Scalable Shared Encoding Architecture for Learning-Based Error Detection in Robotic Wiring Harness Assembly

Scalable Shared Encoding Architecture for Learning-Based Error Detection in Robotic Wiring Harness Assembly

Kevin Galassi[a], Alessio Caporali[a], Gianluca Laudante[b], Gianluca Palli[a]

***Abstract*—This paper focuses on an automatic solution for the detection of manufacturing errors in the context of automatic wiring harness assembly. In the proposed setup, a robot grasp the wires and places them in specific assembly clips according to the wiring harness design. However, due to the deformability of cables, the process outcome is not completely predictable since sometimes the cables remains entangled in other parts of the assembly or do not fit properly into the clips. The proposed error detector verifies the correct insertion of each cables within the clip, considering that the number of cables and their dimension change along the different assembly stage.**

The proposed solution covers possible state-of-the-art network learning model that use point clouds as input source, while the network architecture is designed to offer precision and scalability in the context of a flexible and dynamic automation. The developed solution achieved a 96% precision on a dataset composed by various scenario. Therefore, despite being conceived for a robotic wiring harness manufacturing system, the proposed solution can be potentially applied as an online quality control system in manual wiring harness manufacturing.

***Index Terms*—Deformable Linear Objects, Wiring Harness, Fault Detection, Industrial Manufacturing**

I. Introduction

Wiring harnesses are fundamental components in automotive and aerospace industries, buildings and industrial plants. However, their production is still almost completely manual, rising the production cost and inefficiency. This manufacturing operation is particularly challenging due to the complexity of the product, number of variants and, more important, deformabiltity of the cables.

The automation of wiring harness manufacturing is gaining increasing significance [1]. Recently, robotic solutions for the wiring harness manufacturing have been developed within the REMODEL project. It is worth noticing that this automation process remains a challenge due mainly to the manipulation of Deformable Linear Objects (DLOs), i.e. wires and cables, [2], operation that requires a high level of sensitivity and dexterity that actual robotic hands often lack [3]. From a perception standpoint, DLOs offer limited features that a vision system can leverage for detection, requiring expensive hardware [4] or specialized learning-based algorithms [5], [6].

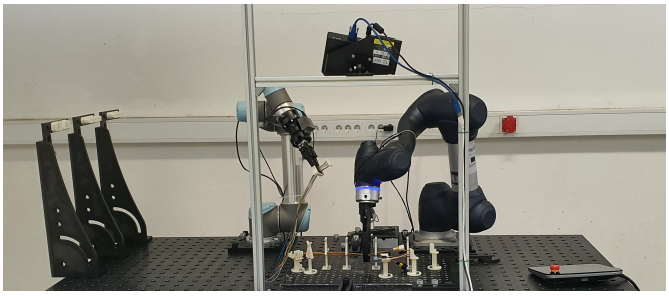


Fig. 1: Workcell for robotic wiring harness assembly.

Wiring harnesses are formed by interconnecting multiple DLOs, resulting in the creation of several branches. The assembly between the different DLO components is then obtained via tape or zip ties. In the context of robotic wiring harness assembly, assessing the quality of the final assembled product, as well as of the intermediate stages of the process, remains an open research problem.

The domain of computer vision has experienced a substantial evolution with the rise of deep learning, enabling machines to understand complex visual information. In recent years, there has been a significant emphasis on employing deep neural networks for the analysis of three-dimensional (3D) data, especially pointclouds [7].

This paper focuses on quality control in wiring harness assembly processes. The proposed approach aims to provide quality feedback in the context of automated wiring harness assembly, as depicted in Fig. 1. More specifically, the objective is to identify errors that occur when one or more wires are incorrectly inserted into a routing clip, resulting in an external path rather than an internal one. The method presented here utilizes a learning-based approach on point cloud data to detect errors during the routing process performed by a robotic system. The proposed implementation features a shared encoder architecture that enables training with a reduced set of samples.

In the remainder of the paper, the related works are discussed in Sec. II. The problem statement is presented in Sec. III while the proposed method is illustrated in Sec. IV. The method is validated in Sec. V. Finally, the conclusions are drawn in Sec. VI.

[a]Kevin Galassi, Alessio Caporali and Gianluca Palli are with DEI - Department of Electrical, Electronic and Information Engineering, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy.

[b]Gianluca Laudanti is with Department of Engineering of the University of Campania "Luigi Vanvitelli", Via Roma 29 - 81031, Aversa (CE), Italy.

Corresponding author: *kevin.galassi2@unibo.it*

This work was supported by the Horizon Europe project *IntelliMan - AI-Powered Manipulation System for Advanced Robotic Service, Manufacturing and Prosthetics* [grant number 101070136].

II. Related Works

A. Wiring harness Manufacturing

Wiring harness manufacturing is a complex and crucial process in today's industry [8]. It comprises multiple steps, including the manufacturing of individual wires with the crimping of connectors if needed, followed by the arrangement of multiple cables and sub-harnesses together. The harness is then assembled by tape or zip ties, and the final stage involves testing the assembly quality and the electrical properties of the harness. Considering the entire manufacturing process, from wire cutting to on-board testing in the vehicle, the assembly of connectors represents the most crucial part where significant time savings can be achieved [9]. The automation of this process using robotic arms is of interest to the automotive and aerospace markets as it enables accelerated production and reduces the cost of the final product.

B. Deformable Linear Objects Perception

Vision-based perception emerges as the predominant choice for DLOs among various sensing modalities, primarily due to the accessibility of diverse sensors and cameras seamlessly integrating into robotic systems [10].

The exploration of semantic and instance segmentation for DLOs, utilizing data-driven methods and synthetic datasets, is undertaken in [6], [11]. The process of shape estimation for DLOs involves extracting the DLO's state from vision-based data to obtain its actual configuration, often represented as a sequence of key-points. Recent approaches with 2D data and real-time capabilities are outlined in [5], [12], [13].

However, achieving a 3D characterization of the shape is typically imperative for tasks involving grasping and manipulation. The direct 3D shape estimation of DLOs has received comparatively less exploration than its 2D counterpart. Practitioners typically address the 2D shape estimation problem first and subsequently utilize depth data to transform the estimated shape into Cartesian space, as discussed in [13].

C. Vision-based Deformable Multi-Linear Objects Perception

The body of research concerning the vision-based perception of Deformable Multi-Linear Objects (DMLOs) is notably sparse, despite the paramount importance of DMLOs in the industrial sector, with less attention compared to DLOs [8].

In [14] is proposed a data-driven architecture tailored for optical inspection tasks related to DMLOs, specifically focusing on segmenting crucial components. However, the study is constrained by a relatively limited training dataset. In a subsequent investigation [15], synthetic data generated from CAD and simulation models are introduced, facilitating a comparative analysis between synthetic and real data, underlying the advantages of integrating synthetic data into the study.

The categorization of DMLO branches is considered in [16]. The work exploits manual data collection and annotation to create a moderately sized dataset, exploring the data augmentation techniques as a means to mitigate challenges arising from the scarcity of available data.

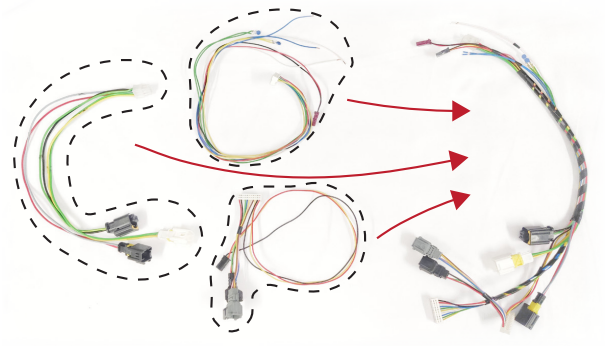


Fig. 2: Wiring harness assembly process example. Transforming 3 separate pieces into the final product through taping.

Several studies adopt a graph-based representation of DMLO configuration centered around *branch-points* [17], [18], [19]. [18] concentrates on correspondence estimation between a known DMLO topology and an actual scene. A directed graph-based representation is constructed from reference information (e.g. CAD), while the scene's topology is derived as an undirected graph through image skeletonization of the actual DMLO. The study delves into the matching problem, accommodating the possibility of partial correspondence.

In [19] a method for localizing and tracking DMLOs is proposed, relying on a combination of rigid and non-rigid registration phases. However, their approaches assume a non-overlapping configuration for DMLOs, limiting their applicability. Finally, [17] addresses the detection of branch points or overlapping points using a data-driven approach. The work introduces a semi-manual annotation procedure to generate necessary training data. However, the semi-manual annotation method lacks a comprehensive user study (involving only one subject), and the evaluation is confined to a single DMLO type, complicating the overall assessment of the proposed method.

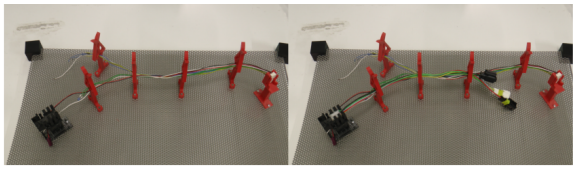
III. Problem Statement

In this research, the problem of error detection in a complex robotic wiring harness manufacturing scenario is considered. Specifically, multiple components of a wiring harness need to be assembled — usually with tape or zip ties — into a single finished product, as demonstrated in Fig. 2. The detection of misplaced wires is of paramount importance to assess the quality of the final product.

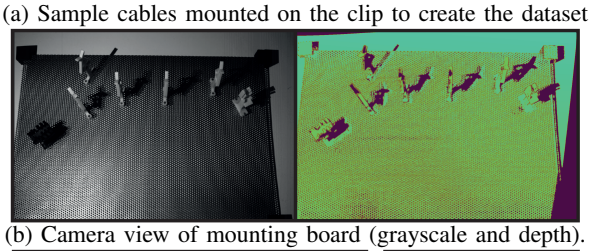
The assembly process is implemented using a robotic platform, depicted in Fig. 1, which is composed of two robotic arms. The first arm is equipped with a specialized gripper to perform routing tasks within the DLO following approach [20]. The second arm is equipped with a spot-taping gun used for the taping process, effectively assembling two separate components.

The mounting board is composed of multiple clips and connector holders, specifically designed to withstand the force exchange during the manipulation and to keep in place the components [21].

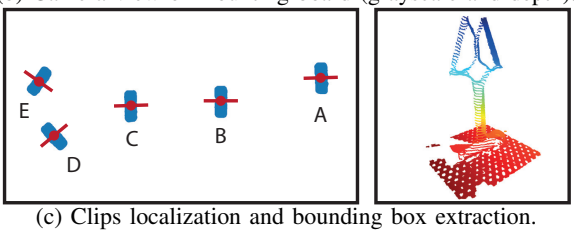
The operation involves a standard routing task [20], where one end of a wiring harness is first inserted into a connector



(a) Sample cables mounted on the clip to create the dataset



(b) Camera view of mounting board (grayscale and depth).



(c) Clips localization and bounding box extraction.

Fig. 3: Data collection setup.

holder, keeping that end fixed. Subsequently, a branch of the wiring harness—such as a set of cables—is routed into a sequence of clips to achieve the desired configuration. This operation is repeated for all the components to obtain the desired final configuration (see Fig. 2).

Despite the complexity associated with the manipulation aspects, the assessment of the success of not in the routing operation is also complex and fundamentally required. Indeed, fault or error detection is of paramount importance in robotic manufacturing and assembly processes. Hence, implementing a method to verify the correct insertion of all cables into the clip at each step becomes crucial.

For the perception system, a 3D camera is utilized, statically mounted on the robotic workcell, as shown in Fig. 1. The camera faces the mounting board, providing an almost top-down view to minimize reflections. The model camera is a Photoneo PhoxiScanner M.

IV. Method

The method analyzes the point cloud of the mounting board, which consists of a set of clips and wires. It outputs a fault signal if a misplaced wire is detected in the scene. A learning-based approach is employed, and the details of data collection and labeling processes are provided in Sec. IV-A. The model architecture is illustrated in Sec. IV-B.

A. Data Collection and Labeling

The process of acquiring the training dataset is achieved by a three steps approach: 1) localizing the individual clips; 2) acquiring the set of samples; 3) labeling the samples.

1) Clips Localization: By leveraging on CAD files, the system can align the point cloud scene with the mounting board setup, facilitating the identification of individual clips. This alignment process is carried out without the insertion of wiring harness branches to simplify the scene and reduce the likelihood of mislocalization, see Fig. 3b.

Specifically, the process starts by processing the point cloud, removing the plane so that only the points belonging to the clips and connector holders remain. This segmentation process is simplified by transforming the point cloud into a world reference frame fixed to the table.

An orthogonal projection is then applied to the point set, yielding a mask of the mounting board with a (virtual) top-down-like view. The mask undergoes dilation to highlight the individual *blobs*, which are identified through connected component analysis. The centroid of each blob is used to denote its position. Additionally, principal component analysis is employed to obtain the two axes of each blob.

The centroids of the blobs are registered against the CAD data to associate each blob with a specific known clip, as illustrated in Fig. 3c.

2) Data Collection: The data are collected manually where, for each sample, an operator moves the cables into a different configuration. The 5 localized clips of Sec. IV-A1 and the 3 sub-harnesses of Fig. 2 result in 8 possible configurations of cable placement inside the clips. For each configuration, 45 positive and 45 negative samples are collected. The operation of gathering the data for one configuration requires approximately 20 minutes.

3) Point Cloud Data Labeling: Given the localized clips, a 3D bounding box is computed around each clip, see the left side of Fig. 3c. Notice that, compared to the clip shown in the figure, the data samples also contain cables inside the scene. Each bounding box represents a data sample in the learning pipeline. In this way, both the amount of data and the complexity of the learning process are reduced.

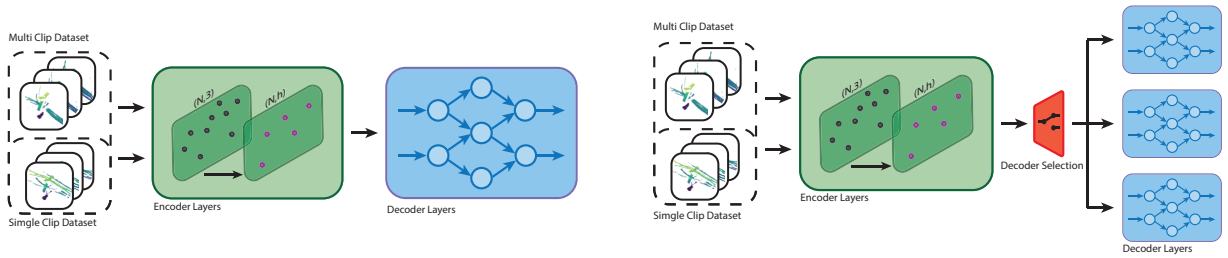
The farthest point sampling algorithm [22] is applied to reduce the number of points to 6000 for each data sample, aligning data dimensions across the entire dataset during pre-processing. This is done instead of using run-time padding techniques, which increase complexity during training and may introduce noise.

Several data augmentation techniques are also applied to help alleviate the reduced scale of the dataset. First, the cutting region of the bounding box is changed during training by applying random Gaussian noise to the center of the box and scaling the box dimensions to increase variability. The extracted points are augmented by adding random noise to the position and rotation of the point cloud. Finally, the point cloud is normalized between 0 and 1 to help stabilize the learning process.

B. Model Architecture

The network model is based on PointNet [22], representing a state-of-the-art approach in terms of data-driven methods applied to point cloud data.

The model is structured as an encoder-decoder system, as shown in Fig. 4. The encoder consists of layers from [23], which take the pointcloud as input, reducing its dimension while increasing features size. A multi-layer perception (MLP)



(a) Clip-Specific/General Model

(b) Shared-Encoder Model

Fig. 4: Illustration of the evaluated model architectures based on the training style of Tab. I. Using the model in Fig.4a the network is trained based on the dataset composed by a single clips or a larger dataset containing multiple samples. In Fig. 4b a common shared encoder is used to learn the features, while the detection is left to a clip-specifi decoder to improve the precision and the scalability.

TABLE I: Training style qualitative evaluation in terms of expected accuracy, training time, scalability and dimension.

Training Style	Accuracy	Training Time	Scalability/ Adaptability	Number Parameters
Clip-Specific Model	+++	+++	+	+
General Model	+	++	++	+++
Shared-Encoder Model	++	+	++	++
Shared-Encoder Model + Fine-Tuning	+++	++	+++	+++

is used for feature embedding and it is shared among all points. During this process, the number of points is reduced by performing a sampling of the network. Features are sampled using Single-Scale Grouping (SSG) introduced in PointNet.

The second part of the network is the decoder, and since it is a classification task, it is composed of multiple linear layers with BatchNorm [24] and dropout [25] to enhance learning robustness. The final output dimension is 2, corresponding to the probability of predicting positive or negative results.

Alternative encoder structures are investigated to study the system's accuracy and robustness. The studied encoders are: 1) Relation-Shape Convolutional Neural Network (RS-CNN) [26]; 2) Point Transformer Layers [27].

RS-CNN employs Multi-Scale Grouping (MSG) to sample features between the following layers. The key difference compared to SSG lies in the fact that the feature vector is a combination of multiple MLPs, each designed to encode a subset of the feature vector. In this work, MSG is preferred over Multi-Resolution Grouping (MRG), an alternative sampling technique, since the pointclouds dimensions in our training set is mostly constant.

The PointTransformer can replace each PointNet layer on the encoder side. With this modification, the attention mechanism is applied directly to the pointcloud itself, and the resulting features are subsequently assessed through cross-attention using the same points from the pointcloud as queries. The objective is to evaluate features both locally and globally to achieve a more comprehensive understanding of the scene.

C. Training Style

The model architectures illustrated in Sec. IV-B are commonly employed to learn and classify/segment datasets with multiple distinct objects (such as airplanes and chairs), leading to significantly different exhibited features. Conversely, in our scenario, all scenes feature the same clip model and objects (DLOs), differing only in the number and dimensions of the wires composing the considered clip/wiring harness pair.

It is reasonable to assume that a common feature extraction encoder can be employed to generate a sequence of features for classification. The responsibility of error detection can then be assigned to a detector specifically trained for each clip. Therefore, the possible model architectures and training styles can be categorized as follows:

1) Clip-Specific Network: This network is trained on a specific clip to maximize the efficiency of the detector (single clip dataset of Fig. 4a);
2) General Model Network: The entire dataset (multiple clips) is used to train a single model (multi-clip dataset of Fig. 4a);
3) Shared Encoder: A network consisting of a shared encoder and clip-specific decoders for classification (see Fig. 4b).

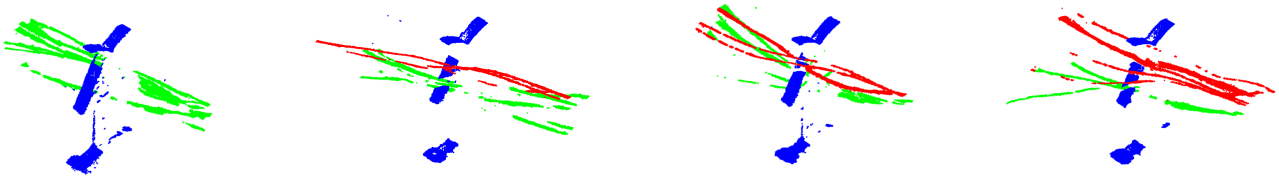
A qualitative evaluation of the differences between these training approaches can be found in Tab. I. The table provides a qualitative analysis of network accuracy, training time, scalability/adaptability, and number of parameters.

Ideally, adopting the shared encoder approach represents the most efficient and scalable strategy. In this configuration, the encoder side is trained to capture the general features of both the clips and the cables, while each decoder is dedicated to performing the clip-specific detection task.

V. Results

To validate the approach, a mockup of a real platform presented in Fig.2 is used for robotic wiring harness assembly. The mentioned model architectures (Sec. IV-B), are trained according to the styles detailed in Sec. IV-C. The collected dataset (Sec. IV-A) of 720 samples is split into train and validation sets according to a 70/30 ratio. Common hyper-parameters employed are batch size of 6, 200 epochs, and learning of 0.001. As loss function, binary cross entropy is employed. The best model is selected as the one associated with the minimum validation loss.

Test samples are gathered to assess the proposed method, employing the same procedure outlined in Sec. IV-A. For each



(a) Pos (b) N1 (c) N2 (d) N3

Fig. 5: Example of dataset test samples for clip B. Colors meaning: blue (clip), red (wrong wires), green (correct wires).

clip configuration, 28 samples are acquired, with specific attention given to the complexity of each sample. While positive cases demonstrate minimal variation, instances of failure are further classified into three groups equally distributed based on increasing difficulty. In total, 4 types of test samples are considered, as shown in Fig. 5. In details:

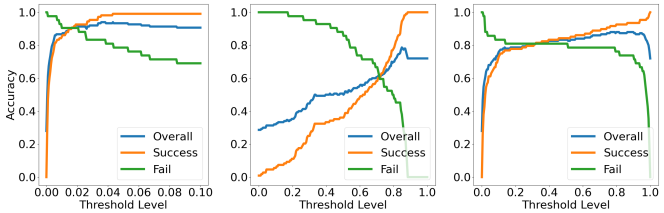
- P: Positive case with all wires inside the clip;
- N1: Simpler cases where most of the cables are outside the clip, easily noticeable;
- N2: Scenarios where most of the cables are correctly inserted;
- N3: The most challenging scenarios involving only 1 or 2 thin cables outside a clip.

Considering the 8 possible configurations, in total 224 samples are employed within the test set. The dataset not include the robot in the scene, since can be easily removed from the pointcloud by projecting a convex hull corresponding to the robot position to remove the non-relevant information.

A. BackBone Comparison

To assess the differences among the three detailed backbones (see Sec. IV-B), we conduct model training using a general classification approach (one model for all examples, General Model of Fig. 4a). The decoder side remains shared across all models, comprising a three-layer MLP with an input feature size of 1024 and an output of 2.

The accuracy results presented in Fig. 6 reveal a higher overall accuracy when utilizing PointNet++ as the backbone. Slightly inferior results are observed with the RS-CNN layers, while PointFormer performs less favorably, probably due to the higher number of parameters involved leading to fitting problems. Indeed, a significant difference lies in the number of parameters for each model. Specifically, the PointNet++ model weighs approximately 545K parameters, the PointFormer model have 2.895K params while the RS-CNN model has 1.737K params.



(a) PointNet++ (b) PointTransformer (c) RS-CNN

Fig. 6: Backbone accuracy comparison.

TABLE II: Experiment of Single Clip vs General Model

Test Clip	Model Type	Mean	Succ	N1	N2	N3
A	Single [A]	100.00	100.00	100.00	100.00	100.00
	Single [B]	83.33	66.67	83.33	83.33	100.00
	General	91.67	100.00	100.00	83.33	83.33
B	Single [A]	83.33	100.00	100.00	66.67	66.67
	Single [B]	91.67	100.00	100.00	83.33	83.33
	General	87.50	83.33	83.33	100.00	83.33

B. Training Style Comparison

1) Single Clip vs General Model: The first comparison pertains to the model illustrated in Fig. 4a. The results are provided in Tab. II, where the training of a clip-specific model versus a general model is compared. The model architecture employed is based on PointNet++. Initially, it is trained for specific clips, namely clip A and clip B, resulting in models *Single (A)* and *Single (B)*.

Subsequently, a general model incorporating data from both clip A and clip B is trained and denoted as *General*. While the *Single (A/B)* models are trained to predict specific scenarios, the *General* model benefits from more extensive data during training, as its dataset comprises the combination of both clips in the assembly platform.

The result, in Tab. II, shows the accuracy of the prediction of each model w.r.t. to each test samples subclass. During cross-testing, comparing the performance of a model trained on one clip against the other, it becomes evident that the *General* model can address this scenario with higher accuracy.

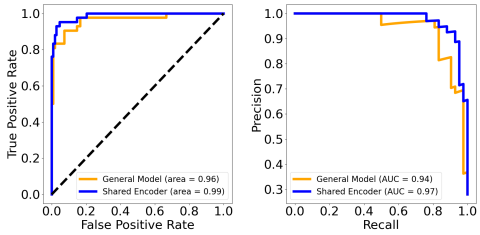
2) General Model vs Shared Encoder: The second evaluation concerns the model illustrated in Fig. 4b. Given the variability in the number and dimensions of cables to be inserted during wiring, it is pertinent to explore the potential of employing a shared encoder and specific decoders for each clip. Like in the previous case, the reported results present the accuracy of each model in the prediction of each subclass of the test samples.

To ensure a more equitable comparison, we train the model using the same dataset and a comparable number of 200 epochs. Tab. III presents the results of the following experiments: 1) we assess the difference between the General Model and the Shared-encoder Model trained from scratch; 2) we explore the feasibility of utilizing the pre-trained encoder obtained from the General Model's training.

The use of the pre-trained encoder is examined after 100 and 200 epochs of training, employing the same methodology as before, and is further fine-tuned for an additional 100 or

TABLE III: Experiment of General Model versus Shared-encoder Model. Px denoted the number of pre-training epochs. FTx denotes the number of fine-tuning epochs.

Model	Training	Mean	Succ.	N1	N2	N3
General Model	P0	93.33	83.33	100.00	97.22	94.44
Shared-Encoder Model	P0 + FT200	63.33	95.24	50.00	47.22	55.56
	P100 + FT100	96.67	92.86	100.00	100.00	94.44
	P100 + FT200	94.67	95.24	94.44	94.44	94.44
	P200 + FT100	95.33	92.86	97.22	97.22	94.44
	P200 + FT200	96.00	92.86	100.00	97.22	94.44



(a) ROC Curve (b) PR Curve

Fig. 7: Receiver Operating Characteristic (ROC) Curve and Precision-Recall (PR) Curve comparing General Model and Shared-Encoder Model.

200 epochs. The ROC and PR curves are shown in Fig. 7 for both the General Model and the Shared-Encoder Model (variant P200 + FT200).

VI. Conclusion

In this paper, a scalable shared encoding architecture for error detection is proposed, aiming to provide a solution to fill the gap in industrial solutions for automotive wiring production. The method employs a learning-based approach to detect faults, particularly misplaced wires around routing clips. During the study, multiple architecture are evaluated, to improve the scalability of the method without affecting the accuracy of the model or the applicability in industrial scenario. The final solution uses a combination of shared encoder across the dataset combined with a specific decoder to improve the effectiveness and reduce the training time in case of addition of a new scenario, reaching a mean accuracy of 96%. The solution has been proven to work with a single camera across all the clips in the scene, making it applicable to real-world applications. The data collection process needed by the application is minimally costly in terms of time and does not require specialized workers, moreover, by using the pre-trained encoder new scenario can be easily added by fine-tuning the decoding part of the network resulting in faster setup time. In future works, an error recovery policy will be investigated alongside the possibility of using synthetic data for training purposes.

References

[1] T. P. Nguyen, D. Kim, H.-K. Lim, and J. Yoon, "Revolutionizing robotized assembly for wire harness: A 3d vision-based method for multiple wire-branch detection," *J. of Manufacturing Systems*, 2024.

[2] A. Caporali, P. Kicki, K. Galassi, R. Zanella, K. Walas, and G. Palli, "Deformable linear objects manipulation with online model parameters estimation," *IEEE Robotics and Automation Letters*, 2024.

[3] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li, J. Pan, W. Yuan, and M. Gienger, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics Automation Magazine*, 2022.

[4] K. P. Cop, A. Peters, B. L. Žagar, D. Hettegger, and A. C. Knoll, "New metrics for industrial depth sensors evaluation for precise robotic applications," in *RSJ Int. Conf. IROS*. IEEE, 2021.

[5] A. Caporali, K. Galassi, B. L. Žagar, R. Zanella, G. Palli, and A. C. Knoll, "RT-DLO: Real-time deformable linear objects instance segmentation," *IEEE Transactions on Industrial Informatics*, 2023.

[6] A. Caporali, M. Pantano, L. Janisch, D. Regulin, G. Palli, and D. Lee, "A weakly supervised semi-automatic image labeling approach for deformable linear objects," *IEEE Robotics and Automation Letters*, 2023.

[7] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2021.

[8] J. Trommnau, J. Kühnle, J. Siegert, R. Inderka, and T. Bauernhansl, "Overview of the state of the art in the production process of automotive wire harnesses, current research and future trends," 2019.

[9] S. Olbrich and J. Lackinger, "Manufacturing processes of automotive high-voltage wire harnesses: State of the art, current challenges and fields of action to reach a higher level of automation," *Procedia CIRP*, 2022.

[10] K. P. Cop, A. Peters, B. L. Žagar, D. Hettegger, and A. C. Knoll, "New metrics for industrial depth sensors evaluation for precise robotic applications," in *RSJ Proc. of IROS*. IEEE, 2021.

[11] J. Dirr, D. Gebauer, J. Yao, and R. Daub, "Automatic image generation pipeline for instance segmentation of deformable linear objects," *Sensors*, 2023.

[12] A. Choi, D. Tong, B. Park, D. Terzopoulos, J. Joo, and M. K. Jawed, "mbest: Realtime deformable linear object detection through minimal bending energy skeleton pixel traversals," *arXiv preprint arXiv:2302.09444*, 2023.

[13] P. Kicki, A. Szymko, and K. Walas, "Dloftbs – fast tracking of deformable linear objects with b-splines," in *Proc. of ICRA*. IEEE, 2023.

[14] H. G. Nguyen and J. Franke, "Deep learning-based optical inspection of rigid and deformable linear objects in wiring harnesses," *Procedia CIRP*, 2021.

[15] H. G. Nguyen, R. Habiboglu, and J. Franke, "Enabling deep learning using synthetic data: A case study for the automotive wiring harness manufacturing," *Procedia CIRP*, 2022.

[16] P. Kicki, M. Bednarek, P. Lembicz, G. Mierzwiak, A. Szymko, M. Kraft, and K. Walas, "Tell me, what do you see?—interpretable classification of wiring harness branches with deep neural networks," *Sensors*, 2021.

[17] M. Zürn, A. Kienzlen, L. Klingel, A. Lechler, A. Verl, S. Ren, and W. Xu, "Deep learning-based instance segmentation for feature extraction of branched deformable linear objects for robotic manipulation," in *Proc. of CASE*. IEEE, 2023.

[18] M. Zürn, M. Wnuk, A. Lechler, and A. Verl, "Topology matching of branched deformable linear objects," in *Proc. of ICRA*. IEEE, 2023.

[19] M. Zuern, M. Wnuk, A. Schneider, A. Lechler, and A. Verl, "Localization and tracking of deformable linear objects with self organizing maps," in *ISR Europe 2022; 54th Int. Symposium on Robotics*, 2022.

[20] K. Galassi and G. Palli, "Robotic wires manipulation for switchgear cabling and wiring harness manufacturing," in *Proc. of ICPS*. IEEE, 2021.

[21] A. Govoni, G. Laudante, M. Mirto, C. Natale, and S. Pirozzi, "Towards the automation of wire harness manufacturing: A robotic manipulator with sensorized fingers," in *Int. Conf. on Control, Decision and Information Technologies (CoDIT)*, 2023.

[22] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," 2017.

[23] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2017.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, 2014.

[26] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *IEEE/CVF Proc. of CVPR*.

[27] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3d object detection with pointformer," in *IEEE/CVF Proc. of CVPR*, 2021.