EasyChair Preprint

№ 12250

Study on Electric Vehicle Charging Socket Detection Using YOLOv8s Model

Vladimir Tadic, Akos Odry, Zoltan Vizvari, Zoltan Kiraly, Imre Felde and Peter Odry

February 24, 2024

Study on Electric Vehicle Charging Socket Detection using YOLOv8s model

Vladimir Tadic[1,2,3], Akos Odry[4], Zoltan Vizvari[3,5], Zoltan Kiraly[1,2], Imre Felde[1], Peter Odry[2,3]

[1]University of Obuda, John von Neumann Faculty of Informatics, Becsi ut 96/B., 1034. Budapest, Hungary; tadity.laszlo@uni-obuda.hu; laslo.tadic@gmail.com; kiru.zoltan@gmail.com; felde.imre@uni-obuda.hu

[2]University of Dunaujvaros, Tancsics Mihaly u. 1/A Pf.: 152, 2401. Dunaujvaros, Hungary; tadityv@uniduna.hu; kiru@uniduna.hu; podry@uniduna.hu

[3]Symbolic Methods in Material Analysis and Tomography Research Group, Faculty of Engineering and Information Technology, University of Pecs, Boszorkany str. 6, H-7624 Pecs, Hungary; vizvari.zoltan@mik.pte.hu

[4]Faculty of Engineering, University of Szeged, Mars ter 7., 6724. Szeged, Hungary; odrya@mk.u-szeged.hu

[5]Department of Environmental Engineering, Faculty of Engineering and Information Technology, University of Pecs, Boszorkany Str. 2, H-7624 Pecs, Hungary; vizvari.zoltan@mik.pte.hu

Abstract: This paper introduces the utilization of the latest small You Only Look Once version 8 – YOLOv8s convolutional neural network in an automatic electric vehicle charging application study. The employment of a deep learning based object detector is a novel and significant aspect in robotic applications, since it is both, the initial and the fundamental step in a series of robotic operations, where the intent is to detect and locate the charging socket on the vehicle's body surface. The aim was to use a renowned and reliable object detector to ensure the reliable and smooth functioning of the deployed robotic vision system in an industrial environment. The experiments demonstrated, that the deployed YOLOv8s model detects the charging socket successfully under various image capturing conditions, with a detection rate of 97.23%.

Keywords: YOLOv8s; Electric vehicle charging socket; Image processing; Object detection; Robotic applications; Automotive applications.

1 Introduction

The global surge in electric vehicle adoption is an ongoing trend, with projections indicating a substantial increase in their presence on roads worldwide. One of the primary limitations to wider electric vehicle (e-vehicle) adoption lies in their batteries. Despite ongoing efforts, there has yet to be a significant breakthrough in battery technology, resulting in limitations in both capacity and lifespan. Consequently, this leads to a restricted driving range for electric vehicles. To address these challenges, there is active research into the development of high-speed chargers and novel charging techniques for electric vehicles. Present-day automotive consumers demand products that cater to their mobile information and entertainment needs. These offerings should be seamlessly integrated into innovative automotive applications, including those related to autonomous electric car charging and automated vehicle washing, etc. [1-4].

The accelerated advancement of electric vehicles will lead to a growing demand for associated applications in the coming years. [4-6]. Certainly, one of the fundamental and decisive task is the recharging of electric vehicle batteries. This underscores the significance of applications designed to facilitate this process. Given that the charging of electric vehicles is a time-consuming endeavor, there exists a clear demand from both users and operators for automation to align with customer needs. Consequently, once a vehicle is parked at a charging station, the user's involvement in the charging process is minimal, limited to simply opening the charging port door. The entirety of the procedure is orchestrated by robots, affording the user the freedom to engage in other activities while the vehicle charges. [7-8]. Thus, the concept of seamless electric vehicle charging, devoid of human involvement, holds great appeal for customers, prompting numerous companies to delve into related research. Moreover, as advancements in autonomous driving and driverless parking technologies continue to unfold, automated charging will assume even greater significance. In these scenarios, a robot will assume full responsibility for the charging process once the vehicle autonomously parks, eliminating the need for human intervention in the process. Embracing automation technology signifies a departure from the manual charging practices of today. Naturally, specific challenges inherent to this application must be thoroughly examined and addressed. These include the need for precise parking, maneuvering of robots around parked vehicles, ensuring adequate lighting for charging socket detection cameras, as well as the ability to promptly halt and disconnect the charging process, etc.

This paper presents an initial study within an automotive industrial project whose aim is to develop a robotic-based application for automatic charging of electric vehicles utilizing image processing and object detection techniques. As a result, a novel approach for electric vehicle charging socket detection using the novel state-of-the-art You Only Look Once version 8 (YOLOv8) [9] object detection framework will be introduced.

The main task of this study is the deployment of a lightweight and efficient object detector for the detection of the Combined Charging System 2 (CCS2) socket of electric vehicles using a renowned and reliable object detector model. Thus, the socket detection procedure in complex image scenes is based on the latest YOLOv8 framework introduced in 2023 by Ultralytics, which is the sequel of the earlier versions of well known YOLO object detectors. The main demand of this task is to initially detect the charging socket in compound scenes in order, that in the later stage of the operation, the robotic arm with a specialized short range 3D camera will approach the detected socket in order to accurately determine its position in space. Further, one of the main project requirements was to use a well known, reliable and fast object detector framework intending to ensure the smooth operation of the whole system. It should be noted, that in the subsequent stages, the Universal Robot 10e (UR10e) equipped with a built-in force-torque sensor will be employed as the robotic arm for the implementation of an autonomous charging application [1-3]. A detailed description of the robot and its work is not within the scope of this paper, and it will be fully depicted in a future research paper.

Finally, using the strict instructions about the simplicity and reliability by the project client, a novel YOLOv8-based procedure was developed for the automatic CCS2 socket detection. This initial study entirely fulfilled the goal set by the project task, and in the future the testing will be performed with a special camera mounted on the robot's arm.

The contribution of this study in terms of an industrial research project is a development of novel, trained and reliable object detector for the detection of a CCS2 charging socket for the automated electric vehicle charging application. The use of the new YOLOv8 for charging socket detection purposes is not yet published in the scientific literature, as well as in papers related to any industrial research with the aim of the charging socket detection. Thus, this approach, offers a solution to the challenge of electric vehicle charging socket detection for upcoming industrial applications with a novel state-of-the art object detector.

The paper can be summarized as follows. The first section is the introduction, the second section is the literature overview, the third section introduces the YOLOv8 framework and the proposed method. Section four shows the experiments and results. Finally, the conclusions are drawn with the future works plan.

2 Related Works

The identification and isolation of the object's position is a common challenge in robotic vision systems [15-26]. This task becomes particularly critical in specific applications where the determination and extraction of certain shapes from the background are imperative. Multiple techniques exist to carry out this segmentation process. It's worth noting, that there is a limited body of research and solutions

addressing the detection of electric vehicle charging sockets in the existing literature. Further, the YOLOv8 framework and its predecessors are very popular for the development of highly accurate, precise and flexible object detectors with acceptably fast execution [9]. Thus, due to all the mentioned features, YOLOv8 framework is suitable for a wide range of applications, which include the detection of various shapes and objects. This section will provide a concise overview of the relevant studies in these domains.

Pan et al. [27] introduced a charging socket detection algorithm with three main steps: recognition, localization and inlay. For the charging socket localization, a convolutional neural network-based (CNN) approach is used. During the socket localization procedure an adapted pose solving method was utilized based on circle features. In the insertion step an AUBO-i3 robot was exploited. Authors stated an accuracy rate of 98.9%. Zhang and Jin [28] developed a new method built on computer vision for electric vehicle charging socket detection and localization with an aim to solve the space and low efficacy limitations in charging operation for e-vehicles. A specific image segmentation process is used based on Hue Saturation Intensity (HSI) color model to educe the features of the charging socket targeted the subpixel precision. Further, the image segmentation method involves thresholding in the Hue component of the input image, edge detection using Canny edge operator and morphological operations. The HALCON vision platform is utilized for the development. Authors reported, that the proposed method can successfully detect and locate the charging socket position with the accuracy rate of 100%. Mišeikis et al. [29] proposed an automatic robot-based vehicle charging application using 3D computer vision operations. The system is established on a 3D computer vision system, an UR10 cobot and a charging station. A shape-based matching procedure is utilized for identification and exact pose determination of the socket. An analogous approach is used for camera-cobot system calibration. Finally, a three-step cobot motion planning procedure is applied for charger plugin. According to the experiments, the introduced method operates in laboratory conditions under indoor lighting with a self-made charging socket holder. Quan et al. [30] suggested an automatic model for the recognition and positioning of charging sockets of e-vehicles. The system is divided into two parts: the coarse and the precise positioning. The coarse positioning relies on the Hough circle and the Hough line transformations, and it locates the position data of the charging socket itself. The precise positioning step utilizes the Canny edge operator to ascertain the contour data of the input and edge-detected images respectively. Finally, the Perspective-n-Point (PNP) algorithm is executed to determine the pose data of the charging socket. The AUBO-i10 6-DOF (degree of freedom) robot is used to test the recognition and inlay accuracies in various conditions and environments. Authors claimed, that the detection rate of the coarse positioning is 97.9%, while the average success rate is 94.8%. Quan et al. [31] proposed a group of efficient and exact procedures for determining the position of an e-vehicle charging connector. The process is divided in two steps: the search step and the aiming step. In the search step, the feature circle procedure is utilized to fit the ellipse information to obtain the pixel coordinates of

the feature point. In the aiming step the contour matching and logarithmic assessment indicators are utilized in the cluster template matching algorithm proposed in their study to obtain the matching position of the socket itself. In the end, the Efficient PNP algorithm is used to determine the pose information of the charging socket. The reported accomplished plug-in rate is 95%. Lou and Di [32] proposed a 4-DOF cable-guided automatic-charging robot built from a 3-DOF cable-guided serial manipulator with a moving platform. The end-effector of their robot is an flexible plug that has the ability to resist negligible flexible deformation. Authors showed in their testing the achievability and the efficacy of utilizing the cable-guided automatic-charging robot to realize automated e-vehicle charging application. Lin et al. [33] presented a model-independent collision detection and classification procedure for cable-guided serial manipulators. First, relied on the dynamic features of the manipulator, data sets of terminal collision were implemented. Next, the collected data sets were applied to build and train a collision localization and classification system, which involves a double layer CNN and a Support Vector Machine (SVM). Authors reported, that the developed system can extract features without human intervention and can handle with collision when the contact surface is irregular. The experiments and results demonstrated the validity of their model with promising prediction accuracy. Li et al. [34] introduced a high-precision method to detect and localize the charging ports based on Scale-Invariant Feature Transform (SIFT) and Semi-Global Block Matching (SGBM) algorithms. The feature detection procedure based on SIFT was adjusted to produce the Difference of Gaussian (DOG) for scale space construction, and the feature matching algorithm with nearest-neighbor search was used to get the set of matching points. The disparity calculation has been done with a semi-global matching (SGM) procedure to get high-precision positioning data for the charging socket pose. The viability of the procedure was verified using OpenCV and MATLAB platforms. Chablat et al. [35] presented a robotic system with parallel structure for automatic e-vehicle charging where the charging socket of the vehicle is at its front side. They used a QR code stuck next to the plug in order to localize the socket on the vehicle. When the robot moves, the QR code detected by the vision sensor is employed to tune the trajectory before starting the inlay of the plug. A prototype of the robot was successfully accomplished. Authors noted, that the research on the robotic charging system will be continued. Tadic [36] utilized the ZED 2i depth sensor for the detection and extraction of the charging socket on the electric vehicle's body surface. The socket detection and extraction were performed using common image processing and morphological. The test showed, that the developed method extracts the charging socket and determines the tilt angles and socket coordinates successfully under various depth measurement conditions, with the success rate of 94%. Hussain [37] provided a detail review of the YOLO evolution from the original YOLO framework to the recent release YOLOv8 from the perspective of industry. His review analyzes the main architectural advantages of the YOLO object detectors for industrial applications. Slimani et al. [38] employed YOLOv8 to improve the efficiency and precision of rust disease

classification in a fava bean field images. They demonstrated, that the proposed model developed with transfer learning has a higher recognition rate than other models. They claimed, that the detection accuracy in their model is reaching 95.1%. Sharma et al. [39] proposed a real-time parking time violation tracking procedure with closed-circuit camera and deep learning models with a tracking algorithm to persist the information from one frame to its subsequent frame. The algorithm employs the state-of-the-art object detection model YOLOv8 to identify vehicles within a parking lot. In their future works, they will explore the synchronization of the proposed algorithm with two or more cameras. Talaat and ZainEldin [40] introduced an advanced fire detection approach for smart cities based on the YOLOv8 model, which utilizes the strengths of deep learning to detect fire specific features in real time using cameras. Authors claimed, that the proposed method achieved a state-of-the-art performance in terms of precision rate of 97.1%. Bai et al. [41] showed a comprehensive investigation and improvement of the YOLOv8n algorithm for object detection, focusing on the integration of Wasserstein Distance Loss, FasterNext, and Context Aggravation strategies. During the experiments, each approach was evaluated individually and collectively in detail to assess its contribution to the model's performance. As it was shown in their study, the proposed YOLOv8n framework achieved a good balance between accuracy and model complexity, and it outperforms other frameworks in terms of model complexity, model accuracy and model inference speed. Finally, it should be noted, that numerous deep learning models [38-44] are analyzed and tested in recent years for various applications and the development of many deep learning models is expected in the upcoming period.

3 Methods

3.1 YOLOv8 Framework

YOLOv8 is the latest version of the renowned real-time object detection and image segmentation model. Since it is introduced in 2023, a related official documentation about the framework and its architecture is not published yet. Only the Ultralytics website provides formal information related to the model.

YOLOv8 leverages the forefront of deep learning and computer vision, delivering unmatched speed and precision. Its refined architecture enables versatility across diverse applications and seamless adaptation to various hardware platforms [45]. Unlike the Region-based Convolutional Neural Network (R-CNN) and Fast R-CNN models, which use a multi-stage process to detect objects in image, all YOLO models use a single neural network (single shot detection) to predict both, the bounding boxes and class probabilities of objects in images [46]. This property makes YOLO models mostly faster than other object detection models, however

also potentially less accurate in some examples [46]. Further, all YOLO models use direct prediction to predict the class probabilities and bounding boxes of objects in images, without learning on region proposals. This feature enables each YOLO model to accomplish object detection in a single forward pass of the network, which is much faster than models that require multiple detection stages [45-46]. YOLO uses a grid-based prediction process, where it splits the original input image into a grid of cells and predicts the presence of shapes in each cell. Then, each cell is responsible for predicting a set of bounding boxes and class probabilities for the shapes within its area. This enables YOLO models to manage multiple objects of different scales in a single image, or video frame [37], [46].

YOLOv8 can be used for the three main computer vision tasks:

1. Classification; Classification is a straightforward task where the object detector model is tasked with identifying and providing a single class that is predominantly present in the input image. The result of a classification consists of a class index and a corresponding confidence score. Typically, classification is beneficial when the goal is to ascertain the presence of a specific class in the input image. It's essential to emphasize that, in this context, the location of the object is not determined, thus the object detector model can only confirm its presence.
2. Object detection; The object detection is an evolution of the classification task. Here, the goal is to not only identify various classes within an image, but also to precisely locate them. The specific locations of the detected objects are visually indicated with bounding boxes.
3. Segmentation; Segmentation represents a step beyond the object detection. While object detection involves pinpointing the location of objects and estimating their position using bounding boxes, than the segmentation takes it further by identifying individual pixels belonging to each object in the image.

Earlier YOLO models use anchor boxes [45], which are predefined bounding boxes in order to enhance the accuracy of its predictions. The anchor boxes are used to represent the prior knowledge of the network about the shapes and aspect ratios of objects. In the YOLOv8 anchor boxes are not needed, since YOLOv8 directly predicts the bounding boxes and class probabilities for each object in the input image [45]. Hence, YOLOv8 is an anchor-free model and this means, that the model predicts directly the center of an object inside the image instead of the offset from a known anchor box. The anchor free object detection reduces the number of box predictions, which speeds up the Non-Maximum Suppression (NMS), a complicated post processing step in the model that sifts through candidate detections after inference [45]. Thus, this method reduces the complexity of the model and enables for more flexibility in detecting objects of diverse sizes and aspect ratios [45-46].

The second key difference is the use of mosaic augmentation in YOLOv8 [45]. Data augmentation is the addition of new artificially derived data from existing training data. These techniques include resizing, flipping, rotating, cropping, padding, etc. It helps to address issues like overfitting and data scarcity, and it makes the model robust with better performance. One of the well-known and powerful annotation and augmentation tools is the Roboflow Framework [47] that is used in this research. Hence, in the training process of YOLOv8 models, various pre-processing tasks and augmentations are applied to training images, including the mosaic data augmentation too. This technique involves combining four distinct images, creating a mosaic, and then presenting this composite image as input to the model during training [45]. Thus, the model learns to recognize actual objects in diverse positions and under conditions of partial occlusion. Where each quadrant of the image includes an arbitrary crop from one of the four input images, then this image is used as an input for the YOLOv8 model [45,47]. Despite its advantages, the mosaic augmentation is disabled for the last 10 epochs in YOLOv8, since it can reduce performance if it is used throughout the entire training process [45].

The activation function utilized in YOLOv8 is the Sigmoid Linear Unit (*SiLU*) function [45-46]:

$$SiLU(x) = x\sigma(x) \tag{1}$$

where the $\sigma(x)$ is the Sigmoid function defined as [46]:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2}$$

The activation function plays a crucial role in determining whether a neuron should be activated. It is achieved by computing the weighted sum and incorporating bias. The primary objective of the activation function is to inject non-linearity into the neuron's output [45-46,48].

Further, the loss of the YOLOv8 model is determined with two functions, the Binary Cross Entropy (*BCE*) calculates the classification loss, while for the bounding box loss the Complete Intersection over Union (*CIoU*) and the Distribution Focal Loss (*DFL*) is calculated [46,48]. A loss function serves as a metric to assess the disparity between the predicted and target output values, quantifying how effectively the neural network represents the training data. During the training process, the goal is to minimize this loss, thereby enhancing the network's ability to accurately predict target outputs [45-46,48].

The *BCE* and *CIoU* functions are defined as:

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\big(p(y_i)\big) + (1-y_i)\log(1-p(y_i)) \tag{3}$$

$$CIoU = 1 - IoU + \left|\frac{\rho^2(b,b^{gt})}{c^2}\right| + \beta v \tag{4}$$

where b and b^{gt} are the central point of the predicted bounding box B and the central point of the ground-truth box B^{gt}, where gt denotes the ground-truth and N is the

number of object classes. The ρ parameter denotes the Euclidean distance, $y \in \{1,0\}$ specifies the ground-truth class, $p \in [0,1]$ is the probality for the class label $y = 1$ and *c* denotes the diagonal length of the smallest enclosing box that masks the two boxes [46,48]. The β parameter represents the trade-off and the *v* parameter measures the consistency of the aspect ratio. The *v* and β are represented as follows [46]:

$$v = \frac{4}{\pi}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \tag{5}$$

$$\beta = \frac{v}{1-IoU+v} \tag{6}$$

where *w* and *h* are the width and the height of the bounding box respectively. The Intersection over Union (*IoU*) is determined with the predicted bounding box *B* and the ground-truth box B^{gt} with the next equation [46]:

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}} \tag{7}$$

The *CIoU* in equation (4) consideres the overlapping area, the aspect ratio and the central point distance into account. It is an improved variant of *IoU*, and its consequence is, that it converges more faster and it is more efficient in executing the bounding box regression [45-46].

Further, the problem with a classic loss functions such as the *BCE* loss function is that suchlike functions handle the missclassifications equally. In the object detection this can be an issue, since the huge majority of the image regions do not contain any object/shape and this could lead to a class inequity problem. Thus, the focal loss function treats this issue by down-weighting the loss allotted to well-classified samples. Further, the *DFL* is built upon by including the class distribution information into the focal loss function. The aim is to learn a dynamic weighting scheme for the loss function built on the distribution of classes in the training data. This allows the YOLOv8 model to assign more weights to the low-represented classes and less weights to the over-represented classes which can lead to more precise bounding box assessment [46,48].

When the continuous distribution of the regression value is converted to the discrete domain, the assessed regression value can be written as follows [46]:

$$\hat{y} = \sum_{i=0}^{n} P(y_i)\, y_i \tag{8}$$

where the $\hat{y}$ is the estimated regression value and the *n* is the number of classes in this case.

Using the Softmax functions [48]:

$$S_i = \frac{y_{i+1}-y}{y_{i+1}-y_i} \tag{9}$$

$$S_{i+1} = \frac{y-y_i}{y_{i+1}-y_i} \tag{10}$$

$\hat{y}$ can be written as follows [46,48]:

$$\hat{y} = \sum_{i=0}^{n} P(y_i)\, y_i = S_i y_i + S_{i+1} y_{i+1} = \frac{y_{i+1}-y}{y_{i+1}-y_i} y_i + \frac{y-y_i}{y_{i+1}-y_i} y_{i+1} = y \quad (11)$$

Finally, the *DFL* can be expressed as [46,48]:

$$DFL(S_i, S_{i+1}) = ((y_{i+1} - y)(S_i) + (y - y_i)\log(S_{i+1})) \quad (12)$$

As an evaluation metric, the image recognition community decided to use the mean Average Precision (*mAP*) for object detector models [45-46]. The *mAP* is a combination of recall and precision values determined over multiple confidence thresholds, the *IoU*. The variation of the *IoU* threshold will result in different True Positives (*TP*) and False Positives (*FP*) predictions in image.

The precision is defined as the fraction of *TP* detections among all detections made at a specific *IoU* threshold [46]:

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

The recall is defined as the fraction of *TP* detections found among all possible detections made at a specific threshold [46]:

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

where *FN* are False Negatives predictions in the image.

Finally, the formula for *mAP* is defined as [46]:

$$mAP = \frac{1}{N} \sum_{n=1}^{N} AP_i \quad (15)$$

where AP_i is the average precision for the *i*-th class and *N* is the number of object classes.

The architecture of YOLOv8 is built upon the previous models of YOLO object detectors. The best description of the YOLOv8 model architecture is provided by RangeKing [49]. The architecture is presented in Figure 1. YOLOv8 uses a CNN network that can be divided into two main parts: the backbone and the head [45-49]. The backbone of YOLOv8 is based on a modified version of the Cross Stage Partial (CSP) Darknet53 architecture, featuring 53 convolutional layers [45, 49]. This architecture incorporates cross-stage partial connections to enhance the flow of information between the various layers [45, 49]. In the head of YOLOv8 there are several convolutional layers followed by a sequence of fully connected layers. These layers play a crucial role in predicting bounding boxes, object scores, and class probabilities for the detected objects in an image [45, 49]. YOLOv8 incorporates an important feature in its head: the integration of a self-attention mechanism. Positioned in the network's head, this mechanism enables the model to selectively focus on distinct areas of the image, dynamically adjusting the significance of various features based on their relevance to the given task [45, 49]. In addition, YOLOv8 has the capability of performing multi-scaled object

detection. To achieve this task, the model employs a feature pyramid network that facilitates the detection of objects across various sizes and scales within an image. This network comprises multiple layers dedicated to detecting objects at different scales, enabling the model to effectively identify both large and small objects present in the image [45, 49].

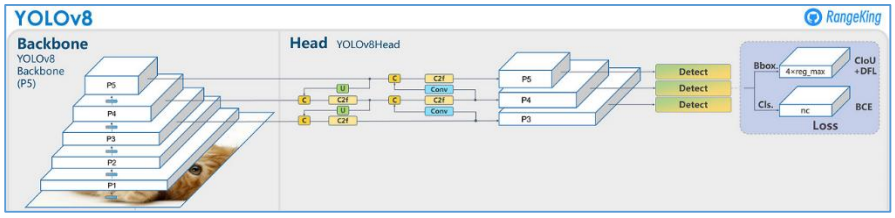

Figure 1

YOLOv8 model architecture (provided by RangeKing [48])

Further, YOLOv8 supports five different neural network sizes that vary in the amount of parameters present in the neural network: YOLOv8n (n-nano), YOLOv8s (s-small), YOLOv8m (m-medium), YOLOv8l (l-large) and YOLOv8x (x-extra large) [45]. It is obvious, that "n" model has the smallest number of parameters, while the "x" model has the largest number of parameters. The parameters are referring on the number of biases and weights in the network [45-49]. While YOLOv8n is the smallest and the fastest model, on the other hand the YOLOv8x is the most accurate and slowest among the YOLOv8 models. Based on the requirements of the application itself, the appropriate model could be selected [45]. Since the aim of the research is to detect the charging socket of the e-vehicle with reliable accuracy in a short time, the YOLOv8s model has been chosen to accomplish this task. According to Ultralytics test on Microsoft Common Objects in Context (MS COCO) dataset [45-48], the YOLOv8s model is to a lesser extent slower than the YOLOv8n model, and it is significantly faster than the other YOLOv8 models according to Open Neural Network Exchange (ONNX) results, while the mean Average Precision on the validation dataset (mAP^{VAL}) is not significantly lower compared to larger models (Table 1). Thus, based on the recommendations [45-46] and test results, the decision fell on the YOLOv8s model in this research.

Table 1

Comparison of YOLOv8 models on COCO dataset [45]

Model	Image Size (pixels)	mAP^{VAL}	Speed CPU ONNX (ms)
YOLOv8n	640	37.3	80.4
YOLOv8s	640	44.9	128.4
YOLOv8m	640	50.2	234.7
YOLOv8l	640	52.9	375.2
YOLOv8x	640	53.9	479.1

3.2 Dataset Preparation and YOLOv8s Training

The overall pipeline of the dataset preparation, training and deployment of the YOLOv8 models is presented in Figure 2. The first step is the dataset preparation which includes the collecting of images and their annotation which can include some arbitrary pre-processing such as contrast adjustment, resizing, etc. [45, 47]. After annotation and pre-processing, the next step involves augmentation, where training examples are generated based on selected augmentation options such as mosaics, rotation, shear, bounding box orientation, etc. [45, 47]. As it was noted, the Roboflow Framework and its annotation tool was utilized in this research which includes optional pre-processing and augmentation options [47]. Later, the annotated images can be split in training, validation and testing folders, while the annotated labels are saved in *.txt files, according to the YOLOv8 models requirements [47]. Finally, the prepared dataset can be exported to YOLOv8 models format and used for training, validation and testing [47]. The second step is the training of the YOLOv8s model object detector according to Ultralytics guidelines [45] in this research and finally the third step is the deployment of the trained network on sample images.

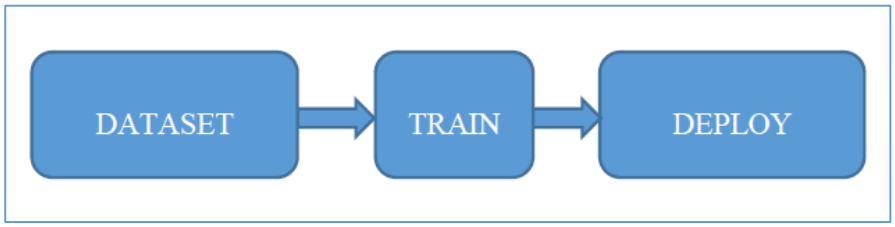

Figure 2

The overall pipeline of the YOLOv8 model [45]

The dataset utilized in this initial study contains 1125 self-created images in order to avoid any authorship issues. Further, all the images were captured with a modest quality camera, since the industrial camera will be procured with the UR10e cobot in future steps of the project. Since the goal was to train a robust object detector, the images were captured under various capturing conditions which includes various lighting, shadow, distance, background, etc. It should be mentioned, that the CCS2 socket is black and its near background is very dark, mostly black. This fact is important, since it is obvious that the detection of this kind of socket is a difficult task for all object detection frameworks. Thus, even the training of the YOLOv8s is a challenging job. After the images were uploaded to Roboflow, the annotation is performed with the frameworks built-in annotation tool [47]. The part of the annotated image dataset can be seen in Figure 3. All the annotated images are labelled appropriately and the labelling data is saved in the corresponding *.txt file [47]. The annotation was followed by an optional pre-processing steps that are included in the online platform [47]. The resizing to 128x70 pixels, automatic contrast adjustment and automatic orientation options were chosen for the image dataset pre-processing and it was executed according to the Roboflow's built-in pre-processing algorithm [47].

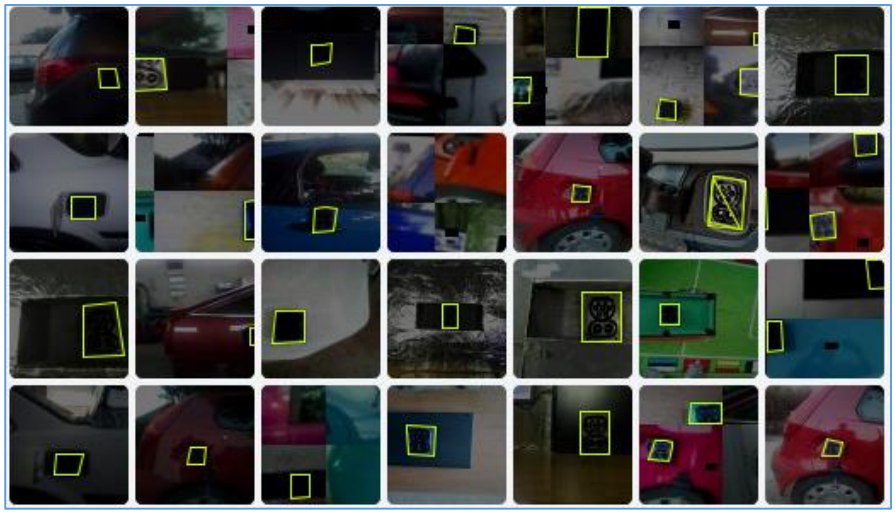

Figure 3

The examples of annotated images [47]

The pre-processing is followed by the augmentation step. As it was mentioned, the data augmentation is an addition of new artificially derived data from existing training data with a goal to enhance the training of the neural network [45, 47]. In this study 10 augmentation options were included as it can be seen in Table 2. All these augmentation operations are randomly applied to the whole dataset according to Roboflow's built in algorithm [47]. Finally, the dataset was split in three folders

according to Roboflow's recommended splitting procedure. After the split is performed, the new folders are: the training folder, validation folder and test folder. Hence, the training folder contains 975, the validation 93 and the test 57 images respectively. The next step is the exportation of the dataset in ZIP file in the appropriate YOLOv8 format provided by Roboflow with the corresponding *.yaml file that contains the information related to dataset folders, number of object classes ("nc") and the objects names ("names") [47]. Later, the ZIP file should be extracted in a main folder, where the training process will be performed. All this information are required during the training, validation and testing process of the YOLOv8s model, thus the *.yaml file should be included in the main folder with the dataset folders [45-49]. If the *.yaml file is not provided, or it is incorrectly configured, the YOLOv8s network will not find the suitable data for the training. Finally, after the YOLOv8 framework is installed and set up in the main folder, the training process can start, followed by the validation and testing steps in later stages [45-49].

Table 2

Data augmentation using Roboflow [47]

Augmentation	**Options/Values**
Rotation	Between -5° and +5°
Shear	±5° Horizontal, ±5° Vertical
Hue	Between -25° and +25°
Saturation	Between -25% and +25%
Brightness	Between -10% and +10%
Exposure	Between -10% and +10%
Cutout	1 box with 5% size each
Bounding Box: Orientation	Between -5° and +5°
Bounding Box:Shear	±5° Horizontal, ±5° Vertical
Bounding Box: Exposure	Between -25% and +25%
Mosaic	Applied

The training process is performed according to Ultralytics instructions, mainly with the default settings provided by the framework itself in Command Line Interface (CLI) in Command Prompt on Windows 10 platform [45]. The utilized hardware platform is CPU Intel CORE(TM) i7-10700 2.90GHz with 16GB RAM (without Graphical Processing Unit - GPU), while the software platform is Ultralytics Yolov8.0.2.212, Python-3.10.9, torch-1.13.1. The training process was done with the YOLOv8s model, with 100 epochs, image size of 640 pixels and number of images per batch 32 [45-48], while all the other parameters were remained default

[45]. The one epoch is when an entire dataset is passed forward and backward through the neural network only once, while the number of batches is a divided dataset into smaller sets or parts (called batches) that are passed through the neural network [45, 47]. The batch size is the total number of training examples present in a single batch [45, 47]. The training time was 21.133 hours and the achieved mAP50 was 0.928, the mAP50-95 was 0.745, the recall was 0.889, the precision was 0.947 and the inference time was 100.6ms on the training dataset. The mAP50 refers to the calculation of the average precision across different levels of recall, up to a limit of 50 detections per image. This metric helps assess how well a model is performing in terms of both precision and recall, with a focus on the top 50 predictions [45]. The mAP50-95 is an extension of the mean mAP metric, specifically considering a range of IoU thresholds. The mAP50-95 is calculated by averaging the Average Precision (AP) values over a range of IoU thresholds, typically from 0.5 to 0.95, in increments of 0.05. The diagrams of the training results are shown in Figure 4, where the X axis represents the number of epochs, and the Y axis represents the corresponding parameter: the precision, recall, mAP50 and mAP50-95 respectively. Obviously, the training time would be shorter with an available GPU [45].

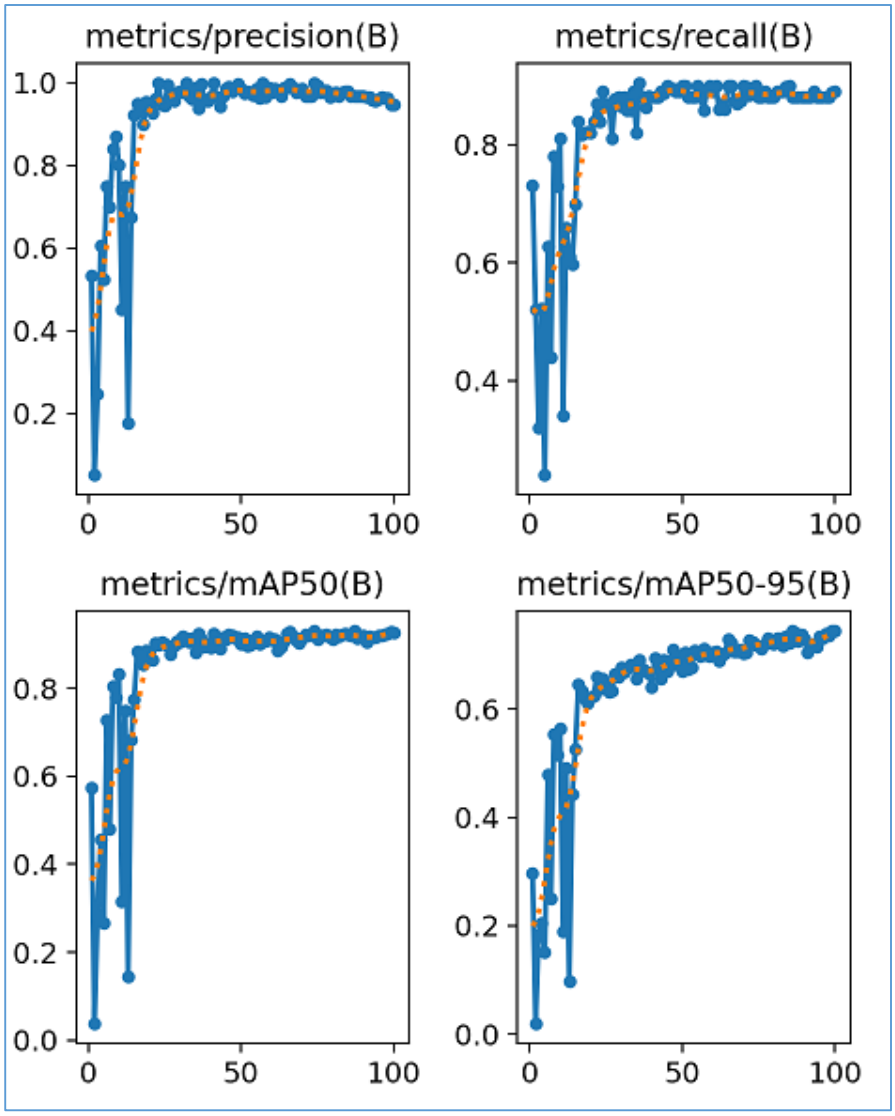

Figure 4

The training results diagrams [45]

In order to show the detecting capabilities of the trained YOLOv8s model, Figure 5. presents several detecting results from the validation dataset. As it can be noticed, the detection results are considerably accurate and acceptable, even in the examples were the CCS2 socket is poorly visible and distinguishable from the background.

Figure 5

The results from the validation dataset

Further, Figure 6. presents several detecting results from the test dataset. As it can be seen, the detection results are also very accurate and acceptable, although the images are of poor quality.

Figure 6

The results from the test dataset

In the end, the training results of the YOLOv8s are considered as acceptable and they fulfilled the goal of the study. In later developments, an industrial computer with an GPU should be acquired for the work and the control of the whole robotic system that should provide an adequate hardware support for the possible novel YOLOv8s model training. Also, it should be noted that when the UR robot arrives, the industrial camera will be installed on the robotic arm and the testing will be executed in real time on the vehicle body model with a built-in CCS2 socket and with e-vehicles. In the final stage when the system is verified and certified, the testing will be extended to other available e-vehicles.

4 Experiments and Results

In this section, a comprehensive explanation of the conducted experiments and the corresponding results will be presented. As it was highlighted in the Introduction, the primary objective of this initial study is to exclusively employ established and verified CNN object detector in the development of the CCS2 socket detection

procedure. Hence, the trained YOLOv8s model's performance was assessed using artificial vehicle body model equipped with CCS2 socket. A diverse range of capturing conditions were deliberately examined to thoroughly assess and delineate the capabilities and constraints inherent in the implemented object detector. It should be noted, that one of the main aspects in the development of the object detector model is the generation of a high-quality, usable input image appropriate for industrial applications that would be ensured in the real application. In addition to the use of high-quality capturing devices, the main requirement is the formation of an appropriately illuminated environment without disturbing effects, which can ensure the repeatability of a quality image capturing later in the commercial use on the parking lot. Herein, in the experiments, a modest quality camera was utilized in order to determine and examine the possibilities of the trained and deployed YOLOv8s model for an initial study purposes. The study dataset was generated internally using a test vehicle body model equipped with an original CCS2 socket, and it contains 975 images. The images were captured across diverse recording conditions, deliberately including instances under less-than-ideal capturing conditions in numerous examples in order to examine the robustness and the limitations of the trained YOLOv8s object detector. Different capturing conditions includes: various camera distance and angle position, intentional shading, various illumination conditions, hazy images, etc.

Since this is an initial study, utilizing the self-created internal image database serves the purpose of avoiding potential legal repercussions that may arise from the use of images depicting proprietary vehicles. Thus, the testing on real e-vehicles will be arranged at a later stage of the project, when all the equipment with the constructed charging station will be available, with legally rented and insured vehicles by the project management. The self-created dataset is not public since it is a part of a commercial industrial project (2020-1.1.2-PIACI-KFI-2020-00173), and it can be provided only with the permission of the project management and the project client.

The testing and the prediction with YOLOv8 object detector has been executed automatically on the whole dataset placed in a custom folder, with a proposed prediction options [45]. The prediction was launched with the command line interface (CLI), where the confidence threshold was set to 0.2, the image augmentation to prediction sources was turned on during the testing and the result saving option was activated [45]. All the other parameters remained the default and the detection results were saved in a separate folder for evaluation purposes [45].

The testing and prediction results achieved a considerable accuracy, since in 948 images the detection was correct while in 27 images the detection failed (in 13 samples were no detection, and in 14 samples were false detection). This detection resulted with a 97.23% accuracy on the available self-developed image dateset.

Figure 7

The detection results from the self-created dataset

Figure 6. displays 25 samples with correctly detected CCS2 sockets (from the 948 correct samples). As it can be noticed, all the samples are captured under various capturing conditions, mostly with poor quality, since the aim was to assess the capabilities and limitations of the trained YOLOv8s object detector. Further, it can

observed, that even under excessive slant, hazy image, or shadow the socket is successfully detected with a lower confidence level, however the bounding box is correctly drawn around the detected object. Also, in examples where the small part of the CCS2 is missing due to the excessive slant, a correct detection is achieved. One of the reasons of successful detection under inordinate image capturing conditions is the activated augmentation option during the prediction according to the related documentation [45]. Therefore, this is also a great advantage of the YOLOv8s model in applications where an incorrect input images are expected during the work. Naturally, in the real application an appropriate lighting source will be mounted on the parking lot, and an industrial AD/3D camera is intended to be used with a special speckle-free blue laser. All these will contribute to a much better quality input image, which will certainly facilitate and improve the detection result of the trained YOLOv8s object detector.

In the end, this initial study entirely fulfilled the aim of the project, and the detection of the charging socket with a renowned YOLOv8s object detector was achieved. The practical application of the obtained results will be tested in the future, where the experiments will be executed with an UR10e robot on electric vehicles with adequate industrial vision and sensing equipment. The future managerial implications are the legal rent of a certain number of electric vehicles for testing purposes and the construction of an adequate real-world parking lots for experiments with the robot in an industrial environment.

Conclusions

Herein, both the training concepts and the deployment of the YOLOv8s object detector for the detection and extraction of the CCS2 charging socket for the automated electric vehicle charging application were introduced. The main steps of the dataset preparation, training and deployment process were presented. The aim of this study was to develop a robot vision system with a well-known and reliable CNN object detector to secure the work of the robot's running process. Suitable experiments were conducted on self-created vehicle body model with a built-in CCS2 socket. All experiments were achieved successfully and the trained YOLOv8s model showed considerable accuracy, as well as an adequate robustness and resistance to adverse illumination conditions, slant and poor capturing conditions. Based on the experiments, the main limitations of the algorithm in terms of inadequate lighting, excessive slant and capturing conditions were determined, and in the future, they will be avoided with a special, high quality industrial camera on the built e-charging station. During the deployment of the YOLOv8s object detector-based system, all the project requirements and instructions were utilized. As a result, the goal of this study was fully achieved, and the further development of the system will be continued with an installed industrial camera on the UR10e robot and electric vehicles on the conveniently built parking lot.

Acknowledgement

This research is a part of projects GINOP_PLUSZ-2.1.1-21-2022-00249 of University of Obuda and 2020-1.1.2-PIACI-KFI-2020-00173 of University of Dunaujvaros.

References

[1] P. Akella et al., "Cobots for the automobile assembly line", Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), 1999, pp. 728-733 vol.1, doi: 10.1109/ROBOT.1999.770061.

[2] Asif, Seemal, Philip Webb, "Realtime Calibration of an Industrial Robot", Applied System Innovation 5, no. 5: 96., 2022, https://doi.org/10.3390/asi5050096

[3] https://www.universal-robots.com/, Accessed: 08.09.2023.

[4] K. W. E. Cheng, "Recent development on electric vehicles", 2009 3rd International Conference on Power Electronics Systems and Applica-tions (PESA), 2009, pp. 1-5., ISBN:978-1-4244-3845-7

[5] X. Zhou et al., "The current research on electric vehicle," 2016 Chinese Control and Decision Conference (CCDC), 2016, pp. 5190-5194, doi: 10.1109/CCDC.2016.7531925.

[6] H. S. Matharu, V. Girase, D. B. Pardeshi and P. William, "Design and Deployment of Hybrid Electric Vehicle, "2022 International Con-ference on Electronics and Renewable Systems (ICEARS), 2022, pp. 331-334, doi: 10.1109/ICEARS53579.2022.9752094.

[7] W. Luo and L. Shen, "Design and Research of an Automatic Charging System for Electric Vehicles", 2020 15th IEEE Conference on In-dustrial Electronics and Applications (ICIEA), 2020, pp. 1832-1836, doi: 10.1109/ICIEA48937.2020.9248188.

[8] H. Wang, "A New Automatic Charging System for Electric Vehicles", 2021 2nd International Conference on Computing and Data Science (CDS), 2021, pp. 19-26, doi: 10.1109/CDS52072.2021.00011.

[9] https://docs.ultralytics.com/

[10] Tadic, V.; Odry, A.; Burkus, E.; Kecskes, I.; Kiraly, Z.; Klincsik, M.; Sari, Z.; Vizvari, Z.; Toth, A.; Odry, P., “Painting Path Planning for a Painting Robot with a RealSense Depth Sensor”, Appl. Sci. 2021, 11, 1467.

[11] Tadic, V.; Odry, A.; Burkus, E.; Kecskes, I.; Kiraly, Z.; Vizvari, Z.; Toth, A.; Odry, P., “Application of the ZED Depth Sensor for Painting Robot Vision System Development”, IEEE Access 2021, 9, 117845–117859.

[12] Tadic, V.; Toth, A.; Vizvari, Z.; Klincsik, M.; Sari, Z.; Sarcevic, P.; Sarosi, J.; Biro, I., "Perspectives of RealSense and ZED Depth Sensors for Robotic Vision Applications", Machines 2022, 10, 183. https://doi.org/10.3390/machines10030183

[13] R. C. Gonzales and R. E. Woods, Digital Image Processing, 4th ed., Pearson, NJ, USA, 2018.

[14] R. C. Gonzales, R. E. Woods, and S. L. Eddins, Digital Image Processing Using MATLAB, 3rd ed. Knoxville, TN, USA: Gatesmark, 2020.

[15] Fabrizio Flacco, Torsten Kroger, Alessandro De Luca, Oussama Khatib, "A Depth Space Approach to Human-Robot Collision Avoid-ance", 2012 IEEE International Conference on Robotics and Automation RiverCentre, Saint Paul, Minnesota, USA May 14-18, 2012

[16] Ashutosh Saxena, Sung H. Chung, Andrew Y. Ng, "3-D Depth Reconstruction from a Single Still Image", International Journal of Computer Vision, 2008, Volume 76, Issue 1, pp 53–69

[17] Vladimiros Sterzentsenko, Antonis Karakottas, Alexandros Papachristou, Nikolaos Zioulis, Alexandros Doumanoglou, Dimitrios Zarpalas, Petros Daras, "A low-cost, flexible and portable volumetric capturing system", 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2018, doi: 10.1109/SITIS.2018.00038

[18] Nicole Carey, Radhika Nagpal, Justin Werfel, "Fast, accurate, small-scale 3D scene capture using a low-cost depth sensor", 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), DOI: 10.1109/WACV.2017.146

[19] Mathieu Labbé, François Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation", J Field Robotics. 2018;1–31., DOI: 10.1002/rob.21831

[20] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, Michael Beetz, "Towards 3D Point cloud based object maps for household environments", Robotics and Autonomous Systems 56, 2008, 927_941. doi:10.1016/j.robot.2008.08.005

[21] Tobias Schwarze, Martin Lauer, "Wall Estimation from StereoVision in Urban Street Canyons", Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics, pages 83-90, DOI: 10.5220/0004484600830090

[22] Jean-Emmanuel Deschaud, François Goulette, "A Fast and Accurate Plane Detection Algorithm for Large Noisy Point Clouds Using Filtered Normals and Voxel Growing", 3DPVT, May 2010, Paris, France, hal-01097361

[23] Aghi, D.; Mazzia, V.; Chiaberge, M., “Local Motion Planner for Autonomous Navigation in Vineyards with a RGB-D Camera-Based Algorithm and Deep Learning Synergy”, Machines, 2020, 8, 27. https://doi.org/10.3390/machines8020027l

[24] Kin-Choong Yow1, Insu Kim, “General Moving Object Localization from a SingleFlying Camera”, Applied Sciences, 2020, 10, 6945; doi:10.3390/app10196945

[25] Xianyu Qi, Wei Wang1, Ziwei Liao, Xiaoyu Zhang, Dongsheng Yang, Ran Wei, “Object Semantic Grid Mapping with 2D LiDAR and RGB-D Camera for Domestic Robot Navigation”, Applied Sciences, 2020, 10, 5782; doi:10.3390/app10175782

[26] Vladimir Tadic, Akos Odry, Attila Toth, Zoltan Vizvari, Peter Odry, “Fuzzified Circular Gabor Filter for Circular and Near-Circular Object Detection”, IEEE Access, DOI: 10.1109/ACCESS.2020.2995553

[27] Mingqiang Pan, Cheng Sun, Jizhu Liu, Yangjun Wang, “Automatic recognition and location system for electrlc vehicle charging port in complex environment”, IET lmage Processing, 2020, Vol.14 188.10, pp. 2263-2272, doi: 10.1049/iet-ipr.2019.1138

[28] Hui Zhang, Xiating Jin, “A Method for New Energy Electric Vehicle Charging Hole Detection and Location Based on Machine Vision”, 5th International Conference on Environment, Materials, Chemistry and Power Electronics, EMCPE, 2016, Atlantis Press

[29] Justinas Mišeikis, Matthias Rüther, Bernhard Walzel, Mario Hirz and Helmut Brunner, “3D Vision Guided Robotic Charging Station for Electric and Plug-in Hybrid Vehicles”, Proceedings of the OAGM&ARW Joint Workshop, 2017, doi: 10.3217/978-3-85125-524-9-13

[30] P. Quan, Y. Lou, H. Lin, Z. Liang and S. Di, "Research on Fast Identification and Location of Contour Features of Electric Vehicle Charging Port in Complex Scenes", in IEEE Access, vol. 10, pp. 26702-26714, 2022, doi: 10.1109/ACCESS.2021.3092210.

[31] Quan, P.; Lou, Y.; Lin, H.; Liang, Z.; Wei, D.; Di, S., “Research on Fast Recognition and Localization of an Electric Vehicle Charging Port Based on a Cluster Template Matching Algorithm”, MDPI Sensors 2022, 22, 3599. https://doi.org/10.3390/s22093599

[32] Y. Lou and S. Di, "Design of a Cable-Driven Auto-Charging Robot for Electric Vehicles", in IEEE Access, vol. 8, pp. 15640-15655, 2020, doi: 10.1109/ACCESS.2020.2966528.

[33] Lin, H.; Quan, P.; Liang, Z.; Lou, Y.; Wei, D.; Di, S., “Collision Localization and Classification on the End-Effector of a Cable-Driven Manipulator Applied to EV Auto-Charging Based on DCNN–SVM”, MDPI Sensors, 2022, 22, 3439. https://doi.org/10.3390/s22093439

[34] Li, T.; Xia, C.; Yu, M.; Tang, P.; Wei, W.; Zhang, D., "Scale-Invariant Localization of Electric Vehicle Charging Port via Semi-Global Matching of Binocular Images", Appl. Sci. 2022, 12, 5247. https://doi.org/10.3390/app12105247

[35] Damien Chablat, Riccardo Mattacchione, Erika Ottaviano, "Design of a robot for the automatic charging of an electric car", ROMANSY 24 - Robot Design, Dynamics and Control, Springer, 2022, hal-03624780 Vladimir

[36] Vladimir Tadic, "Study on Automatic Electric Vehicle Charging Socket Detection using ZED 2i Depth Sensor", MDPI Electronics, 2023, 12, 912. https://doi.org/10.3390/electronics12040912

[37] Muhammad Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection", MDPI Machines 2023, 11, no. 7: 677., https://doi.org/10.3390/machines11070677

[38] Hicham Slimani, Jamal El Mhamdi, Abdelilah Jilbab, "Artificial Intelligence-based Detection of Fava Bean Rust Disease in Agricultural Settings: An Innovative Approach", International Journal of Advanced Computer Science and Applications, Vol. 14, No. 6, 2023

[39] Nabin Sharma, Sushish Baral, May Phu Paing, and Rathachai Chawuthai, "Parking Time Violation Tracking Using YOLOv8 and Tracking Algorithms", MDPI Sensors, 2023, 23, no. 13: 5843. https://doi.org/10.3390/s23135843

[40] Talaat, F.M., ZainEldin, H., "An improved fire detection approach based on YOLO-v8 for smart cities", Neural Computing & Application, 35, 20939–20954, 2023, Springer, https://doi.org/10.1007/s00521-023-08809-1

[41] Ruihan Bai, Feng Shen, Mingkang Wang, Jiahui Lu, Zhiping Zhang, "Improving Detection Capabilities of YOLOv8-n for Small Objects in Remote Sensing Imagery: Towards Better Precision with Simplified Model Complexity", 22 June 2023, Preprint Version 1 available at Research Square [https://doi.org/10.21203/rs.3.rs-3085871/v1]

[42] Fatemeh Rashidi Fathabadi, Janos L. Grantner, Ikhlas, Abdel-Qader, Saad A. Shebrain M.D., "Box-Trainer Assessment System with Real-Time Multi-Class Detection and Tracking of Laparoscopic Instruments, using CNN", Acta Polytechnica Hungarica, Vol. 19, No. 2, 2022, DOI:10.12700/APH.19.2.2022.2.1

[43] Lei Kou, "A Review of Research on Detection and Evaluation of the Rail Surface Defects", Acta Polytechnica Hungarica, Vol. 19, No. 3, 2022, DOI:10.12700/APH.19.3.2022.3.14

[44] Man-Wen Tian, Ardashir Mohammadzadeh, Jafar Tavoosi, Saleh Mobayen, Jihad H. Asad, Oscar Castillo, Annámaria R. Várkonyi-Kóczy, "A Deep-learned Type-3 Fuzzy System and Its Application in Modeling Problems",

Acta Polytechnica Hungarica, Vol. 19, No. 2, 2022, DOI:10.12700/APH.19.2.2022.2.9

[45] https://docs.ultralytics.com/, Accessed: 05.10.2023.

[46] Frederik Fritsch, "Deep Neural Networks for Object Detection in Satellite Imagery", UPTEC IT23 014 Master Thesis, Uppsala Universitet, 2023.

[47] https://roboflow.com/, Accessed: 05.08.2023.

[48] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang and Jian Yang, "Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection", arXiv:2006.04388, 2020.

[49] Yolov8 model architecture layout. https://github.com/RangeKing, Accessed: 08.10.2023.