

Detector de fallas en placas PCB mediante imágenes de microscopio utilizando el algoritmo de YOLOv5

A. Cervantes-Marquez*

*Universidad Autónoma de Querétaro

aldocema04@gmail.com

Resumen—El presente trabajo muestra el uso de la inteligencia artificial en la detección de fallas en placas PCB, con el fin de poder auxiliar a los ingenieros, técnicos, personal de calidad de producto y personal de reparación de equipos electrónicos, en donde es necesario realizar una re-ingeniería o reparación de algún componente o pista de la placa. Teniendo como base la red neuronal YOLOv5 para poder detectar este tipo de anomalías y desperfectos en las mismas, a partir de imágenes tomadas por un microscopio electrónico especial para visualizar este tipo de circuitos.

Index Terms— Detector YOLOv5, Placa PCB, Métricas de evaluación.

I. INTRODUCCIÓN

El uso de placas PCB ha sido intensamente usado a lo largo de los años, debido a su simplicidad y utilidad para reducir espacios y encapsular los componentes en áreas de trabajo bien definidas, proporcionando estabilidad y confianza en el componente. Sin embargo, es posible que se encuentren defectos y problemas en estas placas al momento de fabricarse o repararse que afecten su funcionamiento. Estos problemas básicamente se pueden dar desde fábrica como son: Defectos del maquinado y ensamble, defectos de la consistencia de los materiales, etc. Por otro lado, los defectos de reparación pueden ser: soldadura incorrecta, manejo indebido de la placa, factores externos que afecten la integridad física de la placa, entre otros.

Todo esto con el fin de poder identificar las principales fallas del proceso de fabricación o reparación.

Por lo que el uso de detectores basados en inteligencia artificial como YOLOv5 son una opción atractiva para la solución de problemas con imágenes específicas [1]. Pues han estado comprobadas de tener una buena velocidad y estabilidad en sus pruebas con otro tipo de objetos que requieren un área pequeña de detección [2].

II. MARCO TEÓRICO

Para el desarrollo del trabajo, es necesario conocer los siguientes conceptos:

II-A. Placa PCB

Por sus siglas en inglés (Printed Circuit Board), consiste en un circuito cuyos componentes y líneas se encuentran almacenadas en una estructura mecánica. Hechas de un material laminado entre las capas del mismo. La estructura general es chapada y cubierta con una máscara de soldadura no conductora y una pantalla de impresión para la ubicación de leyenda de componentes electrónicos [3].

Actualmente se diseñan en softwares elaborados que permiten generar más de una capa de cobre, teniendo varios niveles de pistas, en este caso únicamente se revisará una sola capa por separado. Sus dimensiones son variadas según la aplicación pero en general se busca tener el tamaño más pequeño gracias a la capacidad de la existencia de componentes de montaje superficial (SMD).

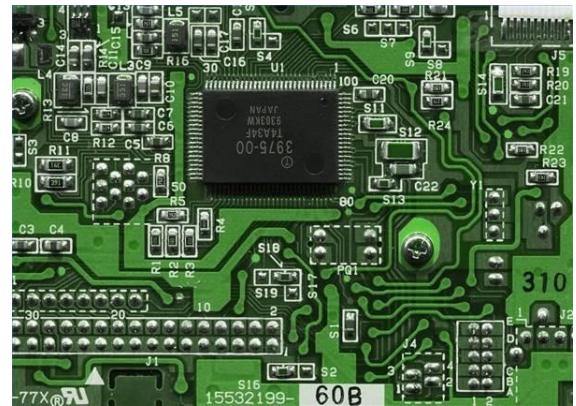


Figura 1. Placa PCB de circuito.

II-B. Microscopio electrónico

Debido a la complejidad y dimensión de las placas PCB es necesario tener una herramienta que permita observar los pequeños componentes que contienen la placa y por lo tanto es necesario revisar la geometría de las pistas y soldaduras con el propósito de calidad y mantenimiento gracias al zoom y calidad de imagen que pueden tener [4]. Estos permiten realizar capturas de los circuitos y grabar, por lo que es posible aplicar software externo para la detección de fallas en la placa (véase Figura 2).



Figura 2. Microscopio para placas PCB.

II-C. Base de Datos

La base de datos consiste en una colección de imágenes de placas que fueron obtenidas a partir de fotos tomadas por un microscopio electrónico en diferentes resoluciones mínimamente variantes y que son en algunos casos inclusive mayores a HD, por lo que se obtienen imágenes de alta calidad, en cada una hay errores en las placas, dirigiéndose a 6 categorías que representan los mayores problemas dentro de la revisión de una placa [5].

- **Agujero sin soldadura (miss_hole)**: ocurre cuando no existe soldadura en la patilla donde el componente es colocado (véase Figura 3).

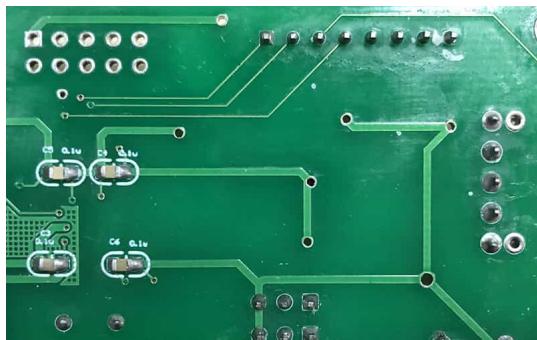


Figura 3. Agujero sin soldadura.

- **Pista incompleta (mouse bite)**: Consiste en una inconsistencia en la pista donde se observa que no es una linea recta la pista, es decir falta material. Esto se debe a un mal proceso de maquinado, inconsistencia del material o una mala aplicación de la mascara de color (véase Figura 4).

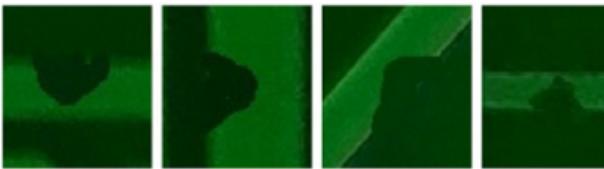


Figura 4. Pista incompleta.

- **Circuito Abierto (Open circuit)**: Consiste en una abertura total en la línea, impidiendo el paso de corriente, esto se debe a un mal maquinado o a un maltrato externo con herramientas inadecuadas principalmente (véase Figura 5).

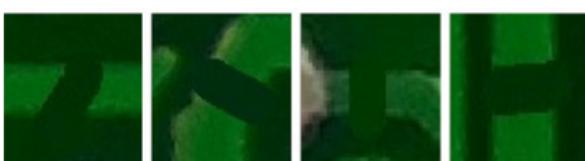


Figura 5. Circuito abierto.

- **Corto circuito (Shortcircuit)** : Es un caso contrario al circuito abierto, en este se cruzan dos pistas que no deberían hacerlo, esto se debe a errores de diseño o bajas tolerancias del maquinado (véase Figura 6).



Figura 6. Corto circuito.

- **Exceso de material (Spur)**: Consiste en un exceso de material en la pista, donde se observa principalmente irregularidades en la forma de la misma con un exceso de material, principalmente se debe a errores en el maquinado de la placa (véase Figura 7).

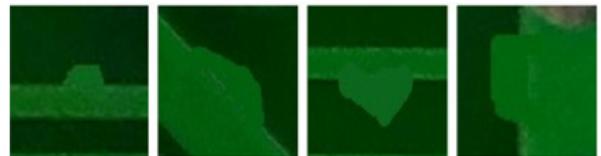


Figura 7. Exceso de material.

- **Falsa pista (Spurious copper)**: Consiste en un segmento de pista que no aporta ninguna utilidad al circuito paralela a otra pista que si es útil. Causando confusiones, retrabajo en el maquinado y una posible causa de ruido electromagnético. Esta es causada principalmente por errores en el software de trazado de las pistas, pues debe haber un espaciado mínimo de material entre pistas según la aplicación (véase Figura 8).



Figura 8. Falsa pista.

La base de datos obtenida tiene 693 imágenes de los circuitos con estas fallas, por lo que se etiquetaron manualmente y posteriormente se le aplicó un aumento de datos para incrementar la base de datos a 1477 imágenes. Los aumentos de datos fueron mediante técnicas de rotación, ruido y rotaciones de las cajas de identificación (véase Figura 9).

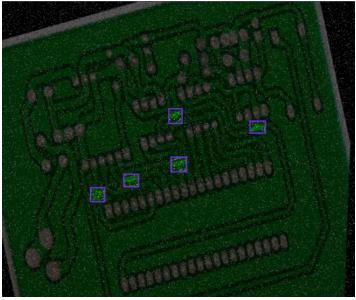


Figura 9. Ejemplo de imagen en la base de datos aumentada.

Finalmente fue normalizada a un conjunto de valores para su utilización en la red YOLOv5.

II-D. Red YOLOv5

Es un algoritmo que consiste en dividir imágenes en un sistema de mallas. Cada malla corresponde a una detección por si misma. Es uno de los algoritmos más famosos debido a su velocidad y su exactitud [6], [7]. Utilizando un optimizador basado en el gradiente descendente [8].

II-D1. Métricas de evaluación: Para comprender que tan bueno fue el modelo desarrollado, se tienen básicamente 3 métricas que son *mAP* (Mean average precision), *Precisión* y *Recall* [9]. Las cuales son definidas de manera matemática respectivamente partiendo de una matriz de confusión en la Figura 10 y intersección sobre la unión (IoU) [10].

VALORES PREDICCIÓN			
	Verdaderos positivos	Falsos Positivos	
VALORES REALES	Falsos Negativos		Verdaderos Negativos

Figura 10. Ejemplo de matriz de confusión.

El *mAP* consiste en encontrar la precisión promedio de cada clase y sobre todas las clases.

$$mAp = \frac{1}{N} \sum_{i=1}^N AP_i \quad (1)$$

La precisión consiste en que tan bien se pueden encontrar positivos entre todas las posibilidades positivas detectadas. Por lo que es posible obtener los límites de las detecciones en cajas de tramaño personalizado, a través de un umbral $IoU \leq x$.

$$P = \frac{TP}{TP + FP} \quad (2)$$

El *Recall* permite saber que tan bien los casos positivos de todas las predicciones positivas correctas.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

III. JUSTIFICACIÓN

El uso de una red neuronal que permite detectar las fallas de una PCB y categorizarla como lo puede realizar una basada en YOLOv5, permite que sea posible prevenir fallas de fábrica o reparación y cuidar los componentes que contiene, con el fin de obtener un funcionamiento correcto de la placa. Permitiendo que sea posible ahorrar dinero y detectar más fácilmente detalles en los procesos.

Por lo que su uso para otras aplicaciones parecidas ha funcionado de manera satisfactoria, teniendo ese antecedente y estudio como base de la justificación teórica de esta aplicación [9].

IV. METODOLOGÍA

Para el desarrollo del proyecto fue necesario aplicar la siguiente metodología (véase Figura 11).

Donde primeramente se realizó una búsqueda de la base de datos abierta, para posteriormente aplicar la segmentación y etiquetado de las clases, para posteriormente aplicar el aumento de los datos y normalización y división de los mismos para formato YOLOv5. Posteriormente se propuso el modelo y se le aplicaron los hiperparámetros de épocas y tamaño de la imagen. Para posteriormente proponer el algoritmo y realizar las pruebas de entrenamiento y test sobre otro conjunto de imágenes y un video. Todo esto hasta lograr resultados aceptables.

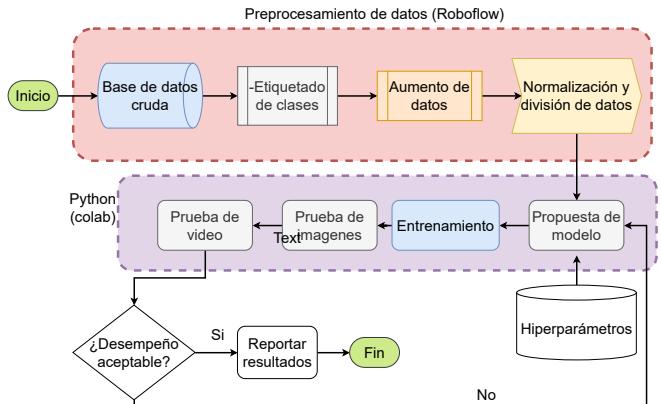


Figura 11. Metodología del trabajo.

V. RESULTADOS

Se realizaron un conjunto de pruebas que permitieron observar un comportamiento interesante en el algoritmo como se muestra en la Tabla I.

Como se observa, mientras se mejora la resolución de la imagen, se incrementan las épocas y se disminuye el batch, se obtienen mejores resultados de *mAP*, precisión y *Recall*.

Tabla I
RESULTADOS BAJO DISTINTAS CONDICIONES DEL ALGORITMO.

Prueba	Tipo de detector	Sub clase	Resolución de imagen	batch	Épocas	mAP (por clase)	Precisión	Recall	Tiempo de entrenamiento
1	YOLO V5	V5s		416	32 (34 imágenes)	0.00738	todas 0.705	todas 0.00184	
					10 (Sin uso de GPU)	miss_hole 0.112	miss_hole 0.23	miss_hole 0.11	
						mouse_bite 0.0005	mouse_bite 0.0005	mouse_bite 0	
						short/circuit 0.0185	short/circuit 1	short/circuit 0	
						Spur 0	Spur 0	Spur 0	
						Spurious_copp 1.00000	Spurious_copp 0	Spurious_copp 0	
2	YOLO V5	V5s		640	16 (78 imágenes)	0.0135	todas 0.0743	todas 0.0085	
					5 (con GPU)	miss_hole 0.0002	miss_hole 0.0002	miss_hole 0	
						mouse_bite 0.0094	mouse_bite 0.0085	mouse_bite 0	
						Open_circuit 0.00455	Open_circuit 0	Open_circuit 0	
						short/circuit 0.00029	short/circuit 0.475	short/circuit 0	
						Spur 0	Spur 0	Spur 0	
						Spurious_copp 0.000486	Spurious_copp 0	Spurious_copp 0	
3	YOLO V5	V5s		640	16 (78 imágenes)	0.345	todas 0.306	todas 0.383	
					20 (con GPU)	miss_hole 0.145	miss_hole 0.145	miss_hole 0	
						mouse_bite 0.147	mouse_bite 0.242	mouse_bite 0.255	
						Open_circuit 0.263	Open_circuit 0.371	Open_circuit 0.371	
						short/circuit 0.00029	short/circuit 0.475	short/circuit 0	
						Spur 0.107	Spur 0.272	Spur 0.103	
						Spurious_copp 0.237	Spurious_copp 0.416	Spurious_copp 0.258	
4	YOLO V5	V5s		640	16 (78 imágenes)	0.519	todas 0.629	todas 0.492	
					50 (con GPU)	miss_hole 0.124	miss_hole 0.124	miss_hole 0	
						mouse_bite 0.324	mouse_bite 0.38	mouse_bite 0.341	
						Open_circuit 0.467	Open_circuit 0.691	Open_circuit 0.466	
						short/circuit 0.00029	short/circuit 0.475	short/circuit 0	
						Spur 3.18e-10	Spur 0.536	Spur 0.260	
						Spurious_copp 0.463	Spurious_copp 0.726	Spurious_copp 0.379	
5	YOLO V5	V5s		640	16 (78 imágenes)	0.511	todas 0.622	todas 0.474	
					100 (con GPU)	miss_hole 0.124	miss_hole 0.124	miss_hole 0	
						mouse_bite 0.367	mouse_bite 0.612	mouse_bite 0.273	
						Open_circuit 0.485	Open_circuit 0.544	Open_circuit 0.471	
						short/circuit 0.00029	short/circuit 0.475	short/circuit 0	
						Spur 0.258	Spur 0.515	Spur 0.181	
						Spurious_copp 0.488	Spurious_copp 0.709	Spurious_copp 0.444	
6	YOLO V5	V5s		640	16 (78 imágenes)	0.622	todas 0.705	todas 0.474	
					200 (con GPU)	miss_hole 0.056	miss_hole 0.832	miss_hole 0.917	
						mouse_bite 0.504	mouse_bite 0.734	mouse_bite 0.375	
						Open_circuit 0.517	Open_circuit 0.772	Open_circuit 0.511	
						short/circuit 0.056	short/circuit 0.812	short/circuit 0.62	
						Spur 0.489	Spur 0.727	Spur 0.406	
						Spurious_copp 0.614	Spurious_copp 0.819	Spurious_copp 0.556	

Obteniendo en la mejor prueba durante las 200 épocas los siguientes comportamientos de las métricas de evaluación.

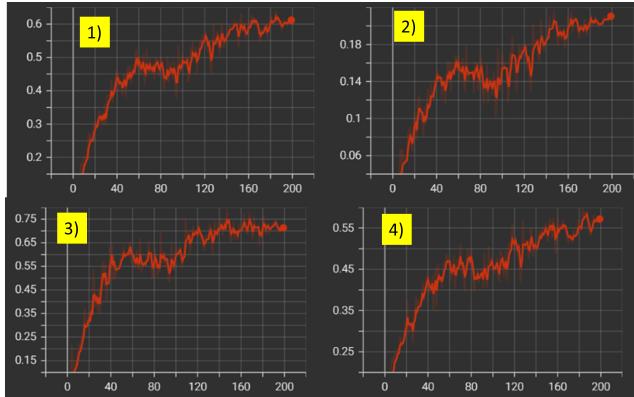


Figura 12. Métricas de los resultados del algoritmo con mejor desempeño.

Donde 1) es la gráfica del mAP , 2) es la gráfica de $mAP_{0.5:0.95}$, 3) es la gráfica de precisión y 4) es la de Recall. Observándose un comportamiento a la alza.

Posteriormente, se observan algunos resultados obtenidos por el algoritmo en la prueba y un video de un microscopio sobre varias placas PCB (véase Figura 13 y 14).

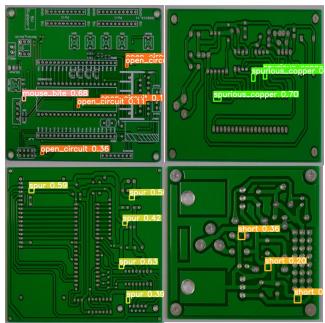


Figura 13. Resultados de las imágenes de prueba de la base de datos.

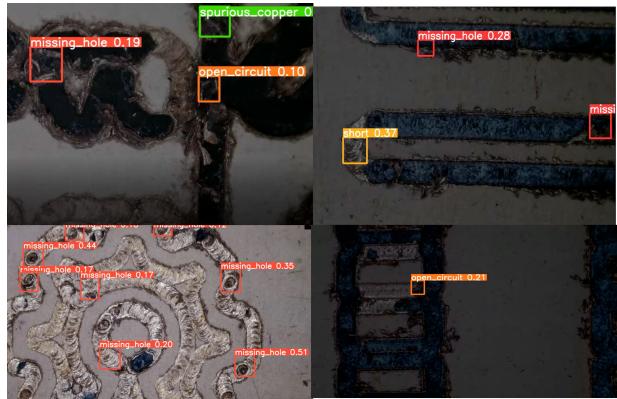


Figura 14. Resultados del video del microscopio.

Como se observa, en la base de datos fue posible obtener una detección buena pero únicamente de grupos muy definidos, por lo que al algoritmo le cuesta trabajo poder realizar detecciones tan específicas de diferentes errores.

Por otro lado, el video tuvo la desventaja de contener casos muy especiales en los que es muy complicado a simple vista saber si el circuito consta de dichos errores, pero al observarse con más detalle, se observan errores de soldadura y desgaste muy evidentes, por lo que las clases que más fueron utilizadas fueron la de exceso de material y ausencia del mismo.

VI. CONCLUSIONES

En este trabajo fue posible implementar un algoritmo de detección basado en YOLOv5 (V5s), el cual permitió detectar las principales fallas en una placa PCB. Sin embargo, los resultados no fueron tan buenos como se esperaban, teniendo como principal problema el computo del entrenamiento de la red puesto que era demasiado tardado y a pesar de ello, se observa un comportamiento a la alza de las métricas de evaluación con una resolución relativamente baja en comparación a las imágenes originales que eran mínimamente HD, por lo que al aplicar una cantidad superior de las mismas, será posible mejorar los resultados con un equipo de computo mucho mas potente, además de que se observa que la composición de la base de datos no era suficientemente grande y variada, atribuyendo a un aprendizaje más lento y con un sobreajuste, puesto que en casi todos los casos únicamente había una clase de falla por placa, sin embargo, esto lo desempeño de una manera correcta a pesar de obtener un mAP relativamente bajo para estas pruebas. En el video se observaron cosas interesantes, puesto que pudo detectar algunas fallas que pueden ser consideradas debido a cambios de iluminación o umbral en las imágenes, así como también confundirse al detectar agujeros en las soldaduras mal realizadas, por lo que es posible observar que si tienen la forma de un agujero. En el caso de la ultima imagen, me pareció interesante como detecta el circuito abierto puesto que no existe una máscara sobre la soldadura (color sobre la pista), lo cual tiene sentido, pues esto de una manera estricta es tomado como un error para cualquier técnico de reparación o calidad. Aún falta depurar

algunos errores en la imagen, pues no se puede asegurar que los errores sean verdaderamente errores sin mirar el circuito de una manera más detallada (tal es el caso de los cortos circuitos que no sabemos exactamente si en verdad deben cruzarse esas pistas). Además de presentar un tiempo de respuesta muy bueno para la detección de las imágenes (en promedio 9ms por frame). Siendo totalmente aceptable y suficiente para esta aplicación, la cual no requiere de una tasa de refresco tan veloz pero si requiere de resultados más certeros, con el fin de identificar de una manera más puntual los errores en la placa.

Considero que para trabajos futuros se deberá adquirir una cámara especial con un conjunto de placas con estas fallas, obteniendo una base de datos más abundante y menos sintética pues las imágenes tenían algunos retoques para mostrar los errores de una manera más clara. También adquirir un equipo de computo más potente para poder entrenar más épocas al algoritmo. Para finalmente ajustar algunos hiperparámetros de una manera más fina y poder realizar más pruebas en tiempo real mediante la cámara.

REFERENCIAS

- 1 Arun, P., “Yolo object detection with opencv and python — by arun ponnusamy — towards data science,” [https://towardsdatascience.com/yolo-object-detection-with-opencv-and-python-21e50ac599e9#:~:text=YOLO%20\(You%20Only%20Look%20Once,in%20C%20from%20the%20author\).](https://towardsdatascience.com/yolo-object-detection-with-opencv-and-python-21e50ac599e9#:~:text=YOLO%20(You%20Only%20Look%20Once,in%20C%20from%20the%20author).), Agosto 2018, (Accessed on 12/04/2022).
- 2 Zhao, Y., Shi, Y., and Wang, Z., “The improved yolov5 algorithm and its application in small target detection,” in *Intelligent Robotics and Applications*, Liu, H., Yin, Z., Liu, L., Jiang, L., Gu, G., Wu, X., and Ren, W., Eds. Cham: Springer International Publishing, 2022, pp. 679–688.
- 3 Zachariah, P., “Significado de pcb ¿qué es un pcb y para qué sirve? — altium,” <https://resources.altium.com/es/p/what-is-a-pcb>, Septiembre 2022, (Accessed on 12/03/2022).
- 4 Guillem, A., “Microscopio digital para inspección de placas - diarioelectronicohoy.com,” <https://www.diarioelectronicohoy.com/microscopio-digital-para-inspeccion-de-placas/>, Junio 2019, (Accessed on 12/05/2022).
- 5 Abhiroop, B. and Sylvain, G. C., “End-to-end deep learning framework for printed circuit board manufacturing defect classification — scientific reports,” <https://www.nature.com/articles/s41598-022-16302-3>, 2022, (Accessed on 12/07/2022).
- 6 “Yolov5 documentation,” <https://docs.ultralytics.com/>, Mayo 2020, (Accessed on 12/07/2022).
- 7 “Yolov5 — pytorch,” https://pytorch.org/hub/ultralytics_yolov5/, (Accessed on 12/06/2022).
- 8 Thuan, D., “Evolution of yolo algorithm and yolov5: The state-of-the-art object detection algorithm,” 2021.
- 9 Zhao, Y., Yang, H., and Feng, H., “An improved yolov5 pcb defect detection,” in *International Conference on Advanced Sensing and Smart Manufacturing (ASSM 2022)*, vol. 12351. SPIE, 2022, pp. 380–387.
- 10 Deval, S., “Mean average precision (map) explained: Everything you need to know,” [https://www.v7labs.com/blog/mean-average-precision#:~:text=let's%20divide%20in!,What%20is%20Mean%20Average%20Precision%20\(mAP\)%3F,values%20from%200%20to%201.](https://www.v7labs.com/blog/mean-average-precision#:~:text=let's%20divide%20in!,What%20is%20Mean%20Average%20Precision%20(mAP)%3F,values%20from%200%20to%201.), Octubre 2022, (Accessed on 12/06/2022).