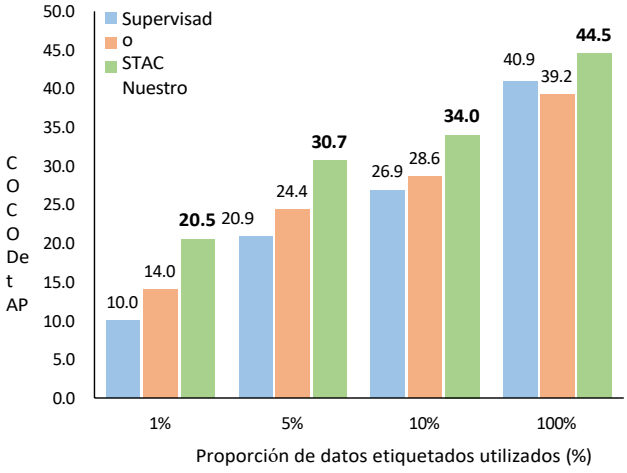
Detección de objetos semi-supervisada de extremo a extremo con un maestro suave

Mengde Xu *[1†] Zheng Zhang[1,2*‡] Han Hu[2‡] Jianfeng Wang[2] Lijuan Wang[2] Fangyun Wei[2]
Xiang Bai[1] Zicheng Liu [21]
Universidad de Ciencia y Tecnología de Huazhong
{mdxu,xbai}@hust.edu.cn
[2]Microsoft
{zhez,hanhu,jianfw,lijuanw,fawe,zliu}@microsoft.com

Resumen

*Este artículo presenta un enfoque de detección de objetos semisupervisado de extremo a extremo, en contraste con los métodos anteriores más complejos de varias etapas. El entrenamiento de extremo a extremo mejora gradualmente la calidad de las pseudoetiquetas durante el currículo, y las pseudoetiquetas, cada vez más precisas, benefician a su vez el entrenamiento de la detección de objetos. También proponemos dos técnicas sencillas pero efectivas dentro de este marco: un mecanismo de profesor suave en el que la pérdida de clasificación de cada caja delimitadora no etiquetada es ponderada por la puntuación de clasificación producida por la red de profesores; un enfoque de "jittering" de cajas para seleccionar pseudocajas fiables para el aprendizaje de la regresión de cajas. En la prueba de referencia COCO, el enfoque propuesto supera a los métodos anteriores por un amplio margen bajo var.
de etiquetado, es decir, el 1%, el 5% y el 10%. Además, nuestro enfoque demuestra que también funciona bien cuando la cantidad de datos etiquetados es relativamente grande. Por ejemplo, puede mejorar un detector de referencia de 40,9 mAP entrenado con el conjunto de entrenamiento de COCO en +3,6 mAP, llegando a 44,5 mAP, aprovechando las 123K imágenes sin etiquetar de COCO. En el detector de objetos basado en la transformada Swin de última generación (58,9 mAP en el test-dev), todavía puede mejorar significativamente la precisión de la detección en +1,5 mAP, alcanzando 60,4 mAP, y mejorar la precisión de la segmentación de la instancia en +1,2 mAP, alcanzando
52,4 mAP. Incorporando además el modelo preentrenado de Object365, la precisión en la detección alcanza los 61,3 mAP y la precisión en la segmentación de instancias llega a los 53,0 mAP, impulsando el nuevo estado del arte. El código y los modelos se harán públicos en https://github.com/microsoft/SoftTeacher.*



1. Introducción

Los datos son importantes. De hecho, los grandes datos como ImageNet han desencadenado en gran medida el auge del aprendizaje profundo en la vi- deografía informática.

*Contribución equitativa. †Este trabajo se realiza cuando Mengde Xu era pasante en MSRA. ‡Persona de contacto.

Figura 1. El método de detección de objetos semisupervisado basado en la pseudoetiqueta propuesto supera al STAC [27] por un amplio margen en la prueba de referencia MS-COCO.

sión. Sin embargo, la obtención de etiquetas puede suponer un cuello de botella, debido al largo y costoso proceso de anotación. Esto ha animado a los métodos de aprendizaje a aprovechar los datos no almacenados para entrenar modelos neuronales profundos, como el aprendizaje autosupervisado y el aprendizaje semisupervisado. Este artículo estudia el problema del aprendizaje semisupervisado, en particular para la detección de objetos.

Para la detección de objetos semisupervisada, nos interesan los enfoques basados en la pseudoetiqueta, que son el estado actual de la técnica. Estos enfoques [27, 36] llevan a cabo un esquema de entrenamiento de varias etapas, con la primera etapa de entrenamiento de un detector inicial utilizando datos etiquetados, seguido de un proceso de pseudo-etiquetado para los datos no etiquetados y un paso de re-entrenamiento basado en los datos pseudo-etiquetados no anotados. Estos enfoques de varias etapas logran una precisión razonablemente buena, sin embargo, el rendimiento final se ve limitado por la calidad de las pseudoetiquetas generadas por un detector inicial y probablemente inexacto entrenado con una pequeña cantidad de datos etiquetados.

detector	método	val2017		test-dev2017	
		mAP$^{de}_{t}$	mAP$^{mas}_{k}$	mAP$^{de}_{t}$	mAP$^{mas}_{k}$
HTC++(Swin-L) con escala única	supervisado	57.1	49.6	-	-
	nuestro	59.1	51.0	-	-
	nuestro*	60.1	51.9	-	-
HTC++(Swin-L) con multiescala	supervisado	58.2	50.5	58.9	51.2
	nuestro	59.9	51.9	60.4	52.4
	nuestro*	60.7	52.5	61.3	53.0

Tabla 1. En el detector de última generación HTC++(Swin-L), nuestro método supera el aprendizaje supervisado tanto en `val2017` como en `test-dev2017`. * indica que los modelos están preentrenados con el conjunto de datos Object365 [24].

Para resolver este problema, presentamos un marco de detección de objetos semisupervisado basado en pseudoetiquetas, que realiza simultáneamente el pseudoetiquetado de las imágenes no etiquetadas y entrena un detector utilizando estas pseudoetiquetas junto con unas pocas etiquetadas en cada iteración. En concreto, las imágenes etiquetadas y no etiquetadas se muestrean aleatoriamente con una proporción preestablecida para formar un lote de datos. Sobre estas imágenes se aplican dos modelos, uno de los cuales realiza el entrenamiento de la detección y el otro se encarga de anotar las pseudoetiquetas de las imágenes no etiquetadas. El primero se denomina también alumno y el segundo maestro, que es una media móvil exponencial (EMA) del modelo alumno. Este enfoque integral evita el complicado esquema de entrenamiento en varias etapas. Además, permite un "efecto volante" por el que los procesos de pseudoetiquetado y de entrenamiento de la detección pueden reforzarse mutuamente, de modo que ambos mejoran a medida que avanza el entrenamiento.

Otro beneficio importante de este marco de extremo a extremo es que permite un mayor aprovechamiento del modelo del profesor para guiar el entrenamiento del modelo del alumno, en lugar de limitarse a proporcionar "algunas pseudocajas generadas con etiquetas categóricas duras" como en los enfoques anteriores [27, 36]. Para poner en práctica esta idea se propone un enfoque de *profesor suave*. En este enfoque, el modelo del profesor se utiliza para evaluar directamente todas las cajas candidatas generadas por el modelo del alumno, en lugar de proporcionar "pseudocajas" para asignar etiquetas de categoría y vectores de regresión a estas cajas candidatas generadas por el alumno. La evaluación directa de estas cajas candidatas permite utilizar una información de supervisión más amplia en el entrenamiento del modelo del alumno. Específicamente, primero catego- rizamos las cajas candidatas como primer plano/fondo por sus puntuaciones de detección con un umbral de primer plano alto para asegurar una alta precisión de las pseudo etiquetas positivas, como en [27]. Sin embargo, este alto umbral de primer plano da lugar a que muchas cajas candidatas positivas se asignen erróneamente como fondo. Para solucionar este problema, proponemos utilizar una medida de *fiabilidad* para ponderar la pérdida de cada caja candidata a "fondo". Encontramos empíricamente que una simple puntuación de detección producida por el modelo del profesor puede servir bien como medida de fiabilidad, y se utiliza en nuestro enfoque. Encontramos que esta medida de aproximación funciona significativamente mejor que las anteriores de tipo "hard fore".

métodos de asignación de terreno/fondo (véanse el Cuadro 3 y el Cuadro 4), y lo denominamos "*maestro blando*".

Otro enfoque que ejemplifica esta idea es la selección de cajas delimitadoras para el entrenamiento de la rama de lo- calización del estudiante, por medio de un enfoque de *jittering de cajas*. Este enfoque primero hace saltar una caja candidata de pseudo-primer plano varias veces. A continuación, estas cajas salteadas se regresan de acuerdo con la rama de localización del modelo del profesor, y la varianza de estas cajas regresadas se utiliza como *medida de fiabilidad*. La caja candidata con una fiabilidad adecuadamente alta se utilizará para el entrenamiento de la rama de localización del alumno.

En la prueba de detección de objetos MS-COCO [16], nuestro enfoque logra 20,5 mAP, 30,7 mAP y 34,0 mAP en `val2017` con 1%, 5% y 10% de datos etiquetados utilizando el marco Faster R-CNN [22] con ResNet-50 [8] y FPN [14], superando al mejor método anterior STAC [27] en
+6,5, **+6,4** y **+5,4** mAP, respectivamente.

Además, también realizamos la evaluación en un entorno más difícil en el que los datos etiquetados han sido suficientemente grandes para entrenar un detector de objetos razonablemente preciso. En concreto, adoptamos el conjunto completo COCO `train2017` como datos etiquetados y el conjunto `no etiquetado2017` como datos no etiquetados. Con esta configuración, mejoramos la línea de base supervisada de un enfoque R-CNN más rápido con las bases ResNet-50 y ResNet- 101 en **+3,6** mAP y **+3,0** mAP, respectivamente.

Además, en un detector de última generación basado en Swin-Transformer [18] que alcanza 58,9 mAP para la detección de objetos y 51,2 mAP para la segmentación de instancias en el `test COCO-dev2017`, el enfoque propuesto puede mejorar la precisión en +1,5 mAP y +1,2 mAP, respectivamente, alcanzando **60,4** mAP y **52,4** mAP. Además, al incorporar el modelo preentrenado de Object365 [24], la precisión de la detección alcanza los **61,3** mAP y la de la segmentación de instancias los **53,0 mAP**, lo que supone el nuevo estado del arte en esta prueba.

2. Obras relacionadas

Aprendizaje semi-supervisado en la clasificación de imágenes El aprendizaje semi-supervisado en la clasificación de imágenes se puede clasificar a grandes rasgos en dos grupos: basado en la consistencia y basado en la pseudo-etiqueta. Los métodos basados en la consistencia [1, 23, 19, 11]

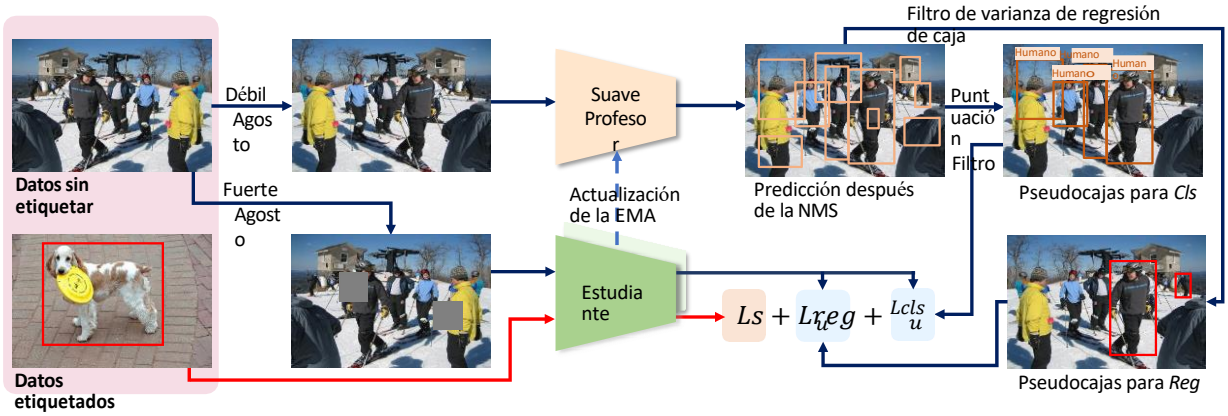


Figura 2. Visión general del marco de seudoetiquetado de extremo a extremo para la detección de objetos con semisupervisión. Las imágenes no etiquetadas y las imágenes etiquetadas forman el lote de datos de entrenamiento. En cada iteración de entrenamiento, se aplica un maestro suave para realizar el pseudo-etiquetado en imágenes no etiquetadas débiles sobre la marcha. Se producen dos conjuntos de pseudocajas: uno se utiliza para la rama de clasificación filtrando las cajas según la puntuación de primer plano, y el otro se utiliza para la rama de regresión de cajas filtrando las cajas según la varianza de regresión de cajas. El modelo del profesor se actualiza mediante el modelo del alumno a través de la media exponencial (EMA). La pérdida final es la suma de la pérdida de detección supervisada Ls y la pérdida de detección no supervisada $_{Lu}$.

aprovechar las imágenes no etiquetadas para construir una pérdida de regularización que fomente que diferentes perturbaciones de una misma imagen produzcan predicciones similares. Hay varias formas de implementar las perturbaciones, incluyendo la perturbación del modelo [1], el aumento de las imágenes [23] o el entrenamiento publicitario [19]. En [11], el objetivo de entrenamiento se ensambla mediante la predicción de diferentes pasos de entrenamiento. En [29], desarrollan [11] ensamblando el propio modelo en lugar de la predicción del modelo, el llamado promedio exponencial (EMA) del modelo de estudiante. Los enfoques de pseudo-etiquetas [33, 7, 12] (también llamados de auto-entrenamiento) anotan imágenes no etiquetadas con pseudo-etiquetas por un modelo de clasificación inicialmente entrenado, y el detector es refinado por estas imágenes pseudo-etiquetadas. A diferencia de nuestro método, que se centra en la detección de objetos, la pseudo-etiqueta no tiene que resolver el problema de la asignación de etiquetas de primer plano/fondo y la regresión de cajas al clasificar las imágenes. Recientemente, algunos trabajos [32, 3, 2, 26] exploran la importancia del aumento de datos en el aprendizaje semi-supervisado, lo que nos inspira a utilizar el aumento débil para generar pseudo-etiquetas y el aumento fuerte para el aprendizaje de modelos de detección.

Aprendizaje semisupervisado en la detección de objetos Al igual que el aprendizaje semisupervisado en la clasificación de imágenes, los métodos de detección de objetos semisupervisados también tienen dos categorías: los métodos de consistencia [10, 28] y los métodos de pseudoetiqueta [20, 36, 13, 27, 31]. Nuestro método pertenece a la categoría de las pseudo-etiquetas. En [20, 36], las predicciones de diferentes aumentos de datos se ensamblan para formar las pseudoetiquetas de las imágenes no etiquetadas. En [13], se entrena una SelectiveNet para seleccionar la pseudo-etiqueta. En [31], una caja detectada en una imagen no etiquetada se pega en una imagen etiquetada, y

la estimación de la consistencia de la localización se realiza sobre la imagen de la etiqueta pegada. Como la propia imagen se modifica, en [31] se requiere un proceso de detección muy exhaustivo. En nuestro método, sólo se procesa la cabeza de detección ligera. STAC [27] propone utilizar un aumento de datos débil para el entrenamiento del modelo y se utiliza un aumento de datos fuerte para realizar la pseudo-etiqueta. Sin embargo, al igual que otros métodos de pseudo-etiquetado [20, 36, 13, 27, 31], también sigue el esquema de entrenamiento en varias etapas. En cambio, nuestro método es un marco de pseudoetiquetado de extremo a extremo, que evita el complicado proceso de entrenamiento y también consigue un mejor rendimiento.

Detección de objetos La detección de objetos se centra en el diseño de un marco de detección eficiente y preciso. Existen dos corrientes principales: los detectores de objetos de una sola etapa [17, 21, 30] y
detectores de objetos de dos etapas [6, 22, 14, 34, 35]. La principal diferencia entre los dos tipos de métodos es el uso de una cascada para filtrar un gran número de candidatos a objetos (propuestas). En teoría, nuestro método es compatible con ambos tipos de métodos. Sin embargo, para permitir una comparación justa con trabajos anteriores [28, 27] sobre la detección de objetos semi-supervisada, utilizamos Faster R-CNN [22] como nuestro marco de detección por defecto para ilustrar nuestro método.

3. Metodología

La figura. 2 ilustra una visión general de nuestro marco de formación de extremo a extremo. Hay dos modelos, un modelo de estudiante y un modelo de profesor. El modelo del alumno se aprende mediante las pérdidas de detección en las imágenes etiquetadas y en las imágenes no etiquetadas utilizando pseudocajas. Las imágenes no etiquetadas tienen dos conjuntos de pseudocajas, que se utilizan para dirigir el entrenamiento de la rama de clasificación y la rama de regresión, respectivamente.

tivamente. El modelo del profesor es una media móvil exponencial (EMA) del modelo del alumno. Dentro de este marco de trabajo de extremo a extremo, hay dos diseños cruciales: el maestro *suave* y *el jittering de caja*.

3.1. Marco de pseudoetiquetado de extremo a extremo

En primer lugar, presentamos el marco integral para la detección de objetos semisupervisada basada en pseudoetiquetas. Nuestro enfoque sigue el esquema de entrenamiento profesor-alumno. En cada iteración de entrenamiento, las imágenes etiquetadas y las no etiquetadas se muestrean de forma continua según un ratio de muestreo de datos s_r para formar un lote de datos de entrenamiento. El modelo del profesor se realiza para generar las pseudocajas en las imágenes sin etiquetar y el modelo del alumno se entrena tanto en las imágenes etiquetadas con la verdad fundamental como en las imágenes sin etiquetar con las pseudocajas como verdad fundamental. Así, la pérdida global se define como la suma ponderada de la pérdida supervisada y la pérdida no supervisada:

$$L = L_s + \alpha L_u, \tag{1}$$

donde L_s y L_u denotan la pérdida supervisada de las imágenes etiquetadas y la pérdida no supervisada de las imágenes no etiquetadas, respectivamente, α
controla la contribución de la pérdida no supervisada. Ambas están normalizadas por el número respectivo de imágenes en el lote de datos de entrenamiento:

$$L_s = \frac{1}{N_l} \sum_{i=1}^{N_l} (L_{cls}(I_l^i) + L_{reg}(I_l^i)), \tag{2}$$

$$L_u = \frac{1}{N_u} \sum_{i=1}^{N_u} (L_{cls}(I_u^i) + L_{reg}(I_u^i)), \tag{3}$$

donde I_l^i indica la *i-ésima* imagen etiquetada, I_u^i indica la
como negativos,
i-ésima imagen sin etiquetar, L_{cls} es la pérdida por clasificación, L_{reg} es la pérdida por regresión de caja, N_l y N_u denotan el número de
imágenes etiquetadas e imágenes sin etiquetar, respectivamente.

Al principio del entrenamiento, tanto el modelo del profesor como el del alumno se inicializan aleatoriamente. A medida que avanza el entrenamiento, el modelo del profesor se actualiza continuamente mediante el modelo del alumno, y seguimos la práctica común [29, 26] de que el modelo del profesor se actualiza mediante la exponencial de medias móviles (EMA).

A diferencia de tomar una simple distribución de probabilidad como pseudo-etiqueta en la clasificación de imágenes, crear una pseudo-etiqueta para la detección de izquierda. Por lo tanto, sólo los candidatos con la puntuación de primer plano[1] superior a un umbral son retenidos como las pseudocajas.

Para generar pseudocajas de alta calidad y facilitar el entrenamiento del modelo del alumno, nos basamos en FixMatch [26], que es el último avance en la tarea de clasificación de imágenes semisupervisada. El aumento fuerte se aplica para el entrenamiento de la detección del modelo del alumno y el aumento débil se utiliza para el pseudo-etiquetado del modelo del profesor.

En teoría, nuestro marco es aplicable a los principales detectores de objetos, incluidos los detectores de objetos de una etapa [15, 17, 21, 30] y los detectores de objetos de dos etapas [22, 9, 5, 35, 34]. Para permitir una comparación justa con los métodos anteriores, utilizamos Faster R-CNN [22] como marco de detección por defecto para ilustrar nuestro método.

3.2. Profesor suave

El rendimiento del detector depende de la calidad de la pseudoetiqueta. En la práctica, encontramos que el uso de un umbral más alto en la puntuación de primer plano para filtrar la mayoría de los candidatos a caja generados por los estudiantes con baja confianza puede lograr mejores resultados que el uso de un umbral más bajo. Como se muestra en la Tabla 9, el mejor rendimiento se consigue cuando el umbral se fija en 0,9. Sin embargo, mientras que la cri- teria estricta (umbral más alto) conduce a una mayor preci- sión del primer plano, el recuerdo de las cajas candidatas retenidas también cae

rápidamente. Como se muestra en la figura 3 (a), cuando el umbral de la tierra delantera se fija en 0,9, el recuerdo es bajo, como el 33%, mientras que la precisión alcanza el 89%. En este caso, si utilizamos
IoU entre los candidatos a la caja generados por los estudiantes y los profesores

generados para asignar etiquetas de primer y segundo plano, como hace un marco general de detección de objetos cuando se proporcionan anotaciones de cajas reales, algunos primeros planos
Los candidatos de la caja serán asignados erróneamente

lo que puede dificultar el entrenamiento y perjudicar el rendimiento. Para paliar este problema, proponemos un *profesor suave* ap-
de la información del profesor.
modelo, gracias a la flexibilidad del marco de trabajo de extremo a extremo. En concreto, evaluamos la fiabilidad de cada caja candidata generada por el estudiante para ser un fondo real, lo que se utiliza para ponderar su pérdida de clasificación del fondo. Dado
dos conjuntos de cajas $\{b_i^{fg}\}$ y $\{b_i^{bg}\}$, con $\{b_i^{fg}\}$ denotando las cajas as-
objetos es más complicado, ya que una imagen suele contener múltiples objetos y la anotación de

i

firmado como primer plano y $\{b^{bg}\}$ que denota las cajas asignadas como fondo, la pérdida de clasificación de una imagen no etiquetada
objetos consiste en la ubicación y la categoría. Dada una unla-
imagen, el modelo del profesor se utiliza para detectar objetos

y se predicen miles de cajas candidatas. A continuación, se realiza la supresión no máxima (NMS) para eliminar inateredundancia. Aunquela mayoría de las casillas redundantes se han eliminado, todavía hay algunas candidatas no destacadas

con la ponderación fiable se define como:

$$L_u^{cls} = \frac{1}{N^{fg}} \left(\sum_{i=1}^{N^{fg}} l^{cls}(b_i^{fg}, G^{cls}) + \sum_{j=1}^{N^{bg}} w_j^{cls} l^{cls}(b_j^{bg}, G^{cls}), \right. \tag{4}$$

[1] La puntuación de los primeros planos se define como la máxima probabilidad de todas las categorías que no son de fondo.

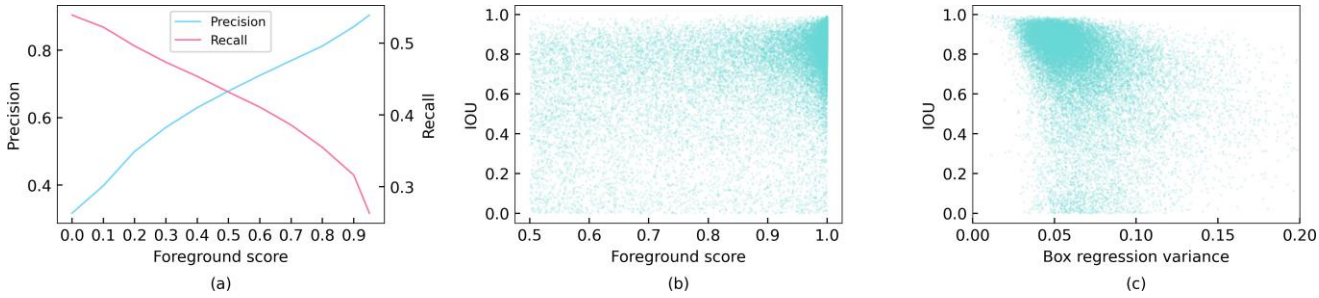


Figura 3. Tomamos una muestra aleatoria de 10k imágenes de entrenamiento no etiquetadas de `train2017` para dibujar figuras basadas en el modelo entrenado con un 10% de imágenes etiquetadas. (a) precisión y recuperación del primer plano bajo diferentes umbrales de puntuación de primer plano. (b) la correlación entre el IoU con el ground-truth y el box foreground score. (c) la correlación entre el IoU con la verdad sobre el terreno y la varianza de regresión de la caja. Cada punto de (b) y (c) representa una caja candidata.

$$w_j = \frac{r_j}{\sum_{k=1}^{N_b^{bg}} r_k}, \tag{5}$$

donde G_{cls} denota el conjunto de pseudocajas (generadas por el profesor) utilizadas para la clasificación, l_{cls} es la clasificación de cajas pérdida, r_j es la puntuación de fiabilidad de la j-ésima caja de fondo can- didato, N^{fg} y N^{bg} son el número de cajas candidatas de el conjunto de cajas $\{b^{fg}_i\}$ y $\{b^{bg}\}$, respectivamente.

La estimación de la puntuación de fiabilidad r es un reto. Nosotros Hemos comprobado empíricamente que la puntuación de fondo producida por el modelo del profesor con una imagen débilmente aumentada puede servir como indicador indirecto de r y se obtiene fácilmente en nuestro marco de entrenamiento de extremo a extremo. En concreto, dada una caja candidata generada por un alumno, su puntuación de fondo puede obtenerse simplemente utilizando el modelo maestro (BG-T) para procesar la caja a través de su cabeza de detección. Cabe destacar que este enfoque, a diferencia de los enfoques de minería negativa dura ampliamente utilizados, por ejemplo, OHEM [25] o Focal Loss [15], es más bien una minería negativa "simple". A modo de comparación, también examinamos otros indicadores:

- *Puntuación de fondo del modelo del estudiante (BG-S)*: Otra forma natural de generar la puntuación de fondo es utilizar directamente la predicción del modelo del alumno.
- *Diferencia de predicción (Pred-Diff)*: La diferencia de predicción entre el modelo del alumno y el modelo del profesor es también un posible indicador. En nuestro enfoque, utilizamos simplemente la diferencia entre las puntuaciones de fondo de los dos modelos para definir la puntuación de fiabilidad:

$$r = 1 - |p^{bg}_S(b) - p^{bg}_T(b)|, \tag{6}$$

donde p^{bg}_S y p^{bg}_T son la probabilidad prevista del para la asignación de primer plano/fondo. Hay dos hipótesis diferentes sobre cómo utilizar el IoU para medir si una caja candidata pertenece al fondo. En la primera hipótesis, si el IoU entre una caja candidata y una caja verdadera es menor que un umbral (por ejemplo, 0,5), un IoU mayor indica que la caja candidata tiene una mayor probabilidad de estar en el fondo. Esto puede verse como una minería negativa dura basada en IoU que es adoptada por Fast R-CNN [6] y Faster R- CNN [22] en la primera implementación. Por el contrario, la otra hipótesis sugiere que las cajas candidatas con un IoU más pequeño con las verdades del terreno tienen más probabilidades de ser fondos. En nuestros experimentos, validamos ambas hipótesis y las denominamos *IoU* y *Reverse-IoU*.

3.3. Caja Jittering

Como se muestra en la Figura 3 (b), la precisión de la localización y la puntuación de primer plano de las cajas candidatas no muestran una fuerte correlación positiva, lo que significa que las cajas con una alta puntuación de primer plano pueden no proporcionar información de localización precisa. Esto indica que la selección de las pseudocajas generadas por el profesor en función de la puntuación de primer plano no es adecuada para la regresión de cajas, y se necesita un criterio mejor.

Introducimos un enfoque intuitivo para estimar la fiabilidad de localización de una pseudocaja candidata midiendo la consistencia de su predicción de regresión. Específicamente, dada una pseudocaja candidata generada por el profesor b_i, muestreamos una caja jittered alrededor de b_i y alimentamos la caja jittered en el modelo del profesor para obtener la caja refinada $\hat{b}_i$, que se formu- lizado de la siguiente manera:

$$b_i = \text{refine}(\text{jitter}(b_i)). \tag{7}$$

El procedimiento anterior se repite varias veces para recoger una conjunto de cajas de jitters refinado

s $Njitter\ \{\hat{b}$ $\}_{i,j}$, y definimos el clase de fondo del estudiante y del modelo del profesor, respectivamente.

- *Intersección sobre la Unión*: El IoU entre las verdades del suelo y los candidatos a la caja es un criterio comúnmente utilizado

fiabilidad de la localización como la varianza de regresión de la caja:

$$\sigma_i = \frac{1}{4}\sum_{k=1}^{4} \hat{\sigma}_k \quad , \tag{8}$$

$$\hat{\sigma}_k = \frac{\sigma_k}{0{,}5(h(b_i) + w(b_i))}, \quad (9)$$

donde σ_k es la derivación estándar de la coordenada k-ésima del conjunto de cajas jittered refinadas $\{\hat{b}_{i,j}\}$, $\hat{\sigma}_k$ es la σ normalizada$_k$, $h(b_i)$ y $w(b_i)$ representan la altura y la anchura de la caja candidato b_i, respectivamente.

Una varianza de regresión de caja más pequeña indica una mayor fiabilidad de lo- calización. Sin embargo, calcular las varianzas de regresión de las cajas de todas las pseudocajas candidatas es insoportable durante el entrenamiento. Por lo tanto, en la práctica, sólo calculamos la fiabilidad de las cajas con una puntuación de primer plano superior a 0,5. De este modo, el número de cajas que hay que estimar se reduce de una media de cientos a unas 17 por imagen y, por tanto, el coste de cálculo es casi nulo.

En la Figura 3 (c), ilustramos la correlación entre la precisión de la localización y nuestra varianza de regresión de caja. En comparación con la puntuación del primer plano, la varianza de la regresión de cajas puede medir mejor la precisión de la localización. Esto nos motiva a seleccionar cajas candidatas cuya varianza de regresión de cajas es menor que un umbral como pseudo-etiqueta para entrenar la rama de regresión de cajas en imágenes no etiquetadas. Dadas las pseudocajas G_{reg} para entrenar la regresión de cajas en datos no etiquetados, la pérdida de regresión se formula como:

$$L_u^{\text{reg}} = \frac{1}{N^{\text{fg}}} \sum_{b}^{N^{\text{fg}}} \sum_{i=1}^{\text{fg}} l_{\text{reg}}(b_i^{\text{fg}}, G_{\text{reg}}), \quad (10)$$

donde b^{fg} es la i-ésima casilla asignada como primer plano, N^{fg} es el total de i número de caja de primer plano, l_{reg} es la pérdida de regresión de la caja. Por lo tanto, sustituyendo la Equ. 4 y Equ. 10 en la Ec. 3, la pérdida de las imágenes no etiquetadas es

$$L_u = \frac{1}{Nu} \sum_{i=1}^{Nu} (L_u^{\text{cls}}(I_u^i, G_{\text{cls}}^i) + L_u^{\text{reg}}(I_u^i, G_{\text{reg}}^i)). \quad (11)$$

Aquí utilizamos las pseudocajas G_{cls} y G_{reg} como entradas de la pérdida para destacar el hecho de que las pseudocajas utilizadas en la clasificación y la regresión de caja son diferentes en nuestro enfoque.

4. Experimentos

4.1. Conjunto de datos y protocolo de evaluación

Validamos nuestro método en la prueba de referencia MS-COCO [16]. Se proporcionan dos conjuntos de datos de entrenamiento, el conjunto `train2017` contiene 118k imágenes etiquetadas y el conjunto `unlabeled2017` contiene 123k imágenes sin etiquetar. muestreados como datos de entrenamiento etiquetados, y las restantes imágenes no muestreadas `del tren2017` se utilizan como datos no etiquetados datos. Para cada protocolo, STAC proporciona 5 datos diferentes pliegues y el rendimiento final es la media de los 5 pliegues. **Datos totalmente etiquetados**: En esta configuración, el `tren2017 se` utiliza como datos etiquetados y `el tren2017 sin etiquetar se` utiliza como datos adicionales sin etiquetar. Esta configuración es más difícil. Su objetivo es utilizar los datos no etiquetados adicionales para mejorar un detector bien entrenado en datos etiquetados a gran escala.

Evaluamos nuestro método en ambas configuraciones y seguimos la convención de informar del rendimiento en `val2017` con la precisión media estándar (mAP) como métrica de evaluación.

4.2. Detalles de la implementación

Utilizamos la R-CNN más rápida [22] equipada con FPN [14] (red de pirámide de características) como marco de detección por defecto para evaluar la eficacia de nuestro método, y se adopta una ResNet-50 preentrenada en ImageNet [8] como base. Nuestra implementación e hiperparámetros se basan en MMDetection [4]. Se utilizan anclajes con 5 escalas y 3 relaciones de aspecto. Se generan propuestas de 2k y 1k regiones con un umbral de supresión no máximo de 0,7 para el entrenamiento y la inferencia. En cada paso de entrenamiento, se generan 512 propuestas muestreo de 2k propuestas como los candidatos de la caja para entrenar RCNN. Dado que la cantidad de datos de entrenamiento de la configuración de **Datos Parcialmente Etiquetados** y la configuración de **Datos Completamente Etiquetados** tiene grandes diferencias, los parámetros de entrenamiento bajo las dos configuraciones son ligeramente diferentes.

Datos parcialmente etiquetados: El modelo se entrena para 180k iteraciones en 8 GPUs con 5 imágenes por GPU. Con el entrenamiento SGD, la tasa de aprendizaje se inicializa en 0,01 y se divide por 10 en 110k iteraciones y 160k iteraciones. El decaimiento del peso y el impulso se fijan en 0,0001 y 0,9, respectivamente. El umbral de primer plano se fija en 0,9 y el muestreo de datos Además, el conjunto `val2017` con 5k imágenes también se proporciona para la validación. En los métodos anteriores [27, 28, 10], hay dos configuraciones para validar el rendimiento:

Datos parcialmente etiquetados: STAC [27] introdujo por primera vez esta configuración. El 1%, el 5% y el 10% de las imágenes `del` conjunto `train2017` son

relación s_r se fija en 0,2 y disminuye gradualmente hasta 0 en las últimas 10k iteraciones.

Datos totalmente etiquetados: El modelo se entrena durante 720k iteraciones en 8 GPUs con 8 imágenes por GPU. En el entrenamiento SGD, la tasa de aprendizaje se inicializa en 0,01 y se divide por 10 en 480k iteraciones y 680k iteraciones. El decaimiento del peso y el impulso se fijan en 0,0001 y 0,9, respectivamente. El umbral de primer plano se fija en 0,9 y el ra- tio de muestreo de datos$_r$ se fija en 0,5 y disminuye gradualmente hasta 0 en las últimas 20k iteraciones.

Para estimar la fiabilidad de la localización de las cajas, fijamos $Njitter$ en 10 y el umbral en 0,02 para seleccionar las pseudoetiquetas para la regresión de las cajas. Las cajas jittered se muestrean aleatoriamente sumando los desplazamientos en cuatro coordenadas, y los desplazamientos se muestrean uniformemente desde [-6%, 6%] de la altura o la anchura de las pseudocajas candidatas. Además, seguimos a STAC y FixMatch para utilizar diferentes aumentos de datos

Aumento	Entrenamiento de imágenes etiquetadas	Entrenamiento de imágenes no etiquetadas	Generación de pseudoetiquetas
Escala de fluctuación	borde corto ∈ (0,5, 1,5)	borde corto ∈ (0,5, 1,5)	borde corto ∈ (0,5, 1,5)
Solarizar el jitter	p=0,25, relación ∈ (0, 1)	p=0,25, relación ∈ (0, 1)	-
Fluctuación de luminosidad	p=0,25, relación ∈ (0, 1)	p=0,25, relación ∈ (0, 1)	-
Fluctuación del contraste	p=0,25, relación ∈ (0, 1)	p=0,25, relación ∈ (0, 1)	-
Fluctuación de la nitidez	p=0,25, relación ∈ (0, 1)	p=0,25, relación ∈ (0, 1)	-
Traducción	-	p=0,3, relación de traslación ∈ (0, 0,1)	-
Girar	-	p=0,3, ángulo ∈ (0, 30°)	-
Turn o	-	p=0,3, ángulo ∈ (0, 30°)	-
Recorte	num ∈ (1, 5), ratio ∈ (0,05, 0,2)	num ∈ (1, 5), ratio ∈ (0,05, 0,2)	-

Tabla 2. Resumen del aumento de datos utilizado en nuestro enfoque. Seguimos la práctica de STAC [27] y FixMatch [26] para proporcionar diferentes aumentos de datos para la generación de pseudo-etiquetas, entrenamiento de imágenes etiquetadas y entrenamiento de imágenes no etiquetadas. "-" indica que el aumento no se utiliza.

Método	1%	5%	10%
Línea de base supervisada (la nuestra)	10.0 ± 0.26	20.92 ± 0.15	26.94 ± 0.111
Línea de base supervisada (STAC) [27]	9.83 ± 0.23	21.18 ± 0.20	26.18 ± 0.12
STAC [27]	13.97 ± 0.35	24.38 ± 0.12	28.64 ± 0.21
Nuestro	**20.46±0.**39	**30.74±0.**08	**34.04±0.**14

Tabla 3. Comparación a nivel de sistema con STAC en `val2017` bajo la configuración de **Datos Parcialmente Etiquetados.** Todos los resultados son la media de los 5 pliegues. Para la evaluación comparativa, también comparamos el rendimiento de la evaluación supervisada entre nuestro método y STAC, y su rendimiento es similar.

Método	Conjunto de datos adicional	mAP
Aprendizaje de propuestas [28]	sin etiqueta2017	37.4 → 38.4
STAC [27]	sin etiqueta2017	39.5 → 39.2
Autoformación [36]	ImageNet+OpenImages	41.1 → 41.9
Nuestro	sin etiqueta2017	40.9 → 44.5

Tabla 4. Comparación con otros estados de la técnica utilizando todos los datos del conjunto `train2017`. En particular, Self-training utiliza ImageNet (1,2 millones de imágenes) y OpenImages (1,7 millones de imágenes) como imágenes no etiquetadas adicionales, lo que supone un aumento de 20 veces respecto a `unlabeled2017` (123 mil imágenes).

para la generación de pseudo-etiquetas, el entrenamiento de imágenes etiquetadas y el entrenamiento de imágenes no etiquetadas. Los detalles se resumen en Ta- ble . 2.

4.3. Comparación de sistemas

En esta sección, comparamos nuestro método con el estado de la técnica anterior en MS-COCO. En primer lugar, evaluamos la configuración de **Datos Parcialmente Etiquetados** y comparamos nuestro método con STAC. Para la evaluación comparativa, comparamos la línea de base supervisada de nuestro método con los resultados reportados en STAC y encontramos que se desempeñan de manera similar, los resultados se muestran en la Tabla. 3. En este caso, comparamos además nuestro método con STAC a nivel de sistema, y nuestro método muestra una mejora de rendimiento significativa en diferentes protocoles. En concreto, nuestro método supera al STAC en **6,5** puntos, **6,4** puntos y **5,4** puntos cuando hay un 1%, 5% y 10% de datos etiquetados, respectivamente. Los resultados cualitativos de nuestro método en comparación con la línea de base supervisada son

1% Propo

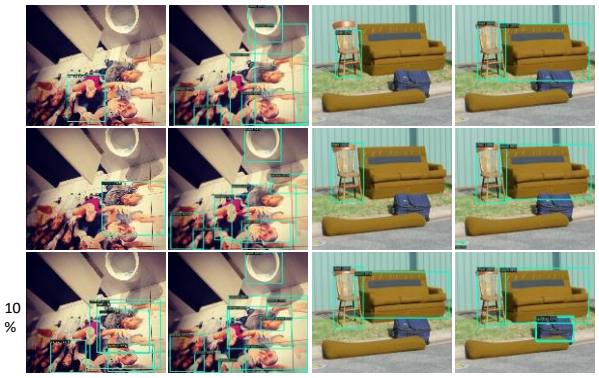


Figura 4. Los resultados cualitativos de nuestro método. (a), (c) son los resultados de la línea de base supervisada. (b), (d) son los resultados de nuestro método.

que se muestra en la figura. 4.

detector	columna vertebral	método	mAPdet	mAPmask
R-CNN más rápido	ResNet-50	supervisado nuestro	40.9 44.5(+**3.6**)	- -
R-CNN más rápido	ResNet-101	supervisado nuestro	43.8 46.8(+**3.0**)	- -
HTC++	Swin-L	supervisado nuestro	57.1 59.1(+**2.0**)	49.6 51.0(+**1.4**)
HTC++(multiescala)	Swin-L	supervisado nuestro	58.2 59.9(+**1.7**)	50.5 51.9(+**1.4**)

Tabla 5. Comparación con varios detectores entrenados supervisados en `val2017`. Se utiliza todo el `tren2017` como imágenes etiquetadas, y el `unlabeled2017` se utiliza como las imágenes adicionales sin etiquetar.

Método	mAP	mAP@0.5	mAP@0.75
Supervisado	27.1	44.6	28.6
Múltiples etapas	28.7	47.0	30.9
E2E	30.0	47.4	32.4
E2E+EMA	**31.2**	**48.8**	**34.0**

Tabla 6. Multietapa *vs. Fin a* Fin. El método de extremo a extremo (E2E) supera al marco multietapa. La actualización de la red de profesores mediante la estrategia de media móvil exponencial (EMA) mejora aún más el rendimiento.

A continuación, comparamos nuestro método con otros métodos de última generación en un entorno de **datos totalmente etiquetados**. Dado que el rendimiento de la línea de base supervisada varía en diferentes trabajos, presentamos los resultados de los métodos de comparación y su línea de base al mismo tiempo. Los resultados se muestran en la Tabla. 4.

Primero lo comparamos con la Propuesta de Aprendizaje [28] y el STAC [27], que también utilizan datos no etiquetados2017 como datos adicionales no etiquetados. Debido a los mejores hiperparámetros y a un entrenamiento más adecuado, nuestra línea de base supervisada logró un mejor rendimiento que otros métodos. Bajo la línea de base más fuerte, nuestro método sigue mostrando una mayor ganancia de rendimiento (+3,6 puntos) que Proposal Learning (+1,0 puntos)
y STAC (-0,3 puntos). El autoentrenamiento [36] utiliza Ima- geNet (1,2M de imágenes) y OpenImages (1,7M de imágenes) como datos adicionales sin etiquetar, que son 20× más grandes que el `unlabeled2017` (123k imágenes) que utilizamos. Con simi-
En la línea de base, nuestro método también muestra mejores resultados con menos datos sin etiquetar.

Además, evaluamos nuestro método con otros detectores más potentes, y los resultados evaluados en el conjunto `val2017` se muestran en la Tabla. 5. Nuestro método mejora sistemáticamente el rendimiento de los distintos detectores por un margen notable. Incluso en el detector de última generación HTC++ con columna vertebral Swin-L, mostramos una mejora de 1,8 en la detección AP y de 1,4 en la máscara AP. Además, también informamos de los resultados en el conjunto de `prueba-dev2017`. Como se muestra en la Tabel. 1, nuestro método mejora el HTC++ con la columna vertebral Swin-L en 1,5 mAP en la detección, que es el primer trabajo para superar los 60 mAP en la prueba de detección de objetos COCO.

4.4. Estudios de ablación

En esta sección, validamos nuestros diseños clave. Si no se especifica, todos los experimentos de ablación se realizan en el pliegue de datos único proporcionado por [27] con un 10% de imágenes etiquetadas del conjunto `train2017`.

Multietapa *vs. Final.* Comparamos nuestro método de extremo a extremo con el marco multietapa, como se muestra en la Tabla 6. Al pasar simplemente del marco multietapa a nuestro marco de extremo a extremo, el rendimiento se incrementa en
1,3 puntos. Actualizando el modelo del profesor con el modelo del alumno mediante la estrategia de la media móvil exponencial (EMA), nuestro método consigue además 31,2 mAP.

Efectos del maestro blando y del jittering de la caja. Aplacamos los efectos del maestro blando y del jittering de la caja. Los resultados se muestran en la tabla. 7. Basándonos en nuestro modelo de extremo a extremo equipado con EMA (E2E+EMA), la integración del soft teacher mejora el rendimiento en 2,4 puntos. Aplicando además el box jittering, el rendimiento alcanza 34,2 mAP, lo que supone 3 puntos más que E2E+EMA.

Diferentes indicadores en el profesorado blando. En la sección 3.2, se exploran varios indicadores diferentes para la es- tación de la fiabilidad. Aquí, evaluamos los diferentes indicadores y los resultados se muestran en la Tabla. 8. La puntuación de fondo predicha por el modelo del profesor consigue el mejor rendimiento. El simple hecho de cambiar el modelo de profesor a alumno empeora el rendimiento. Además, la mejora de IoU y Revearse-IoU es insignificante en comparación con BG-T. Estos resultados demuestran la necesidad de aprovechar el modelo del profesor.

Efectos de otros hiperparámetros. Estudiamos los efectos de los hiperparámetros utilizados en nuestro método. La tabla 9 estudia los efectos de diferentes umbrales de puntuación de primer plano. El mejor rendimiento se obtiene cuando el umbral se fija en 0,9, y los umbrales más bajos o más altos causarán una significativa

Profesor blando	Caja de vibración	mAP	mAP@0.5	mAP@0.75
		31.2	48.8	34.0
✓		33.6	**52.9**	36.6
✓	✓	**34.2**	52.6	**37.3**

Tabla 7. Estudiamos los efectos de las técnicas de *"soft teacher"* y "*box jittering"*.

Indicador	mAP	mAP@0.5	mAP@0.75
sin peso	31.2	48.8	34.0
IoU	31.7	51.4	34.2
Reverse-IoU	31.6	49.5	34.1
Pred-Diff	32.3	51.0	34.6
BG-S	25.9	44.4	27.0
BG-T	**33.6**	**52.9**	**36.6**

Tabla 8. Comparación de los diferentes indicadores en el profesorado blando.

Umbral	mAP	mAP@0.5	mAP@0.75
0.70	29.9	48.6	32.1
0.80	33.2	52.8	35.9
0.90	**33.6**	**52.9**	**36.6**
0.95	32.1	50.6	34.7

Tabla 9. Estudio de ablación sobre los efectos de diferentes umbrales de primer plano.

Umbral	mAP	mAP@0.5	mAP@0.75
0.04	33.8	52.3	36.7
0.03	34.0	52.5	36.9
0.02	**34.2**	**52.6**	**37.3**
0.01	32.9	52.2	35.8

Tabla 10. Estudio de ablación sobre los efectos de diferentes umbrales de selección de pseudocajas para la regresión de cajas según la varianza de la regresión de cajas.

Njitter	mAP	mAP@0.5	mAP@0.75
5	34.0	52.3	37.0
10	**34.2**	**52.6**	37.3
15	**34.2**	52.5	**37.4**

Tabla 11. Estudio de ablación sobre los efectos de un número diferente de cajas jit- teradas utilizadas para estimar la varianza de la regresión de cajas.

degradación del rendimiento. En la Tabla 10, estudiamos el umbral de varianza de la regresión de caja. El mejor rendimiento se consigue cuando el umbral se fija en 0,02. En la Tabla. 11, estudiamos los efectos de diferentes números de cajas jittered, y el rendimiento se satura cuando *Njitter se fija* en 10.

5. Conclusión

En este artículo, proponemos un marco de entrenamiento de extremo a extremo para la detección de objetos semisupervisada, que descarta el complicado esquema de varias etapas adoptado por anteriores

enfoques. Nuestro método mejora simultáneamente el detector y las pseudoetiquetas aprovechando un modelo de estudiante para el entrenamiento de detección, y un modelo de profesor que se actualiza continuamente por el modelo de estudiante a través de la estrategia de media móvil exponencial para el pseudoetiquetado en línea. Dentro del entrenamiento de extremo a extremo, presentamos dos técnicas sencillas denominadas soft teacher y box jittering para facilitar el aprovechamiento eficiente del modelo de profesor. El marco propuesto supera a los métodos más avanzados con un amplio margen en la prueba de referencia MS-COCO, tanto con datos parcialmente etiquetados como con datos totalmente etiquetados.

6. Agradecimiento

Nos gustaría agradecer a Yue Cao sus valiosas sugerencias y discusiones; a Yutong Lin y a Yixuan Wei su ayuda en los experimentos del transformador Swin.

Referencias

[1] Philip Bachman, Ouais Alsharif y Doina Precup. Learning with pseudo-ensembles. *arXiv preprint arXiv:1412.4864*, 2014. 2, 3

[2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang y Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 3

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver y Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019. 3

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection tool-box and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[5] Yihong Chen, Zheng Zhang, Yue Cao, Liwei Wang, Stephen Lin y Han Hu. Reppoints v2: Verification meets regres- sion for object detection. *NIPS*, 2020. 4

[6] Ross Girshick. Fast r-cnn. En *ICCV*, 2015. 3, 5

[7] Yves Grandvalet, Yoshua Bengio, et al. Aprendizaje semi-supervisado de por minimización de entropía. En *CAP*, 2005. 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren y Jian Sun. Aprendizaje residual profundo para el reconocimiento de imágenes. En *CVPR*, 2016. 2, 6

[9] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai y Yichen Wei. Redes de relación para la detección de objetos. En *CVPR*, 2018. 4

[10] Jisoo Jeong, Seungeui Lee, Jeesoo Kim y Nojun Kwak. Aprendizaje semisupervisado basado en la consistencia para la de- tección de objetos. *NIPS*, 2019. 3, 6

[11] Samuli Laine y Timo Aila. Ensamblaje temporal para el aprendizaje supervisado semi . *ICLR*, 2016. 2, 3

[12] Dong-Hyun Lee et al. Pseudo-etiqueta: El sencillo y eficiente método de aprendizaje semi-supervisado para redes neuronales profundas.
En el *Taller del ICML*, 2013. 3

[13] Yandong Li, Di Huang, Danfeng Qin, Liqiang Wang y Boqing Gong. Mejora de la detección de objetos con autoentrenamiento selectivo supervisado. En *ECCV*, 2020. 3

[14] Tsung-Yi Lin, Piotr Dolla'r, Ross Girshick, Kaiming He, Bharath Hariharan y Serge Belongie. Redes de pirámide de características para la detección de objetos. En *CVPR*, 2017. 2, 3, 6

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He y Piotr Dolla'r. Pérdida focal para la detección de objetos densos. En *ICCV*, 2017. 4, 5

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dolla'r y C Lawrence Zitnick. Microsoft coco: Objetos comunes en contexto. En *ECCV*, 2014. 2, 6

[17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu y Alexander C Berg. Ssd: Detector de caja múltiple de un solo disparo. En *ECCV*, 2016. 3, 4

[18] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin y Baining Guo. Swin trans- former: Hierarchical vision transformer using shifted win- dows. *arXiv preprint arXiv:2103.14030*, 2021. 2

[19] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama y Shin Ishii. Virtual adversarial training: a regulariza- tion method for supervised and semi-supervised learning.
TPAMI, 2018. 2, 3

[20] Ilija Radosavovic, Piotr Dolla'r, Ross Girshick, Georgia Gkioxari y Kaiming He. Destilación de datos: Hacia el aprendizaje supervisado omni- . En *CVPR*, 2018. 3

[21] Joseph Redmon, Santosh Divvala, Ross Girshick y Ali Farhadi. Sólo se mira una vez: Unified, real-time object detection. En *CVPR*, 2016. 3, 4

[22] Shaoqing Ren, Kaiming He, Ross Girshick y Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. En *NIPS*, 2015. 2, 3, 4, 5, 6

[23] Mehdi Sajjadi, Mehran Javanmardi y Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *arXiv preprint arXiv:1606.04586*, 2016. 2, 3

[24] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li y Jian Sun. Objects365: Un conjunto de datos a gran escala y de alta calidad para la detección de objetos. En *CVPR*, 2019. 2

[25] Abhinav Shrivastava, Abhinav Gupta y Ross Girshick. Training region-based object detectors with online hard example mining. En *CVPR*, 2016. 5

[26] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang y Colin Raffel. Fixmatch: Simplificando el aprendizaje semi-supervisado con consistencia y confianza. *NIPS*, 2020. 3, 4, 7

[27] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee y Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 1, 2, 3, 6, 7, 8

[28] Peng Tang, Chetan Ramaiah, Yan Wang, Ran Xu y Caiming Xiong. Proposal learning for semi-supervised object detection. En *WACV*, 2021. 3, 6, 7, 8

[29] Antti Tarvainen y Harri Valpola. Los maestros medios son mejores modelos de conducta: Los objetivos de consistencia promediados por peso mejoran los resultados del aprendizaje profundo semisupervisado de . *NIPS*, 2017. 3, 4

[30] Zhi Tian, Chunhua Shen, Hao Chen y Tong He. Fcos: Detección de objetos de una etapa totalmente convolucional. En *ICCV*, 2019. 3, 4

[31] Keze Wang, Xiaopeng Yan, Dongyu Zhang, Lei Zhang y Liang Lin. Hacia la cooperación hombre-máquina: Minería de muestras autosupervisada para la detección de objetos. En *CVPR*, 2018. 3

[32] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong y Quoc V Le. Unsupervised data augmentation for consistency training. *NIPS*, 2020. 3

[33] Qizhe Xie, Minh-Thang Luong, Eduard Hovy y Quoc V Le. Self-training with noisy student improves imagenet classification. En *CVPR*, 2020. 3

[34] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang y Stephen Lin. Reppoints: Representación de conjuntos de puntos para la detección de objetos.
En *ICCV*, 2019. 3, 4

[35] Ze Yang, Yinghao Xu, Han Xue, Zheng Zhang, Raquel Urtasun, Liwei Wang, Stephen Lin y Han Hu. Dense reppoints: Representación de objetos visuales con conjuntos de puntos densos.
ECCV, 2019. 3, 4

[36] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk y Quoc V Le. Repensando el preentrenamiento y el autoentrenamiento. *NIPS*, 2020. 1, 2, 3, 7, 8