AJEA- Actas de Jornadas y Eventos Académicos de UTN
DOI: https://doi.org/10.33414/ajea.1145.2022

Detección de pose usando Deep Learning en ambientes industriales

Pose Detection using Deep Learning in Industrial Environments

Presentación: 26 y 27 de octubre de 2022

Rebeca Yuan
Universidad Tecnológica Nacional Facultad Regional San Francisco
ryuan@facultad.sanfrancisco.utn.edu.ar

Micaela Mulassano
Universidad Tecnológica Nacional Facultad Regional San Francisco

Javier Redolfi
Universidad Tecnológica Nacional Facultad Regional San Francisco
javierredolfi@sanfrancisco.utn.edu.ar

Resumen

Cada vez es más común aplicar visión artificial en distintos entornos industriales, para establecer la ubicación, forma y calidad de los objetos. En trabajos anteriores se analizaron algoritmos de *machine learning* que permitieron detectar objetos y su presentación (frente/dorso). La presente investigación analiza la aplicación de una red convolucional profunda conocida como U-Net, para la segmentación de imágenes. El objetivo es lograr obtener la pose de los objetos, completando así la información de pose (ubicación) para utilizar métodos de *bin picking* (recogida aleatoria de contenedores) que completen el proceso de embalaje del objeto.

Palabras Claves: redes convolucionales, pose, U-Net

Abstract

It is becoming more and more common to apply machine vision in different application domains, to establish the location, quality, formation of objects. In previous works, machine learning algorithms were analyzed to detect objects and their presentation (front/back). The present research analyzes the application of a deep convolutional network defined under the U-Net algorithm, for image segmentation. The objective is to obtain the pose of the objects, thus completing the pose information (location) in order to use bin picking methods to complete the object packing process.

Keywords: convolutional networks, pose, U-Net

Introducción

Ya sea en medicina, industria, control de procesos, calidad, etc., la visión artificial está tomando cada vez más resonancia por los excelentes resultados que se obtienen en detección y clasificación. En estudios previos realizados por el equipo de investigación (Redolfi, 2020), se buscó obtener datos de paquetes de salchicha que viajaban en cintas transportadoras. El objetivo era poder detectar los paquetes, conocer su presentación (frente/dorso) y posición, para que estos datos sean receptados por un robot tipo tridop/araña que, en punta de la línea de embalaje, pueda ubicar los paquetes de salchicha en forma correcta dentro de la caja para su embalaje. Los métodos de *bin picking*, que detectan y extraen automáticamente las piezas situadas aleatoriamente dentro de una cinta o contenedor, proporcionan una mejora en la cadena productiva, reduciendo el tiempo del proceso, aumentando la productividad y mejorando la calidad laboral (Rodriguez, 2019). Estos métodos giran en torno a la visión por computadora y la robótica. La visión por computadora, permite el reconocimiento de la imagen, para esto, en investigaciones anteriores se utilizó YOLO.v3 (Yuan Rebeca, et al. 2021)(Redolfi, 2020)(Hmrishav Bandyopadhyay, 2022). A diferencia de otros algoritmos de

AJEA- Actas de Jornadas y Eventos Académicos de UTN
DOI: https://doi.org/10.33414/ajea.1145.2022

reconocimiento que reutilizan los clasificadores para realizar la detección, YOLO utiliza una red neuronal haciendo predicciones de cada cuadro y probabilidad de clases de pertenencia de la imagen al mismo tiempo. La principal ventaja de este algoritmo es la velocidad, pudiendo correr a 15 cuadros por segundo en una PC común y llegando a 120 cuadros por segundo en PC con placas de video. Los enfoques basados en redes neuronales convolucionales (CNN: *convolutional neural network*) tienen gran éxito en el reconocimiento de imágenes. Uno de los avances de las redes convolucionales (CNN: *convolutional neural network*) se corresponde a R-CNN (Regiones con CNN) (Evan Shelhamer, Jonathan Long, 2016). Los algoritmos como Faster R-CNN funcionan detectando posibles regiones de interés utilizando *Region Proposal Network* y luego realizan el reconocimiento de esas regiones por separado, YOLO realiza todas sus predicciones con la ayuda de una sola capa completamente conectada. Si bien los resultados fueron positivos, no se logró establecer la pose del objeto dentro de la cinta. La presente investigación, busca completar los datos necesarios para el empaque de objetos que se mueven en cintas transportadoras, para su empaque final.

Pose

La pose se define como la ubicación y orientación que caracteriza al objeto y que parte de un punto de referencia ubicado en un punto estratégico dentro de dicho objeto, ya sea el centro del objeto o una zona característica. Para este caso en particular en donde trabajamos en un plano de referencia como la cinta transportadora, la pose está definida como las posiciones x e y del objeto y la orientación con respecto a la cinta transportadora. En el trabajo citado se utilizó Yolo para encontrar la posición x e y del objeto, pero el problema de usar esta red es que es incapaz de encontrar la orientación del objeto. Debido a esta limitación se decidió encarar el problema como uno de segmentación semántica. La segmentación semántica consiste asignar una etiqueta de clase a cada píxel perteneciente al objeto. Para hacer esto se decidió utilizar el algoritmo U-NET (Libin Jiao, Lianzhi Huo, 2020).

U-NET

Los modelos basados en CNN (Evan Shelhamer, Jonathan Long, 2016), como YOLO utilizan extractores de características para aprender de forma adaptativa características dentro de las imágenes y, posteriormente, mapearlas. Las CNN clasifican cada píxel de una imagen de forma individual, presentándolo como parches extraídos alrededor del píxel concreto, luego producen un mapa de probabilidad multicanal del mismo tamaño que la imagen de entrada. Para mitigar el gran consumo de memoria, se añade una capa de muestreo (como la agrupación de máximos y la agrupación de promedios) después de varias capas convolucionales. Sin embargo, esto último afecta la resolución en la salida. Uno de los métodos propuestos para solucionar este inconveniente son las FCN (*Fully Convolutional Networks*). Las FCN pueden aceptar cualquier tamaño de imagen de entrada; la última capa convolucional puede restaurar la dimensión de sus entradas al mismo tamaño que la imagen inicial, de modo que se puede generar una predicción para cada píxel, conservando la información espacial de la imagen de entrada original (Ver Figura 1).

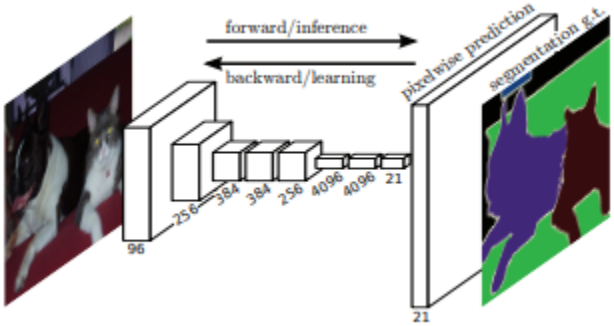


Figura 1. Las redes totalmente convolucionales pueden aprender a realizar predicciones para tareas complejas.

Al igual que las FCN, la U-Net (Olaf Ronneberger, Philipp Fischer, 2015) consta de capas convolucionales, capas de muestreo descendente y capas de muestreo ascendente. Una característica distintiva con respecto a las FCN es el número de capas de muestreo descendente y capas de muestreo ascendente y capas de convolución, entre ellas, en la U-Net es el mismo. Además, U-net utiliza la operación de conexión de salto para conectar cada par de capas de muestreo descendente y la capa de muestreo ascendente, lo que que hace que la información espacial se aplique directamente a capas mucho más profundas y que el resultado de la segmentación sea más preciso. La arquitectura U-Net se deriva de la llamada "red totalmente convolucional" propuesta por primera vez por Long, Shelhamer y Darrell (Evan Shelhamer, Jonathan Long, 2016).

AJEA- Actas de Jornadas y Eventos Académicos de UTN
DOI: https://doi.org/10.33414/ajea.1145.2022

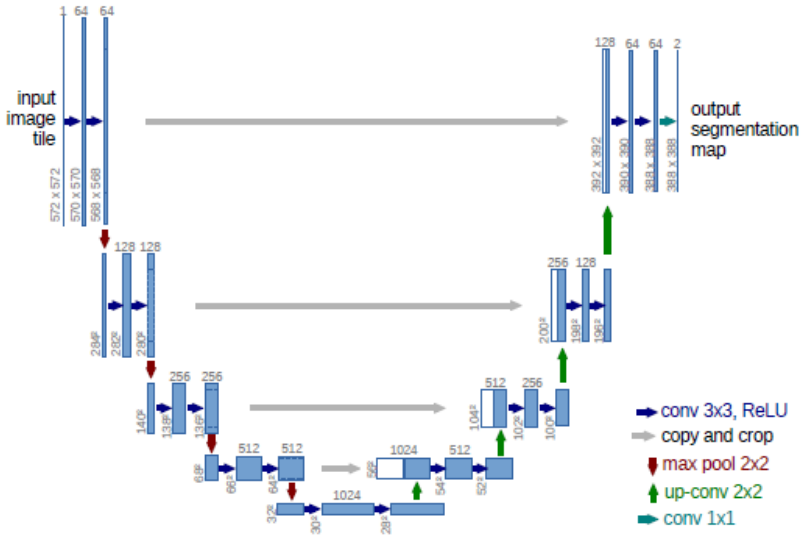


Figura 2. Arquitectura U-net (ejemplo para 32x32 píxeles en baja resolución)

La arquitectura de la red se ilustra en la Figura 2. Consta de una ruta de contracción (lado izquierdo) y un camino expansivo (lado derecho). La ruta de contracción sigue la arquitectura típica de una red convolucional. Consiste en la aplicación repetida de dos convoluciones de 3x3, cada una de ellas seguida de una unidad lineal rectificada (ReLU) y una operación de *pooling* para el muestreo descendente. En cada paso de muestreo descendente se duplica el número de canales. Cada paso de la ruta expansiva consiste en un *upsampling* del mapa de características seguido de una convolución 2x2 ("*up-convolution*") que reduce a la mitad el número de canales de características. En la capa final, se utiliza una convolución 1x1 para asignar cada vector de características de 64 componentes al número deseado de componentes. En total, la red tiene 23 capas convolucionales.

Desarrollo

Para el entrenamiento de la red, es necesario tener imágenes etiquetadas en donde se indique en que parte de la imagen se encuentran los paquetes a detectar. El etiquetado de las imágenes se realizó a través del software Labelimg (https://github.com/heartexlabs/labelImg), de característica *open source* que permitió hacer factible el etiquetado como se observa en la imagen (Figura 3).

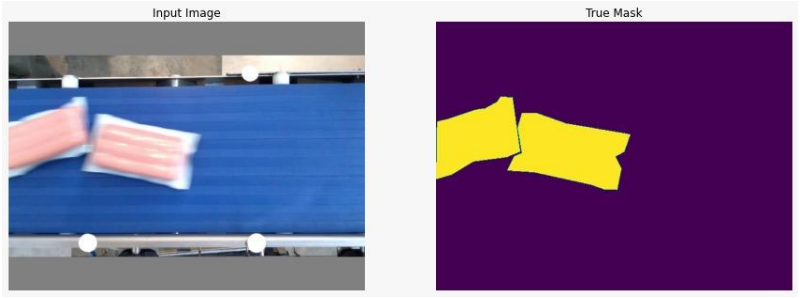
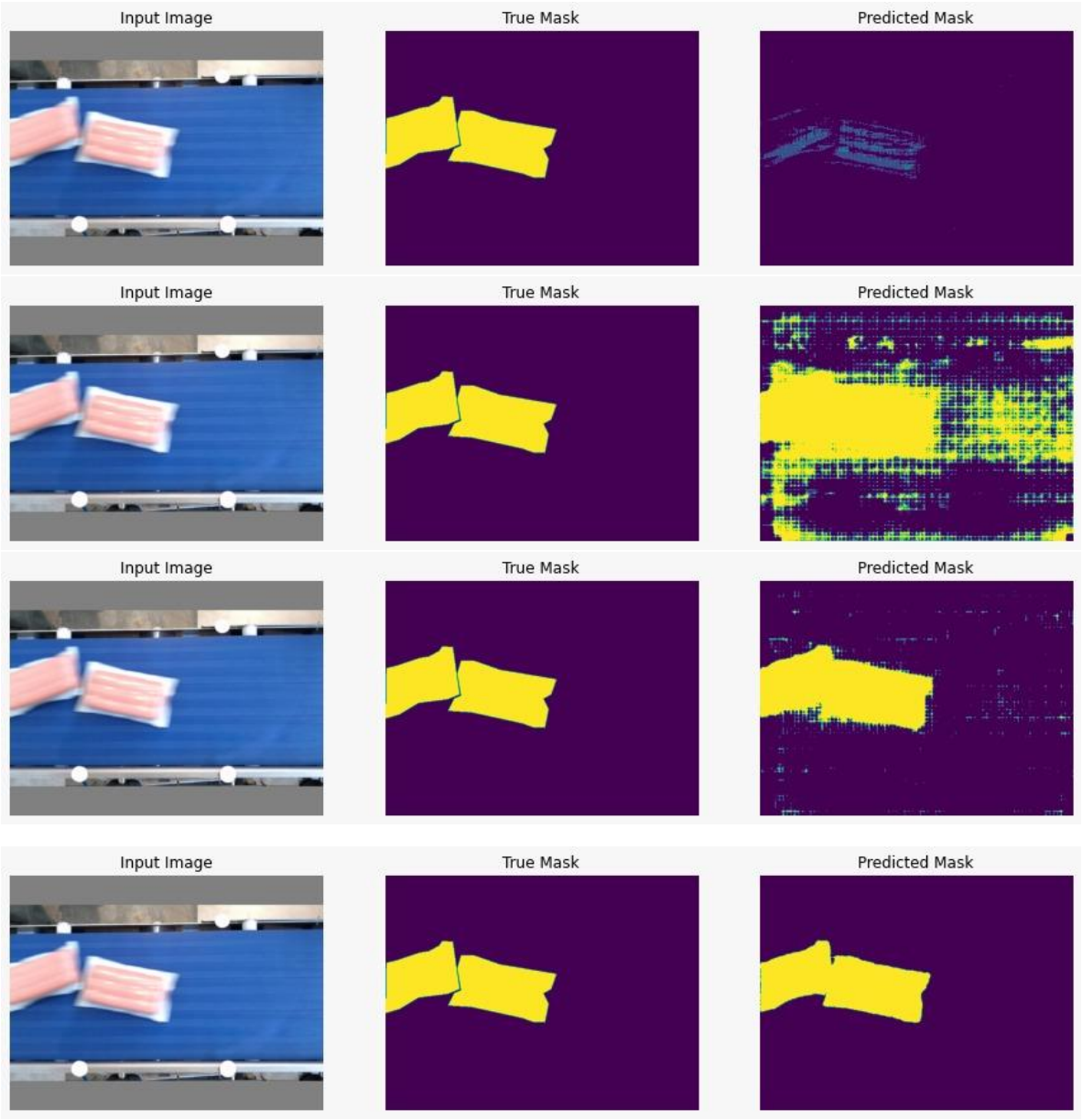


Figura 3. Etiquetado a través de Labelimg

Después del etiquetado se procedió al entrenamiento de la red. Dicho entrenamiento se realizó en la plataforma Colab de Google (https://colab.research.google.com). Se utilizaron 18 imágenes de entrenamiento y se corrieron 20 iteraciones de entrenamiento.

A continuación, en la secuencia de imágenes, se muestran diferentes predicciones de la red a medida que se ejecutaban diferentes pasos de entrenamiento. En un comienzo se puede observar que la predicción es muy baja (*Predicted Mask*), pero a medida que la red se entrena va comenzando a detectar los paquetes.

AJEA- Actas de Jornadas y Eventos Académicos de UTN
DOI: https://doi.org/10.33414/ajea.1145.2022



Otra forma de analizar la evolución de aprendizaje es a través de la función de pérdida o *loss function*, la cual evalúa la desviación entre las predicciones realizadas por la red neuronal y los valores reales de las observaciones utilizadas durante el aprendizaje. Cuanto menor es el resultado de esta función, más eficiente es la red neuronal. La siguiente gráfica (Figura 4) muestra la evolución de la función pérdida de la red neuronal a medida que aumentan los pasos de entrenamiento.

AJEA- Actas de Jornadas y Eventos Académicos de UTN
DOI: https://doi.org/10.33414/ajea.1145.2022

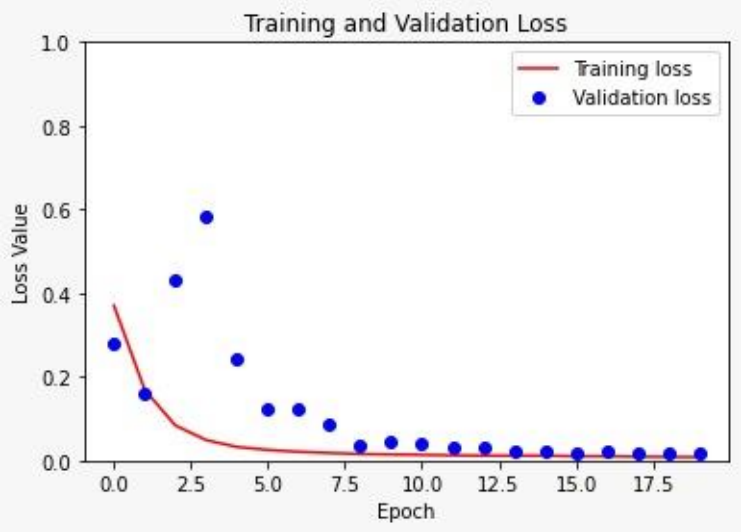


Figura 4. Función de Pérdida

Resultados experimentales

Una vez entrenada la red, se la aplicó al reconocimiento de nuevos paquetes de test, no utilizados en el entrenamiento. En la siguiente imagen (Figura 5), se pueden ver algunos ejemplos de las detecciones realizadas).

Figura 5. Reconocimiento de pose

Conclusiones

A través del entrenamiento de la red U-Net se logró obtener la pose de los paquetes en el conjunto de imágenes de test. Lo que resta para poder pasarle esta información a un sistema de *bin picking* es calcular el centroide y la orientación de cada paquete en base a la máscara obtenida a través de U-Net. A fututo, se planea capturar nuevos videos para entrenar el algoritmo con un mayor número de imágenes, además de etiquetar la orientación de cada paquete para compararla con la obtenida usando el algoritmo propuesto.

Referencias

Evan Shelhamer, Jonathan Long, and T. D. (2016). Fully Convolutional Networks for Semantic Segmentation. *IEEE Xplorer*. https://doi.org/10.1109/TPAMI.2016.2572683

Hmrishav Bandyopadhyay. (2022). *https://www.v7labs.com/blog/yolo-object-detection*. https://www.v7labs.com/blog/yolo-object-detection

Libin Jiao, Lianzhi Huo, C. H. and P. T. (2020). Refined UNet: UNet-Based Refinement Network for Cloud and Shadow Precise Segmentation. *Remote Sens.* https://doi.org/10.3390/rs12122001

Olaf Ronneberger, Philipp Fischer, and T. B. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *Springer*, *Vol.9351*, 234–241. http://lmb.informatik.uni-freiburg.de/

Redolfi, R. Y. I. J. B. J. C. J. A. (2020). Detección de Paquetes en Movimiento sobre una Cinta Transportadora Usando Visión por Computadora. *IEEE Xplorer*. https://doi.org/10.1109/ARGENCON49523.2020.9505530

AJEA- Actas de Jornadas y Eventos Académicos de UTN
DOI: https://doi.org/10.33414/ajea.1145.2022

Rodriguez, A. C. (2019). *Sistema de detección y estimación de pose de objetos basado en visión por computador para planta piloto Industria 4.0*. Universidad de Oviedo.

Yuan Rebeca, Mulassano Micaela, Chiabrando Bruno, Jaime Ibrahim, Cervetti Gonzalo, R. J. (2021). *Detección de pose de objetos usando cámaras RGB para aplicaciones industriales*. https://doi.org/: https://doi.org/10.33414/ajea.1.871.2021