http://dx.doi.org/10.15446/ing.investig.xxxx

Automatic segmentation of industrial tools using Deep Learning

Segmentación automática de herramientas industriales mediante Deep Learning

Pedro Alejandro Montaño Herrera [1], William Alberto Ramirez Ruiz [2], Juan Pablo Sosa Esquivel[3], Marco Antonio Jinete Gómez [4]

ABSTRACT

Semantic segmentation in simulated scenarios related to robotics has been widely used to devise and justify new implementations related to process automation. In the field of robotics, studying simulated environments allows exploring a wide variety of techniques related to artificial intelligence, seeking optimization in models related to Deep Learning. The present work proposes the use of Deep Learning for segmentation in the detection of tools and objects in a simulated environment for the use of a robotic arm. For this purpose, the corresponding author presents a synthetic dataset using the ROBODK platform, presenting a total of 10 classes commonly present in production lines. The Unet++ and Deeplab V3+ models together with their backbones will be evaluated, emphasizing the use of the Mean iou metric. As a result, the present work presents a new perspective on the use of Deep Learning in computational vision problems in simulated scenarios, obtaining mean iou values of up to 94% in the segmentation of multiple tools.

Keywords: Deep Learning, Deeplab V3, Unet++, Tools segmentation.

RESUMEN

La segmentacion semantica en escenarios simulados relacionados con la robotica se ha utilizado ampliamente para idear y justificar nuevas implementaciones relacionadas con la automatizacion de procesos. En el campo de la robotica, el estudio de entornos simulados permite explorar una amplia variedad de tecnicas relacionadas con la inteligencia artificial, buscando la optimizacion en modelos relacionados con el Deep Learning. El presente trabajo propone el uso de Deep Learning para la segmentacion en la deteccion de herramientas y objetos en un entorno simulado para el uso de un brazo robotico. Para ello, el autor correspondiente presenta un conjunto de datos sinteticos empleando la plataforma ROBODK, presentando un total de 10 clases comunmente presentes en las lıneas de produccion. Se evaluaran los modelos Unet++ y Deeplab V3+ junto con sus backbones, enfatizando el empleo de la metrica Mean iou. Como resultado, el presente trabajo presenta una nueva perspectiva sobre el uso del Deep Learning en problemas de vision computacional en escenarios simulados, obteniendo valores medios de iou de hasta 94 % en la segmentacion de multiples herramientas comunes en la industria.

Palabras clave: Aprendizaje profundo, Deeplab V3, Unet++, Segmentacion de herramientas.

Received: January _th 2022
Accepted: January _th 2022

Introduction

The inclusion of automated systems in production lines has captured the interest of industries, showing a wide variety of applications around costs and risk reduction. Some tasks related to greater physical demands, such as overexertion, repetitive motion, and awkward body posture, represent a challenge for the manufacturing industry. That generates a wide variety of high-risk incidents for humans and where the implementation of robotics has proven to be a great support tool (Li, Gul, & Al-Hussein, 2019), (Colim et al., 2020).

From the point of view of electronic control, a robotic system requires prior knowledge regarding kinematics and dynamics modeling, and specialization regarding the work that the device will perform (Kurdila & Ben-Tzvi, 2019). For this reason, in past times, the implementation of these devices required an extensive process of specialized study by operators. Such perspective limited the devices to execute a series of programmed sequence movements in a certain area of interest, due to the kinematic redundancy that robots have when a manupulator has more degrees of freedom than necessary to execute a given task, for example, a manipulator with six degrees of freedom becomes redundant with respect to five-dimensional end-effector tasks, such as arc welding, laser cutting, spray painting, because it has a sixth roll angle, generating infinite solutions for a single point in these tasks (Xie, Jin, Luo, Li, & Xiao, 2021).

The implementation of robotic systems can involve different levels of study around the modeling of systems and the use of drivers. Consequently, the operation of this

[1]Mechatronic Engineer, Universidad Piloto de Colombia, Colombia. E-mail: pedro-montano@upc.edu.co
[2]Electrical Engineer, Universidad Antonio Nariño, Colombia. M.Sc. Electrical Engineering, Pontifícia Universidade Católica do Rio de Janeiro, Brasil. E-mail: williamramirez@uan.edu.co
[3]Mechatronic Engineer, Universidad Piloto de Colombia, Colombia. E-mail: juan-sosa@upc.edu.co
[4]Electronic Engineer, Universidad Santo Tomas, Colombia, M.Sc. Industrial automation, Universidad Nacional de Colombia, Colombia. Affiliation: Associate Professsor, Universidad Piloto de Colombia, Colombia. E-mail: marco-jinete@upc.edu.co

How to cite: Montaño, P., Ramirez , W., Sosa, J., Jinete , M. (2022). Automatic segmentation of industrial tools using Deep Learning. Ingeniería e Investigación, xx(xx), xx-xx. DOI: 10.15446/ing.investig.xxxx

type of system may involve constantly having input from various sensors of multiple objects in an area, knowing the kinematics to plan a trajectory (Lee, Li, Shen, & Chuang, 2018; Yenorkar & Chaskar, 2018), having spatial information for the use of coordinates and definition of speed in a movement (using a driver, (Soehartanto, Imran, & Purwitosari, 2017)).

Around this perspective, the use of artificial intelligence has made a difference factor in the automation of tasks. Considering possible limitations oriented to the definition of trajectories, artificial intelligence has sought to provide decision support around the study of the interaction of the machine in an environment (Wagner et al., 2022). Consequently, the use of robotics in the industry has become intelligent and self-sufficiency, being able to recognize changes in production lines. For this reason, the use of simulated environments has become more and more necessary, being crucial stages for the implementation and innovation of new approaches. As presented in Lee et al. (2018); Yenorkar and Chaskar (2018), multiple tools allow studying the behavior of a system around the study of kinematics, perturbations for the construction of a stable control system.

Once a control system is defined, trajectory planning will turn to decision-making based on spatial recognition. The input to the control system will represent coordinates, force and speed of the multiple actuators modeled to execute some actions (Reiter, Muller, & Gattringer, 2018). Considering this from this stage, the focus turns to a study of computational vision, where the designer will search from coordinates and detect objects to execute some action. From this perspective, the use of Deep Learning for object detection, classification and segmentation has become a highly desirable approach in decision making (X. Chen & Guhl, 2018; Mallick, del Pobil, & Cervera, 2018).

The use of Deep Learning has been inspired by human vision, seeking the extraction of information for decision making. Based on the use of a large number of data and convolutional neural networks considering multiple attribute extraction techniques, Deep Learning seeks to propose learning based on the extraction of relevant information (Khan, Laghari, & Awan, 2021). The extraction of these scenes may involve a deep analysis in the reconstruction of information from optical systems, commonly in the industry the use of cameras (Tang et al., 2020). Such information can provide a trend for decision making using Deep Learning, considering a 3-dimensional environment.

The use of optical data can involve a great variety in the study of attributes, depending on the depth and sensors used in data capture. A dataset in an integrated optical system, such as kinect, can return information relevant to shape, depth and orientation (Ma, Liu, & Cai, 2020). Consequently, around defining and prioritizing relevance in the attributes present in a data set, it is common to apply techniques related to data processing. As shown in Amado, Gomes, Amaro, Wolf, and Osório (2019), approaches based on detection and segmentation of objects can involve a deep analysis of attributes for the implementation of architectures based on Deep Learning. Consequently, Amado et al. (2019) shows how having a greater amount of data, depth and resolution can help to obtain a model with better performance, however, as it was evidenced by concatenating so much data, the computational cost becomes quite high.

Yeh et al. (2020) shows how the use of a robotic arm through a driver can be controlled through the support of artificial intelligence. Through a simulated environment having depth attributes, the robotic arm managed to detect a series of objects with different shapes using an R-CNN architecture. As a contribution, the author proposes the use of tools for the construction of datasets, for the learning process using pre-entered weights of a great variety of scenarios.

Other types of applications have demonstrated the use of object detection for learning movement and coordinates. Recognizing multiple objects such as tools, obstacles, and areas can lead to the use of algorithms for automation in the abstraction of trajectories. Amado et al. (2019) presents a simulated approach for the implementation of a control system based on Q learning, oriented to the detection of objects for the implementation of reinforcement learning. In the first stage he emphasizes the importance of image processing for the detection of multiple objects in a scene, looking for a starting point for the extraction of trends. In this way, Amado et al. (2019) presents how the use of simulated environments around a computer vision problem can provide enough information for the implementation of systems related to learning in decision-making.

Based on current research interests, this paper will emphasize the study of simulated scenarios related to an automated production plant. Considering this as an approach based on image segmentation will be proposed, delimiting for each object its position and area or bounding box, emphasizing the use of a simulated environment in ROBODK. The main contributions of this work will be represented in:

- Present a novel dataset oriented to segmentation tasks, representing common classes present in a production line with the presence of a robotic arm.
- Contextualizing how the use of artificial intelligence can be highly desirable and efficient in industries that require a large number of production tasks to be performed.
- Present through a simulated environment the feasibility of using convolutional networks in computational vision problems oriented to decision making in robotic devices.

The remaining parts of this work are presented as follows: Methods introduce the use of Deep Learning in computational vision problems in the field of robotics exposing the models and metrics used. Next is the study area of interest, and also the methodology employed for our experiments. After that, the experimental setup, obtained results, and finally summarizes the conclusions.

Methods

In numerous applications, Deep Learning has shown high performance supporting decision-making tasks of high complexity or repetitiveness for humans. Some of these tasks are the recognition and automation of tasks related to the use of tools, location status and product packaging in production stations. Around this, the study of new and efficient techniques based on deep learning has been crucial, being widely of interest in the present decade.

Unet ++

Inspired by the Unet proposed by the University of Freiburg in Germany, Unet ++ architecture (Huang et al., 2020) is a fully convolutional network where 2 phases are considered, first of reduction (encoder) and second of expansion (decoder). In this version three new modules are added, redesigned skip pathways, dense skip connections and deep supervision, as shown in Figure 1.

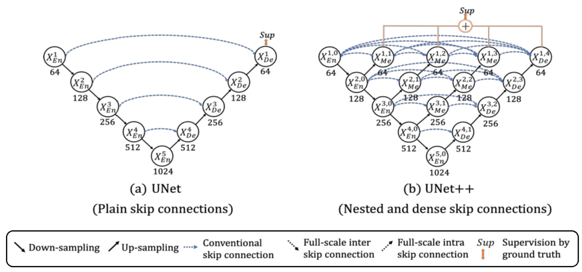

Figure 1. UNet (a) vs UNet++(b).
Source: (Huang et al., 2020)

In this way, the first module will reduce the semantic gap of the feature maps, solving some aspects emphasized in optimization. The second module looks for the accumulation of previous feature maps, using a dense convolution block along each skip pathway. Finally, the third module will relate the complexity of the model, to adjust its performance. From this perspective, the complexity of the model will vary around the versions L1 with 0.1 million (M) of parameters, L2 with 0.5M, L3 with 2.2M and L4 with 9.0M.

Figure 1 presents the variation between Unet and Unet++, mainly highlighting the implementation of Re-designed Skip Pathways (represented by the blue dotted lines) and deep supervision (represented in the upper red line). The deep supervision will be able to several the mode of operation, the first precise mode averaging all the results in the segmentation branches, or a fast mode where only one segmentation map of a segmentation branch will be selected.

Deeplab v3 + and the use of Backbones

Presented by Google, Deeplab V3 plus (L.-C. Chen, Zhu, Papandreou, Schroff, & Adam, 2018) proposes use of atrous convolution and Depthwise separable convolution to consider a larger receptive field without entailing a significant increase in parameters. Figure 2 shows the base architecture of Deeplab V3+, showing a multiscale hide section where the atrous convolution is implemented in the encoder stage and the decoder stage where the result of the segmentation is refined.

In its operation, the use of Backbones (attribute extractors based on other architectures) is proposed to improve performance in terms of speeds and/or accuracy. In this direction, the use of backbones inspired by ResNet, Xception and MobileNet can lead to improvements in metrics or

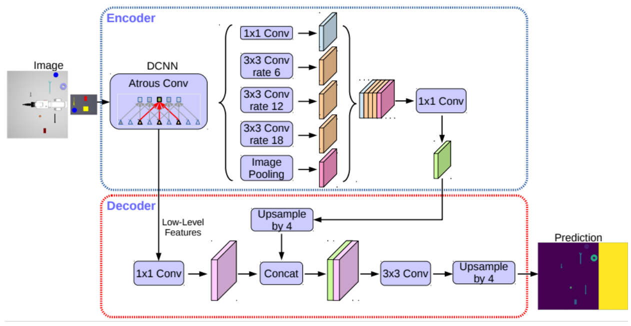

Figure 2. DeepLabv3+ structure.
Source: (L.-C. Chen et al., 2018)

performance of Deeplab V3 plus, especially in the use of the Xception backbone.

Quantitative Metrics

In the evaluation stage of the studied methods, the use of recall, precision and F1-score metrics was considered. On this way, the Precision metric presents the ratio of true positives to all positives, the objective is to see how correct the inference of a sample. Equation 1 shows how to calculate *Precision*.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

where TP(*True positive*) are the samples of the class of interest correctly classified, and FP(*False positive*) samples that do not belong to the class and are classified incorrectly.

Recall evaluates the number of true positives correctly classified. The Equation 2 shows how Recall is calculated.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

where FN (*False negative*) are samples of the class misclassified.

The F1-score metric study the harmonic mean between the metrics *Precision* and *Recall*, as show in Equation 3.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

Study area

Datasets

This dataset proposes the study of a simulated environment of a production line, where a robotic arm will have the function of performing a pick and place function of a series of products and tools. For this, the ROBODK tool was used, proposing an object detection approach for decision-making.

A total of 10 classes were considered having a total of 6 classes (4 to 9) that describe the presence of tools, 4 (0 to 3) of products present in a line, and Background (10), as shown in Table 1.

The objective of this dataset is to represent the presence of multiple tools present in a production line, various items

Table 1. Ten common classes present in a production line.

Class	Id
Recipient 1 (Blue Circle)	0
Recipient 2 (Yellow square)	1
Recipient 3 (Red rectangle)	2
Recipient 4 (Brown rectangle)	3
Wrench	4
Screw	5
Barrette	6
Cradle ball bearing	7
Screwdriver	8
Nut	9
Background	10

Source: Authors

represented in more common morphologies and obstacles present during the displacement of a robotic arm (Montaño, Ramirez, & Esquivel, 2021b).

Methodology

As shown in Figure 3, the present work proposes an approach emphasized in the study of multiple representative forms of tools and obstacles present in a simulated environment in ROBODK, establishing an initial stage for the automation of trajectories.

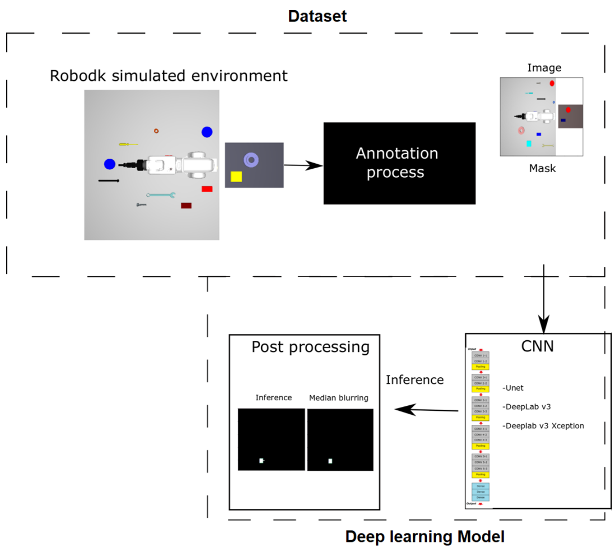

Figure 3. Proposed methodology in the study of a simulated environment for an industrial production line.
Source: Authors

The stages addressed involved the construction of a data set based on the use of the ROBODK software, generating a total of 2000 image/mask samples. Subsequently, a training stage based on the use of 3 possible architectures was proposed. Finally, in the representative output of the inference, a post-processing stage was proposed for the refinement of the inference using the median blurring technique.

Experimental setup

Considering the scenario of interest, 3 architectures were studied in this approach, proposing the experimental setup outlined in Table 2. Among the architectures of interest, the use of Unet++ (Huang et al., 2020), Deeplab V3 (L.-C. Chen, Papandreou, Schroff, & Adam, 2017) and Deeplab v3 + Xception (L.-C. Chen et al., 2018) was considered. Taking into consideration the scenario of interest, the selection of the approaches was based on the detection of edges, deep extraction of attributes in objects with various scales and optimization regarding computational cost.

As shown in Table 2, for the purpose of training, a data augmentation based on the transformations zoom in, flips and rotations was applied, this considering that the dataset generated in the present work has only 2000 images with the presence of the classes of interest. Additionally, an early stop was considered based on the mean iou metric, seeking to avoid overfitting scenarios and a min max normalization considering the nature of the image (8-bit RGB).

Table 2. Training parameters.

Parameter	Value
Normalization	Min max
Optimizer	Adam (momentum = 0.9)
learning rate	0.001
Epochs (max)	200
Distribution	Train = 70%, Validation =20%, Test =10%
Early stopping	Mean iou tolerance 6 epochs.
Data augmentation	Vertical flip, horizontal flip, Zoom

Source: Authors

This study was done with an Intel xeon processor, 16GB of RAM, and a K80 GPU.

Results

The data set presented, includes the presence of 10 classes, considering a multiclass scenario. Based on this, a total of 200 epochs and 20 iterations were trained for the Unet++ and Deeplab V3+ models using the Xception backbones, as shown in Figure 4. For the selection of the architectures studied, their good performance was considered in similar scenarios, where the detection of edges and objects at different scales was considered.

Figure 4 presents the learning process of the studied architectures, emphasizing the analysis to the study of the mean of intersection Over Union metric. In Figure 4 Unet++ architecture was represented through the green color, DeepLab V3+ using the Purple color and Deeplab V3+ using the Xception backbone through the blue color, showing the variability index in the 20 experiments executed by architecture. From the training graph, it is possible to show that the Deeplab v3+ Xception architecture had a faster convergence with respect to the study metric. Regarding mean values, Unet++ obtained higher scores, having a worse variability around 20 training sessions (+/- 0.05).

In the inference process, each model obtained was evaluated considering the distribution established for the test set. Figure 5 presents the analysis of the results obtained in the inference through the analysis of the F1-score metric. Also, it indicates a box plot of the results obtained for the test set, showing the median of the values obtained in the inference of each of the 20 models obtained by architecture. The lower and upper limits of the box plot will represent trends regarding the variability in the metric studied in

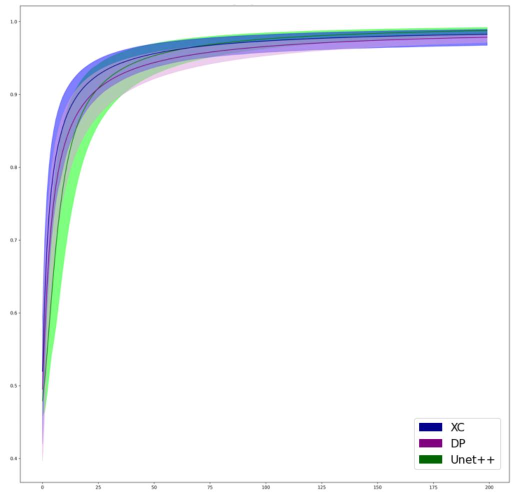

Figure 4. Evolution of the *F1-score* during training of each model in the robotic environment dataset. Each curve represents the average of the *F1-score* along the 20 experiments.
Source: Authors

the inference, and the points present will show anomalous results, where the inference had a tendency that was too high/low.

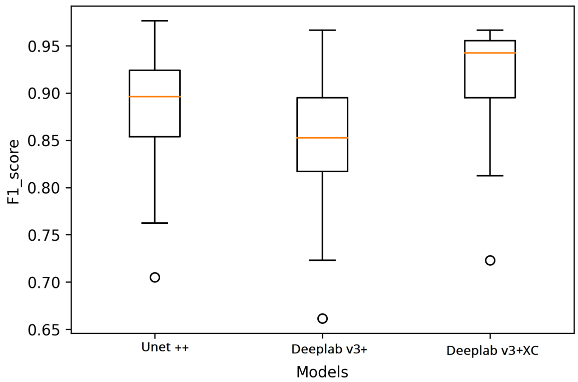

Figure 5. Box plot graph, evaluating the inference of each model in the robotic environment dataset. Each box represents the variability among data distribution along the 20 experiments.
Source: Authors

As presented in Figure 5, DeepLab v3 + backbone Xception (DeepXC) had the best performance, obtaining a value for the median closer to the upper limit of the box plot. In comparison, DeepXC presented a lower variability in the test metrics, showing great reliability with respect to the expected inference in the segmentation of the 10 classes studied.

The Deeplab architecture with the absence of a backbone, presented the worst behavior compared to the other architectures studied. The performance of this architecture can be observed through the box plot (Figure 5), having a higher distance between the upper and lower limits, representing a high variability with respect to the trained models.

Table 3 presents the results expressed in Recall for the analysis of the inference maps, presenting the mean of 20 interactions per architecture. The analysis of the Recall metric expresses the number of true positives labeled as positive, showing the performance of the models obtained in the specific segmentation of the class of interest.

Table 3. Mean and Standard Deviation of *Recall* as a percentage for each model obtained in the test set after running each experiment 20 times with the ROBODK Tools dataset.

Class	Unet++	Deeplab V3+	Deeplab v3 Backbone
Recipient 1	0.987	0.973	0.982
Recipient 2	0.962	0.936	0.979
Recipient 3	0.692	0.626	0.912
Recipient 4	0.986	0.951	0.971
Wrench	0.966	0.957	0.960
Screw	0.921	0.938	0.951
Barrette	0.929	0.934	0.967
Cradle ball bearing	0.844	0.809	0.949
Screwdriver	0.674	0.668	0.938
Nut	0.759	0.793	0.921

Source: Authors

As presented in Table 3, the Deeplab architecture in this scenario presented a high difficulty in the segmentation of the rectangular obstacle, due to its great similarity with other obstacles regarding shape and tonality. About the screwdriver and nut classes, it can be seen that both the Unet++ and Deeplab v3+ models presented a high difficulty in the segmentation of areas with fewer pixels.

Considering this, it should be noted that in initial experiences, due to the great similarity between obstacles and tools with respect to color and dimensions, it presents a challenge, having a clear dependence on the analysis of contours and differentiation by texture. The presence of objects with similarities in a real environment is highly common, especially with regard to tools and products. The corresponding detection can involve different actions, such as differentiating a container labeled for polymers and another for metals, for their corresponding storage, and even the presence of the same tool with different dimensionality. Considering this, the implementation of the backbone was emphasized to obtain a model with greater precision even to small changes in the objects.

As evidenced in Table 3, using a backbone emphasized the deep extraction of attributes and generated a scenario with better learning of representations. The presence of true positives was greater in the architecture with the Xception backbone, having a clear differentiation between objects with extremely close attributes.

As a result, it could be observed that the segmentation oriented to the detection of objects can be significant in the planning of a trajectory of a robotic arm in a production plant. As was expressed, optimizing the trajectory automatically can represent a great improvement in logistics and human cost in the adaptability of multiple processes. About the low availability of samples and effects related to the increase of data, the use of a deeper network for segmentation represented a better performance around the learning of different abstractions, having a better approach to the variability in the representations.

Post processing

The output of the obtained models will show the probability that a pixel belongs or not to a class. For this scenario, a small presence of false positives was observed in classes

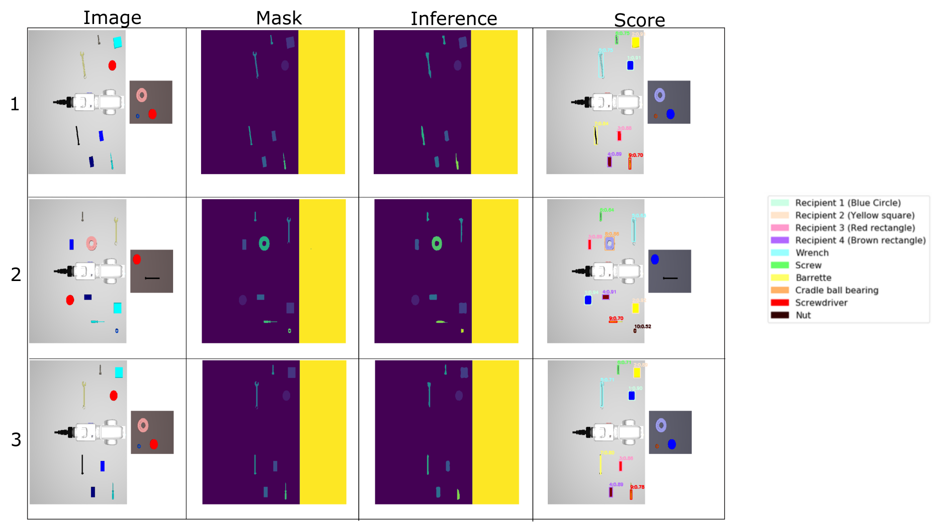

Figure 6. Inference map for three test images using Deeplab V3+ Xception. This image contains input image , masks, inference and box segmentation area information for the three scenarios shown.

with some type of similarity, showing this effect mainly in Unet++ and Deeplab V3+. Around this and the main focus proposed (detection of objects for decision making), a refinement of the inference map around a statistical method (median blurring) was proposed.

In this way, the output of the trained model, will show a map expressed in segmentation, where each pixel will represent a class. For analysis and trajectory definition purposes, a box detection based on matching by the shape of each class in the segmentation was built (see Figure 7), evaluating its correspondence with the defined mask. Figure 6 relates the executed process, showing image, mask, inference and score in the correspondence of the segmentation for each element.

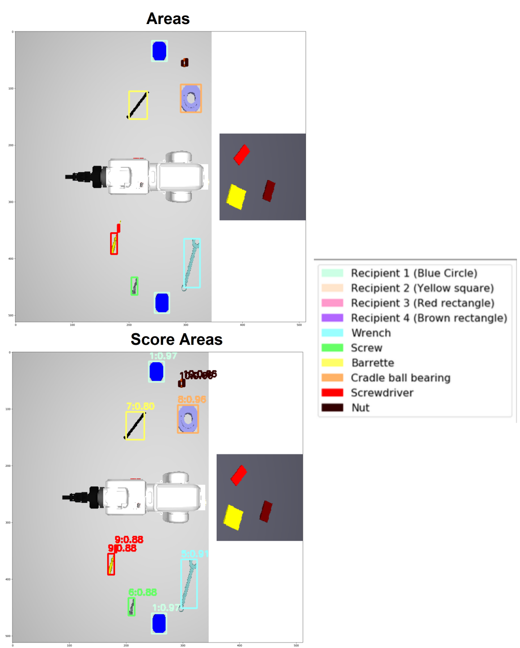

Figure 7. Expected output after processing using test data set, Deeplab V3+ Xception architecture.
Source: Authors

As shown in Figure 6 and Figure 7, the output of the model will be represented in the detection of objects based on segmented areas, to have the representative areas and pixels available for each class. With the aforementioned information, it will be possible to perform recognition of a work area, facilitating the use of some method in the automation of trajectories, for example, genetic algorithms from the coordinates obtained from centroids.

Conclusions

From the study of the simulated environment of elements in the industry, a dataset with 10 common classes of objects of interest for mapping trajectories was presented. In this process, an approach based on the detection and segmentation of objects was emphasized, seeking the exploration and automatic identification of tools and obstacles.

Through the experimental phase, it was observed how the use of backbones in the extraction of attributes can provide a model with better performance. Considering the low representativeness of some classes together with similar tendencies between classes, emphasizing a more robust architecture provided sufficient reliability for tasks related to detection.

Based on the results obtained, and the dataset provided, the present work proposed as future work the application of some techniques related to trajectory planning. As an approach, the use of techniques related to reinforcement learning is proposed, taking advantage of the availability of a feasible reference for the calculation of coordinates, shape, and area. This work is fully reproducible, as the source code is publicly available in our repository (Montaño, Ramirez, & Esquivel, 2021a).

References

Amado, J. A. D., Gomes, I. P., Amaro, J., Wolf, D. F., & Osório, F. S. (2019). End-to-end deep learning applied in autonomous navigation using multi-cameras system with rgb and depth images. *2019 IEEE*

Intelligent Vehicles Symposium (*IV*), 1626–1631. https://doi.org/10.1109/IVS.2019.8813897

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*. https://doi.org/10.48550/ARXIV.1706.05587

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In (pp. 801–818). https://doi.org/10.48550/ARXIV.1802.02611

Chen, X., & Guhl, J. (2018). Industrial robot control with object recognition based on deep learning. *Procedia CIRP*, *76*, 149-154. https://doi.org/10.1016/j.procir.2018.01.021

Colim, A., Sousa, N., Carneiro, P., Costa, N., Arezes, P., & Cardoso, A. (2020). Ergonomic intervention on a packing workstation with robotic aid - case study at a furniture manufacturing industry. *Work*, *66*(1), 229–237. https://doi.org/10.3233/WOR-203144

Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., . . . Wu, J. (2020). Unet 3+: A full-scale connected unet for medical image segmentation. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), 1055-1059. https://doi.org/10.1109/ICASSP40776.2020.9053405

Khan, A. A., Laghari, A. A., & Awan, S. A. (2021, 4). Machine learning in computer vision: A review. *EAI Endorsed Transactions on Scalable Information Systems*, *8*(32), e4. https://doi.org/10.4108/eai.21-4-2021.169418

Kurdila, A. J., & Ben-Tzvi, P. (2019). *Dynamics and control of robotic systems*. John Wiley & Sons
.

Lee, J.-D., Li, W.-C., Shen, J.-H., & Chuang, C.-W. (2018). Multi-robotic arms automated production line. *2018 4th International Conference on Control, Automation and Robotics* (*ICCAR*), 26-30. https://doi.org/10.1109/ICCAR.2018.8384639

Li, X., Gul, M., & Al-Hussein, M. (2019). An improved physical demand analysis framework based on ergonomic risk assessment tools for the manufacturing industry. *International Journal of Industrial Ergonomics*, *70*, 58-69. https://doi.org/10.1016/j.ergon.2019.01.004

Ma, Y., Liu, D., & Cai, L. (2020). Deep learning-based upper limb functional assessment using a single kinect v2 sensor. *Sensors*, *20*(7), 1903. https://doi.org/10.3390/s20071903

Mallick, A., del Pobil, A. P., & Cervera, E. (2018). Deep learning based object recognition for robot picking task. In (pp. 1–9). https://doi.org/10.1145/3164541.3164628

Montaño, P. A., Ramirez, W. A., & Esquivel, J. P. S. (2021a). *Robodk-cnn-for-segmentation*. Github. Retrieved from https://github.com/williamalbert94/Robodk-CNN-for-segmentation

Montaño, P. A., Ramirez, W. A., & Esquivel, J. P. S. (2021b). *Tools in industry*(*robodk*). Zenodo. https://doi.org/10.5281/zenodo.5768611

Reiter, A., Muller, A., & Gattringer, H. (2018). On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators. *IEEE Transactions on Industrial Informatics*, *14*(4), 1681–1690. https://doi.org/10.1109/TII.2018.2792002

Soehartanto, T., Imran, I. F., & Purwitosari, L. A. (2017). Control design for direct-drive robotic ARM using sliding mode control. *2017 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation* (*ICAMIMIA*), 277-282. https://doi.org/10.1109/ICAMIMIA.2017.8387601

Tang, Y., Chen, M., Wang, C., Luo, L., Li, J., Lian, G., & Zou, X. (2020). Recognition and localization methods for vision-based fruit picking robots: A review. *Frontiers in Plant Science*, *11*, 510. https://doi.org/10.3389/fpls.2020.00510

Wagner, M., Bodenstedt, S., Daum, M., Schulze, A., Younis, R., Brandenburg, J., . . . Speidel, S. (2022). The importance of machine learning in autonomous actions for surgical decision making. *Artificial Intelligence Surgery*, *2*(2), 64–79. https://doi.org/10.20517/ais.2022.02

Xie, Z., Jin, L., Luo, X., Li, S., & Xiao, X. (2021). A data-driven cyclic-motion generation scheme for kinematic control of redundant manipulators. *IEEE Transactions on Control Systems Technology*, *29*(1), 53-63. https://doi.org/10.1109/TCST.2019.2963017

Yeh, C.-Y., Shi, Z.-H., Chuang, J.-T., Hsu, K.-H., Liu, S.-W., Wu, R.-W., . . . Lee, J.-D. (2020). 3D cameras and algorithms for multi-angle gripping and control of robotic arm. *2020 IEEE Eurasia Conference on IOT, Communication and Engineering* (*ECICE*), 363-365. https://doi.org/10.1109/ECICE50847.2020.9301931

Yenorkar, R., & Chaskar, U. M. (2018). Gui based pick and place robotic arm for multipurpose industrial applications. *2018 Second International Conference on Intelligent Computing and Control Systems* (*ICICCS*), 200-203. https://doi.org/10.1109/ICCONS.2018.8663079