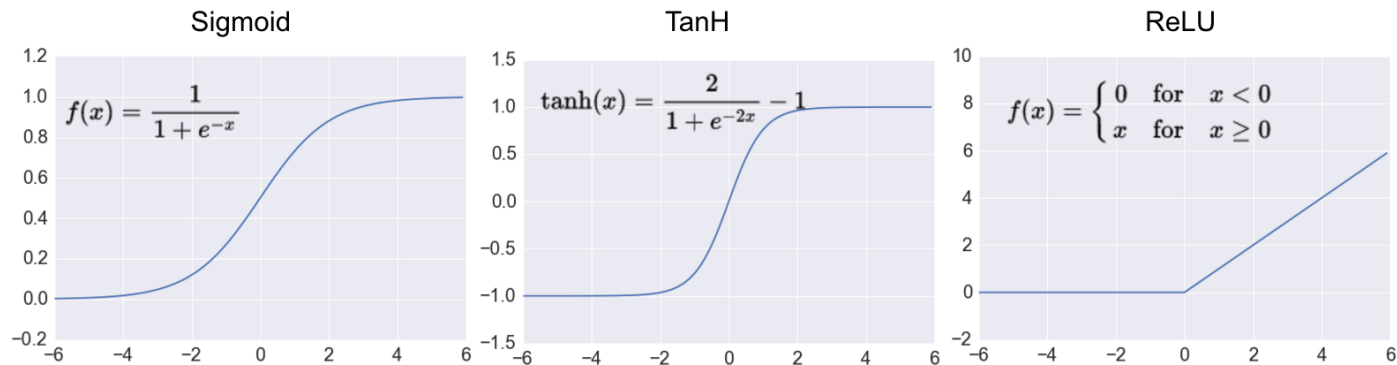
5	13	14
17	42	35
16	32	15

3x3

Experimentación Clasificación

Capa Activación

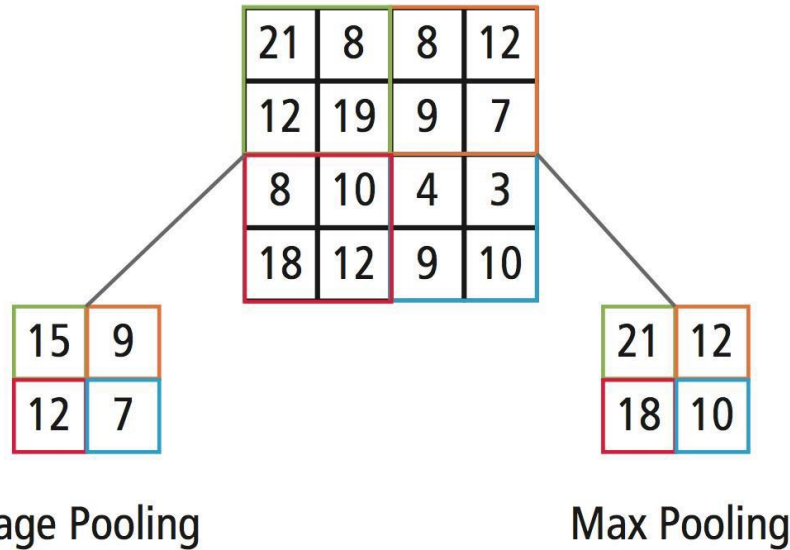
➤ Función a utilizar



Experimentación Clasificación

◎ Capa Pooling

- Tamaño de la ventana
- Stride

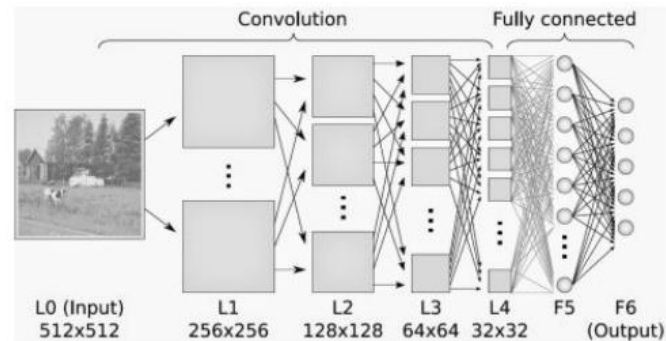


Experimentación Clasificación

◎ Capa FC

- Número de neuronas

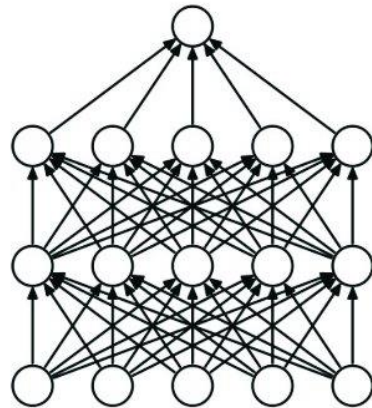
Fully-Connected (FC) layer



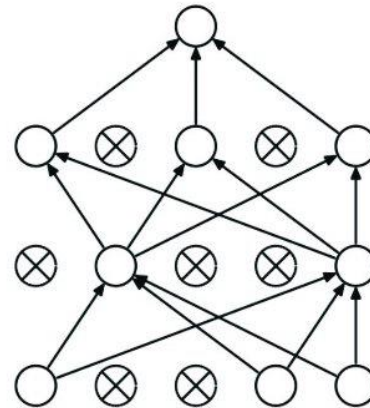
Experimentación Clasificación

◎ Capa Dropout

- Porcentaje de activaciones a desactivar



(a) Standard Neural Network



(b) Neural Net with Dropout

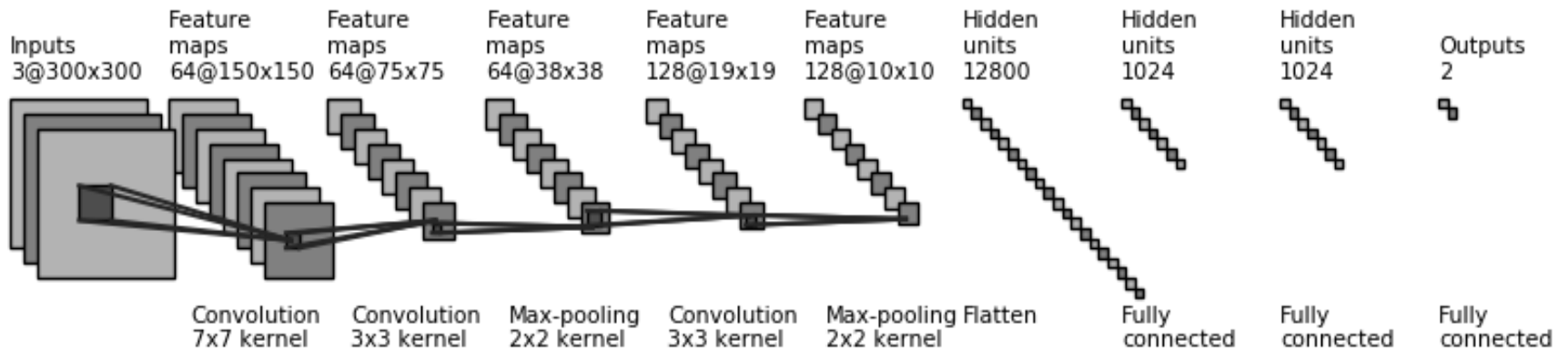
Experimentación Clasificación

1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. **Experimentación (clasificación)**
 - Base de datos
 - **Arquitectura (Implementación)**
 - Entrenamiento y resultados
 - Análisis
4. Conclusiones



Experimentación Clasificación

© Arquitectura



- Con capas dropout entre las capas FC
- Sin weight Decay
- 2 clases

Experimentación Clasificación

© Arquitectura

- Salida de la red:
 - Un array de dos posiciones por imagen de entrada, que nos indicará la probabilidad de que pertenezca a una clase u otra



[Probabilidad persona, Probabilidad no persona]

Experimentación Clasificación

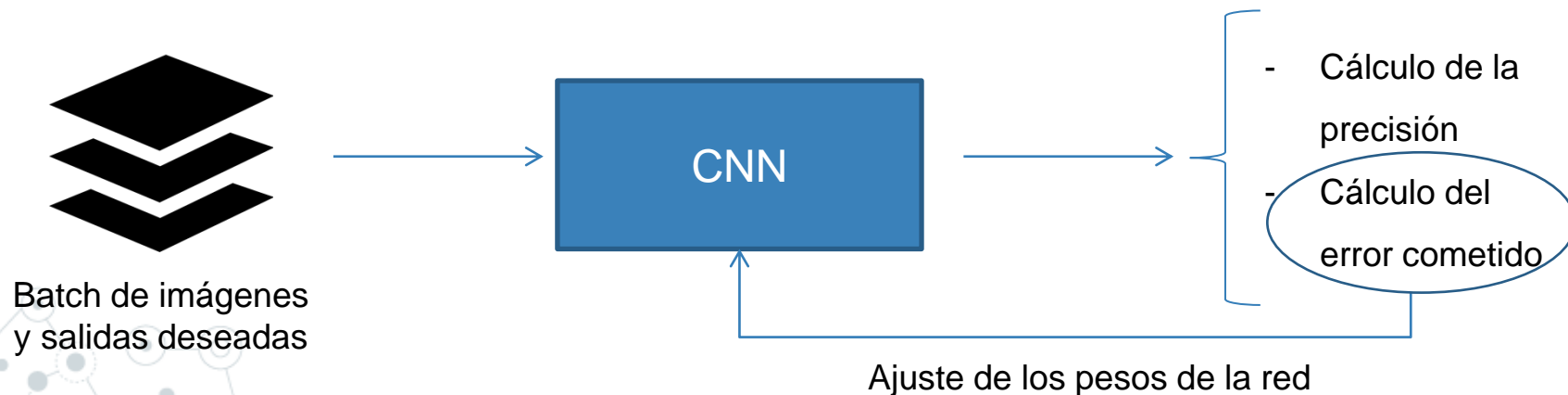
1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. **Experimentación (clasificación)**
 - Base de datos
 - Arquitectura
 - **Entrenamiento y resultados**
 - Análisis
4. Conclusiones



Experimentación Clasificación

© Entrenamiento y resultados

- Batch de 14 imágenes
 - Salida deseada en imágenes de personas $\rightarrow [1.0, 0.0]$
 - Salida deseada en imágenes que no hay personas $\rightarrow [0.0, 1.0]$
- Ajuste de los pesos de la red



Experimentación Clasificación

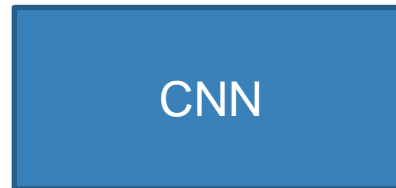
© Entrenamiento y resultados

- Error cometido (entropía cruzada):

```
error = -tf.reduce_mean(y_true * tf.log(y_dev_soft + 1e-10))
```



Salida deseada:
[1.0, 0.0]



Salida de la red:
[0.6, 0.4]

Experimentación Clasificación

© Entrenamiento y resultados

➤ Optimizador Adam:

- Según un estudio (<https://arxiv.org/pdf/1705.08292.pdf>):
 - ☐ Puede no ser la mejor opción, pero suele trabajar bien para diferentes datasets
 - ☐ De los optimizadores que más rápido convergen y por lo tanto la duración del entrenamiento es menor
- El más utilizado por la comunidad de las CNN

Experimentación Clasificación

© Entrenamiento y resultados

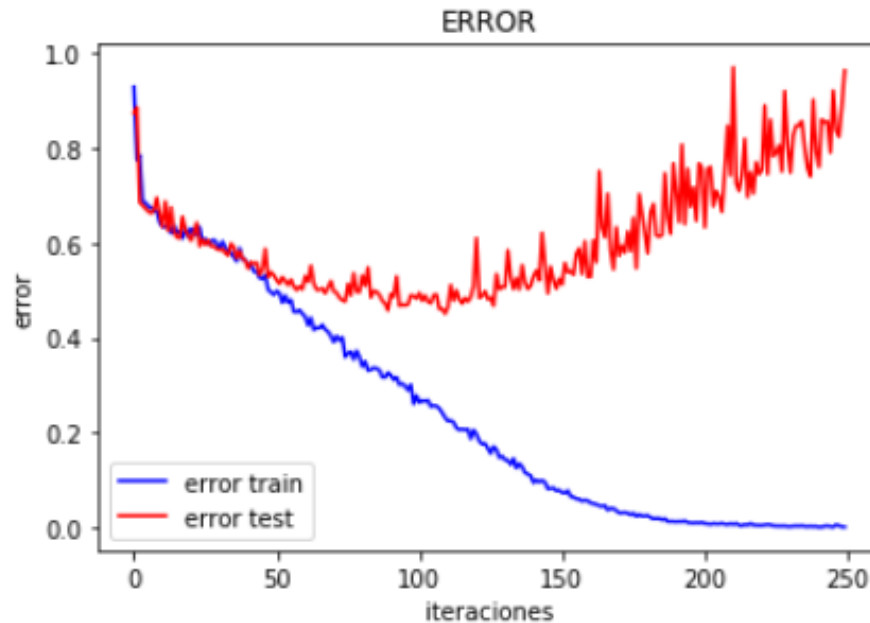
➤ Precisión:

```
prec_img = tf.equal(tf.argmax(y_dev_soft, 1), tf.argmax(y_true, 1))
```

```
prec_conj = (( $\sum$  Prec_img) / Num_imgs_conj) * 100
```

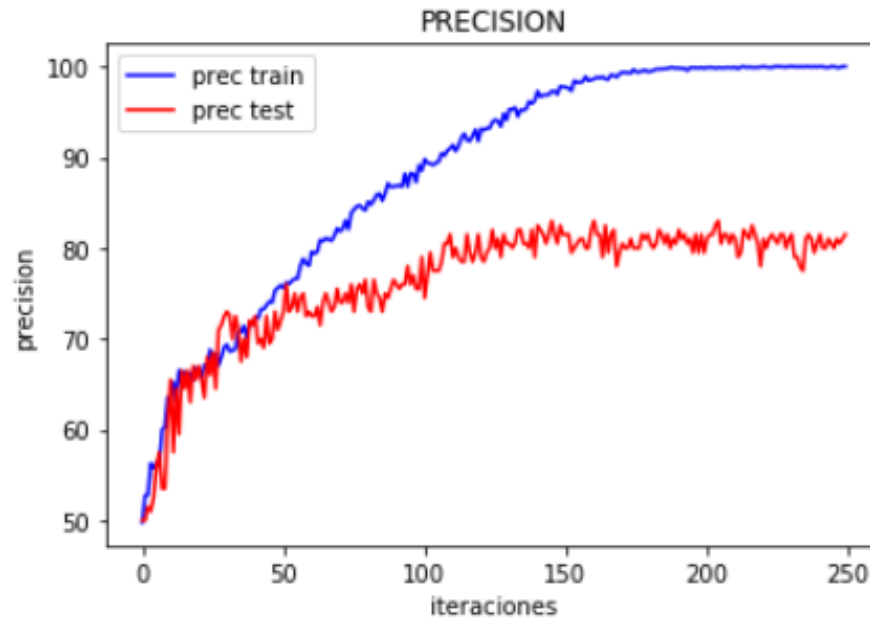
Experimentación Clasificación

© Entrenamiento y resultados



Experimentación Clasificación

© Entrenamiento y resultados



Experimentación Clasificación

© Entrenamiento y resultados

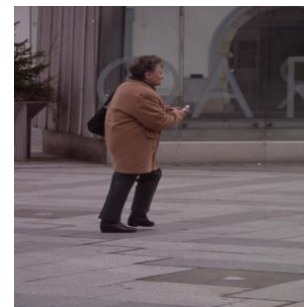
TRAIN



[0.6, 0.4]



[0.94, 0.06]



[0.74, 0.26]



[0.35, 0.65]



[0.14, 0.86]



[0.32, 0.68]

Experimentación Clasificación

© Entrenamiento y resultados

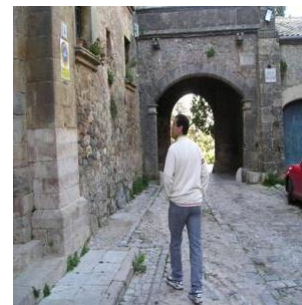
TEST



[0.56, 0.44]



[0.78, 0.22]



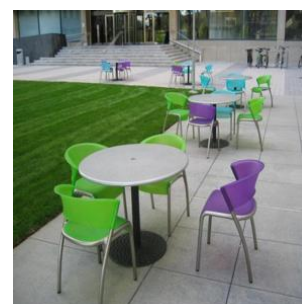
[0.36, 0.64]



[0.32, 0.68]



[0.34, 0.66]



[0.26, 0.74]

Experimentación Clasificación

1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. **Experimentación (clasificación)**
 - Base de datos
 - Arquitectura
 - Entrenamiento y resultados
 - **Análisis**
4. Conclusiones



Experimentación Clasificación

© Análisis

- Sobreaprendizaje (overfitting)
- Posible solución al sobreaprendizaje producido

Experimentación Detección

1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. **Experimentación (detección)**
 - **Base de datos**
 - Arquitectura
 - Entrenamiento y resultados
 - Análisis
4. Conclusiones



Experimentación Detección

© Base de datos (imágenes)

- Sólo las imágenes con personas



Etiquetado

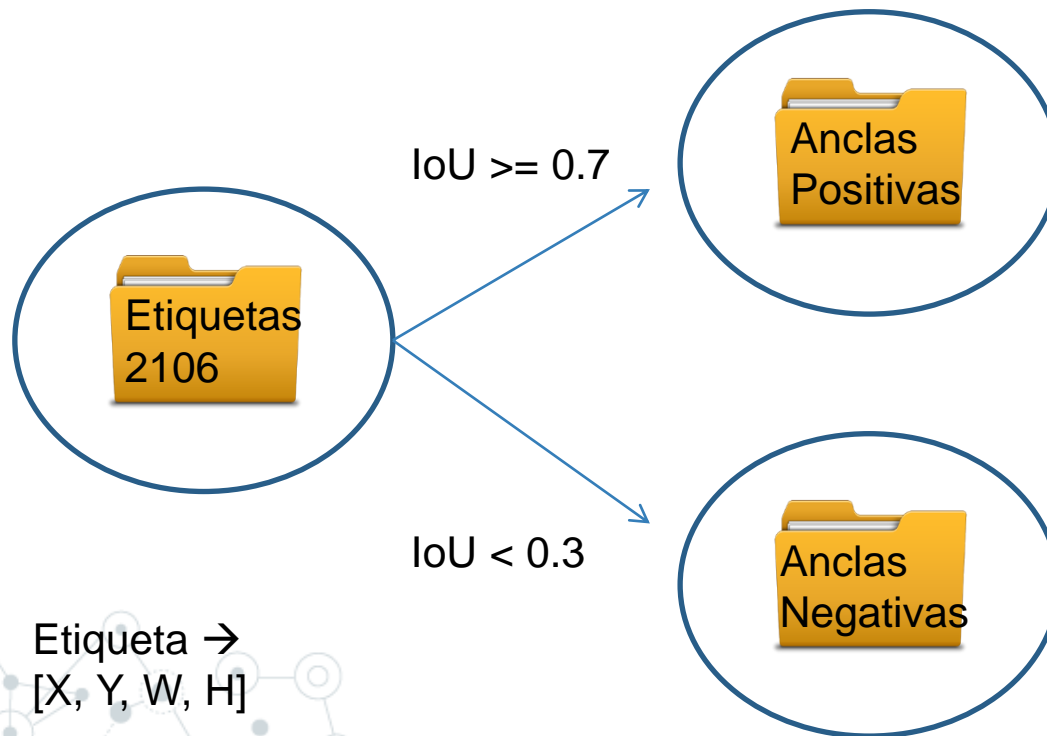
Labeling
(<https://github.com/zutalin/labelImg>)




Experimentación Detección

◎ Base de datos (imágenes)

○ Etiquetado de Anclas



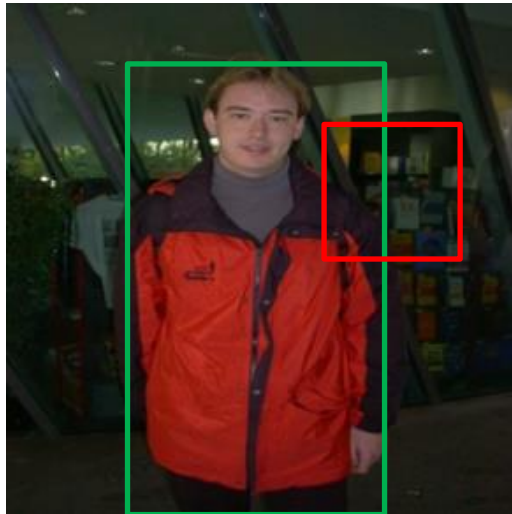
$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Intersection over Union (IoU)

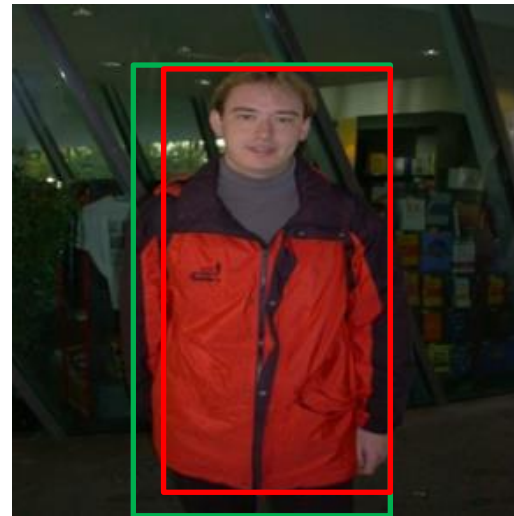
Experimentación Detección

◎ Base de datos (imágenes)

- Etiquetado de Anclas



$\text{IoU} = 0.2$ (Ancla negativa)



$\text{IoU} = 0.9$ (Ancla positiva)

Experimentación Detección

1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. **Experimentación (detección)**
 - Base de datos
 - **Arquitectura (Fundamentos teóricos)**
 - Entrenamiento y resultados
 - Análisis
4. Conclusiones

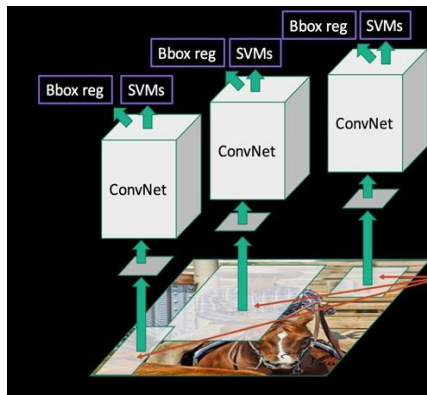


Experimentación Detección

Arquitecturas CNN para detección

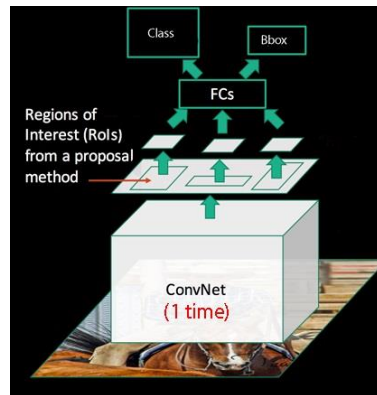
◎ R-CNN

- 50 segundos por img.



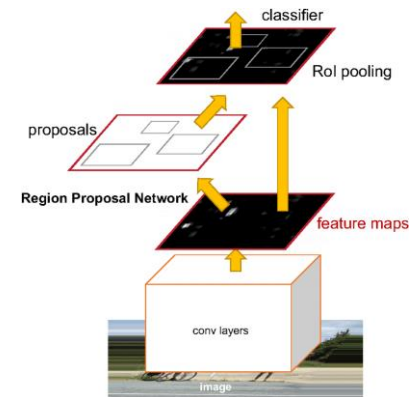
◎ Fast R-CNN

- 2 segundos por img.



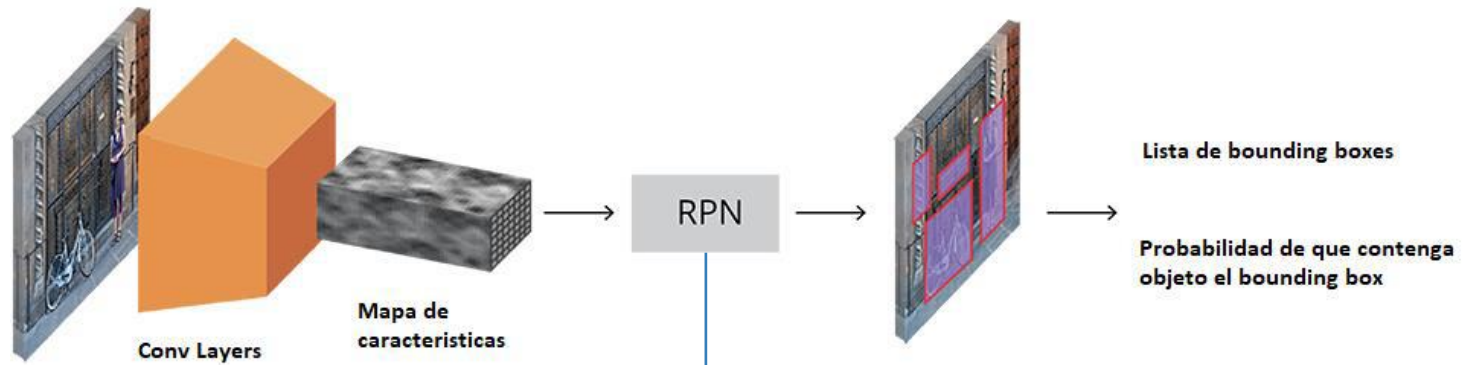
◎ Faster R-CNN

- 0.2 segundos por img.



Experimentación Detección

© Arquitectura Faster R-CNN (<https://arxiv.org/pdf/1506.01497.pdf>)



- Capas preentrenadas
- Congelar pesos o dejar ajustándose en el nuevo entrenamiento

«Region Proposal Network»

Experimentación Detección

© Arquitectura Faster R-CNN

Anclas

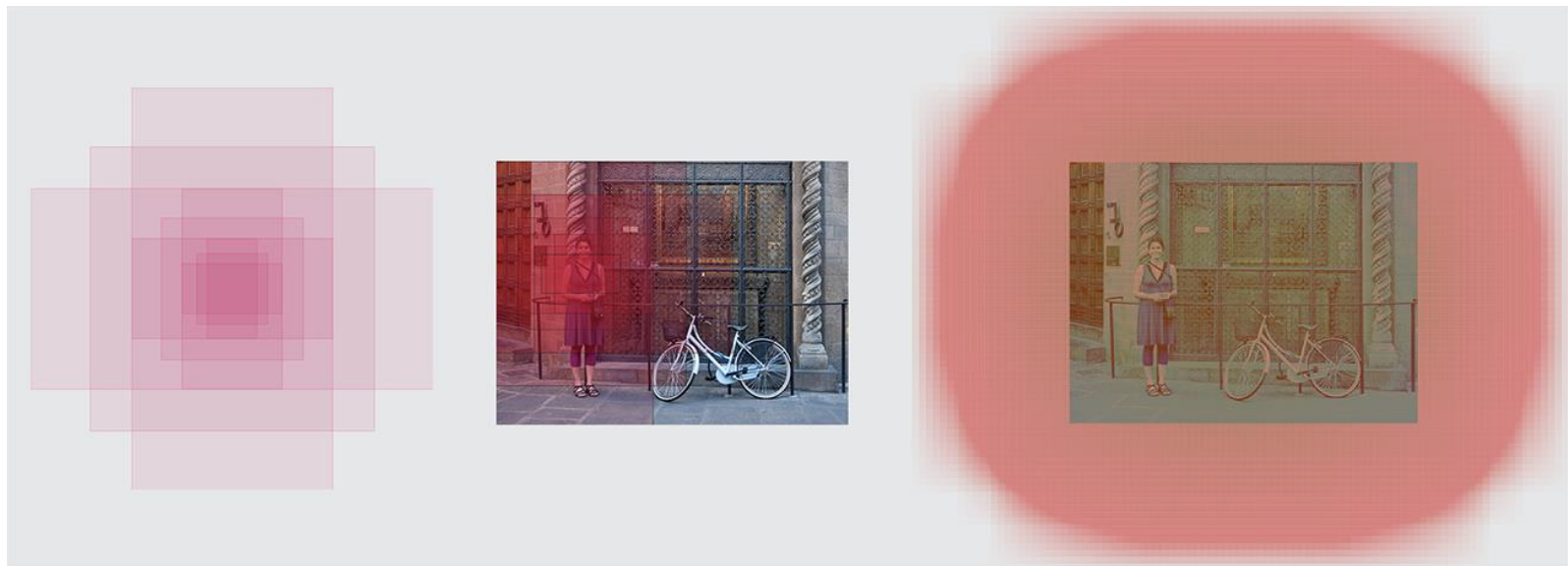
- Sirven de referencia
- Fijas para cualquier imagen de entrada
- Distintos tamaños y relaciones de aspecto
- Separadas por 'r' píxeles

$$r = \frac{\text{tamaño imagen}}{\text{tamaño mapa características}}$$



Experimentación Detección

© Arquitectura Faster R-CNN



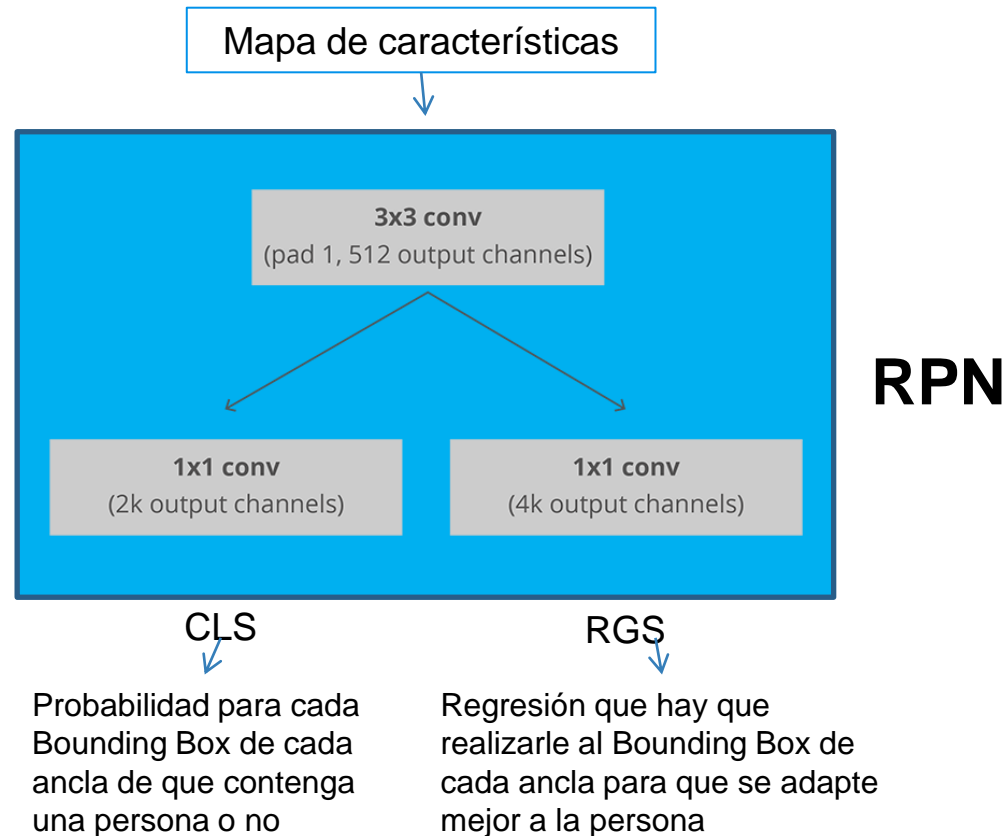
Bounding Boxes
por cada ancla

Bounding Boxes de
un ancla sobre la
imagen de entrada

Bounding Boxes de
todas las anclas sobre
la imagen de entrada

Experimentación Detección

© Arquitectura Faster R-CNN



Experimentación Detección

© Arquitectura Faster R-CNN

- CLS:

Matriz CLS (19x19x9)

[0.4, 0.6]	[0.3, 0.7]	...	[0.0, 1.0]	
[0.9, 0.1]	[0.7, 0.3]	...	[0.1, 0.9]	
...	
[0.2, 0.8]	[0.3, 0.7]	...	[0.7, 0.3]	

Experimentación Detección

© Arquitectura Faster R-CNN

- RGS:

Matriz RGS (19x19x9)

[5, 6, 9, 15]	[-3, 7, 6, 10]	...	[-5, 1, 6, 2]
[1, -1, 3, -5]	[7, -3, 3, 10]	...	[1, -9, 5, 10]
...
[2, 8, 6, 7]	[3, 7, 4, -3]	...	[-7, 3, 4, 20]

+

Bounding Boxes anclas
(19x19x9)

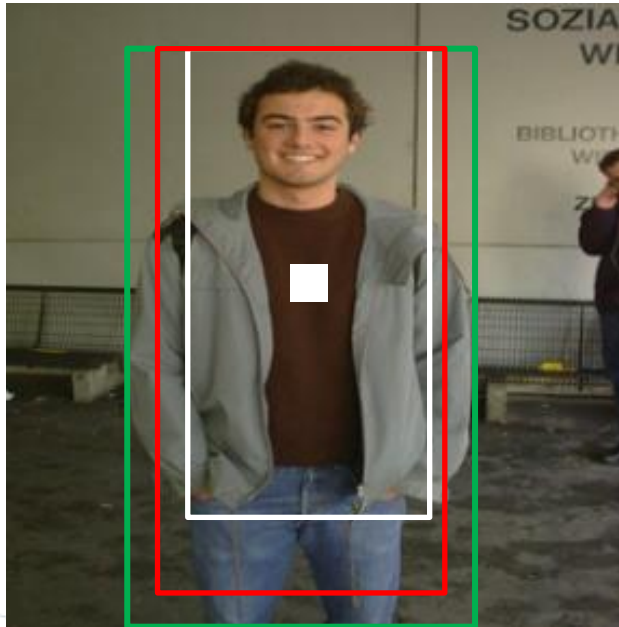
[-50, -50, 100, 100]	[-35, -50, 100, 100]	...	[220, -50, 100, 100]
[-50, -35, 100, 100]	[-35, -35, 100, 100]	...	[220, -35, 100, 100]
...
[-50, 220, 100, 100]	[-35, 220, 100, 100]	...	[220, 220, 100, 100]

Ajuste →
[X, Y, W, H]

Experimentación Detección

© Arquitectura Faster R-CNN

- RGS:



Bounding Box Real



Bounding Box Devuelto
(BB Ancla + RGS)



Bounding Box Ancla

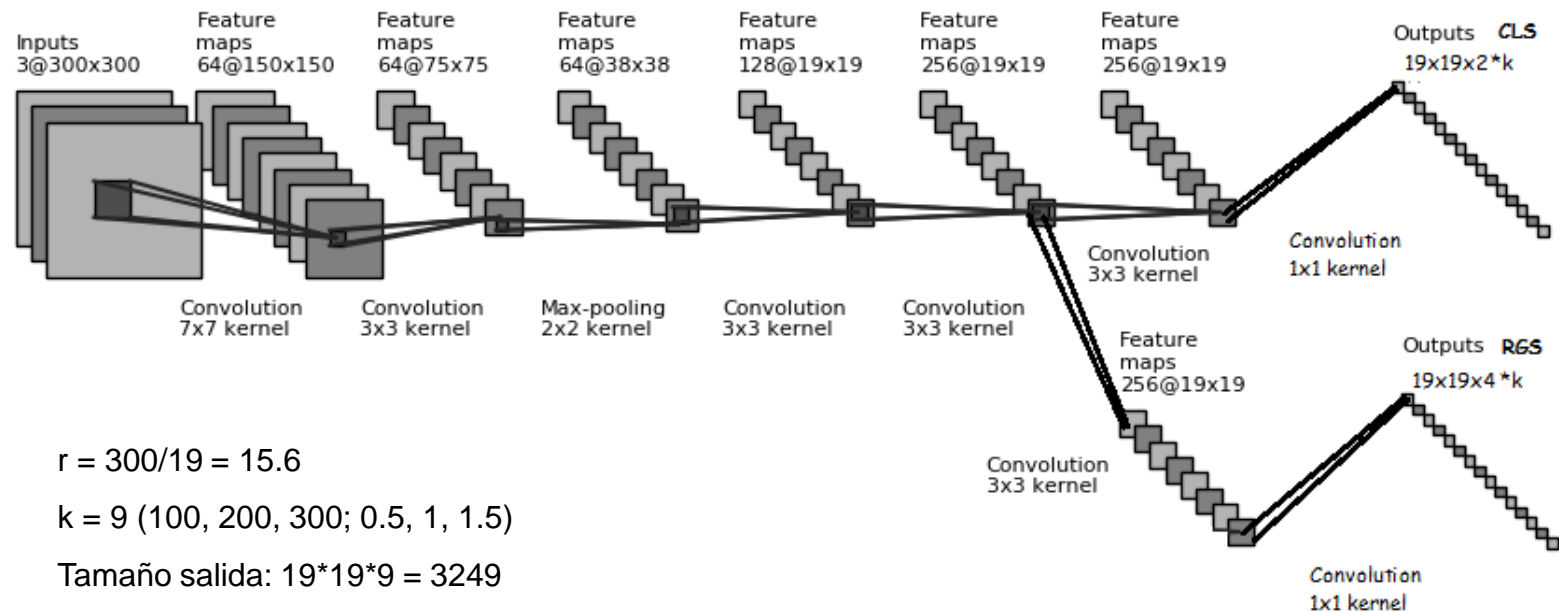
Experimentación Detección

1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. **Experimentación (detección)**
 - Base de datos
 - **Arquitectura (Implementación)**
 - Entrenamiento y resultados
 - Análisis
4. Conclusiones



Experimentación Detección

© Arquitectura proyecto



$$r = 300/19 = 15.6$$

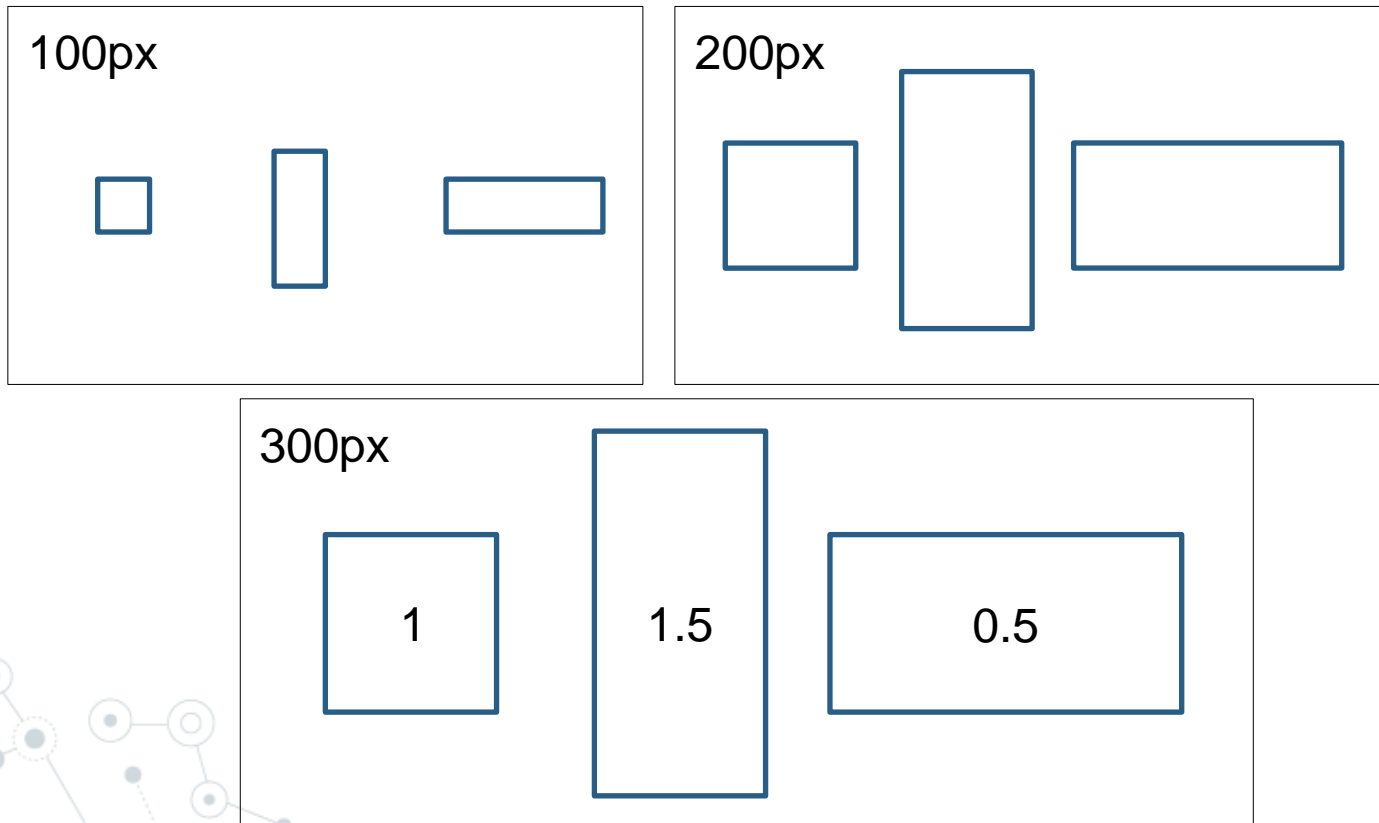
$$k = 9 (100, 200, 300; 0.5, 1, 1.5)$$

$$\text{Tamaño salida: } 19 \times 19 \times 9 = 3249$$

Sin weight Decay ni capas Dropout

Experimentación Detección

© Arquitectura proyecto (Anclas)



Experimentación Detección

1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. **Experimentación (detección)**
 - Base de datos
 - Arquitectura (Implementación)
 - **Entrenamiento y resultados**
 - Análisis
4. Conclusiones



Experimentación Detección

© Entrenamiento y resultados

- Batch de anclas positivas y negativas
- Cálculo de las pérdidas CLS y RGS.
- Optimizador Adam (Loss CLS + Loss RGS)
 - CLS: Pérdida → Entropía cruzada (muestras + y -)
 - RGS: Pérdida → Smooth_L1 (Sólo muestras +)

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

$$\text{Diff} = \text{BB_real} - \text{BB_pred} = [x_1, x_2, x_3, x_4]$$

$$\text{Loss_Tot} = \sum \text{smooth_L1}(x_i)$$

Experimentación Detección

© Entrenamiento y resultados

➤ Pérdida CLS:

Matriz CLS (19x19x9)

[0.4, 0.6]	[0.3, 0.7]	...	[0.0, 1.0]
[0.9, 0.1]	[0.7, 0.3]	...	[0.1, 0.9]
...
[0.2, 0.8]	[0.3, 0.7]	...	[0.7, 0.3]

Batch:

[1.0, 0.0] → Pos. 2,1,1

[1.0, 0.0] → Pos. 1,1,3

[0.0, 1.0] → Pos. 5,9,5

[0.0, 1.0] → Pos. 7,2,8

Experimentación Detección

Entrenamiento y resultados

➤ Pérdida RGS:

Matriz RGS (19x19x9)

[5, 6, 9, 15]	[-3, 7, 6, 10]	...	[-5, 1, 6, 2]
[1, -1, 3, -5]	[7, -3, 3, 10]	...	[1, -9, 5, 10]
...
[2, 8, 6, 7]	[3, 7, 4, -3]	...	[-7, 3, 4, 20]

+

Bounding Boxes anclas
(19x19x9)

[-50, -50, 100, 100]	[-35, -50, 100, 100]	...	[220, -50, 100, 100]
[-50, -35, 100, 100]	[-35, -35, 100, 100]	...	[220, -35, 100, 100]
...
[-50, 220, 100, 100]	[-35, 220, 100, 100]	...	[220, 220, 100, 100]

Batch:

[32, 16, 100, 100] → Pos.

2,1,1

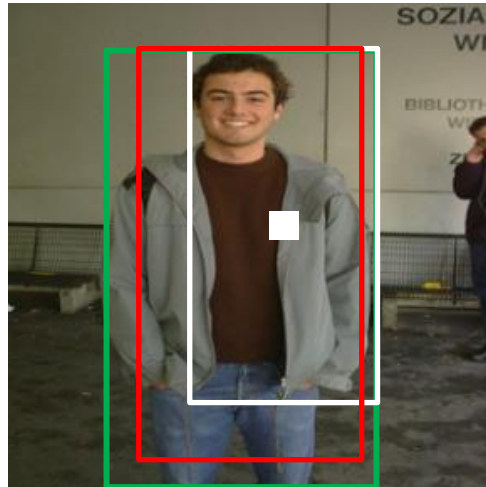
[16, 16, 100, 200] → Pos.




1,1,3

Experimentación Detección

© Entrenamiento y resultados

➤ Pérdida RGS:



-  Bounding Box Real
-  Bounding Box Ancla
-  Bounding Box Devuelto (BB Ancla + RGS)

Bounding Box Devuelto = [55, 30, 150, 265]

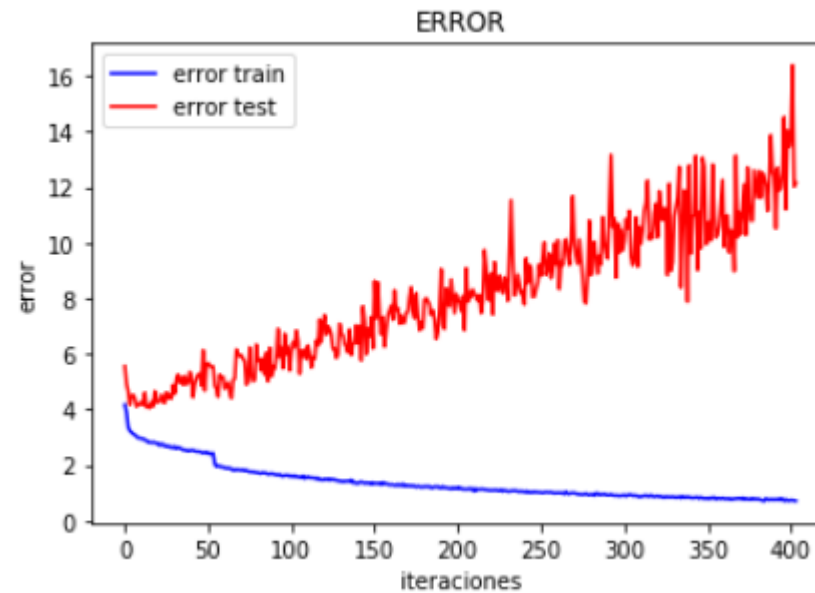
Bounding Box Real = [48, 32, 160, 270]

Diferencia = $BB_Dev - BB_Real = [7, -2, -10, -5]$

Error = $Smooth_L1(7) + Smooth_L1(-2) + Smooth_L1(-10) + Smooth_L1(-5)$

Experimentación Detección

© Entrenamiento y resultados



Experimentación Detección

© Entrenamiento y resultados

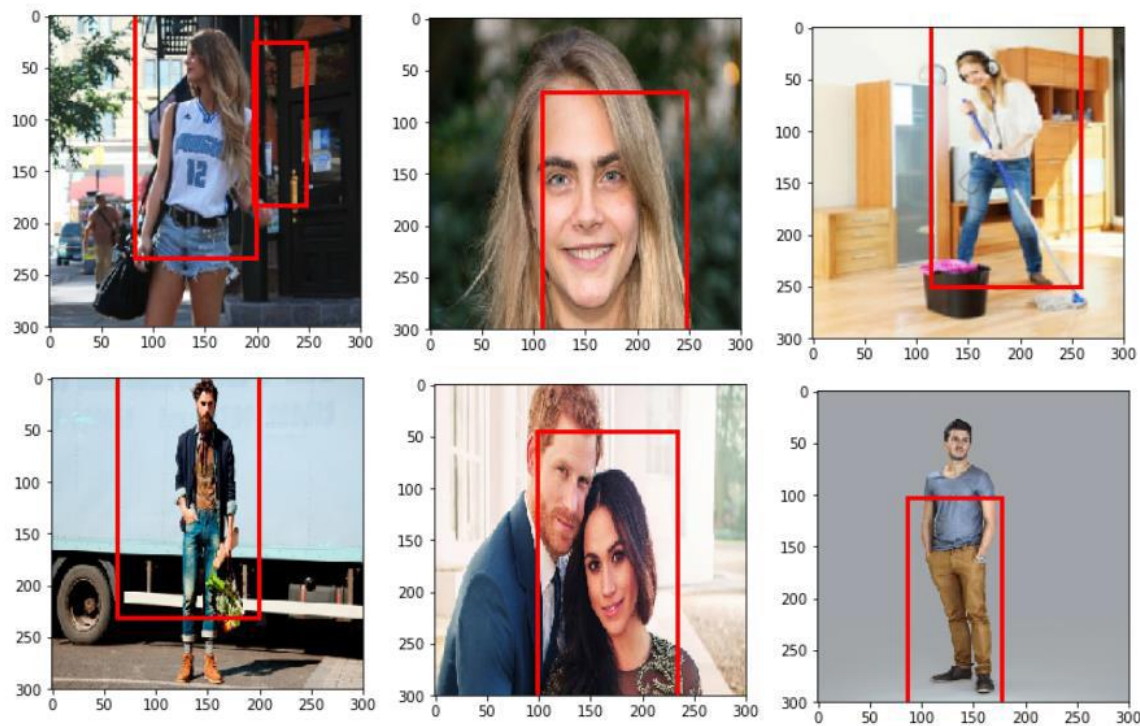
TRAIN



Experimentación Detección

© Entrenamiento y resultados

TEST



Experimentación Detección

1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. **Experimentación (detección)**
 - Base de datos
 - Arquitectura (Implementación)
 - Entrenamiento y resultados
 - **Análisis**
4. Conclusiones



Experimentación Detección

© Análisis

- Correcto funcionamiento a pesar de la baja precisión.
- Sobreaprendizaje (overfitting)
- Posible solución al sobreaprendizaje producido

Experimentación Detección

1. Propuesta del proyecto
2. Introducción al Deep Learning y CNN
3. Experimentación
 - Base de datos
 - Arquitectura (Implementación)
 - Entrenamiento y resultados
 - Análisis
4. **Conclusiones**



Conclusiones

◎ Objetivos conseguidos

- Fundamentos de DL y CNN
- Lenguaje de programación Python y TensorFlow
- Uso de la API de Tensorflow
- Técnicas de aumento de datos para imágenes
- Implementación de un sistema de clasificación y de detección de personas en imágenes mediante CNN

Conclusiones

🎯 Líneas futuras

- Reentrenamiento del modelo con una base de datos con mayor cantidad de imágenes
- Transferencia del aprendizaje mediante la API de TensorFlow

Preguntas

