Universitat Oberta de Catalunya (UOC)

Máster Universitario en Ciencia de Datos (*Data Science*)

TRABAJO FINAL DE MÁSTER

Área: 2. Machine Learning

Lista de la compra automática de fruta mediante la detección de objetos en imágenes con *Mask* R-CNN

Autor: Paula León Gil-Gibernau

Tutor: Jerónimo Hernández González

Profesor: Jordi Casas Roma

Barcelona, 22 de marzo de 2020

Créditos/Copyright

Una página con la especificación de créditos/copyright para el proyecto (ya sea aplicación por un lado y documentación por el otro, o unificadamente), así como la del uso de marcas, productos o servicios de terceros (incluidos códigos fuente). Si una persona diferente al autor colaboró en el proyecto, tiene que quedar explicitada su identidad y qué hizo.

A continuación se ejemplifica el caso más habitual, aunque se puede modificar por cualquier otra alternativa:

FICHA DEL TRABAJO FINAL

Título del trabajo:	Lista de la compra automática de fruta mediante la detección de objetos en imágenes con *Mask R-CNN*
Nombre del autor:	Paula León Gil-Gibernau
Nombre del colaborador/a docente:	Jerónimo Hernández González
Nombre del PRA:	Jordi Casas Roma
Fecha de entrega (mm/aaaa):	06/2020
Titulación o programa:	Máster Universitario en Ciencia de Datos (*Data Science*)
Área del Trabajo Final:	M2.879 - TFM - Área 2 aula 1 - *Machine Learning*
Idioma del trabajo:	Español
Palabras clave	Red neuronal convolucional profunda, detección de frutas, aprendizaje automático

Resumen

El objetivo de este estudio es realizar de forma automática una lista de la compra de fruta. Mediante un histórico de imágenes realizadas diariamente de un cajón de fruta, sin apilar, de una nevera, se aplicará una red neuronal convolucional profunda para detectar seis tipos de frutas en las imágenes. En concreto se usará una *Mask R-CNN*, que permite resolver la segmentación por instancias en imágenes obteniendo así las clases, las máscaras y las cajas delimitadoras de las frutas en las imágenes.

Palabras clave: Red neuronal convolucional profunda, detección de frutas, detección de objetos, máscaras, segmentación de instancias, aprendizaje automático, lista de la compra automática

Abstract

The aim of this project is to generate an automatic fruit shopping list. To do so, we will use a history of images taken daily, of a refrigerator fruit drawer, containing only unstacked fruit. We will apply a deep convolutional neural network to detect six different classed of fruit in the images. In particular, we will use Mask R-CNN, which allows solving the instance segmentation in images, being able to obtain the bounding boxes, the masks and the classes of the fruits which appear in the images.

Keywords: Deep convolutional neural network, Fruit detection, Object detection, Masks, Instance segmentation, Machine learning, Automatic shopping list

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Contexto y justificación del Trabajo

Este trabajo pretende realizar de forma automática una lista de la compra de fruta. Para ello se deberán procesar imágenes, reconociendo objetos en ellas, en este caso frutas. Se deberán aplicar algoritmos de aprendizaje profundo para crear modelos que permitan reconocer y contar las frutas en las imágenes.

El reconocimiento de objetos en imágenes está ampliamente usado para solucionar problemas de visión por computación con infinitas finalidades, desde el reconocimiento facial, la detección de peatones hasta para seguir una pelota en los partidos de fútbol.

En la actualidad, la tecnología se aplica cada vez más en el sector de la agricultura. En concreto, los sistemas automáticos inteligentes que se pueden usar para contar la fruta de una cosecha, para la detección de plagas que puedan dañarla o para analizar el rendimiento de dicha cosecha. Un sistema preciso de reconocimiento de fruta en imágenes es el elemento clave para poder realizar cualquiera de estas tareas. Aplicar dichos sistemas les permite a los agricultores, ganar tiempo, precisión y dinero en las tareas citadas anteriormente.

Existen ya muchas empresas que intentan revolucionar la cesta de la compra automática. Desde Amazon con su cesta de la compra por suscripción, que permite recibir con la frecuencia deseada productos seleccionados [6] a la empresa Caper con un carro de la compra que detecta los productos introducidos en él y calcula el importe total [7]. Todo esto ayuda a mejorar la experiencia del cliente al comprar y le ayuda a ahorrar tiempo con la tarea de la compra. Este trabajo es otra idea que busca ayudar al cliente a saber qué, cuánto y cuándo debe comprar fruta, que se podría extender a muchos más productos.

A partir de este trabajo se podrían llegar a aplicaciones de control de existencias de uso doméstico, en una casa, y empresarial, por ejemplo un supermercado.

1.2. Motivación

Estudié Ingeniería Informática en Mención de Computación en la UAB, y lo que más me interesó fue todo lo relacionado con el procesado y reconocimiento de imágenes. He podido seguir profundizando en temas de computación durante la realización de este master en Ciencia de Datos en la UOC, y me parece muy interesante concluir dicho master con un trabajo en el que aplicar todo lo aprendido durante estos años y que está relacionado con un tema que me gusta, la visión por computación.

1.3. Objetivos del Trabajo

1.3.1. Hipótesis

Este proyecto pretende demostrar la siguiente hipótesis:

Automatizar la generación de la lista de la compra de fruta, usando un histórico de imágenes realizadas diariamente del cajón de la fruta, sin apilar, de una nevera.

Se tratarán seis clases de frutas:

- Plátano
- Manzana
- Pera
- Piña
- Limon
- Naranja

1.3.2. Objetivos parciales

Existen dos objetivos parciales que nos ayudarán a demostrar y confirmar la hipótesis anteriormente descrita.

- Detección de existencias. ¿Cuántas piezas de cada fruta hay?. Esto implica la detección de fruta en una imagen. Para ello se propone aplicar un modelo *Mask R-CNN* que usa tanto máscaras como segmentación de instancias.
- Predicción de la compra. ¿Qué he de comprar hoy cuando vaya al supermercado?

1.4. Enfoque y método seguido

Para entender el enfoque y el método seguido para la realización de este proyecto se deben tratar varios puntos:

- La base de datos de imágenes. Este es uno de los puntos más importantes a tratar. Se suele pensar que cuanto mayor es el número de muestras mejor resolverá el modelo. Pero se debe encontrar el punto medio para no caer ni en sobreajuste ni en subajuste. El sobreajuste o sobreentrenamiento puede deberse a un exceso de datos o a un numero de parámetros muy elevado respecto a la cantidad de muestras. Por el contrario el subajuste, en inglés *underfitting*, puede producirse por falta de datos. La falta de datos puede hacer que nuestro modelo sea demasiado general y no tenga buenos resultados, el exceso de datos podría ocasionar que sea demasiado específico a los datos que tenemos y por tanto dará excelentes resultados con esos mismos datos pero no tan buenos para datos distintos a estos usados para entrenar el modelo. Para poder controlar y evitar los dos casos anteriores y para poder validar los resultados del modelo, usaremos un dataset de entrenamiento, otro de test y por último uno para la validación. En relación al último, se generará un dataset para validar el modelo de imágenes de fruta, sin apilar, dentro de un cajón de la nevera. Al no estar apilada, eliminaremos mucho ruido y disminuiremos la probabilidad de error.

- Las clases de frutas. Es importante escoger frutas con características distintas entre sí y otras que se asemejen. Sería interesante ver que el modelo se confunde más entre peras y manzanas que entre plátanos y naranjas. Escoger demasiadas clases de frutas puede aumentar mucho la complejidad del problema, y escoger muy pocas hacerlo demasiado sencillo. Seis clases me parecen razonables y además las frutas escogidas tienen tanto otras que se asemejan como totalmente distintas.

- El modelo para la detección de frutas en la imagen. Existen infinidad de modelos para la detección de objetos en imágenes. El modelo *Mask R-CNN* es el estado del arte en cuanto a la segmentación de instancias, y creo que puede dar resultados excelentes en la detección de frutas en imágenes.

- La detección y predicción de existencias: con ayuda del modelo anterior se calculará, dada una imagen el número de unidades de cada clase de fruta para cada imagen. El dataset de validación mencionado anteriormente será el usado para este punto, siendo este un histórico de imágenes generadas diariamente. Con ello tendremos la cantidad de cada fruta por día durante un periodo concreto de tiempo. A partir de esto se pueden extraer patrones y predecir las compras a realizar.

1.5. Planificación del Trabajo

La planificación de este proyecto se divide principalmente en las entregas parciales que se deben realizar, llamadas PECs. A partir de cada entrega se han desglosado las tareas a realizar, a continuación se puede ver el diagrama de Gantt de las PECs 1 y 2 1.1, el de la PEC 3 1.2 y el de las PECs 4 y 5 1.3:

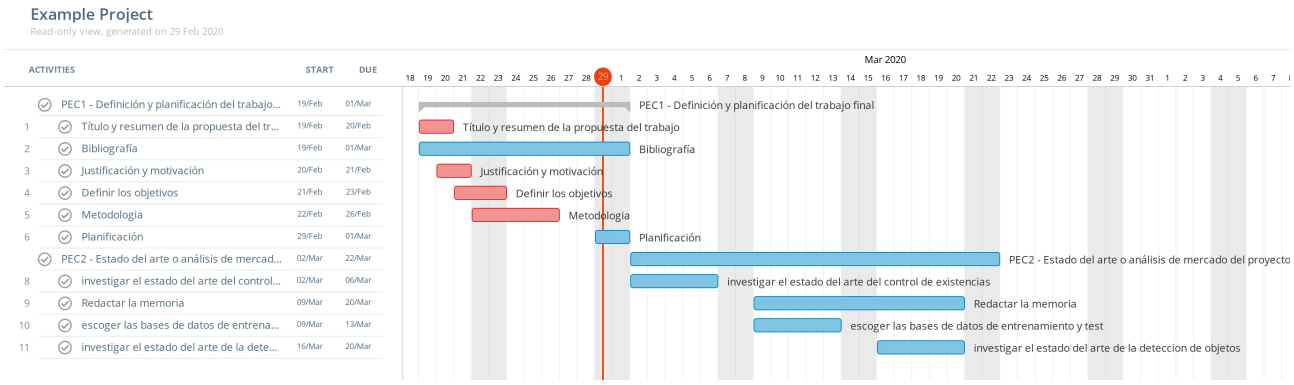

Figura 1.1: Diagrama de Gantt de la planificación del proyecto PECs 1 y 2.

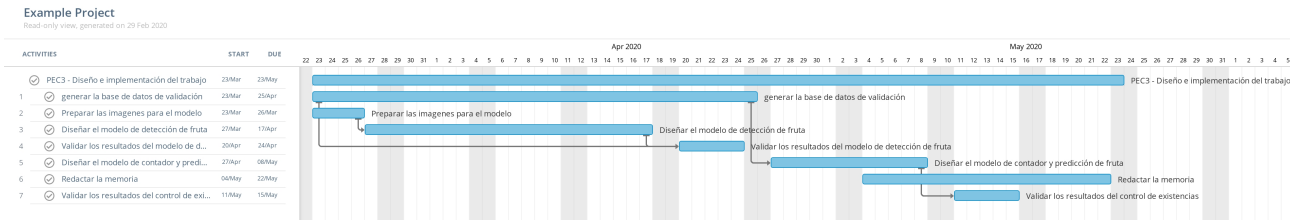

Figura 1.2: Diagrama de Gantt de la planificación del proyecto PEC 3.

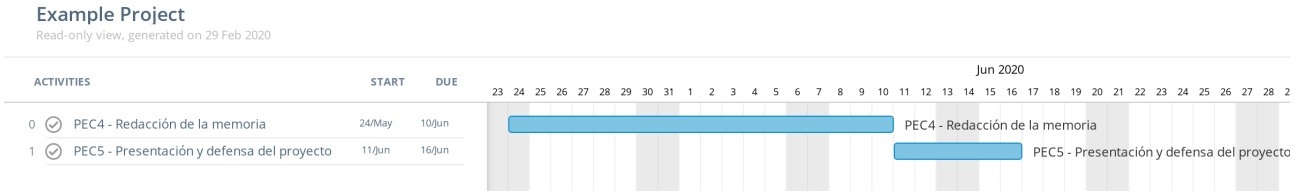

Figura 1.3: Diagrama de Gantt de la planificación del proyecto PECs 4 y 5.

1.6. Breve descripción de los capítulos de la memoria

Este trabajo empieza con una introducción, en la que se explica su contexto y justificación, realizar de forma automática una lista de la compra de fruta. Después se justifica la motivación personal, es decir por qué se quiere hacer este proyecto. A continuación se exponen tanto la hipótesis principal como los objetivos parciales. Por último, en este apartado se explica el enfoque y el método seguido para la realización de este proyecto junto con la planificación y el diagrama de Gantt de las tareas con su duración.

El segundo capítulo de esta memoria es el estado del arte, en el que se relata de forma sintetizada cómo han abordado los investigadores, hasta el momento, el problema que pretende resolver este trabajo. En primer lugar se explican las técnicas del estado del arte de aprendizaje automático para la detección de objetos en imágenes, en segundo lugar las técnicas de aprendizaje profundo y por ultimo como se ha abordado de forma más concreta la detección de fruta en imágenes.

Capítulo 2

Estado del Arte

La detección de objetos en imágenes es un problema que ha atraído mucha atención durante estos años, pues tiene infinitas aplicaciones. Dicho problema se ha resuelto tanto con algoritmos convencionales de aprendizaje automático como con algoritmos de aprendizaje profundo. Pero la mayor parte de los modelos que son el estado del arte usan aprendizaje profundo. El objetivo de la detección de un objeto en imágenes implica por un lado clasificar y etiquetar el objeto y por otra localizarlo en la imagen, obteniendo la caja delimitadora de dicho objeto. La detección de más de un objeto en una imagen requiere por un lado de la detección del objeto, pero además de la segmentación de instancias, que es la detección de los objetos en las imágenes a nivel de pixel. En la siguiente imagen 2.1, se muestra la diferencia entre el problema de la detección de un solo objeto o varios objetos en una imagen:

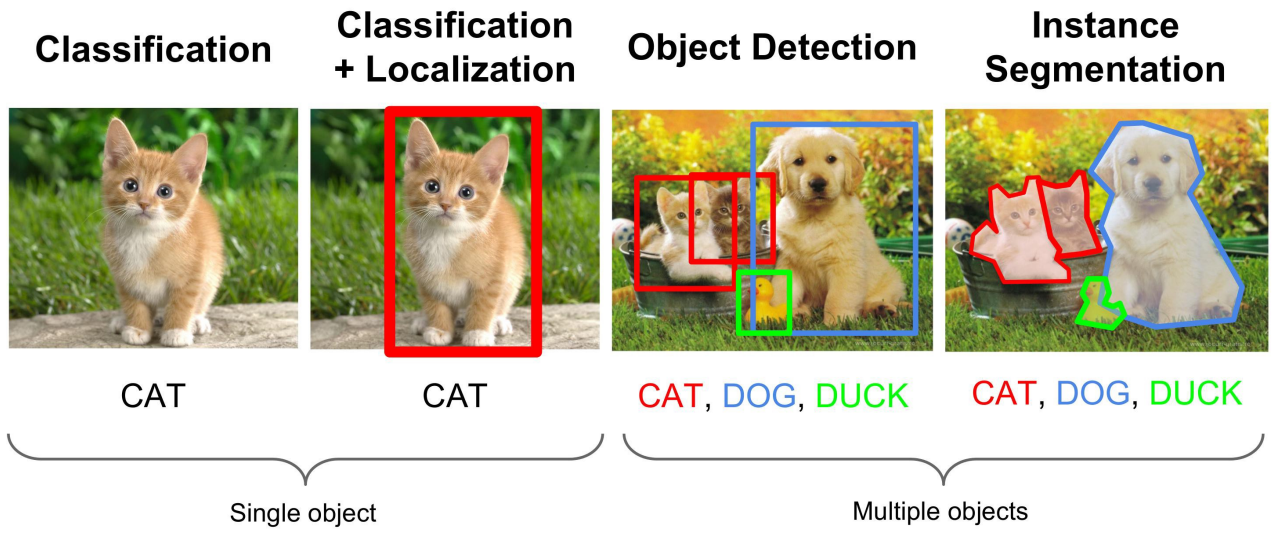

Figura 2.1: Diferencia entre la detección de un solo objeto o varios en imágenes.[1]

2.1. Aprendizaje automático

El primer detector de objetos con resultados robustos, fue propuesto en 2001 por Viola y Jones [8]. El objetivo era la detección de caras en tiempo real, pero se ha usado para infinidad de aplicaciones, desde la detección de manos a coches, señales de tráfico y otros objetos. Usaron un algoritmo de aprendizaje basado en *AdaBoost*, que es capaz de obtener clasificadores muy eficientes seleccionando un número pequeño de características visuales relevantes, y además combinaron clasificadores en cascada que permite descartar las regiones de la imagen que no son de interés, para centrarse en las que sí lo son. *AdaBoost*, la abreviatura en inglés de *Adaptive Boosting*, es un algoritmo de aprendizaje automático creado por Yoav Freund y Robert Schapire en 1996, que se usa habitualmente en problemas de clasificación, con el objetivo de convertir un conjunto de clasificadores débiles en fuertes.

El Histograma Orientado a Gradientes, conocido por sus siglas en inglés como *HOG*, es un descriptor de características enfocado a la detección de objetos, que apareció en 1986. La técnica se basa en contar las apariciones de orientación de gradientes en regiones localizadas de una imagen. No fue hasta 2005, que cobro más interés, cuando Dalal y Triggs presentaron su trabajo de detección de peatones usando *HOG* [9], dicho trabajo también fue aplicado a la detección de animales y vehículos en imágenes.

Para reconocer los descriptores *HOG*, se usan las Máquinas de vectores de soporte, por sus siglas en inglés *SVM*, que son algoritmos de aprendizaje supervisado que se usan para clasificar. Todos estos algoritmos de aprendizaje automático requieren del etiquetado previo de las imágenes al igual que de la selección previa de las características relevantes de la imagen.

2.2. Aprendizaje profundo

El aprendizaje profundo, es denominado así por la profundidad del número de capas por las que los datos se transforman. Consiguen desenredar las abstracciones extraídas en cada capa para elegir las características que mejoran el rendimiento.

Las arquitecturas básicas y más representativas de aprendizaje profundo son las redes neuronales profundas y más concretamente las redes neuronales convolucionales, conocidas por sus siglas en inglés como *CNN*. Las redes neuronales profundas son redes neuronales con varias capas ocultas entre las capas de entrada y salida. Las redes neuronales convolucionales, son un tipo de redes neuronales profundas aplicadas principalmente a problemas de clasificación y segmentación de imágenes. El motivo por el que estas redes se llaman así, es porque se emplea una operación matemática llamada convolución en, como mínimo, una de sus capas.

El objetivo que se busca con estas redes es entrenarlas con imágenes etiquetadas para que

el modelo pueda abstraer características que le permitan posteriormente clasificar una nueva imagen sin etiquetar. Por tanto ya no se requiere de la selección previa de dichas características de la imagen, como en los métodos de aprendizaje automático anteriormente explicados.

Las unidades de procesamiento gráfico, conocidas por sus siglas en inglés como *GPU*, permiten ejecutar los métodos de aprendizaje profundo en ella y como son muy paralelizables, permite aumentar altamente la capacidad de cómputo. Además también existen servicios en la nube que ofrecen procesamiento con *GPU*. Por otro lado Google creó una plataforma de aprendizaje automático en el que proporciona modelos ya pre entrenados y la posibilidad de personalizar modelos.

Gracias al desarrollo de las redes neuronales de convolución profunda y a la potencia de cómputo que proporcionan las *GPUs*, las técnicas de detección de objetos en imágenes han teniendo una rápida y gran evolución. En la siguiente figura 2.2, podemos ver los trabajos más importantes relacionados con la detección de objetos que usan *CNNs* desde 2012 hasta la actualidad :

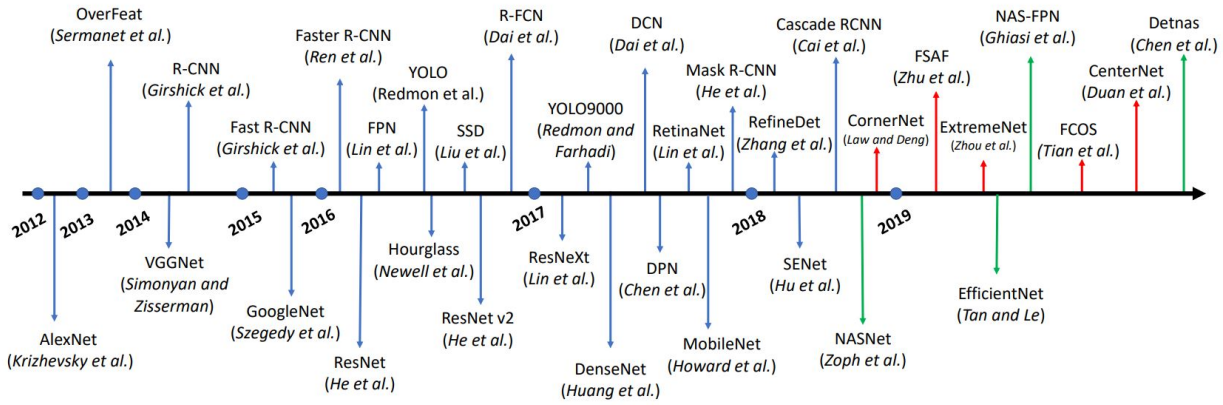

Figura 2.2: Trabajos de detección de objetos basados en *CNNs* desde 2012 hasta 2019.[2]

Las bases de datos más comúnmente usadas para la evaluación de los modelos de detección de objetos son:

- *Pascal VOC2007*: tiene un total de 20 categorías, está compuesto por tres conjuntos, uno de entrenamiento con 2501 imágenes, otro de test con 5011 y uno de validación con 2510 imágenes.
- *Pascal VOC2012*: tiene las mismas 20 categorías que Pascal VOC2007, está compuesto por tres conjuntos, uno de entrenamiento con 5717 imágenes, otro de test con 10991 y

uno de validación con 5823 imágenes.

- *MSCOCO*, tiene un total de 80 categorías, está compuesto por tres conjuntos, uno de entrenamiento con 118287 imágenes, otro de test con 40670 y uno de validación con 5000 imágenes.

- *Open Images*: contiene 1.9M de imágenes con 15M de objetos y 600 categorías. La mayoría de estas categorías tiene alrededor de 1000 muestras de entrenamiento.

- *ImageNet*: contiene más de 20.000 categorías y un total de 14 millones de imágenes. Desde 2010 se celebra una competición de los algoritmos de análisis de imágenes sobre esta base de datos llamada *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)*.

Los modelos usados para resolver la detección de objetos en imágenes se pueden clasificar en dos tipos, por un lado aquellos tradicionales, que proponen la región del objeto y luego lo clasifican y por otro lado afrontar el problema como un problema de regresión o clasificación, obteniendo resultados finales directamente, tanto la localización como la categoría a la que pertenece el objeto detectado.

2.2.1. Métodos tradicionales

Los métodos tradicionales más conocidos para la detección de objetos en imágenes son:

- *AlexNet*: un trabajo publicado por Krizhevsky, Sutskever y Hinton, en 2012 [10] fue el primero en usar redes neuronales convolucionales para la resolución del problema de la detección de objetos en imágenes. Aplicado a la base de datos *ImageNet* y posicionándose entre los cinco mejores de la competición *ILSVRC* del año 2012, consiguiendo una tasa de error del 15,3 %. La arquitectura de *AlexNet* tiene ocho capas de las cuales cinco son convolucionales, algunas de estas capas seguidas por capas *max-pooling*, funciones que permiten reducir el tamaño de los datos, y las últimas tres capas completamente conectadas. Usaron además, la función de activación de unidad lineal rectificada, por sus siglas en inglés, *ReLU*, no saturante, que resultó tener mejor rendimiento que las funciones tan y sigmoidea. Consiguieron mejorar el tiempo de entrenamiento gracias al uso de *multi-GPU*, poniendo a entrenar la mitad de las neuronas en una *GPU* y la otra mitad en la otra *GPU*. Además se enfrentaron a problemas de sobre entrenamiento, usando técnicas como el *dropout*, que consiste en desactivar neuronas con una probabilidad determinada y el aumento de datos, generando más imágenes.

- *OverFeat*: arquitectura presentada en 2013 por Sermanet et al. [11], en la que propusieron un algoritmo con una ventana deslizante de escala múltiple usando redes neuronales convolucionales (*CNNs*).

- *R-CNN*: Poco después de la publicación de *OverFeat*, Girshick et al publicaron en 2014 un trabajo en el que presentaron el método llamado *R-CNN*, regiones con características *CNN* [3]. Proponían un modelo de tres pasos. En el primero se extraen posibles objetos con un método de propuesta de regiones. En el segundo, se extraen las características de cada región usando redes neuronales convolucionales (*CNNs*), y por último en el tercer paso, se clasifica cada región usando máquinas de vectores de soporte (*SVMs*). En la siguiente figura 2.3 se muestran estos pasos:

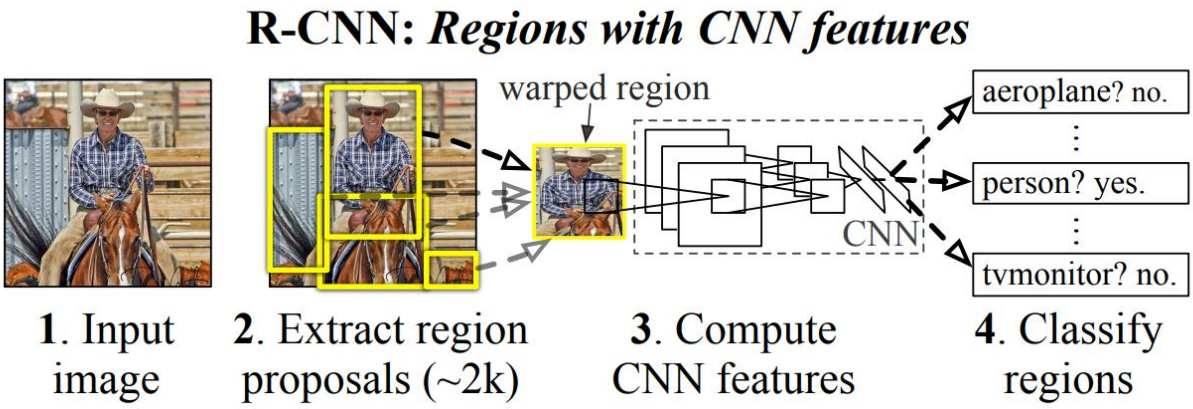

Figura 2.3: Arquitectura *R-CNN*.[3]

- *Fast R-CNN*: Girshick, publicó en 2015 [12], una mejora del anterior modelo *R-CNN*, para hacer el modelo más rápido, aplicando la red neuronal convolucional en toda la imagen de entrada en lugar de sobre cada región.

- *GoogLeNets*: Szegedy et al. publicaron en 2015 [13] una arquitectura también llamada *Inception*, en la que mejoraron el uso de los recursos informáticos dentro de la red, aumentando la profundidad y el ancho de la red.

- *Faster R-CNN*: en 2016, Shaoqing, Kaiming, Girshick y Sun, publicaron otro trabajo [14], la tercera iteración del *R-CNN*, añadiendo el concepto de red de propuesta de regiones, por sus siglas en inglés *RPN*, para hacer que el modelo pudiera entrenarse de principio a fin.

- *ResNets*: en 2016, He, Zhang, Ren y Sun, publicaron un trabajo [15], en el que presentaron la arquitectura *ResNets*, introduciendo el nuevo concepto de funciones residuales de aprendizaje para poder entrenar redes neuronales muy profundas. Ganaron la competición *ILSVRC* de 2015 con el primer puesto en el problema de clasificación.

- *R-FCN*: en 2016, Dai, Li, He y Sun publicaron un trabajo [16] en el que presentaron la arquitectura *R-FCN*, a diferencia de los detectores basados en regiones anteriores, como el *R-CNN*, no aplicaban una red por cada región, sino que compartían los cálculos por toda la imagen.

- *Mask R-CNN*: en 2017 se publicó un trabajo de He, Gkioxari, Dollar y Girshick [4], sobre la arquitectura *Mask R-CNN* para la segmentación de instancias en objetos y la generación de máscaras de los objetos. Ganaron la competición *COCO* en 2016, en problemas de segmentación de instancias y en encontrar las cajas delimitadoras de objeto. La arquitectura permite crear la segmentación a nivel de pixel para cada objeto y además puede separar cada objeto de su fondo. A continuación podemos ver la arquitectura de la segmentación de instancias 2.4.

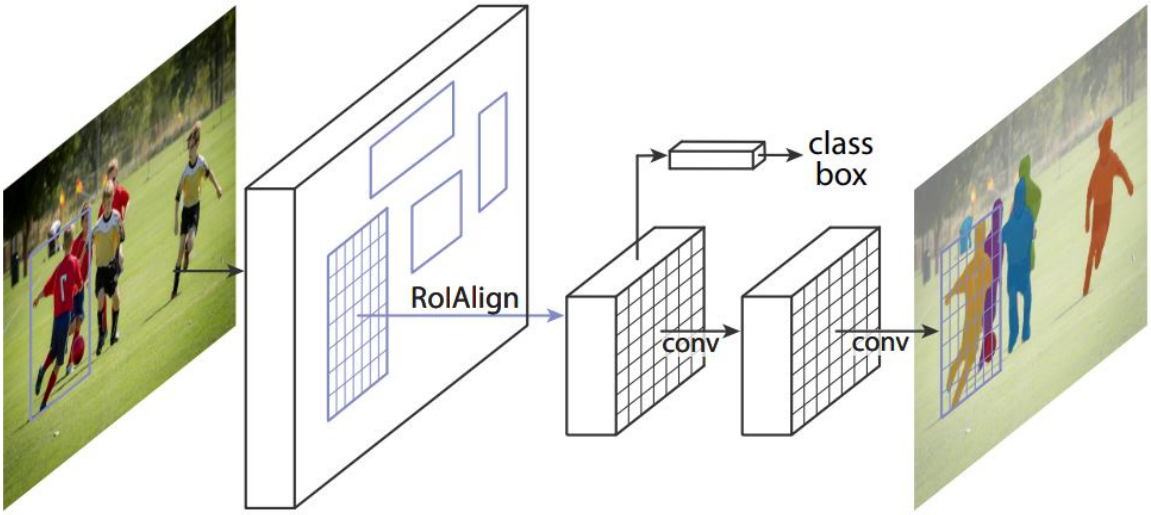

Figura 2.4: Arquitectura de *Mask R-CNN* para la segmentación de instancias.[4]

2.2.2. Métodos de regresión y clasificación

Los métodos de regresión y clasificación más conocidos para la detección de objetos en imágenes son:

- *YOLO (You Only Look Once)*: en 2016, se publicó el trabajo [5] de Redmon, Divvala, Girshick y Farhadi 2016, en el que presentaron una arquitectura basada en una simple red neuronal convolucional, con buenos resultados y rápida de entrenar. Permitiendo por primera vez la detección de objetos en tiempo real. En esta figura 2.5 se puede ver su arquitectura:

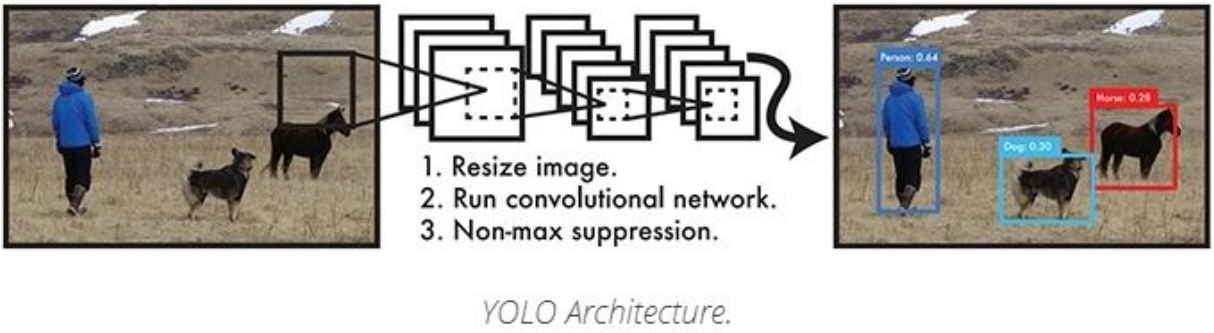

Figura 2.5: Arquitectura *YOLO*.[5]

- *SSD*: en 2016 Liu et al. publicaron un trabajo [17] en el que presentan una arquitectura llamada *SSD (Single Shot MultiBox Detector)*, con una única red neuronal profunda, usando mapas de características convolucionales con múltiples tamaños, haciendo el método más rápido que el *YOLO* y mejorando sus resultados.

- *YOLO900*: En 2017, Redmon y Farhadi publicaron la arquitectura YOLO900 [18], mejorando la velocidad y los resultados de *YOLO*.

De los últimos trabajos presentados entorno a la detección de objetos, en 2018, Zoph, Vasudevan, Shlens y Le, publicaron la arquitectura *NASNet* [19] que propone un nuevo espacio de búsqueda que permite la transferibilidad. Y en 2019, Chen et al. presentaron su trabajo sobre la arquitectura *DetNAS* [20], en el que usan también la búsqueda de arquitectura neural, conocida por sus siglas en inglés como *DetNAS*, para el diseño de mejores conexiones principales para la detección de objetos.

2.3. Detección de fruta

Existen muchos trabajos realizados sobre la detección de fruta en imágenes, en concreto existe uno presentado por Payne, Walsh, Subedi y Jarvis en 2014 [21], en el que se analizaba automáticamente el rendimiento de un cultivo de mango, analizando imágenes nocturnas.

Pero no fue hasta hace pocos años que se aplicaron, de forma más extensiva, métodos de aprendizaje automático y de aprendizaje profundo para la resolución de dicho problema. En 2016, Inkyu et al. [22], publicaron un trabajo en el que se detectaba fruta en imágenes, usando redes neuronales convolucionales, en concreto usando un método basado en *Faster R-CNN*.

En 2017, Rahnemoonfar y Sheppard [23], publicaron un trabajo en el que se usaron también redes neuronales convolucionales pero esta vez con un modelo basado en *Inception-ResNet*, para contar frutas en imágenes.

Bibliografía

[1] A. Ouaknine, "Review of deep learning algorithms for object detection.," 2018. URL https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852.

[2] D. S. y. S. C. H. Xiongwei Wua, "Recent advances in deep learning for object detection.," URL https://arxiv.org/pdf/1908.03673.pdf.

[3] T. D. y. J. M. Ross Girshick, Jeff Donahue, "Rich feature hierarchies for accurate object detection and semantic segmentation.," 2014. URL https://arxiv.org/abs/1311.2524.

[4] P. D. y. R. B. G. K. He, G. Gkioxari, "Mask r-cnn.," 2017. URL https://arxiv.org/pdf/1703.06870.pdf.

[5] R. G. y. A. F. Joseph Redmon, Santosh Divvala, "You only look once: Unified, real-time object detection.," 2016. URL https://arxiv.org/abs/1506.02640.

[6] V. M. OSORIO, "Amazon lanza la cesta de la compra por suscripción.," URL https://www.expansion.com/economia-digital/companias/2017/02/01/589187f0e5fdeabb0d8b45f1.html.

[7] F. RETAIL, "¿un carro de la compra inteligente? caper lo consigue.," URL https://www.foodretail.es/industria-auxiliar/carro-inteligente-compra-caper_2_1290790909.html.

[8] P. V. y Michael J. Jones, "Robust real-time object detection," 2001. URL https://www.researchgate.net/publication/215721846_Robust_Real-Time_Object_Detection.

[9] N. D. y Bill Triggs, "Histograms of oriented gradients for human detection," 2005. URL https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf.

[10] I. S. y. G. E. H. A. Krizhevsky, "Imagenet classification with deep convolutional neural networks.," 2012. URL https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[11] X. Z. M. M. R. F. y. Y. L. Pierre Sermanet, David Eigen, "Overfeat: Integrated recognition, localization and detection using convolutional networks.," 2014. URL https://arxiv.org/abs/1312.6229.

[12] R. Girshick, "Fast r-cnn.," 2015. URL https://arxiv.org/abs/1504.08083.

[13] Y. J. P. S. S. R. D. A. D. E. V. V. y. A. R. Christian Szegedy, Wei Liu, "Going deeper with convolutions.," 2015. URL https://arxiv.org/abs/1409.4842.

[14] R. G. y. J. S. Shaoqing Ren, Kaiming He, "Faster r-cnn: Towards real-time object detection with region proposal networks.," 2016. URL https://arxiv.org/abs/1506.01497.

[15] S. R. y. J. S. Kaiming He, Xiangyu Zhang, "Deep residual learning for image recognition.," 2016. URL https://arxiv.org/abs/1512.03385.

[16] K. H. y. J. S. Jifeng Dai, Yi Li, "R-fcn: Object detection via region-based fully convolutional networks.," 2016. URL https://arxiv.org/abs/1605.06409.

[17] D. E. C. S. S. R. C.-Y. F. y. A. C. B. Wei Liu, Dragomir Anguelov, "Ssd: Single shot multibox detector.," 2016. URL https://arxiv.org/abs/1512.02325.

[18] J. R. y A. Farhadi, "Yolo9000: Better, faster, stronger.," 2017. URL https://arxiv.org/pdf/1612.08242.pdf.

[19] J. S. y. Q. V. L. Barret Zoph, Vijay Vasudevan, "Learning transferable architectures for scalable image recognition," 2018. URL https://arxiv.org/abs/1707.07012.

[20] X. Z. G. M. X. X. y. J. S. Yukang Chen, Tong Yang, "Detnas: Backbone search for object detection.," 2019. URL https://arxiv.org/abs/1903.10979.

[21] W. K. S. P. y. J. D. Payne, A., "Estimating mango crop yield using image analysis using fruit at 'stone hardening' stage and night time imaging," 2014. URL https://www.researchgate.net/publication/259511023_Estimating_mango_crop_yield_using_image_analysis_using_fruit_at_'stone_hardening'_stage_and_night_time_imaging.

[22] F. D. B. U. T. P. y. C. M. Inkyu Sa, Zongyuan Ge, "Deep-fruits: a fruit detection system using deep neural networks.," 2016. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5017387/.

[23] M. R. y Clay Sheppard, "Deep count: Fruit counting based on deep simulated learning.," 2017. URL https://www.mdpi.com/1424-8220/17/4/905.