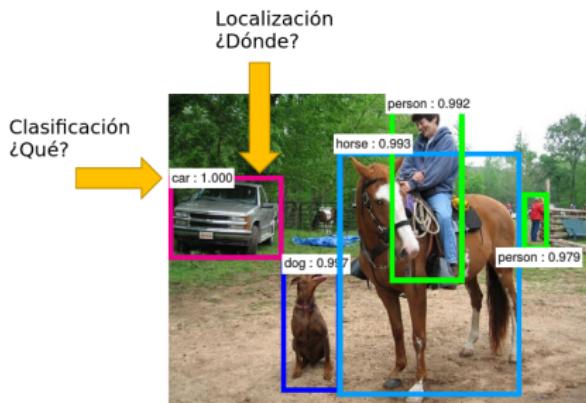


Detección de Objetos Usando Redes Neuronales Convolucionales junto con Random Forest y Support Vector Machines



Alumno : Diego Campanini García

Profesor Guía: Javier Ruiz del Solar
San Martín

Comisión: Pablo Estévez Valencia
Felipe Tobar Henríquez

Universidad de Chile

2 de mayo de 2018

Índice

1 Objetivos

- Objetivo General
- Objetivos Específicos

2 Antecedentes Generales

- CNNs para Clasificación de Imágenes
- CNNs para Detección de Objetos

3 Metodología e Implementación

- Propuesta de Mejoramiento del Sistema Faster R-CNN
- Entrenamiento de los Clasificadores
- Implementación del Sistema Faster R-CNN+RF
- Implementación del Sistema Faster R-CNN+SVM

4 Resultados y Análisis

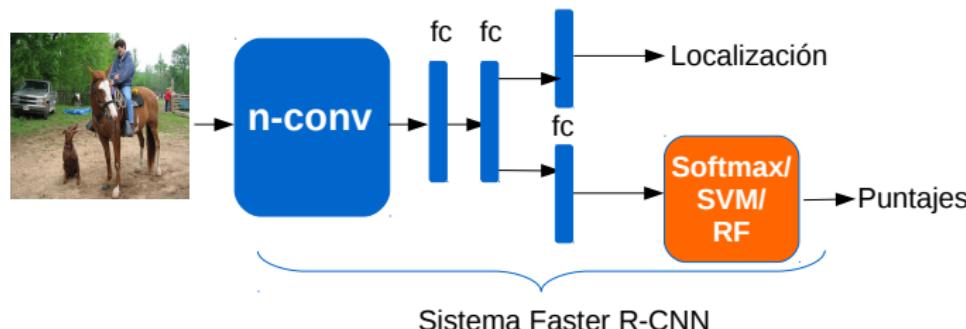
- Replicación Resultados Sistema Faster R-CNN
- Resultados Método de Etiquetado 1
- Resultados Método de Etiquetado 2
- Resumen Mejores Resultados
- Ejemplos de Detecciones

5 Conclusiones

6 Bibliografía

Objetivo General

Mejorar el sistema de detección conocido como **Faster R-CNN**. Para esto se propone usar los clasificadores **Random Forest** y **Support Vector Machines**, reemplazando con estos la función **Softmax** encargada de generar los puntajes de confianza de la clasificación.



Objetivos Específicos

Analizar el efecto de utilizar las mejores implementaciones¹ (librerías) de RF y SVMs.

Evaluar la capacidad del sistema Faster R-CNN para generar vectores de características que permitan entrenar clasificadores como RF y SVMs.

Evaluar distintos métodos de etiquetado de los vectores de características.

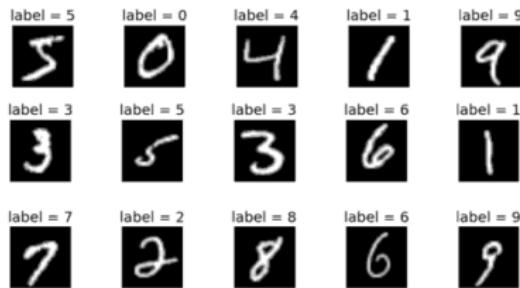
Evaluar con cuál capa completamente conectada se obtiene un mayor rendimiento para SVMs.

Implementar técnicas que permitan el entrenamiento eficiente, en tiempo y en uso de memoria, para SVMs.

¹ Manuel Fernández-Delgado et al. Do we need hundreds of classifiers to solve real world classification problems?

CNNs para Clasificación de Imágenes

Tarea: Asignar la clase correcta a la imagen completa



(a) Clasificación de Dígitos (MNIST)

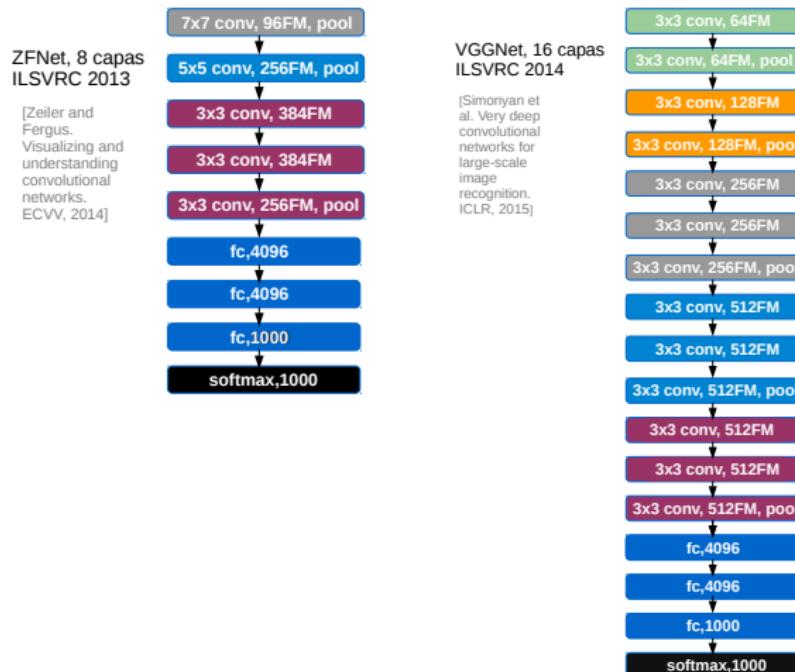


(b) Clasificación de Objetos (PASCAL, ImageNet)

LeNet: Red convolucional desarrollada por Yann LeCun en los 90. Se utilizó con éxito para la clasificación de dígitos manuscritos. [LeCun et al. Handwritten digit recognition with a back-propagation network. NIPS, 1990.](#)

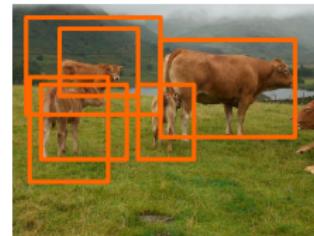
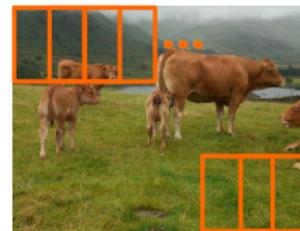
AlexNet: El primer trabajo que popularizó las CNNs en el área de visión por computador. Superó significativamente al segundo lugar en Imagenet 2012 (top-5 error de 15.3 % comparado con 26.2 % del segundo lugar) [Krizhevsky, et al. Imagenet classification with deep convolutional neural networks. NIPS, 2012.](#)

CNNs para Clasificación de Imágenes



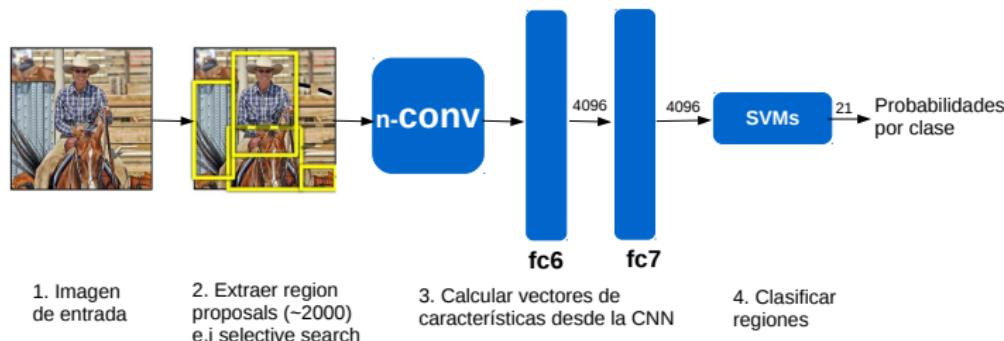
CNNs para Detección de Objetos

- ▶ Dado el éxito en clasificación cómo aplicar las **CNNs para detección**.
- ▶ **Problema:** cómo localizar un objeto
- ▶ **Soluciones:**
 - ▶ **Ventana deslizante:** Se realiza un barrido exhaustivo por la imagen y se genera una gran cantidad de ejemplos (10^4 a 10^7) ⇒ **Costoso**
 - ▶ **Object proposal:** Se genera un conjunto de *bounding boxes* propuestos para guiar la búsqueda de objetos. ⇒ **Más eficiente**



CNNs para Detección de Objetos: R-CNN

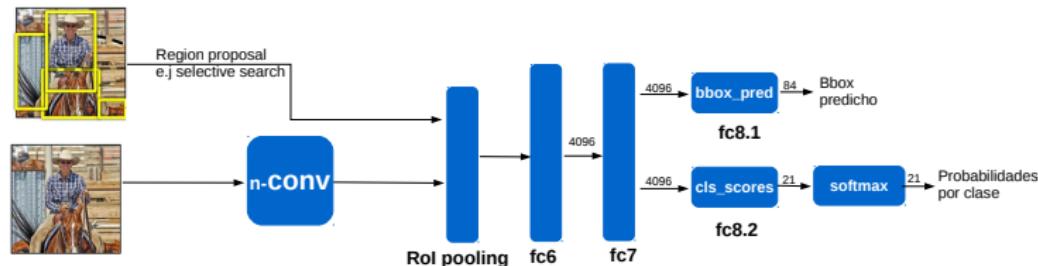
R-CNN: Regions with CNN features



Inconvenientes de R-CNN:

- Depende de un método externo para generar los *region proposals*
- Entrenamiento en múltiples etapas
- Lento en la detección de objetos (13 s por imagen)
- Rendimiento en PASCAL VOC 2007 de 66.0 % mAP

CNNs para Detección de Objetos: Fast R-CNN



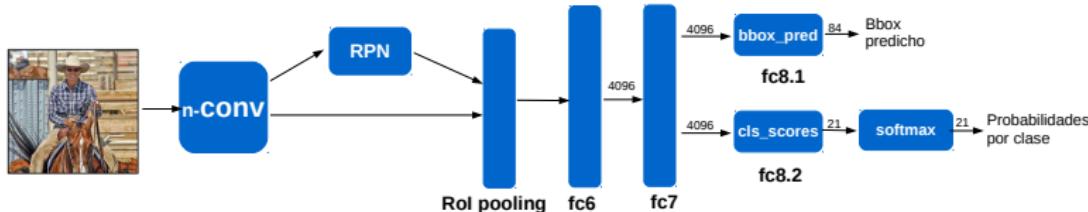
Mejoras con respecto a R-CNN:

- Entrenamiento en una sola etapa. Se utiliza *softmax* y se incluye una capa para localización.
- Las capas convolucionales procesan la imagen completa y no cada *proposal* logrando una detección más rápida que R-CNN.
- El tiempo de ejecución es de 300 ms por imagen
- Rendimiento en PASCAL VOC 2007 de 66.9 % mAP

Persiste un notorio inconveniente:

- Dependencia de un método externo de *region proposals*.

CNNs para Detección de Objetos: Faster R-CNN



Mejoras con respecto a Fast R-CNN:

- La obtención de los *proposals* se incluye en la red convolucional. Se introduce una *Region Proposal Network* (RPN).
- La obtención de los *proposals* es marginal: 10 ms
- El tiempo de ejecución es de 198 ms por imagen
- Rendimiento en PASCAL VOC 2007 de 70.0 % mAP

1 Objetivos

 Objetivo General
 Objetivos Específicos

2 Antecedentes Generales

 CNNs para Clasificación de Imágenes
 CNNs para Detección de Objetos

3 Metodología e Implementación

 Propuesta de Mejoramiento del Sistema Faster R-CNN
 Entrenamiento de los Clasificadores
 Implementación del Sistema Faster R-CNN+RF
 Implementación del Sistema Faster R-CNN+SVM

4 Resultados y Análisis

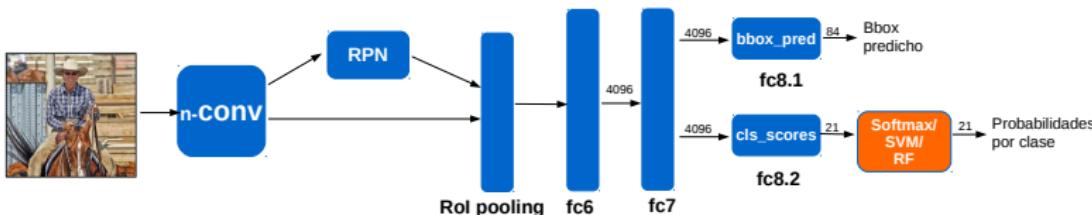
 Replicación Resultados Sistema Faster R-CNN
 Resultados Método de Etiquetado 1
 Resultados Método de Etiquetado 2
 Resumen Mejores Resultados
 Ejemplos de Detecciones

5 Conclusiones

6 Bibliografía

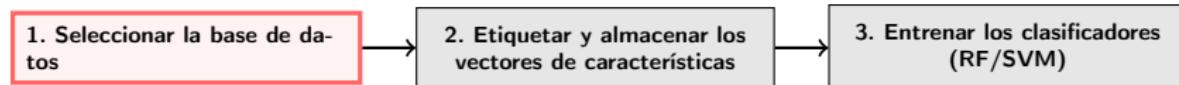
Propuesta de Mejoramiento del Sistema Faster R-CNN

- Generar los puntajes de salida con los clasificadores **SVM** y **RF**, reemplazando la función **softmax**.
- Probar las mejores implementaciones (librerías) de SVM y RF según ²



² Manuel Fernández-Delgado et al. Do we need hundreds of classifiers to solve real world classification problems? JMLR, January 2014.

¿Cómo Entrenar los Clasificadores?

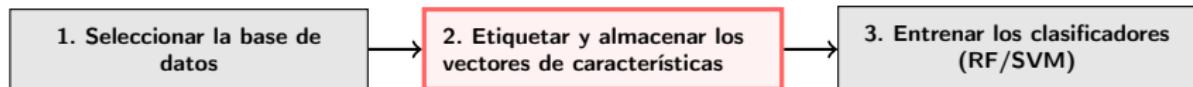


PASCAL VOC 2007

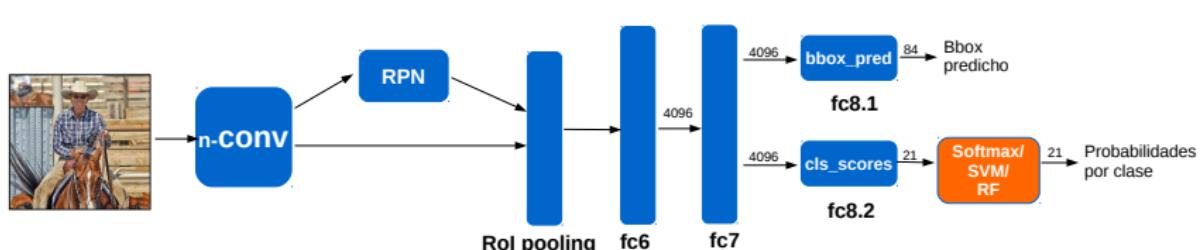


	Img	Obj	aero	bike	bird	boat	bott	bus	car	cat	chr	cow	dtbl	dog	horse	mbk	prsn	plnt	shp	sofa	train	tv
trainval	5.011	12.608	306	353	486	290	505	229	1.250	376	798	259	215	510	362	339	4.690	514	257	248	297	324
test	4.952	12.032	285	337	459	263	469	213	1.201	358	756	244	206	489	348	325	4.528	480	242	239	282	308

¿Cómo Entrenar los Clasificadores?



- ▶ Por cada imagen que entra a la red se generan 300 *proposals*. Por cada uno de estos se tendrá un vector de características.
- ▶ Para entrenar RF se utilizan los vectores de la capa **fc8.2** (dim 21).
- ▶ Para SVM, además se prueban los vectores extraídos desde las capas **fc7** y **fc6** (dim 4096).

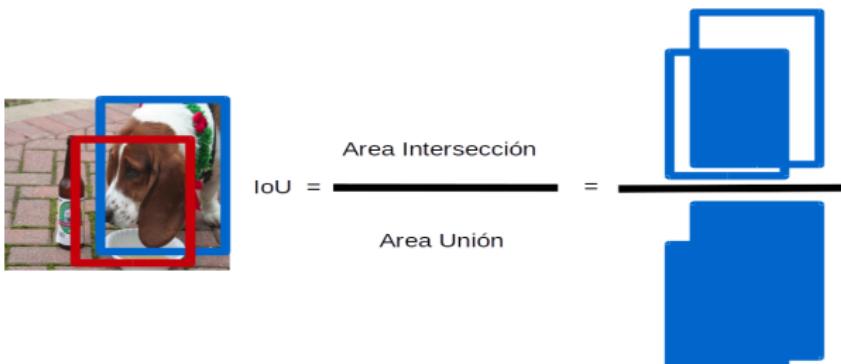


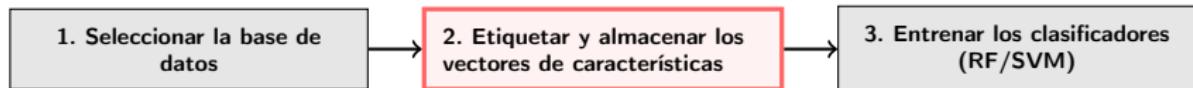
¿Cómo Entrenar los Clasificadores?

Cada *proposal* se debe etiquetar con una de las 21 clases. ¿Cómo? ⇒ Usando **Intersection over Union** (IoU)

- IoU es una métrica usada para evaluar la precisión de la detección de un objeto dado un **bounding box predicho** b_{ij} y el **bounding box ground truth** B_{gt} .

$$\text{IoU}(B_{gt}, b_{ij}) = \frac{\text{Area}(B_{gt} \cap b_{ij})}{\text{Area}(B_{gt} \cup b_{ij})} \quad (1)$$





- ▶ Usando $IoU(B_{gt}, b_{ij})$ se implementan dos métodos de etiquetado:

Método 1 de Etiquetado

- 1 Para cada imagen en el conjunto de entrenamiento etiquetar cada proposal de la siguiente forma:
 - 1.1 Si $IoU(B_{gt}, b_{ij}) \geq 0.5$ para alguno de los bbox del ground truth considerar al proposal como perteneciente a una de las 20 clases.
 - 1.2 $IoU(B_{gt}, b_{ij}) \in [0.1, 0.5]$ ⇒ se considera como fondo
- 2 Redistribuir los ejemplos, tal que la clase fondo corresponda al 90 % de las muestras.

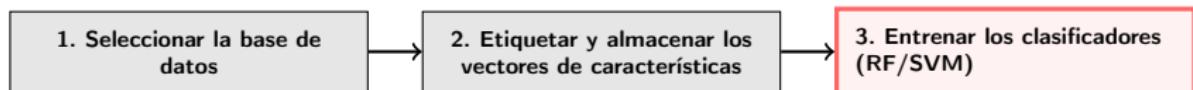
He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV, 2014.
Girshick. Fast r-cnn. ICCV, 2015.

Método 2 de Etiquetado

- 1 Para cada imagen en el conjunto de entrenamiento etiquetar cada proposal de la siguiente forma:
 - 1.1 Si $IoU(B_{gt}, b_{ij}) \in [0.0 ; 0.3]$ ⇒ se considera como fondo
- 2 Retirar las muestras negativas que se solapan entre ellas más de un 70 %
- 3 Considerar como ejemplos positivos solamente a los *bounding boxes ground truth*

Girshick et al. Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR, 2014.
He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV, 2014.

¿Cómo Entrenar los Clasificadores?



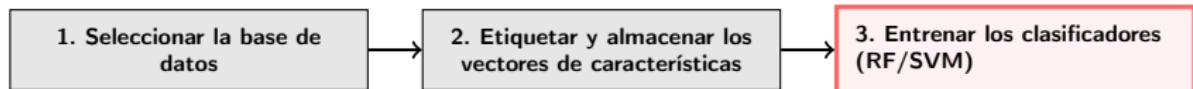
Implementación del Sistema Faster R-CNN+RF

Se utilizan dos librerías para RF:

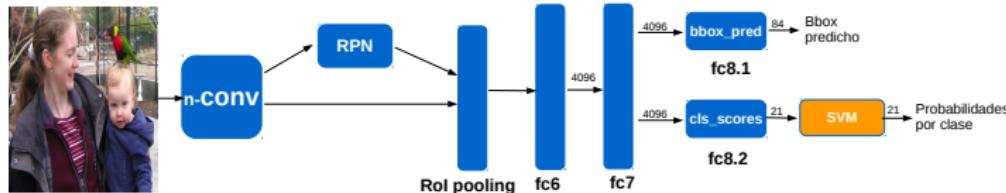
- 1 **randomForest**: Librería que **implementa RF** en el **lenguaje R**. Según ³ es la tercera mejor librería de la familia RF y la quinta al compararla con otras 178 implementaciones.
- 2 **scikit-learn**: Librería que **implementa RF** en **Python**. Según ³ se ubica en la posición 7 en la familia de RF y 18 a nivel global.

³ Manuel Fernández-Delgado et al. Do we need hundreds of classifiers to solve real world classification problems? JMLR, January 2015.

Implementación del Sistema Faster R-CNN+SVM



- 1 Para implementar el sistema se utiliza la **librería LIBSVM**. Calificada según ⁴ como la mejor implementación de SVM y la tercera entre todas las librerías.
- 2 Se utilizan dos métodos para implementar SVM multiclas:
 - 2.1 **One-vs-One (ovo)**: Se construye un clasificador binario por cada par de clases distintas. Se entranan $k(k - 1)/2$ clasificadores.
 - 2.2 **One-vs-All (ova)**: Se construyen k modelos distintos. El i -ésimo modelo considera las muestras de la clase i como positivas y todas las restantes como negativas.



⁴ Manuel Fernández-Delgado et al. Do we need hundreds of classifiers to solve real world classification problems? JMLR, January 2015.

Problema con el Tamaño de los Datos al Trabajar con *fc7/fc6*

- ▶ Problemas al intentar entrenar los clasificadores para la detección de objetos.
 - ▶ Los datos de entrenamiento no se ajustan en memoria.
 - ▶ Se vuelve no factible considerar todos los ejemplos negativos simultáneamente.
 - ▶ **Solución:** utilizar un algoritmo de **bootstrapping** o **Hard Example Mining (HEM)**.
- ▶ **HEM:** Algoritmo de entrenamiento iterativo.
 - ▶ Se comienza con un **caché inicial** (C_1), subconjunto del set de entrenamiento completo D ($C_1 \subseteq D$).
 - ▶ Se entrena un **modelo inicial** (m_1).
 - ▶ Se **retiran** desde el **caché** **ejemplos easy** y se **agregan ejemplos hard**.
 - ▶ **Actualizar el modelo inicial.** Se itera hasta que todos los ejemplos **hard** estén contenidos en el **caché** ($H(m_i, D) \subseteq C_i$).

1 Objetivos

 Objetivo General
 Objetivos Específicos

2 Antecedentes Generales

 CNNs para Clasificación de Imágenes
 CNNs para Detección de Objetos

3 Metodología e Implementación

 Propuesta de Mejoramiento del Sistema Faster
 R-CNN
 Entrenamiento de los Clasificadores
 Implementación del Sistema Faster R-CNN+RF
 Implementación del Sistema Faster R-CNN+SVM

4 Resultados y Análisis

 Replicación Resultados Sistema Faster R-CNN
 Resultados Método de Etiquetado 1
 Resultados Método de Etiquetado 2
 Resumen Mejores Resultados
 Ejemplos de Detecciones

5 Conclusiones

6 Bibliografía

Replicación de los Resultados Sistema Faster R-CNN

Entrenamiento en set train+val PASCAL VOC 2007

Sistema	Red	mAP
Faster R-CNN Replicado	ZF	60.0
Faster R-CNN [paper]	ZF	59.9

Sistema	Red	mAP
Faster R-CNN Replicado	VGG16	69.3
Faster R-CNN [ICCV15]	VGG16	70.0

Método de Etiquetado 1

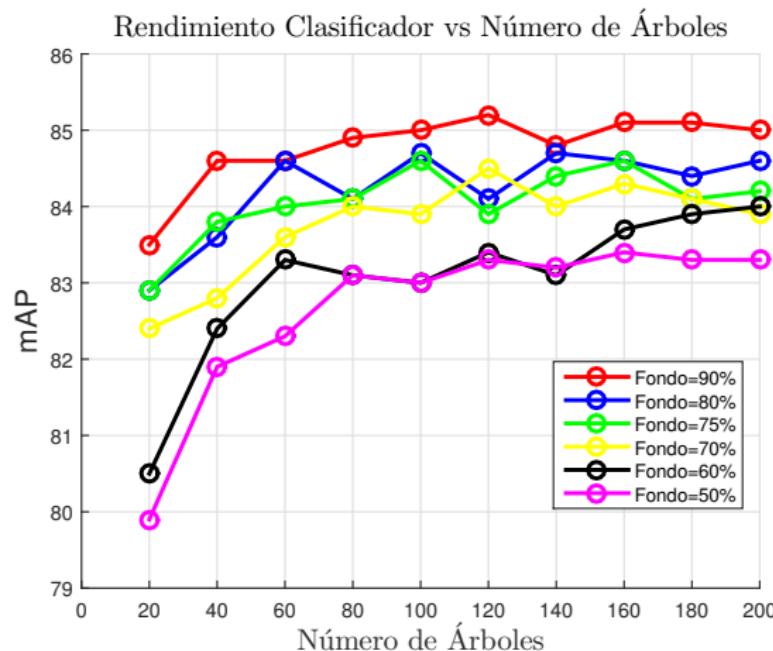
- 1 Para cada imagen en el conjunto de entrenamiento, usando el bounding box ground truth (B_{gt}) y el bounding box predicho por el sistema (b_{ij}), etiquetar cada proposal de la siguiente forma:
 - 1.1 Si $IoU(B_{gt}, b_{ij}) \geq 0.5$ para alguno de los bbox del ground truth considerar al proposal como perteneciente a una de las 20 clases.
 - 1.2 $IoU(B_{gt}, b_{ij}) \in [0.1, 0.5)$ ⇒ se considera como fondo
- 2 Redistribuir los ejemplos, tal que la clase fondo corresponda al 90 % de las muestras.

Para VGG se etiquetan 432.641 ejemplos (389.367 ej fondo y 43.274 ej positivos)

- Se implementa el sistema Faster R-CNN+RF y se entrena con el método de etiquetado 1 realizando las siguientes pruebas:
 - Se analiza usando solo la red ZF la mejor distribución entre los ejemplos de fondo y el resto de las clases.
 - Se prueban **6 distribuciones distintas (50 %, 60 %, 70 %, 75 %, 80 % y 90 %).**
 - Para todas las distribuciones se analiza el **rendimiento v/s el número de árboles.**
 - Los RFs son entrenados usando el set *train* y se mide el rendimiento en el set de validación.

Sistema Faster R-CNN+RF Método de Etiquetado 1

Análisis rendimiento v/s número de árboles en RF usando la librería randomForest y la red ZF



Rendimiento Sistemas Faster R-CNN+RF y Faster R-CNN+SVM Método 1

Sistema Faster R-CNN+RF

- ▶ Usando la redistribución del 90 % y una cantidad de árboles de 160 o más se entrena en el *set trainval* el sistema Faster R-CNN+RF.

Sistema	Librería	Red	mAP
Faster R-CNN+RF160	randomForest	ZF	57.4
Faster R-CNN+RF180	randomForest	ZF	57.5
Faster R-CNN+RF160	sklearn	ZF	57.6
Faster R-CNN+RF180	sklearn	ZF	57.5
Faster R-CNN+RF160	sklearn	VGG	67.3
Faster R-CNN+RF180	sklearn	VGG	67.8

Sistema Faster R-CNN+SVM

- ▶ Se utiliza el *set trainval* redistribuido al 90 %

Sistema	Red	Parámetros	mAP ovo
Faster R-CNN+SVM RBF	VGG	c=1, γ=1/21	68.9
Faster R-CNN+SVM Lineal	VGG	c=1	68.7

Método de Etiquetado 2

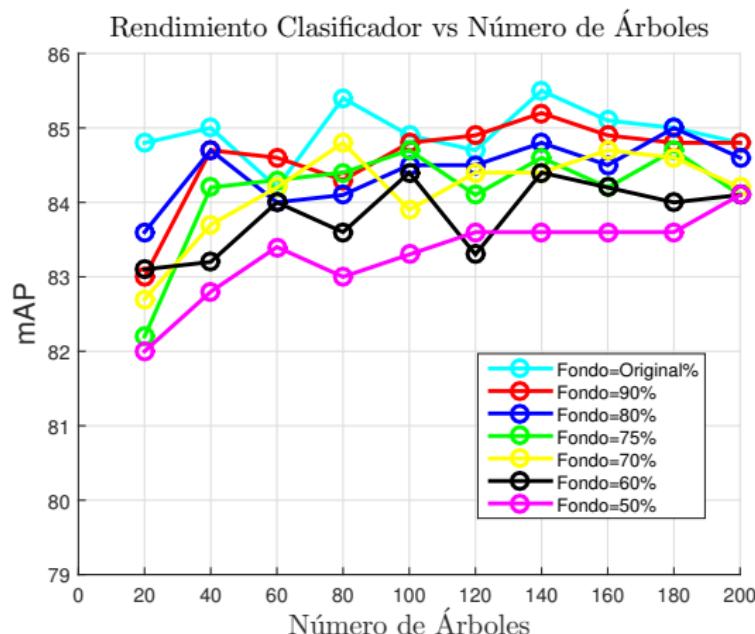
- 1 Para cada imagen en el conjunto de entrenamiento, usando el bounding box ground truth (B_{gt}) y el bounding box predicho por el sistema (b_{ij}), etiquetar cada proposal de la siguiente forma:
 - 1.1 Si $IoU(B_{gt}, b_{ij}) \in [0.0 ; 0.3]$ ⇒ se considera como fondo
- 2 Retirar las muestras negativas que se solapen entre ellas más de un 70 %
- 3 Considerar como ejemplos positivos solamente a los *bounding boxes ground truth*

Para VGG se etiquetan 533.583 ejemplos de fondo

- Se implementa el sistema Faster R-CNN+RF y se entrena con el método de etiquetado 2 realizando las siguientes pruebas:
 - Se analiza usando solo la red ZF la mejor distribución entre los ejemplos de fondo y el resto de las clases.
 - Se prueban **7 distribuciones (50 %, 60 %, 70 %, 75 %, 80 %, 90 %, 96.45 % ⇒ original)**
 - Para todas las distribuciones se analiza el **rendimiento v/s el número de árboles**
 - Los RFs son entrenados en el *set train* y se mide el rendimiento en el *set de validación*.

Sistema Faster R-CNN+RF Método de Etiquetado 2

Análisis rendimiento v/s el número de árboles en RF usando la librería randomForest y ZFNet



Resultados sistema Faster R-CNN+RF

Sistema	Librería	Red	u_1	u_2	mAP
Faster R-CNN+RF180	sklearn	VGG	0.7	0.3	66.4
Faster R-CNN+RF200	sklearn	VGG	0.7	0.3	66.4
Faster R-CNN+RF180	sklearn	VGG	0.8	0.2	66.3
Faster R-CNN+RF200	sklearn	VGG	0.8	0.2	66.2

Resultados sistema Faster R-CNN+SVM

- ▶ Entrenamiento realizado con *hard example mining* (HEM).

Sistema	Red	u_1	u_2	mAP ova		mAP ova HEM		
				fc8	fc8	fc7	fc6	
Faster R-CNN+SVM RBF	ZF	0.8	0.2	56.7	56.5	-	-	
Faster R-CNN+SVM Lineal	ZF	0.8	0.2	54.0	53.7	50.8	49.2	

- ▶ Resultados para red VGG usando características desde fc8.

Sistema	Red	u_1	u_2	mAP	
				ovo	ova
Faster R-CNN+SVM RBF	VGG	0.7	0.3	66.8	67.5
Faster R-CNN+SVM Lineal	VGG	0.7	0.3	67.5	65.2
Faster R-CNN+SVM RBF	VGG	0.8	0.2	66.6	67.4
Faster R-CNN+SVM Lineal	VGG	0.8	0.2	66.9	65.2

Resumen Mejores Resultados para Red VGG

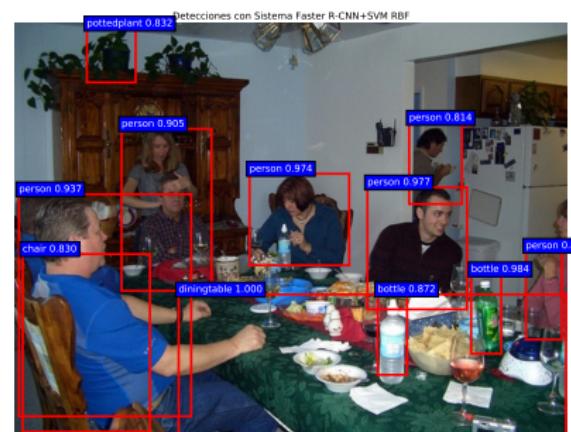
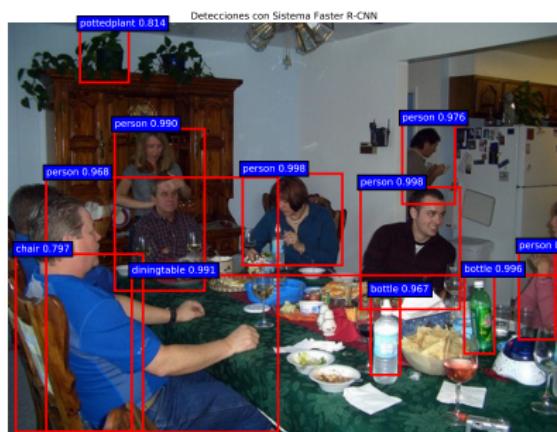
Sistema	Efq	Librería	mAP	aero	bike	bird	boat	bott	bus	car	cat	chr	cow	dtbl	dog	horse	mbk	prsn	plnt	shp	sofa	train	tv
FRCNN	-	Caffe	69.3	69.1	78.0	68.5	58.1	53.5	76.1	79.9	78.6	48.9	75.8	67.8	77.4	80.5	76.4	76.7	42.0	68.0	65.0	76.4	68.6
FRCNN+RF180	m1	sklearn	67.8	67.1	77.3	66.7	54.8	51.7	75.5	79.1	78.3	47.5	71.4	63.4	77.9	82.1	75.1	75.9	37.8	68.4	62.6	73.7	69.1
FRCNN+RF180	m2 ²	sklearn	66.4	65.2	78.2	66.7	53.6	47.5	75.0	79.0	77.9	43.9	73.5	61.1	76.4	80.3	73.6	73.5	34.4	67.5	61.8	75.0	63.8
FRCNN+SVM RBF¹	m1	libsvm	68.9	68.7	78.0	67.3	56.2	51.9	76.5	79.6	77.8	49.2	76.5	63.6	79.5	79.9	76.1	76.1	41.2	68.3	64.4	76.2	71.7
FRCNN+SVM Lineal ¹	m2 ²	libsvm	67.5	68.7	77.1	67.4	52.9	48.4	75.6	79.3	77.6	45.5	74.9	65.7	76.0	80.4	74.2	70.1	42.2	67.3	61.1	76.1	68.6

¹ svm multiclase ovo

² umbral solapamiento=0.7, umbral etiquetado=0.3

- ▶ El sistema Faster R-CNN es el máximo para 11 clases.
- ▶ Para las restantes 9 clases la diferencia con los máximos oscila entre 0.2 para la clase *bike* y 3.1 para la clase *tv/monitor*.
- ▶ En general cuando el Faster R-CNN no logra ser el máximo la diferencia es mínima a favor de otro sistema (menor a 1.0 punto).
- ▶ El sistema Faster R-CNN+SVM lidera en 5 clases: *bus, chair, cow, dog* y *tv/monitor*

Ejemplos de Detecciones



Ejemplos de Detecciones



Conclusiones

- ▶ Usando el **método 1 de etiquetado se obtienen mejores resultados que con el método 2**, fundamentalmente, porque, con el primero se aumenta el número de ejemplos de entrenamiento para las clases positivas.
- ▶ **Resulta mejor usar los vectores de características desde fc8.2, que desde una capa menos profunda**, como fc7 o fc6. El mAP para fc8.2, fc7 y fc6 fue de 53.7, 50.8 y 49.2 respectivamente.
- ▶ **Se comprobó la efectividad de hard negative mining (HNM)**. Permitió entrenar los sistemas cuando se utilizaban los vectores de características desde fc7 y fc6, (dimensión 4.096).
- ▶ **El rendimiento no es afectado significativamente por la elección de la librería**. Se probó el efecto de usar la mejor librería de RF para implementar los sistemas, sin embargo, se llegó a la conclusión que la diferencia no logra ser considerable, al menos entre las primeras 20 implementaciones de clasificadores evaluadas.
- ▶ **El mejor resultado se obtuvo con SVM**, se logró un **mAP=68.9 %** para el sistema Faster R-CNN+SVM con *kernel* RBF.
- ▶ **El segundo mejor resultado fue para Faster R-CNN+RF180**, alcanzando un **mAP=67.8 %**.

Bibliografía

-  Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, R. E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Handwritten digit recognition with a back-propagation network. In Proc. Advances in Neural Information Processing Systems, 1990.
-  Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. IEEE TPAMI, 2010.
-  Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. NIPS, 2012.
-  Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorí. Do we need hundreds of classifiers to solve real world classification problems? JMLR, 2014.
-  Matthew D. Zeiler & Rob Fergus. Visualizing and understanding convolutional networks. ECCV, 2014.
-  Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR, 2014.
-  Ross Girshick. Fast R-CNN. ICCV, 2015.
-  Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. NIPS, 2015.
-  K. Simonyan & A. Zisserman. Very deep convolutional networks for large-scale image recognition. ICLR, 2015.