Detección de orientación de paquetes en movimiento sobre cintas transportadoras

Rebeca Yuan, Ignacio Cipolatti, Alejandro Juarez, Iván Dietta, and Javier Andrés Redolfi

Grupo de Investigación Sobre Aplicaciones Inteligentes (GISAI), Facultad Regional San Francisco, Universidad Tecnológica Nacional

Resumen

Cada vez es más frecuente la utilización de técnicas de visión por computadora en diferentes entornos industriales con el fin de identificar la localización, forma y calidad de los objetos. En este trabajo se presenta un método para encontrar la rotación de objetos que presentan forma rectangular, en particular paquetes de salchichas que se mueven sobre una cinta transportadora. Dicha información junto con la posición del paquete pueden ser utilizadas en punta de línea por un robot que automatice el embalaje de los paquetes en cajas. En investigaciones previas se han estudiado e implementado algoritmos de aprendizaje automático para la detección de objetos y su orientación (frontal o posterior), obteniendo buenos resultados, pero esta información no resultó suficiente para encontrar la rotación exacta del producto. La segmentación de imágenes, no solo detecta objetos sino que logra separar objetos de interés del fondo que los contiene y de otros elementos presentes en la imagen. En este estudio se presenta la aplicación de una red neuronal convolucional profunda conocida como U-Net para segmentar imágenes y en base a esa información obtener la posición y rotación de los objetos y así completar la información de su pose del objeto para poder utilizar métodos de selección aleatoria de contenedores que complementen el proceso de embalaje de los objetos.

1. Introducción

A la hora implementar una solución en procesos de automatización; el tiempo y la precisión en la ejecución de las distintas tareas, resultan métricas distintivas. El requerimiento que dió inicio a esta investigación, promulgó la idea de aplicar algoritmos de aprendizaje de máquina para la detección de paquetes en cintas transportadoras que permitan establecer los atributos de posición y rotación para qué en punta de línea de embalaje un robot tipo araña tome los paquetes y proceda al embalaje. Bajo esta premisa, se establece que la solución vendrá de la mano de algoritmos de visión artificial.La visión artificial se enfoca en desarrollar algoritmos y sistemas capaces de comprender e interpretar imágenes de manera similar a como lo hacen los seres humanos. Tal como conocemos actualmente la fisiología del sistema visual, se puede establecer un paralelismo con las redes convolucionales profundas [1], sobre todo en lo referente al reconocimiento rápido de objetos [10]. Los sistemas de visión basados en redes neuronales convolucionales (CNN) han demostrado buenos resultados. Este campo de estudio no solo permite la detección y reconocimiento de objetos, también da lugar a una clasificación de los mismos en las categorías que se especifiquen. Desde entonces su evolución es cada vez más significativa. En la actualidad nos encontramos con redes categorizadas como convolucional, pero con arquitecturas distinta a la que le diera origen, mejorando cada vez su funcionamiento.En el trabajo realizado por [7] utilizan una red convolucional junto al modelo de máquinas de soporte vectorial (MSV) con el objetivo de localizar y categorizar objetos que se encuentren muy cercanos, nombrando a esta categorización como una discriminación fina, con el objeto de acercarse a la sementación de imágenes. En estudios previos realizados por el equipo de investigación [13], 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 araña que, en punta de la línea de embalaje, pueda agarrarlos y ubicar los paquetes de salchicha en forma correcta

dentro de la caja para su embalaje. Los métodos de recolección automática, 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 [2]. Estos métodos giran en torno a la visión por computadora y la robótica. En el trabajo mencionado utilizamos para el reconocimiento de objetos YOLO. El algoritmo You Only Look Once (YOLO) es rápido y preciso [3] esto se traduce en el pequeño tamaño del modelo y en su rápida velocidad de cálculo. La estructura de YOLO es sencilla, permitiendo la detección de objetos más rápida y eficiente en comparación con otros enfoques. Puede emitir directamente la posición y la categoría del cuadro delimitador a través de la red neuronal. YOLO es capaz de detectar múltiples objetos en una imagen simultáneamente, ya que realiza la detección en una sola pasada. Esto es beneficioso en escenarios en los que es importante detectar y localizar varios objetos en una escena, pero puede tener dificultades con la detección precisa de objetos pequeños y la segmentación semántica. YOLO no se centra en la segmentación semántica, por lo que se comenzó a investigar otros algoritmos, que dieran soporte a la solución buscada. En función a estas características, es que se pensó en la segmentación de imágenes a través de U-NET. La segmentación semántica es un área dentro de la visión artificial que se centra en asignar una etiqueta o clasificación a cada píxel de una imagen. Esto permite identificar y delimitar las regiones de interés en la imagen y es útil para tareas como la identificación de objetos que se encuentren muy próximos.

En este trabajo se plantea la hipótesis de qué usando una red neuronal convolucional de segmentación y un posterior algoritmo de detección de ángulos es posible obtener la orientación de los objetos que viajan sobre cintas transportadoras en ambientes industriales en base imágenes de los mismos.

Las contribuciones principales de este trabajo son la obtención de un método simple para obtener la pose y ángulo de los objetos, utilizando pocas muestras de entrenamiento.

2. Problema

Para que un robot tome un paquete que se traslada por una cinta, se necesita conocer la pose del mismo en un sistema de coordenadas global. 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, donde se trabaja 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. Si el objeto a identificar tuviera forma circular, encontrar el centro del mismo, una vez identificado el objeto no implicaría tanto esfuerzo. En este punto, nuestro problema es que el objeto bajo estudio presenta una estructura rectangular y con límites amorfos.

En investigaciones anteriores [14, 13] se utilizó YOLO.v3 [8] para encontrar la posición x e y del objeto con el objetivo posterior de encontrar la orientación en base a algún algoritmo de regresión. Los modelos basados en CNN [5], 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. Yolo basa su arquitectura en el uso de una CNN para extraer características de la imagen, seguida de una capa de detección que produce regiones de interés (RoI) para el objeto de interés. El trabajo realizado con YOLO permitió detectar, en imágenes, tamaño y forma de los objetos, logrando identificar la posición de los paquetes de salchichas como se observa en la figura 1.

A través de YOLO, se logró la detección de los paquetes en las imágenes, pero no resultó suficiente base para encontrar la orientación de los objetos detectados. Como se observa en la figura 1, el recuadro de detección excede los límites del paquete y esta diferencia va a impactar en un sesgo sobre la obtención de la rotación del mismo.

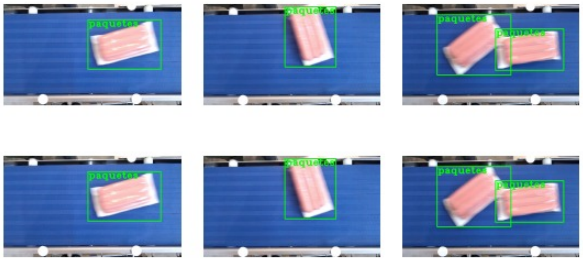

Figura 1: Resultado de la aplicación de YOLO sobre las imagenes de prueba

3. Solución Propuesta

Debido a la limitación expresada en el párrafo anterior, se decidió resolver el problema de la siguiente manera: realizar la detección de los paquetes en base a un algoritmo de segmentación para separar el objeto de interés del fondo (sección 3.1); luego sobre la imagen de segmentación obtenida aplicar un algoritmo para obtener el contorno del paquete en la imagen segmentada (sección 4.2.2) y por último en base a cálculos algebraicos básicos obtener la orientación del paquete (sección 4.2.3).

3.1. Método Propuesto: Segmentación Semántica

"La Segmentación Semántica es un conjunto de técnicas que permiten crear regiones dentro de una imagen y atribuir significado semántico a cada una de ellas" [5]. La misma consiste en asignar una etiqueta de clase a cada píxel perteneciente al objeto.

Por su lado, las redes neuronales convolucionales realizan tareas de reconocimiento de imágenes a través del apilado de muchas capas de procesamiento generando métodos conocidos como de aprendizaje profundo. Cada capa de procesamiento obtiene información relativa a patrones visuales; a medida que se va profundizando o acercando a la salida, se va obteniendo un mayor nivel de rasgos capturados. Las capas finales serán las que capturan el contenido semántico y conceptual de la imagen.

Actualmente se encuentran distintas arquitecturas de CNNs para la segmentación semántica. Al lograr encontrar características abstractas que definen a cada clase semántica, las redes totalmente convolucionales (FCN, por sus siglas en inglés) se utilizan para múltiples soluciones, entre ellas la segmentación semántica, aunque su uso es cada vez más frecuente en distintas áreas [11]. 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 pixel, conservando la información espacial de la imagen de entrada original (ver figura 2).

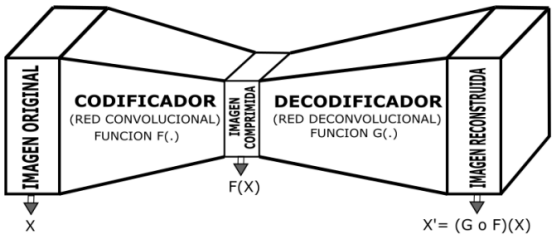

Figura 2: Arquitectura FNC. Tomado de [12].

La FCN toma la imagen original y la comprime en mapas de características que contienen la información semántica de los objetos, pero con resolución espacial y de color reducidas [12]. Sin embargo estas redes presentan dificultades en su aplicación al no considerar la variación espacial que pueda existir en los objetos. Dentro de la variedad de algoritmos de segmentación semántica, se decidió utilizar el algoritmo U-NET [4], para dar solución a esta etapa del proceso.

3.1.1. U-Net

Al igual que las FCN, la U-Net [9] 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 [5]. La arquitectura de la red se ilustra en la figura 3.

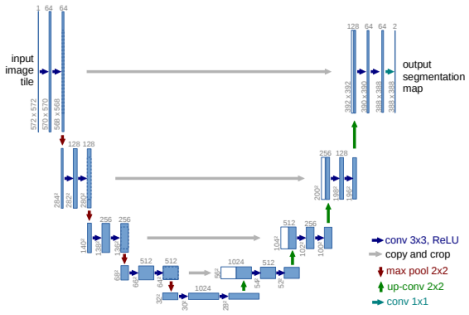

Figura 3: Arquitectura U-net. Figura tomada de [9].

La arquitectura de la red U-Net trata de un proceso de contracción y expansión [9]; la ruta de contracción se puede apreciar en el lado izquierdo y el camino expansivo en el lado derecho de la 3. La ruta de contracción sigue la arquitectura típica de una red convolucional. En cambio, la etapa de contracción realiza una agrupación consecutiva de convoluciones (3X3 conv. y 2X2 max), esto permite extraer características más avanzadas, pero reduce el tamaño de las mismas. Lo anterior también se conoce como sampleo hacia abajo o agrupación, y se encarga de extraer la significación semántica tanto de la imagen como de las diferentes partes de la misma. En cada paso de muestreo descendente se duplica el número de canales. Cada paso de la ruta expansiva consiste en un sampleo hacia arriba del mapa de características seguido de una convolución 2x2 (convolución hacia arriba) que reduce a la mitad el número de canales de características. El proceso de expansión realiza una conversión para recuperar la segmentación, construir el mapa de segmentación semántica y se compone de capas de convolución e incremento de resolución. Después de cada sampleo hacia arriba, logramos obtener la localización del objeto, desde la contracción hasta la expansión. 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.

4. Metodología de los Experimentos

A continuación se describe el material que se utilizó para el entrenamiento de la U-Net, nuestros datos base. Las imágenes se obtuvieron en un ambiente industrial en dónde el sistema trabajaría a futuro respetando las condiciones de iluminación y velocidad de la cinta.

4.1. Material de Trabajo

4.1.1. Conjunto de datos

Para el entrenamiento de la red de segmentación es necesario contar con un conjunto de imágenes etiquetadas del objeto a segmentar. Para esto se capturaron imágenes de paquetes de salchichas que se deslizaban en una cinta transportadora con una ubicación aleatoria. En base a las imágenes obtenidas se construyó un conjunto de datos de trabajo consistente de 45 imágenes de entrenamiento y 12 imágenes de test.

Además de contar con las imágenes, para entrenar UNet se necesita etiquetar en las imágenes cada paquete para distinguirlo del fondo. Este procedimiento se realizó usando el software LabelMe [1].

En la figura 1 se pueden ver algunos ejemplos de los datos de entrenamiento y en la parte central de la figura 4 se puede ver la máscara de etiquetado generada para algunos ejemplos.

4.2. Métodos Utilizados

Para alcanzar nuestro objetivo, realizamos en primer lugar el entrenamiento de la red convolucional U-NET. Trabajar con aprendizaje supervisado implica dividir el set de datos en dos conjuntos, entrenamiento y prueba. El primero se utiliza para entrenar el modelo y el segundo para validar el aprendizaje del mismo. En segundo lugar, se explica el proceso realizado para la obtención de contornos de la imágen. Por último, el cálculo del ángulo de rotación del paquete con respecto a la cinta transportandora.

4.2.1. Entrenamiento de la red

Después del etiquetado se procedió al entrenamiento de la red. Dicho entrenamiento se realizó en la plataforma Colab de Google [2]. Se utilizaron 45 imágenes de entrenamiento y se corrieron 20 iteraciones de entrenamiento. En la figura 4 se muestran algunos ejemplos de las predicciones de la red a medida que se avanzan la etapa de entrenamiento. En un comienzo se puede apreciar que la predicción es poco confiable, pero a medida que la red aprende empieza a detectar mejor los paquetes.

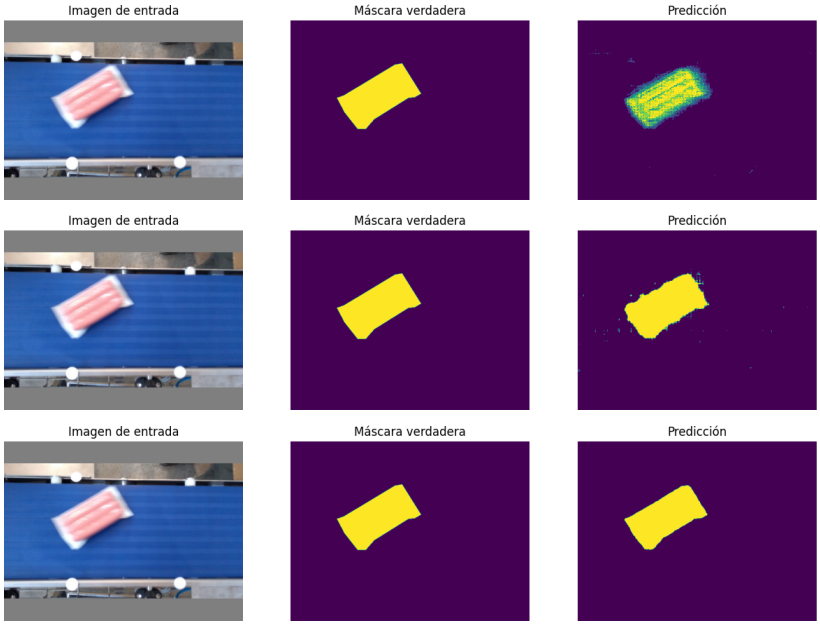

Figura 4: Resultados de la segmentación a medida que la red aprende.

Otra forma de analizar la evolución de aprendizaje es a través de la función de pérdida, 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. En la figura 5 se muestra la evolución de la función pérdida de la red neuronal a medida que aumentan los pasos de entrenamiento. Como se puede apreciar en la figura, con pocos pasos de entrenamiento el error de la red llega a un valor bajo.

4.2.2. Obtención de Contornos

Para la obtención de los contornos se parte de la salida de segmentación que genera la red como se muestra en la parte derecha de la figura 4. Sobre esta imagen se aplica un umbralizado seguido de

[1] https://github.com/wkentaro/labelme
[2] https://colab.research.google.com

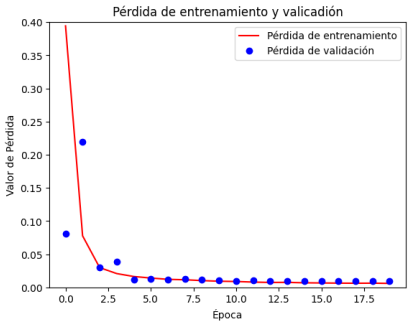

Figura 5: Evolución de la función de pérdida durante el entrenamiento.

una operación de dilatación y erosión con kernels de 5x5 para eliminar pequeños huecos o ruidos que pueden aparecer en la imagen (imagen izquierda de la figura 6). Después de esto se aplica el algoritmo Canny para detectar bordes como se puede apreciar en la parte central de la figura 6 y por último se utiliza la función para encontrar contornos de la librería OpenCV [3]. Con este procedimiento se limpia la imagen y se obtiene de manera fácil el contorno del paquete como se aprecia en la imagen de la derecha de la figura antes mencionada.

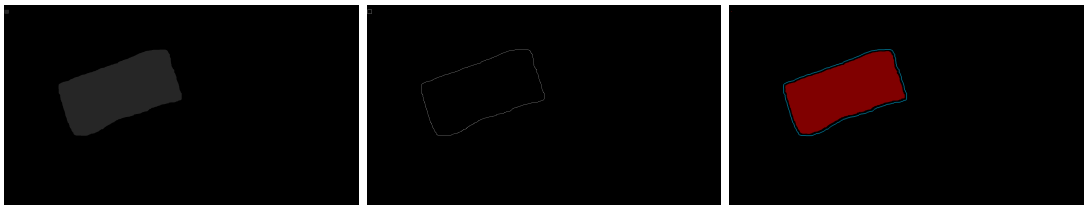

Figura 6: Umbralizado más dilatación y erosión a la izquierda. Detección de bordes usando Canny al centro. Contornos finales y tierra verdadera a la derecha.

4.2.3. Cálculo de la Orientación

En base al contorno del paquete se buscan los 4 vértices del mismo en base a máximos y mínimos en la imagen; luego se busca el lado inferior más largo y en base a ese lado se calcula la orientación con respecto a la horizontal como se muestra en la figura 7.

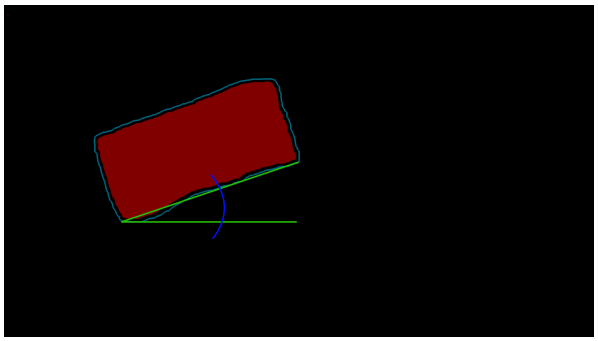

Figura 7: Obtención de la orientación de cada paquete.

[3] https://opencv.org

4.2.4. Cálculo de Orientación Verdadera

Por otro parte, para poder comparar los resultados obtenidos con el algoritmo propuesto se creo un programa que permite marcar manualmente la orientación de cada paquete. Dicho programa muestra una imagen, deja seleccionar los 4 vértices del paquete y en base a ellos calcula la orientación en base al triángulo que se muestra en la figura 7. Estas orientaciones obtenidas son los ángulos "reales" que usaremos como tierra verdadera o GT (por sus siglas en inglés). Este programa se aplica a todas las imágenes de entrenamiento para obtener un vector de orientaciones.

5. Resultados

El Error Cuadrático Medio (MSE) es una métrica utilizada para evaluar la precisión de modelos de regresión. Al trabajar con un modelo supervisado, contamos con un conjunto de datos con los ángulos reales de los paquetes de salchichas en la cinta transportadora. Esta información es de tipo numérica y podemos decir que es nuestra etiqueta, la cual vamos a usar para comparar con los resultados obtenidos en el algoritmo propuesto en estre trabajo anteriormente. El MSE proporciona una medida de magnitud de los errores de predicción. Calcula el promedio de los errores al cuadrado entre los valores predichos y los reales (etiqueta). Para evaluar la exactitud de la solución propuesta, primero se aplica el algoritmo planteado en la sección 4 a todas las imágenes de entrenamiento y se genera un vector de orientaciones con las predicciones realizadas por nuestro algoritmo. Este vector es comparado con el vector de GT planteado en la sección 4.2.4 a través del error cuadrático medio (MSE, por sus siglas en inglés).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\theta_i - \hat{\theta}_i)^2 \tag{1}$$

en donde θ_i es la orientación verdadera y $\hat{\theta}_i$ es la orientación estimada por nuestro algoritmo.

Para las 12 imágenes de test se obtuvo un $MSE = 3{,}60°$ que se corresponde a un $RMSE = 1{,}90°$. El máximo error en ángulo fue de 3,14°.

Si bien se evaluó el algoritmo sobre pocas imágenes de test, se puede decir que el valor obtenido en el MSE y RMSE es bajo; que el error máximo es relativamente bajo, indicando un buen comportamiento en la detección de la inclinación o rotación.

6. Conclusiones y Trabajo a Futuro

En base a los experimentos realizados, podemos confirmar que usando un algoritmo de segmentación de imágenes basado en CNN combinado con un simple algoritmo de procesamiento de imágenes es posible obtener la orientación de paquetes en movimiendo sobre una cinta transportadora. Además podemos decir que el error en la orientación obtenido es bajo para la mayoría de las aplicaciones de empaquetado. De esta manera se presento un algoritmo simple para obtener la pose completa de los paquetes y brindar coordenadas precisas a un robot en punta de línea para continuar con el proceso de embalaje o para realizar cualquier otro proceso.

Como trabajo a futuro se plantean optimizar los resultados en base a Half-UNet la cual propone una solución en la sobrecarga de memoria y cómputo observada en Unet3. La arquitectura propuesta es esencialmente una red de codificador-decodificador basada en la estructura U-Net, en la que se simplifican tanto el codificador como el decodificador [6] (ver figura 8). La arquitectura rediseñada aprovecha la unificación de los números de canal, la fusión de características a gran escala y los módulos fantasma. Los módulos fantasma, se corresponden a una técnica de regulación para redes neuronales para reducir el número de parámetros y mejorar el rendimiento y eficiencia de la red. La técnica utiliza canales "fantasmas"para producir una salida intermedia, en lugar de utilizar todos los canales de salida.

Además se plantea realizar un estudio sobre la posibilidad de generar imágenes sintéticas del problema para aumentar el conjunto de entrenamiento usando técnicas de inteligencia artificial. Y por último se planean implementar formas más robustas de cálculo de la inclinación basándose en el uso de los 4 vértices o en base a la aproximación de las 4 rectas que forman el contorno para luego hacer un promedio de las mismas.

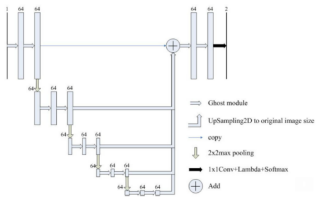

Figura 8: Arquitectura Half Unet. Tomado de [6].

Referencias

[1] Bengio, Y.: Learning deep architectures for ai. Universit´e de Montreal, https://www.iro.umontreal.ca/~lisa/pointeurs/TR1312.pdf

[2] Crespo Rodríguez, A., et al.: Sistema de detección y estimación de pose de objetos basado en visión por computador para planta piloto industria 4.0 (2019)

[3] Jiang, P., Ergu, D., Liu, F., Cai, Y., Ma, B.: A review of yolo algorithm developments. Procedia Computer Science **199**, 1066–1073 (2022). https://doi.org/https://doi.org/10.1016/j.procs.2022.01.135, https://www.sciencedirect.com/science/article/pii/S1877050922001363

[4] Jiao, L., Huo, L., Hu, C., Tang, P.: Refined unet: Unet-based refinement network for cloud and shadow precise segmentation. Remote Sensing **12**(12), 2001 (2020)

[5] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)

[6] Lu, H., She, Y., Tie, J., Xu, S.: Half-unet: A simplified u-net architecture for medical image segmentation. Frontiers in Neuroinformatics **16** (2022)

[7] Ning Zhang, Jeff Donahue, R.G..T.D.: Part-based r-cnns for fine-grained category detection. Computer Vision- ECCV 2014 **8689**(834-849) (2014)

[8] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)

[9] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)

[10] Serre, T., K.G.K.M.C.C.K.U..P.T.: A quantitative theory of immediate visual recognition. In: Progress in Brain Research, Computational Neuroscience: Theoretical Insights into Brain Function, Volumen 165. p. 33–56 (2007)

[11] Sevak, J.S., Kapadia, A.D., Chavda, J.B., Shah, A., Rahevar, M.: Survey on semantic image segmentation techniques. In: 2017 International Conference on Intelligent Sustainable Systems (ICISS). pp. 306–313. IEEE (2017)

[12] Soto-Orozco, O.A., Corral-Sáenz, A.D., Rojo-González, C.E., Ramírez-Quintana, J.A.: Análisis del desempeño de redes neuronales profundas para segmentación semántica en hardware limitado. ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica **8**(2) (2019)

[13] Yuan, R., Jaime, I., Chiabrando, B.J., Redolfi, J.A.: Detección de paquetes en movimiento sobre una cinta transportadora usando visión por computadora. In: 2020 IEEE Congreso Bienal de Argentina (ARGENCON). pp. 1–6. IEEE (2020)

[14] Yuan, R., Mulassano, M., Chiabrando, B., Jaime, I., Cervetti, G., Redolfi, J.: Detección de pose de objetos usando cámaras rgb para aplicaciones industriales