2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Towards Visual Classification Under Class Ambiguity

Viktor Kozák[1,2], Jan Mikula[1,2], Lukáš Bertl[1], Karel Košnar[1] and Libor Přeučil[1]

Abstract— **Visual classification under uncertainty is a complex computer vision problem. We present a thorough comparison of several variants of convolutional neural network (CNN) classification techniques in the context of ambiguous image data interpretation. We explore possible improvements in classification accuracy achieved by insertion of prior ambiguity information during the annotation process. This enables us to harness known similarities between individual classes and use them as probability distributions for soft ground-truth labels. We also present an approach based on Bayesian CNNs, offering the possibility of further interpretation of classification results in a problem where the neural network model is often considered as a black box. The presented techniques are verified on a practical spot weld inspection problem.**

I. Introduction

Recent advancements in robotic automation enable the deployment of intelligent robotic manipulators in previously unsolvable tasks. At the same time, the increasing affordability of robotic solutions is an incentive to increase automation in various industries. However, applications such as visual navigation or inspection often have to account for a certain level of uncertainty. This may include uncertainties in the position, size, or shape of the target object, as well as its visual appearance. External factors, such as changing environmental conditions, also have to be taken into account. Thus arises a need for the application of robust and reliable computer vision techniques for object detection and extraction of its underlying characteristics.

In this paper, we focus on visual classification under class ambiguity, using several techniques based on convolutional neural networks (CNNs). These are applied in a complex robotic solution for autonomous spot weld inspection. The system uses visual localization of spot welds for precise robot navigation and CNNs for their subsequent classification. Since the appearance of spot welds differs greatly depending on the used material, the applied heat and its duration, and even the reflection of ambient light, the visual classification problem is especially challenging. One of the main aspects is the gradual change in appearance due to the heat applied during the resistance spot welding process. If the heat is too low, the resulting weld connection may be too weak. Conversely, excessive heat may damage the material. The goal of the applied classification techniques is to correctly assign the spot welds to one of the three main classes (*cold*, *ok*, and *hot*) representing the estimated quality of the spot weld. Additionally, the classifiers are trained to recognize invalid or corrupted spot weld detections.

Since the valid (and even the corrupted) classes share a large number of underlying characteristics, we explore the use of several CNN techniques suitable for the analysis of ambiguous visual data. We study the influence of the insertion of prior ambiguity information during the training process through the use of soft labels. We also present a stochastic approach based on Bayesian CNNs. The Bayesian approach integrates ambiguity directly into the network model, which is beneficial both for the classification task and the subsequent interpretation of the data. The aforementioned techniques are compared with a basic CNN classifier and evaluated in several scenarios.

The rest of this paper is organized as follows. Section II introduces related work. Sections III and IV present the proposed system and the methodology behind it. Section V describes the experimental setup and results, whereas Section VI is devoted to conclusions and future work.

II. Related work

There is increasing demand for the automation of weld inspection tasks, especially in the automotive industry. Traditional destructive testing methods such as tension, bend, and peel tests are often laborious, non-automated, and time and cost consuming. Consequently, reliable quality inspection through non-destructive testing (NDT) is of great importance [1], [2]. Most of the current NDT methods rely on radiography [3], specialized vision systems [4], or additional sensing technology [5], [1]. Thus, vision-based methods are mostly used as a support tool for other approaches.

The use of visual object detection and localization methods for robot navigation in object manipulation and inspection tasks has become increasingly popular in industrial automation [6], [2]. Many of the developed systems rely on CNN based object detectors such as Faster R-CNN [7] or YOLO [8] which are either used to identify areas of interest or directly for robot navigation. While most object detectors provide initial class estimates, these are generally developed towards fast inference and are unsuitable for fine classification. Spot weld detection is directly addressed in [2], which proposed a network model for small object detection based on YOLOv3 [8], and includes a simple binary quality evaluation distinguishing good and bad spot welds.

The work has been supported by the state budget by the Technology Agency of the Czech Republic and the Ministry of Industry and Trade within the programme TREND, project ID: FW03010600, and by the Grant Agency of the Czech Technical University in Prague, grant No. SGS23/071/OHK3/1T/13.

[1]Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague, Jugoslávských partyzánů 1580/3, 160 00 Praha 6, Czech Republic viktor.kozak@cvut.cz

[2]Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Karlovo náměstí 13, 121 35 Praha 2, Czech Republic

A number of works on vision-based methods for weld quality inspection is shape-oriented and performs weld evaluation through semantic segmentation [4], [9] or salient feature extraction [10]. However, these methods are prone to environmental changes since they use a combination of image processing techniques such as thresholding or contour detection, which rely heavily on parameter tuning. As such, it is difficult to use these methods under different conditions or generalize them for a different problem. More data-oriented methods are based on neural networks or the use of support vector machines (SVM) [11], [2], [1], [5]. Such approaches are based on supervised learning and the use of labeled data. Given suitable training data, these methods are typically generalizable to some extent.

Although the number of works on weld inspection is not small, only some of the visual inspection methods address multi-class classification [3], [1], [4]. This is a challenging problem, especially since the appearance of a spot weld can vary extensively and still retain many of the shared underlying characteristics. In this work, we focus on CNN classification [12] techniques suitable for multi-class ambiguous data classification.

Multi-class classification is commonly used with "hard" ground-truth labels using binary values (1 or 0). Furthermore, most applications use one-hot encoded classes that are mutually exclusive, meaning that each instance can be assigned only to a single class. E.g., for a problem with 4 classes, an instance belonging to the second class would be labeled as $[0, 1, 0, 0]$. The use of label smoothing, recently introduced in [13] and further explored in [14], relaxes these restrictions. It is used to prevent overconfidence in the classification task by relaxing the label values with the use of a uniform distribution, and it can improve the classification accuracy. An example of a "smoothed" label belonging to the second class would be $[0.1, 0.7, 0.1, 0.1]$.

We can further expand on this technique using "soft" labels [15] that can be freely defined. Such labels often take the form of a probabilistic distribution, e.g., $[0.2, 0.5, 0.2, 0.1]$. These can be defined either by using prior knowledge about the class distributions and similarities, or by taking additional steps during the ground-truth label acquisition. A commonly used approach is to have the data labeled by multiple evaluators using one-hot labels and use the average values to obtain soft labels [15]. Other approaches go one step further and improve the technique by modeling the behavior (or credibility) of individual evaluators [16]. Soft labels proved to increase the performance of the trained model, however, the additional difficulty in label acquisition has to be considered. Although our work is focused on CNNs, the use of soft labels can also be applied to other classification approaches (such as SVM) to some extent.

Neural networks using point estimates as weights generally perform well with large datasets, but fail to express uncertainties in regions with little data, which leads to overconfident decisions [17]. The model tends to fit well to the training data, but is not as predictive for new data. Taking into account possible uncertainties, a variational inference scheme for neural networks was derived in [18]. This was later extended in [19], introducing *Bayes by Backprop* for feedforward neural networks, which derives a variational approximation to the true posterior. Later, [17] devised a dual application of convolutional operations, one for the mean and one for the variance, introducing Bayesian learning to convolutional neural networks to add a measure of uncertainty and regularization to their predictions.

III. System specification

The presented techniques address the classification problem in a practical task of spot weld inspection by an autonomous robotic arm. The developed system uses visual localization of spot welds for precise robot navigation and convolutional neural networks for spot weld classification and inspection.

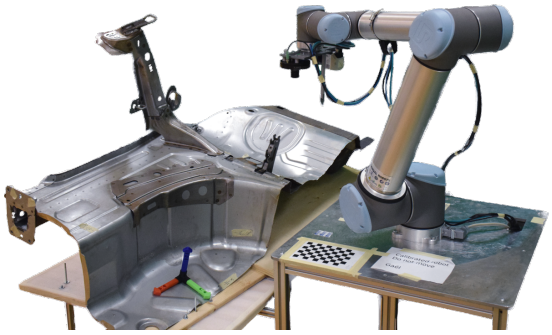

Fig. 1: A laboratory prototype of the robotic workcell for spot weld inspection.

An RGB camera is integrated directly into the robot end effector (Figure 2). This enables us to capture images from a close range with high resolution suitable for visual inspection and precise navigation. We use a daA1280-54uc Basler 2D camera with an onsemi AR0134 color CMOS sensor and a resolution of 1280 × 960 pixels. The lens used in this setup is a Computar M1224-MPW2 with a focal length of 12 mm and an angle of view of 20°. An RL-100W120 SmartView circular light is mounted around the camera to provide direct light from a constant position relative to the camera.

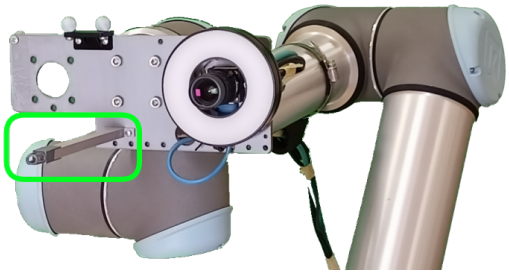

Fig. 2: Laboratory end effector prototype with a plastic model of the ultrasonic probe (marked by a green rectangle).

The final goal of the developed vision system is a precise, safe, and robust navigation of an ultrasonic probe mounted on the robot end effector, which is used for non-destructive testing and inspection of spot welds. This paper covers the

visual navigation and inspection task, omitting the ultrasonic defectoscopy procedure.

IV. Methodology

A. Initialization

Prior to the inspection task, the expected weld positions and their surface normals are extracted from a 3D model of the welded metallic part. We generate a set of desired inspection positions where the z-axis of the camera matches the extracted normal vector at a constant distance ($d = 0.2m$) from the spot weld position.

In order to reduce unnecessary movement, the inspection problem was formulated as the generalized travelling salesman problem (GTSP) and solved with a state-of-the-art generalized large neighborhood search (GLNS) metaheuristic algorithm [20], [21]. Partial trajectories between individual configurations, which are necessary for creating the GTSP input distance matrix, are planned in the robot configuration space by the rapidly exploring random tree (RRT*) algorithm.

B. Visual Navigation

Since the actual position of a spot weld may differ from the expected position from the documentation, it is necessary to apply an additional visual navigation procedure. We use the Faster R-CNN ResNet-50 FPN [7] object detector for precise spot weld localization. The detector was trained on manually labeled spot weld images. It generates a set of bounding boxes containing the detected objects, paired with confidence values ranging from 0 to 1. To prevent false detections, the objects are filtered using an empirically estimated confidence threshold value and the expected size limits of the bounding box. To guarantee an optimal viewing position, only the centermost positioned bounding box is selected for inspection (Figure 3).

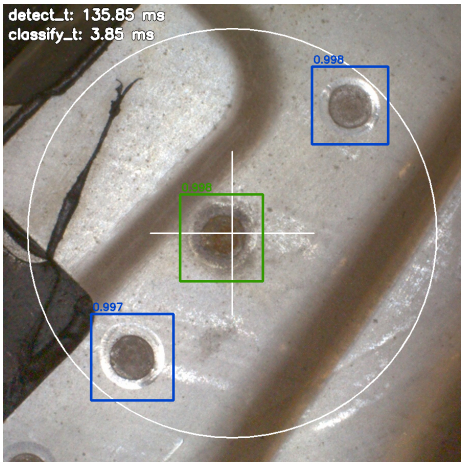

Fig. 3: Spot weld detection.

Next, a precise camera-to-object position is calculated and used for the navigation of the ultrasonic probe for the defectoscopy procedure. The capturing position extracted from the 3D model specification is aligned with the normal vector of the part surface at a fixed distance d from the surface. If we use a calibrated 2D camera, the center of the bounding box can be projected using a pinhole camera model. The object position coordinates x and y relative to the camera are then determined using the following equations:

$$x = \frac{(p_x - c_x)d}{f}, \qquad y = \frac{(p_y - c_y)d}{f}, \tag{1}$$

where d is the distance from the camera to the object, f is the focal length of the camera, c_x and c_y represent the principal point coordinates, and p_x and p_y are the object coordinates in the image plane corresponding to the center of the selected bounding box.

C. Spot Weld Classification

To infer additional information about the detected objects, we extract the image patches generated by the bounding box detector (after the initial filtering phase) and use them as input for an image classification method. The output of the method serves as an evaluation for the visual inspection of the object. Moreover, since object detectors are generally developed towards fast inference, the detection can become unreliable when used with ambiguous objects. As such, undesired objects bearing similarity to the actually desired objects might be detected. The image classification method serves as an additional measure to intercept such detections.

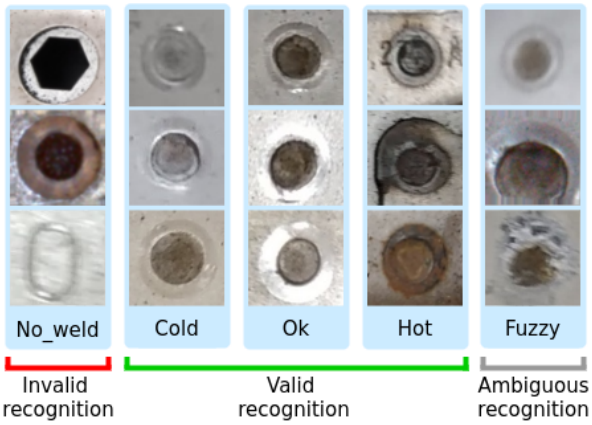

Fig. 4: Defined object classes with example image patches.

Training images for the classification task were obtained using the visual inspection procedure described in Sections IV-A and IV-B. To extract a sufficient number of invalid data, the detection confidence threshold was intentionally lowered prior to the image patch extraction. After initial data evaluation, we have manually defined possible classes for spot weld classification (Figure 4). Three valid classes represent the gradually changing appearance of the spot weld during the welding process. The invalid class contains image patches with incorrect detections. Finally, the ambiguous class contains corrupted images and obscure detections.

In this work, we present three variants of end-to-end CNN models. The implementation is based on the Keras [22] neural network library and all models are trained for the

classification task in a supervised learning fashion through backpropagation. The training data consist of a set of pairs (*X*, *Y*), where *X* is the input of the network (an image patch of the detected object) and *Y* is the output vector. The output has the same dimension as the number of defined classes and each number in the vector represents the estimated probability of the input belonging to each of these classes. For our five-dimensional output vector, the output sequence corresponds to the classes: *no_weld, cold, ok, hot*, and *fuzzy* (l_1, l_2, l_3, l_4, l_5, respectively).

1) Standard CNN: The extracted image patches are resized to 75 × 75 pixels and used as input for the network (Figure 5). The input is processed using four convolutional layers with ReLU activation functions, followed by max-pool layers. This creates the backbone of our CNN model. The last fully connected layer contains five neurons, serving as the classification output.

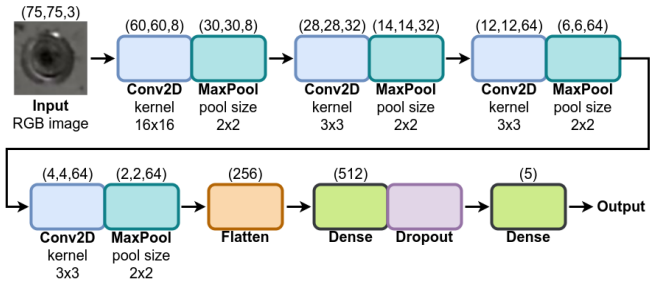

Fig. 5: The structure of the used CNN model.

Standard applications of the CNN classification model use "hard" labels with binary one-hot classes, where each instance can belong only to one category at a time. For example, an instance of an object belonging to the second class (*cold*) would have a label $l_2 = [0, 1, 0, 0, 0]$.

The loss function used for training the CNN is the categorical cross-entropy (CCE) loss used for multi-class classification. The CCE loss is shown in (2), where $\hat{y}$ is the predicted value, y is the ground truth, and n_c is the number of classes.

$$CCE(\hat{y}, y) = -\sum_{i=1}^{\mathrm{n}_c} y_i \log(\hat{y}_i) \tag{2}$$

2) Soft Labels: When working with ambiguous data, it can be argued that one instance of an object can belong to multiple classes. In the same way, the underlying characteristics of an object may occur in multiple classes. In such cases, it is possible (and sometimes desirable) to use soft labels, which can directly propagate the ambiguity into the training data [15].

It can be clearly seen that the valid recognition classes bear similarity. This characteristic can be used when creating labels. For instance, a soft label for the second class l_2 may be defined as $l_2^s = [0, 0.7, 0.2, 0.1, 0]$, which reflects the gradual change in appearance when the weld is burned with increasing intensity. This approach is similar to a commonly used technique called label smoothing [14], [2], used to prevent overconfidence in the classification task. It can be considered as an informed variant of the technique.

CNN models trained with soft labels use the same network model. However, since the CCE loss function rewards output values only in the correct class, it is not suitable for this problem. Therefore, we have decided to use the binary cross-entropy (BCE) function (3), which also penalizes values in other classes. We have also tested the network with the mean squared error (MSE) function (4), commonly used for regression. In our experience, when used with soft labels, models trained with BCE and MSE performed better than those trained with CCE.

$$BCE(\hat{y}, y) = \sum_{i=1}^{n_c} (-y_i \ln(\hat{y}_i) - (1 - y_i) \ln(1 - \hat{y}_i)) \tag{3}$$

$$MSE(\hat{y}, y) = \frac{1}{n_c} \sum_{i=1}^{n_c} (\hat{y}_i - y_i)^2 \tag{4}$$

3) Bayesian CNNs: The output vectors of standard CNNs are deterministic and provide only a single fixed result y. E.g., an output vector y with values [0.01, 0.92, 0.02, 0.02, 0.03] would clearly indicate the classification in the second class, however, it would provide no further information.

Bayesian approach applied to neural networks introduces ambiguity directly in the network model, which results in stochastic behavior [23]. Our Bayesian CNN model differs from traditional CNNs in the first convolutional layer and the last fully connected layer, where weights w_i are generated from normal distributions $N(\mu_i, \sigma_i^2)$. This allows us to obtain an arbitrary number of varying outputs by using a probabilistic weight distribution and running the network multiple times.

Although it is possible to implement a fully Bayesian network, this results in a high increase in the complexity of the model, effectively doubling the number of parameters, yet such approaches often report little change in predictive performance [19]. By applying the Bayesian principle only to the first and last layers, the change in overall complexity is minimal, resulting in an increase of less than 5% of the original number of parameters.

When training the normal distribution values, it is necessary to infer the difference in the prior and posterior kernel and bias parameters. The loss function used for the training of Bayesian CNNs maximizes the likelihood estimate of the mean and the standard deviation of the weights. This is done by minimizing the Kullback-Leibler divergence (KLD) from the true posterior [17].

V. Experiments and Results

This chapter describes the experimental setup, evaluation criteria, and results. Subsection V-A describes the evaluation dataset, together with the experimental setup of individual methods. This includes the necessary input and output definitions. The classification results and their evaluation are presented in Subsection V-B. Subsection V-C offers a discussion on the use of the presented methods and their individual advantages and disadvantages.

A. Experimental Setup

The experiments were carried out using 850 original image patches, each class was represented with an equal number of images (dataset generation is described in Section IV-C). Of these image patches, 70% were used for training and validation data for the neural network classifiers. The other 30% were used to test the trained networks and generate the presented results. Furthermore, the training dataset was augmented to approximately 6380 images by mirroring, rotating, and zooming the images and by contrast adjustment.

All three network models were trained using Adam Optimizer with a learning rate of 0.001, 300 training epochs, and a dropout of 0.2. The training was performed in batches. This means that N different inputs are presented to the network, and the training loss is calculated as the average of individual errors. During each epoch, we randomly split the input data into 30 batches. The calculation of the average loss is shown in (5), where $L(\hat{y}, y)$ represents the given loss function.

$$L_{avg} = \frac{1}{N} \sum_{i=1}^{N} L(\hat{y}, y) \tag{5}$$

Prior to the training, 25% of the training images were set apart as validation data. After each epoch, the accuracy of the trained model is evaluated using the validation data. We designate a patience parameter ($p = 30$) and minimal improvement parameter ($\delta_{min} = 0.002$). If the validation accuracy does not increase by δ_{min} in the last p epochs, the training is stopped and the model values with the highest validation accuracy are saved. This training session is conducted 20 times to obtain the most reliable model, which is selected based on the validation accuracy.

Five possible object classes were defined in Subsection IV-C. Three valid classes (l_2, l_3, l_4) representing spot weld categories, one invalid class (l_1), and one class with corrupted images (l_5). Standard "hard" labels are defined using binary one-hot classes. In contrast, we define soft labels in a way that considers the similarities between the three valid spot weld classes. The proposed values reflect the graudal change in appearance during the welding process, as described in Subsection IV-C.2.

$$\begin{aligned} l_1^s &= [1, 0, 0, 0, 0], \\ l_2^s &= [0, 0.7, 0.2, 0.1, 0], \\ l_3^s &= [0, 0.15, 0.7, 0.15, 0], \\ l_4^s &= [0, 0.1, 0.2, 0.7, 0], \\ l_5^s &= [0, 0, 0, 0, 1]. \end{aligned} \tag{6}$$

Bayesian CNNs allow us to obtain an arbitrary number of varying outputs by using a probabilistic weight distribution and running the network multiple times. In the following experiments, we use 100 forward passes and, unless stated otherwise, we use median values for classification results.

B. Results

1) Direct Class Estimation: Table I presents the classification accuracy for individual classes. We present the *Standard*, *Soft*, and *Bayesian* CNN models trained for all five classes. Additionally, we present the accuracy results for models trained and tested separately on the three valid spot weld classes.

TABLE I: Weld class estimation results. Percent correctness [%].

		No_weld	Cold	Ok	Hot	Fuzzy
Standard	5 classes CCE	75.00	97.95	37.03	83.33	16.66
	3 classes CCE	-	100.00	40.74	83.33	-
Soft	5 classes BCE	77.08	95.91	57.40	83.33	27.08
	5 classes MSE	70.83	95.91	50	83.33	37.50
	3 classes BCE	-	97.95	55.55	85.18	-
	3 classes MSE	-	100.00	38.88	87.03	-
Bayesian	5 classes KLD	70.83	87.75	72.22	83.33	28.00
	3 classes KLD	-	95.91	83.33	87.03	-
Soft Bayesian	5 classes KLD	68.75	95.91	42.59	79.62	22.91
	3 classes KLD	-	95.91	57.40	88.88	-

The *Soft* CNN shows a clear improvement over the *Standard* CNN. While the *Bayesian* CNN shows a higher capacity to distinguish between valid weld classes. We can also observe an increase in the classification accuracy for models trained separately for the valid spot weld classification subproblem. Interestingly, the combination of soft labels with *Bayesian* CNNs did not improve the accuracy of the prediction.

TABLE II: Resulting estimations for individual classes. Bayesian CNN results according to the median. [%]

	Class predictions				
	No_weld	Cold	Ok	Hot	Fuzzy
No_weld	70.8	6.3	6.3	2.1	14.6
Cold	2.0	87.8	6.1	2.0	2.0
Ok	0.0	22.2	72.2	0.0	5.6
Hot	0.0	3.7	3.7	83.3	9.3
Fuzzy	16.0	47.0	2.0	7.0	28.0

Table II shows a confusion matrix with the resulting distribution of the classification results for individual classes (using the *Bayesian* CNN model). The *fuzzy* class clearly has the highest misclassification rate. The image patches assigned to this class contain corrupted weld images. Although these are unreliable for robot navigation, the image patches still retain a large amount of valid weld characteristics, which explains the high level of misclassification.

2) Weld Validity Estimation: We can join estimates clearly indicating the presence of a weld {cold, ok, hot} and invalid estimates {no_weld, fuzzy}, changing the problem to binary classification. Additionally, we present the accuracy for the *Standard* and *Bayesian* models, trained directly for a binary (valid/invalid) spot weld classification. The results of the weld validity classifications can be observed in Table III in the form of the confusion matrix values. Note that the use of soft labels would lose its meaning in binary classification.

The CNN models trained for binary classification clearly outperform most of the 5 class models on this task. As such,

TABLE III: Number of valid and invalid estimations.

		True positive	False positive	True negative	False negative
Binary	Standard	144	18	78	13
	Bayesian	145	28	68	12
5 class	Standard	143	43	53	14
	Soft (BCE)	149	35	61	8
	Bayesian	147	36	60	10
	Soft Bayesian	146	28	68	11

the optimal solution to the problem would be to use a binary model to filter out invalid detections, and a 3 class model for subsequent spot weld classification. Although, this would double the computational complexity. We can also see that soft labels improve the performance of both the *Standard* and *Bayesian* CNNs. This is possibly due to the relaxation of loss penalties between the valid classes.

3) Ambiguity Exploitation: Standard CNN output vectors are deterministic and provide only a single, often overconfident, result for each instance. By using a Bayesian network with a probabilistic weight distribution and running the network multiple times, we can obtain an arbitrary number of varying outputs. This gives us additional information in the form of probability distributions.

Let us use an example shown in Figure 6. The resulting median-based classification of the object is in the *cold* class. However, we can see a significant response in the *fuzzy* class to which it was originally assigned during manual labeling. This indicates that the object instance has characteristics of both classes. To take further advantage of the Bayesian CNN output, we can filter out classes with uncertain predictions for manual evaluation. This could be done, for example, by using an ambiguity threshold (e.g., $t = 0.15$). The resulting probability distributions can then serve as a support tool during evaluation.

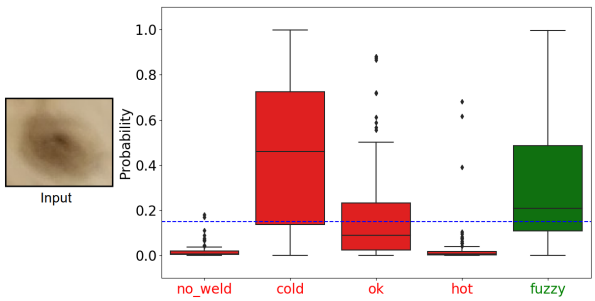

Fig. 6: Results of 100 forward passes through the stochastic 5 class Bayesian CNN for a single image patch input. The resulting distributions are projected over individual classes using a five-point summary. The ambiguity threshold is depicted by a blue line.

Although it is also possible to use the ambiguity threshold to automatically reduce the number of false positive classifications without manual reclassification, this comes at the cost of omitting a certain number of valid (true positive) detections. Conversely, the same approach could be used to reduce false negative classifications.

C. Discussion

Based on the presented results, we can conclude several conjectures.

When used for image classification, the performance of convolutional neural networks can be improved by using well-defined soft labels. However, this is conditioned by the suitability of the data classes. The use of soft labels also requires additional effort during the labeling phase. The soft labels used in our application could be easily defined based on the visual transition of the object caused by the increase in heat during the welding process. This allowed us to map the original one-hot labels to the predefined soft labels. However, such a method may not be suitable for different datasets. In other works, the data is often labeled manually by multiple evaluators using one-hot labels and the final soft labels are generated by averaging the label values [15] or posprocessed by modeling the reliability of individual evaluators [16].

The use of Bayesian CNNs is advantageous for a deeper understanding of the trained model, the distribution of data, and the classification results. As such, these are suitable for ambiguous data classification. Bayesian CNNs do not require additional effort during the training phase. However, multiple forward passes are necessary to extract statistically relevant data. The Bayesian CNN model used in our application is relatively small and one forward pass takes on average *20 ms* on a CPU, as such running the network multiple times is still feasible. Additionally, the number of forward passes can be freely adjusted depending on the application.

Lastly, although it is a well known fact, the accuracy of the predictions can improve if the classification domain is reduced. As proven with the reduced 3 class models for valid spot weld classification and the binary models for the spot weld validity classification.

VI. Conclusions

In this paper, we have elaborated on several variants of CNN classification techniques in the context of ambiguous data analysis. We have demonstrated that harnessing prior information regarding known class similarities in the form of soft ground-truth labels can increase the classification accuracy of a CNN model. We have also presented a stochastic approach based on Bayesian CNNs, which introduces known uncertainties directly in the network model. The Bayesian approach proved to be suitable for ambiguous data classification. It also offers the possibility to further interpret the classification results. The presented techniques were applied in a practical robotic application for spot weld inspection and tested on a custom dataset. However, it is possible to generalize the techniques to other problems.

In future work, we intend to expand the dataset and apply unsupervised machine learning techniques for semi-autonomous data labeling. Furthermore, we plan to explore the use of multi-label CNN classification techniques.

References

[1] D. Younes, E. Alghannam, Y. Tan, and H. Lu, "Enhancement in quality estimation of resistance spot welding using vision system and fuzzy support vector machine," *Symmetry*, vol. 12, no. 8, 2020.

[2] W. Dai, D. Li, D. Tang, Q. Jiang, D. Wang, H. Wang, and Y. Peng, "Deep learning assisted vision inspection of resistance spot welds," *Journal of Manufacturing Processes*, vol. 62, pp. 262–274, 2021.
[3] M. S. H. Nizam, A. R. M. Zamzuri, S. Marizan, and S. A. Zaki, "Vision based identification and classification of weld defects in welding environments: A review," *Indian journal of science and technology*, vol. 9, pp. 1 –15, 2016.
[4] Y. Ou and Y. F. Li, "Quality evaluation and automatic classification in resistance spot welding by analyzing the weld image on metal bands by computer vision," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, pp. 301–314, 2015.
[5] Z. Zhang, G. Wen, and S. Chen, "Weld image deep learning-based online defects detection using convolutional neural networks for al alloy in robotic arc welding," *Journal of Manufacturing Processes*, vol. 45, pp. 208–216, 2019.
[6] V. Kozák, R. Sushkov, M. Kulich, and L. Přeučil, "Data-driven object pose estimation in a practical bin-picking application," *Sensors*, vol. 21, no. 18, 2021.
[7] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2015.
[8] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *ArXiv*, vol. abs/1804.02767, 2018.
[9] T. Zheng, Y. Yang, P. Zheng, L. Benz, and L. Wang, "An appearance inspection method for resistance spot welding based on semantic segmentation," *IOP Conference Series: Materials Science and Engineering*, vol. 790, 2020.
[10] S. Ye, Z. Guo, P. Zheng, L. Wang, and C. Lin, "A vision inspection system for the defects of resistance spot welding based on neural network," in *Computer Vision Systems*. Cham: Springer International Publishing, 2017, pp. 161–168.
[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
[12] R. Yamashita, M. Nishio, R. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, 2018.
[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.
[14] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?" in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.
[15] A. M. Aung and J. Whitehill, "Harnessing label uncertainty to improve modeling: An application to student engagement recognition," in *2018 13th IEEE International Conference on Automatic Face Gesture Recognition*, 2018, pp. 166–170.
[16] M. Y. Guan, V. Gulshan, A. M. Dai, and G. E. Hinton, "Who said what: Modeling individual labelers improves classification," *ArXiv*, vol. abs/1703.08774, 2018.
[17] K. Shridhar, F. Laumann, and M. Liwicki, "A comprehensive guide to bayesian convolutional neural network with variational inference," *ArXiv*, vol. abs/1901.02731, 2019.
[18] A. Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems*, vol. 24. Curran Associates, Inc., 2011.
[19] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, vol. 37, 2015, p. 1613–1622.
[20] S. Smith and F. Imeson, "GLNS: an effective large neighborhood search heuristic for the generalized traveling salesman problem," *Computers & Operations Research*, vol. 87, 05 2017.
[21] D. Woller and M. Kulich, "Path planning algorithm ensuring accurate localization of radiation sources," *Applied Intelligence*, pp. 1–23, jan 2022.
[22] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras
[23] H. Wang and D.-Y. Yeung, "A survey on bayesian deep learning," *ACM Computing Surveys*, vol. 53, no. 5, 2020.