Automated Defect Report Generation for Enhanced Industrial Quality Control

Jiayuan Xie[1], Zhiping Zhou[2], Zihan Wu[2], Xinting Zhang[4], Jiexin Wang[2,3], Yi Cai[2,3*], Qing Li[1]

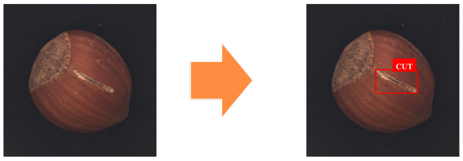
[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR, China
[2]School of Software Engineering, South China University of Technology, Guangzhou, China
[3]Key Laboratory of Big Data and Intelligent Robot (SCUT), MOE of China
[4] Department of Mathematics, The University of Hong Kong, Hong Kong SAR, China
jiayuan.xie@polyu.edu.hk, ycai@scut.edu.cn

Abstract

Defect detection is a pivotal aspect ensuring product quality and production efficiency in industrial manufacturing. Existing studies on defect detection predominantly focus on locating defects through bounding boxes and classifying defect types. However, their methods can only provide limited information and fail to meet the requirements for further processing after detecting defects. To this end, we propose a novel task called defect detection report generation, which aims to provide more comprehensive and informative insights into detected defects in the form of text reports. For this task, we propose some new datasets, which contain 16 different materials and each defect contains a detailed report of human constructs. In addition, we propose a knowledge-aware report generation model as a baseline for future research, which aims to incorporate additional knowledge to generate detailed analysis and subsequent processing related to defects in images. By constructing defect report datasets and proposing corresponding baselines, we chart new directions for future research and practical applications of this task.

Introduction

Industrial defect detection aims to automatically identify and locate defects or anomalies in products of different materials, e.g., glass or steel, etc (Zhang et al. 2023a). These defects or anomalies may include surface defects, cuts, cracks, deformations, size deviations, and other issues that could impact the quality and safety of the materials. Teaching a machine to automatically detect defects has become an emerging task in the computer vision area due to its vast potential applications in industrial production scenarios (Zhang et al. 2023b; Lan and Huang 2023). Specifically, effective defect detection contributes to enhancing monitoring and control of product quality on the production line, which plays a crucial role in ensuring the products meet quality standards and reducing the rate of defective items.



Defect Type: Cut.

Report (easy): There is a long sized cut-type defect in the mid-mid position of the hazelnut.

Report (hard): There is a long sized cut-type defect in the mid-mid position of the hazelnut. Hazelnuts are an important crop and are widely used in various food products.
EFFECTs: **1.** Hazelnuts that have been cut can be contaminated with various substances, such as pesticides. These contaminants can pose a health hazard to consumers if not detected properly. **2.** Defective hazelnuts can impact product quality, taste, and texture. If undetected, lower quality hazelnuts can end up in the final product, compromising its overall quality.
SOLUTIONs: **1.** Implement strict quality control measures to ensure only the highest quality hazelnuts are used in production. This can include manual inspection processes, using advanced technology such as optical sorting machines, or a combination of both. **2.** Provide thorough training to the employees responsible for handling and processing hazelnuts. Teach them how to identify and properly handle defective hazelnuts, and emphasize the importance of quality control at all stages of the manufacturing process. **3.** Regularly maintain and calibrate all machinery used in processing hazelnuts. This will help ensure that the equipment is functioning optimally.

Figure 1: A case of defect detection in Hazelnut. The defect results are displayed in two forms of bounding box and text report, in which the report includes the effect and solution.

Existing studies (Yang et al. 2023; Qiu, Wu, and Yu 2019; Di et al. 2019) predominantly divide defect detection into two consecutive subtasks, i.e., localization followed by classification. Defect localization refers to identifying the position of the defects within an image and framing it with bounding boxes. Defect classification involves matching the defect in the framed bounding box with known candidate defect types to determine the type of defect, e.g., cracks, scratches, and blemishes, etc. Despite these methods are capable of locating and classifying defects, they can only provide limited information about the defects. This makes it challenging to assist producers in gaining a comprehensive understanding of the defect situation and devising solutions. To overcome this limitation, we are inspired by research in natural language processing areas such as medical report generation (Kisilev et al. 2015; Liu et al. 2021a) and attempt to describe the defects in the form of text. Specifically, text can provide detailed descriptions and explanations of defects, offering a more intuitive expression of the causes or solutions to the defects. As shown in Figure 1, the text (i.e., Report (hard)) contains two main contents, i.e., defect description of a hazelnut and comprehensive information of

[*]Corresponding author

"Cut" defect type. Defect description describes the basic information of the location and type of the defect, while comprehensive information encompasses specific descriptions of effects, their underlying causes, or proposed solutions.

In this paper, we propose a novel task, i.e., defect detection report generation (DDRG), which aims to express defect detection results in the format of report text. To the best of our knowledge, no comparable dataset exists for this DDRG task. As a first step in further research to fill this gap and motivate the development of defect detection report generation methods, we manually construct new defect report datasets for the DDRG task. In detail, the construction of datasets involved 16 different materials, e.g., steel, wood, or hazelnut. Each constructed report incorporates defect characteristics and additional knowledge sources (e.g., Wikipedia) of comprehensive information as references for report annotations. The constructed datasets could facilitate training and thorough evaluation of such generation methods. As depicted in Figure 1, based on the differing content contained within the reports, the constructed datasets can be categorized into two classes, i.e., $Easy$ report dataset and $Hard$ report dataset. The $Easy$ report dataset includes solely defect descriptions, while the $Hard$ report dataset additionally incorporates comprehensive information. Both of these datasets contribute to evaluating some defect detection capabilities of the methods. Specifically, the $Easy$ report dataset highlights the traditional abilities of defect detection models, i.e., localization and classification. In contrast, the $Hard$ report dataset goes beyond these traditional capabilities and evaluate the model's ability to acquire additional knowledge to generate comprehensive information.

Different from existing image-to-text tasks such as image captioning (Hirota, Nakashima, and Garcia 2023), the DDRG task has two unique characteristics: i) **Objective.** The DDRG task primarily concentrates on describing and analyzing specific defects present in the image, rather than focusing on depicting the whole image of these image description tasks. ii) **Additional Knowledge.** The DDRG task may involve domain-specific background knowledge to accurately describe defects and solutions, which goes beyond the visual content of image descriptions. To this end, we propose a model targeting the above two characteristics, named knowledge-aware report generation (KRG) model, which is intended to serve as a baseline for future methods. Our model contains four modules, multimodal feature extractor, knowledge retriever, knowledge-aware encoder and decoder. To facilitate the description focusing on the defects in an image, multimodal feature extractor utilizes pre-trained model VisualBERT (Li et al. 2020a) to align the image features and defect type words, which enables image features to incorporate defect information. To acquire the knowledge related to the defect, the knowledge retriever utilizes large-scale language models as enhancers to retrieve the additional knowledge related to all candidate defect types in the material. Considering that not all additional knowledge is valid, the knowledge-aware encoder is designed to extract and encode target knowledge related to the defects in images. Finally, the image features and extracted knowledge are used in the decoder for report generation.

To summarize, our contributions are as follows:

- We propose a novel defect detection report generation (DDRG) task, which enables a model to learn to express defect information in text format in more detail and intuitively.
- We construct new datasets that can be used to evaluate three capabilities of defect detection models, i.e., i) the ability to describe defect description; ii) the ability to integrate additional knowledge; iii) the ability to resolve the zero-shot situation. Specifically, the materials in production are diverse and impossible to be comprehensively covered due to the labor-intensive annotation process. Therefore, in this paper, we try to annotate a wide variety of materials as much as possible, in order to construct a suitable dataset that drives the model towards achieving zero-shot capability.
- We propose two characteristics of DDRG and propose a corresponding model named knowledge-aware report generation, which enables at least 38% improvement in BLEU-4 over the baselines and can be used as a future benchmark. Experimental results show that our model design meets the requirements of two characteristics to a certain extent, i.e., capture valid knowledge for report generation.

Related Work

The DDGR task aims to generate reports according to the defect detection results, which is a combination task of defect detection and image description.

Defect Detection Existing methods on defect detection can be divided into supervised learning, unsupervised learning, and semi-supervised learning. Most of the supervised learning methods make use of the variants of the CNN. Qiu et al. (2019) propose a method composed of 3 stages, and a fully convolutional network (FCN) is used in the segmentation stage. Hu et al. (2020) utilize Faster R-CNN to detect surface defects on printed circuit boards (PCBs), demonstrating its suitability for PCB manufacturing quality control. Shang et al. (2023) propose a DAT-Net that enhances surface defect detection using a defect-aware Transformer network that can efficiently model long-range dependencies. Semi-supervised methods are also widely used in defect detection. Both Li et al. (2019) and He et al. (2019) use semi-supervised methods to tackle the surface defect classification of steels, since most of the steel surfaces are unlabeled. Both Hou et al. (2021) and Gong et al. (2019) adopt an encoder-decoder structure for reconstruction. Besides, some studies attempt to model the multi-class distribution via unsupervised learning, such as You et al. (2022) propose a model called UniAD, under this framework the anomaly detection for multiple classes is accomplished. These methods can solely provide defect location and type, but fail to provide richer information like text format.

Image Captioning Image captioning aims to describe images in the form of meaningful text. Yao et al. (2018) and Yang et al. (2019) explore combining semantic information within the embedding. Yang et al. (2019) propose a variant of the self-attention operator for image captioning. Liu et al. (2021b) apply the self-attention operators on the image patches. Language model component in image captioning aims at predicting a sequence of words with the given

	steel	bottle	cable	capsule	carpet	grid	hazelnut	leather	metal	pill	screw	tile	toothbrush	transistor	wood	zipper	sum
Original	1800	63	92	109	89	57	70	92	93	141	119	84	30	40	60	119	3058
Easy	927	22	56	17	40	47	32	38	46	59	44	34	24	16	55	52	1509
Hard	1543	53	85	76	78	56	56	79	74	112	100	60	30	33	60	107	2602
Len	98	148	166	146	151	175	142	149	165	162	147	148	176	145	176	158	-

Table 1: The number of deduplicated reports included in the datasets and the length of hard reports under 16 different materials.

embedding (Zhou et al. 2020). Li et al. (2020b) and Wang et al. (2023) propose utilizing pre-trained models to embed the tokens for this task. Luo et al. (2023) propose a novel paradigm that leverages semantic prior from cross-modal retrieval to guide Diffusion Transformers in a cascaded manner, yielding improved vision-language alignment and linguistic coherence. These methods can provide a description of the entire image but often ignore details related to defects.

Dataset Construction

Existing datasets for industrial defect detection tasks are insufficient to support the evaluation of the defect detection report generation (DDRG) task. The reasons are as follows: i) **Inadequate for Textual Demands.** Defect reports often necessitate textual descriptions for defects of materials, while image bounding boxes and classifications alone may not fulfill these textual requirements. ii) **Lack of Comprehensive Information.** Merely providing location bounding boxes and classifications may not offer sufficiently detailed comprehensive information of defect, failing to adequately convey the effects, causes, and solutions of defects. However, constructing a dataset suitable for the DDRG task from scratch is labor-intensive.

Therefore, we propose to perform secondary processing on existing datasets based on the defect detection task. In a dataset we constructed, the defect report in each sample mainly contains two parts, i.e., defect description and comprehensive information.

Defect Description Collection

Defect description refers to a detailed description of the detected defect, including its location, defect type or characteristics. To construct the descriptions, we design a series of templates based on the box locations and classifications of defects annotated in existing datasets, e.g., “There is a xxx sized xxx-type defect in the xxx position of the xxx (material like the pill).”. Through this approach, we can reduce the manual effort required for defect description construction.

Comprehensive Information Collection

Comprehensive information contains more complex content related to the defect, e.g., the effect, cause or solution of the defect. Different from defect description collection, comprehensive information collection requires more human involvement rather than being generated by rule templates. Specifically, we build a comprehensive information annotation web page that displays images with bounding boxes of defects and text boxes below for entering comprehensive information like the effects of the defect. We first ask a group of trained annotators to write down unique referring comprehensive information of the defects based on given image, and then another group of trained annotators checks the content quality and makes appropriate revisions if necessary. To ensure the quality of annotation, the annotators in the first group are researchers with professional industrial production or material background knowledge, and they combine a large amount of additional knowledge such as Wikipedia during the annotation process. The annotators in the second group are front-line production personnel in the manufacturing industry. Following the labeling standards for the defect detection series datasets (Bao et al. 2021; Bergmann et al. 2019), we specify some labeling requirements: i) The cause and solution need to be combined with the image background and defect characteristics. For example, for the same “Cut” defect type, we need to generate consistent content based on the specific material in the image. ii) The comprehensive information does not necessarily have to include complete effects, causes, and solutions. We should choose parts with high confidence as much as possible.

There are many existing datasets containing different materials that can be used for this secondary processing. For the material selection in dataset construction, we have two requirements, i.e., important or common. Thus, we select to construct NEU-r and MVTec-r datasets based on the NEU (Bao et al. 2021) and MVTec (Bergmann et al. 2019) datasets, respectively. Specifically, the NEU dataset is about the defect detection of steel, which is one of the most important materials in industrial manufacturing; While the MVTec dataset involves 15 of the most common materials in everyday life, such as leather and wood. In addition, to gradually verify whether the model has the ability of traditional defect detection and incorporating additional knowledge, we divide the two constructed datasets into two types, $Easy$ and $Hard$, respectively. $Easy$ signifies that the report solely includes defect descriptions, whereas $Hard$ encompasses additional comprehensive information.

Data Diversity Analysis

Since defects may vary across images, the reports in our constructed dataset should be as diverse as possible. Therefore, we performed repeated statistics on the constructed data set, and the results are shown in Table 1. The “Original” indicates the number of images each material contained in the original dataset, which is also the number of our constructed reports. We find that $Easy$ report dataset has a large number of repeated reports, with a total repetition rate of about 50%. This is due to the fact that defect descriptions are generated based on rule templates. The $Hard$ report dataset has

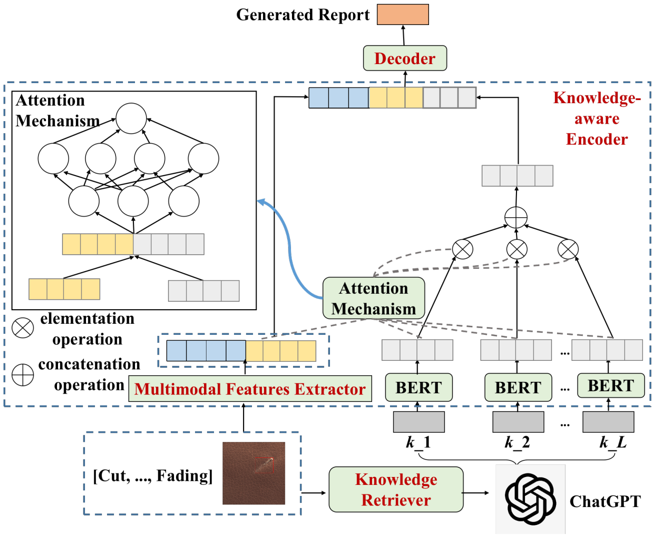


Figure 2: Overview of our model KRG. The four modules are highlighted in red font.

a lower repetition rate of about 14%. This is because the artificially constructed content containing comprehensive information has diversity according to defects or content combinations. And the "Len" represents the average length of each $Hard$ report under different materials. According to this analysis, we consider that the constructed $Hard$ report dataset is diverse in report expressions.

Benchmark

In this DDRG task, given an image I in industrial manufacturing and the L candidate defect types $T = [t_1, ..., t_L]$, our goal is to generate a report R related to the defect. We conduct a thorough evaluation of multiple mainstream methods for image description as an initial benchmark on our datasets. Specifically, the benchmarks contain the neural network model GRNN (Mostafazadeh et al. 2016), and the pretrained models CLIP (Radford et al. 2021) and VisualBERT (Li et al. 2020a). We show that while each method can detect certain types of defects and generate their defect descriptions, none of the evaluation methods excels at generating comprehensive information. This is due to the fact that additional knowledge related to defects usually needs to be integrated when generating comprehensive information. To this end, we design a new baseline model based on VisualBERT, which is intended to serve as a baseline for future methods.

Proposed Method

We propose a knowledge-aware report generation (KRG) model and the overall of our proposed model can be seen in Figure 2. It consists of four modules: (i) Multimodal features extractor, which aims to extract aligned visual features of image I and textual features of defect types T. (ii) Knowledge Retriever, which aims to retrieve knowledge related to the defects of specific material from a knowledge base. (iii) Knowledge-aware encoder, which aims to combine the image features and defect-related knowledge to obtain the embedding vector. (iv) Decoder module, which aims to generate a report based on the output of the knowledge-aware encoder. Details of each part of our framework are presented in the following sections.

Multimodal Features Extractor Unlike the image captioning task, the DDRG task aims to describe the part of the defect in an image. Thus, we employ the VisualBERT to capture the rich semantics in this multimodal information of given images I and defect types T. The VisualBERT integrates the BERT (Devlin et al. 2019) and the ResNet (He et al. 2016) to process the defect type words and the image patches. Specifically, we divide the input images into numerous overlapping segments and apply ResNet for feature extraction on each segment. The extracted features then serve as representations for their respective regions. We opt for BERT and ResNet as the feature extractors owing to their robust performance in text and image analysis domains, proficiently capturing and representing high-level features inherent in both texts and images.

Inputs of defect type word and the image patches features are jointly processed by multiple Transformer layers. The rich interactions between words and patches enable the model to capture complex relationships among them. Furthermore, the model is allowed to implicitly discover useful alignments between image patches and candidate defect types, and then obtain new defect type representations $h^t = \{h_i^t\}_{i=1}^{L}$ and image patches representations $h^v = \{h_i^v\}_{i=1}^{K}$, where K represents the image is divided into K patches.

Knowledge Retriever Defect descriptions in reports can be obtained with visual information, while other comprehensive information is difficult. The generation of comprehensive information usually requires additional knowledge, which can provide the cause or solution ideas related to the defect. Thus, we design a knowledge retriever to retrieve additional knowledge related to each candidate defect type.

With the development of large-scale language models (LLMs), LLMs as enhancers can provide more accurate and comprehensive relevant knowledge than traditional knowledge bases e.g., ConceptNet (Speer, Chin, and Havasi 2017). Following the study of Brown et al. (2020), we utilize ChatGPT with prompt engineering to generate additional knowledge for each candidate defect type, which contains EFFECT, CAUSE and SOLUTION. Specifically, the designed template is as follows, "There are xxx (defect type) on the xxx (material), will this have any effect?". The prompt template introduces the defect type and its product material in detail, and then raises a related question about effect. Finally, we can retrieve relevant knowledge about all defect types corresponding to each material.

Knowledge-aware Encoder For the same material, LLMs may retrieve a series of similar knowledge. However, not all knowledge is contributable, and effective knowledge needs to be consistent with specific details (e.g., type, size, etc.) of defects present in different images. Thus, we designed a knowledge-aware encoder to extract and encode defect-related knowledge. Specifically, the module utilizes a BERT to tokenize each retrieved knowledge into individual sub-

Dataset	Type	Model	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR	ROUGE	CIDEr	Acc
NEU-r	Easy	GRNN	40.20	38.08	35.83	23.42	30.39	56.69	2.59	46.08
		CLIP	**54.37**	47.48	**39.93**	**30.68**	39.57	**60.00**	3.13	48.52
		VisualBERT	50.83	**49.67**	38.48	30.24	**45.32**	58.47	**4.54**	**49.77**
	Hard	GRNN	20.16	12.87	8.06	6.99	13.16	26.17	1.24	47.47
		CLIP	20.55	18.69	10.41	11.01	19.92	26.80	1.53	55.47
		VisualBERT	29.70	18.89	16.83	9.82	25.95	38.16	3.66	65.54
		KRG	**35.31**	**21.70**	**25.27**	**16.88**	**36.83**	**45.91**	**4.12**	**70.75**
	Zero-shot	GRNN	18.16	10.87	7.06	2.91	13.16	26.17	1.23	0
		CLIP	19.90	13.66	9.76	3.43	11.21	26.88	2.30	6.40
		VisualBERT	21.92	14.35	10.43	7.11	23.74	33.94	2.66	7.50
		KRG	**28.31**	**17.71**	**13.61**	**9.42**	**29.08**	**37.96**	**3.19**	**10.25**
MVTec-r	Easy	GRNN	40.40	37.54	29.13	26.32	21.20	48.11	3.02	48.27
		CLIP	50.83	**44.11**	**36.98**	29.07	**30.90**	**52.49**	**3.80**	48.72
		VisualBERT	**51.36**	41.22	31.09	**29.64**	28.10	52.48	3.01	**52.31**
	Hard	GRNN	19.57	26.47	15.95	13.45	14.13	27.89	1.12	50.84
		CLIP	27.26	23.04	18.53	13.27	18.40	23.28	1.05	55.01
		VisualBERT	26.40	24.66	20.71	14.02	17.31	25.94	1.34	57.84
		KRG	**46.51**	**31.70**	**23.98**	**19.43**	**26.93**	**32.38**	**3.85**	**61.65**
	Zero-shot	GRNN	17.09	12.64	9.67	7.97	15.16	20.25	1.22	0
		CLIP	17.26	13.04	10.53	9.46	12.24	21.1	1.05	10.21
		VisualBERT	14.96	13.10	11.50	9.98	16.52	21.16	1.10	13.65
		KRG	**32.60**	**23.67**	**15.82**	**11.26**	**22.30**	**25.86**	**2.54**	**23.65**

Table 2: Main automatic metrics results of baselines and our model. Bold: the best performance in the column for each type.

word tokens using WordPiece tokenization and convert the tokenized input into corresponding word embeddings,

$$h^k_{i,j} = BERT(k_{i,j}), \tag{1}$$

where $k_{i,j}$ represents the j-th word in the knowledge corresponding to the i-th defect type.

Then, a mean pooling over the word embeddings is implemented to obtain a fixed-size representation for the i-th knowledge, i.e.,

$$\bar{h}^k_i = \sum_{j=0}^{M} h^k_{i,j}, \tag{2}$$

where $\bar{h}^k_i$ represents the the i-th knowledge representation and M means that the i-th knowledge contains M words.

Similarly, we also use the mean pooling layer to obtain the representation of the image and candidate defect types,

$$\bar{h}^v = \sum_{i=0}^{K} h^v_i, \tag{3}$$

$$\bar{h}^t = \sum_{p=0}^{L} BERT(t_p), \tag{4}$$

where $\bar{h}^v$ denotes the whole image representation, while $\bar{h}^t$ represents the representation of all candidate defect types. Additionally, K and L mean the image contains K patches and L candidate defect types, respectively.

For a given image $\bar{h}^v$ and their candidate knowledge representations $\bar{h}^k$, the attention mechanism applies a fully-connected layer and the softmax function to calculate the normalized weights for each defect,

$$s_i = \frac{exp(W_s(\bar{h}^v \oplus \bar{h}^k_i) + b_s)}{\sum_{m=0}^{L} exp(W_s(\bar{h}^v \oplus \bar{h}^k_m) + b_s)}, \tag{5}$$

where s_i denotes the weight of the i-th knowledge, W_s and b_s are learned parameters and $\oplus$ denotes the concatenation operation. The additional knowledge $\tilde{h}^k$ can be calculated as the weighted sum of L knowledge representations,

$$\tilde{h}^k = \sum_{i=0}^{L} s_i \bar{h}^k_i. \tag{6}$$

Decoder Module We use an LSTM (Hochreiter and Schmidhuber 1997) as the decoder to generate a report. Then we initialize the decoder state with the image features $\bar{h}^v$ through Equation (3),

$$s_0 = (h_0, c_0) = \bar{h}^v. \tag{7}$$

In each decoder step, the decoder focuses on different image patches. Thus, we set up a dynamic mechanism to force the decoder to focus on different patches of the image, i.e.,

$$l_{t,i} = \frac{exp(W_q(h_t \oplus \bar{h}^v_i) + b_q)}{\sum_{c=0}^{K} exp(W_q(h_t \oplus \bar{h}^v_c) + b_q)}, \tag{8}$$

$$v_t = \sum_{i=0}^{K} l_{t,i} h^v_i, \tag{9}$$

where h_t represents the hidden state at the step t-1, and v_t represents the image representation at step t.

Subsequently, the decoder module leverages the image feature v_t, knowledge representation $\tilde{h}^k$, defect type $\bar{h}^t$, and the last generated word embedding $\widehat{r}_{t-1}$ to generate the current hidden state,

$$s_t = \mathrm{LSTM}(v_t, \tilde{h}^k, \bar{h}^t, \widehat{r}_{t-1}, s_{t-1}). \tag{10}$$

Building upon the approach presented in (Ma et al. 2018), we train a fully connected layer on s_t and subsequently employ a softmax activation to derive the distribution of word probabilities at the step t,

$$\widehat{r}_t = \mathrm{softmax}(W_y s_t + b_y). \tag{11}$$

Experiment

Baseline Models

(i) **GRNN** (Mostafazadeh et al. 2016) is a baseline model. It utilizes GRU to decode image features obtained by VGGNet. (ii) **CLIP** is a model we devised, building upon the foundation laid by Radford et al. (2021). This model integrates the characteristics of text and images by jointly encoding defect type words and visual features, yielding a cohesive cross-modal representation (iii) **VisualBERT** is based on the work of Li et al. (2020a). The model jointly encodes the defect types and patches in the image to obtain the cross-modal representation.

Experimental Details

We implement all models in Pytorch and train them with two P100 GPUs. We divide each image into $K = 196$ patches for CLIP and VisualBERT. The decoder employs 350 hidden units, and dropout layers with a dropout probability of P_{drop}=0.4. During the training process, we fine-tune the model's performance by minimizing the cross-entropy loss function. This optimization is achieved using the gradient descent algorithm with the Adam optimizer (Kingma and Ba 2015), initialized with a learning rate of 0.0001.

The division and usage details of datasets are as follows. The NEU-r dataset comprises solely steel material. For constructing the zero-shot scenario, we train the model using the MVTec-r dataset and then perform predictions on the NEU-r test set. The MVTec-r dataset contains 15 different materials. We shuffle the data of different materials for training and prediction. For the zero-shot scenario, we use the NEU-r dataset for training and the MVTec-r test set for prediction.

Evaluation

Automatic Evaluation Metrics. We measure the performance of the models used to generate questions with five metrics: BLEU (1 to 4) (Papineni et al. 2002), ROUGE$_L$ (Lin 2004), METEOR (Denkowski and Lavie 2014) and CIDEr (Vedantam, Zitnick, and Parikh 2015), which are standard evaluation metrics for natural language generation.

Human Evaluation Criteria. To further refine our assessment, we also invited 5 volunteers with good English and industrial manufacturing education to conduct manual evaluation (Xing et al. 2017). Specifically, we randomly select 50 samples of the same number in each model, and then evaluate them according to the following metrics: Fluency (**F**) mainly reflects the fluency of generated report sentences, as well as whether there are grammatical errors and unknown words (UNKs). Defect description (**D**) indicates whether the generated results can accurately reflect information such as the size, location and type of defects. Comprehensive Information (**C**) indicates whether the generated results can accurately reflect comprehensive information such as the effects, causes or solutions of defects.

Material	**Type**	**Model**	**F**	**D**	**C**
NEU-r	**Easy**	**GRNN**	2.94	0.53	-
		CLIP	3.10	0.61	-
		VisualBERT	**3.15**	**0.63**	-
	Hard	**GRNN**	2.42	0.56	0.38
		CLIP	2.89	0.61	0.27
		VisualBERT	2.95	0.65	0.31
		KRG	**3.26**	**0.67**	**0.58**
		Human	4.70	0.98	0.87
MVTec-r	**Easy**	**GRNN**	3.03	0.42	-
		CLIP	2.97	**0.52**	-
		VisualBERT	**3.10**	0.49	-
	Hard	**GRNN**	2.20	0.47	0.22
		CLIP	2.58	0.56	0.30
		VisualBERT	2.42	0.52	0.19
		KRG	**2.97**	**0.63**	**0.49**
		Human	4.86	0.93	0.83

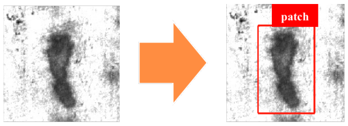
Table 3: The human evaluation results.

Flu takes values from 0 to 5 (Higher values represent higher fluency), while F and C take a binary value (1 or 0).

Results and Analysis

Automatic Evaluation Results Table 2 shows the automatic evaluation results of baselines. The results are divided into two parts, i.e., evaluating the textual quality of the generated results and evaluating the accuracy of defect type classification. For the first part, we have several findings:

- ***Easy*** **report dataset.** i) Compared to $Hard$ report dataset, we find that the three baselines achieve more competitive results on the $Easy$ report dataset. This is because $Easy$ report dataset only contains short text descriptions of defects and is constructed through rule templates, so models can effectively learn relevant information and template formats. ii) The CLIP and VisualBERT models are slightly better than the GRNN model. This is because the CLIP and VisualBERT models introduce candidate defect types as inputs, which have a certain guiding effect on the positioning and classification of defect types.
- ***Hard*** **report dataset.** i) For $Hard$ report dataset, three baselines fail to perform satisfactorily. This is due to the difficulty in obtaining comprehensive information solely from images. Importantly, it presents a challenging task for defect detection. ii) Our model KRG outperforms three baselines in various metrics. For example, KRG improves BLEU 4 by 53% and 38% compared with the best baseline on the NEU-r and MVTec-r datasets, respectively. This indicates that our model can effectively capture additional



Report (Human): There is a presence of small patch defects located at the center of the image. **CAUSEs**: 1. The defects in the patches could lead to localized stress concentration, resulting in a reduction in the strength and toughness of the strip steel, potentially leading to fractures or damage during usage. 2. The defects in the patches could impact the surface quality of the strip steel, making processing more challenging and decreasing overall production efficiency. 3. The defects in the patches could give rise to depressions or protrusions on the surface of the strip steel, thereby diminishing the product's visual appeal and affecting its competitiveness in the market. **SOLUTIONs**: 1. Strengthening the quality inspection of raw materials to ensure their freedom from significant defects; implementing effective treatment of slag to prevent the inclusion of impurities into the strip steel. 2. Exercising proper process control over parameters such as hot rolling temperature, reduction amount, and cooling method to minimize the likelihood of surface defects in strip steel.

Report (Ours): there is a defect of small patches located at the mid-mid of the image. it can cause some problems such as: 1. patch defects lead to localized stress concentration reducing the strength and toughness of the strip steel which may result in fracture or damage during use; 2. patch defects form depressions or protrusions on the surface of the strip steel reducing the appearance quality of the product and affecting its market competitiveness. and some solutions such as: 1. strengthen quality inspection of raw materials to ensure that they are free from significant defects additionally effective treatment of slag should be carried out to prevent inclusion of impurities from entering the strip steel; 2. control the hot rolling temperature reduction amount cooling method and other parameters through proper process control to reduce the possibility of surface defects in strip steel; 3. adopt high-precision non-destructive testing techniques such as ultrasonic testing x-ray testing etc.

Report (CLIP): there is a defect of small patches located at the midmid of the image there is a defect of small patches located at the midlower of the image it can cause some problems such as decreased strength patch defects lead to localized stress concentration reducing the strength.

Figure 3: Case study of sample output report generated by humans, our model KRG, and CLIP.

knowledge related to comprehensive information to a certain extent, and utilize it for report generation.

- **Zero-shot.** Annotating defects in different materials typically incurs significant manual labor costs, while industrial production often involves a diverse range of materials that require detection. Therefore, we also attempted to assess the zero-shot capability of each model. We find that our model KRG is not affected by the dataset partition by utilizing additional knowledge, so our model can still provide some effective information to generate results. But overall all models perform poorly in this situation, which also suggests an important direction for future research.

For the second part, we have several findings:

- **Accuracy.** i) We observe that all models yielded moderate results in defect category determination. This is due to the fact that the generative approach tends to perform comparatively less effectively than supervised classification methods. ii) The accuracy rate obtained by the $Hard$ report dataset is higher than that of the $Easy$ report dataset, which indicates that the generation of comprehensive information will also have a certain impact on the content of the defect description. iii) Both perform badly in the zero-shot situation. There are different candidate defect types for each material, and a GRNN model would fail to generate correct results without such information.

Human Evaluation Results To provide a clearer depiction of the efficacy of human evaluations, we initially compute the Fleiss' Kappa Coefficients for each criterion. We find that the results are high (i.e., greater than 0.4), which indicates that our human evaluations are reliable. Table 3 shows the human evaluation results. We find that:

- First, the fluency of the results for the $Easy$ type is significantly higher compared to the $Hard$ type. This is attributed to the longer length of the Hard type dataset, which presents a challenge commonly encountered in natural language generation tasks.
- Second, for metric **D**, the results for the $Hard$ type exhibit a slightly better performance than the $Easy$ type. This indicates that the generation of comprehensive information to some extent prompts the model to produce more accurate defect descriptions.
- Third, the baseline models perform poorly in metric **C**, which is consistent with the results for the $Hard$ report dataset in automated evaluation. Our designed model manages to capture relevant additional knowledge to some extent, thereby outperforming the baselines in metric **C**.
- Fourth, the results constructed by humans performed well on the three indicators, which shows that the quality of the datasets we constructed are recognized by multiple groups of annotators and evaluators.

Case Study

Figure 3 shows defect detection report generated from the human-constructed, CLIP, and our model KRG. We find that (i) the model has some fluency problems in the process of generating long text, such as CLIP will generate repeated words. (ii) The baseline CLIP excels in generating defect descriptions, whereas the model exhibits minimal capability to generate comprehensive information. (iii) Our model KRG is able to generate a report that include the effect and solution of defect. This indicates that the knowledge integration method of our model can capture and utilize effective additional knowledge for report generation to a certain extent.

Conclusion

In this paper, we propose a new task called defect detection report generation, which can describe defects in detail and give additional information in the form of text. Existing datasets on defect detection only contain defect location boxes and defect categories. For this task, we first propose corresponding new datasets, which contain 16 different materials. Specifically, each report in the constructed dataset contains two parts, i.e., defect description and comprehensive information. In addition, we design a knowledge-aware report generation model. Our model extracts relevant knowledge from large-scale models according to candidate defect types, and then focuses on the parts related to image content for generation. Experiments show that the DDRG task we proposed is difficult to be solved by existing methods. Our model provides a way to incorporate additional knowledge that can serve as a baseline for future research.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (62076100), Fundamental Research Funds for the Central Universities, SCUT (x2rjD2230080), the Science and Technology Planning Project of Guangdong Province (2020B0101100002), CAAI-Huawei MindSpore Open Fund, CCF-Zhipu AI Large Model Fund, and the Hong Kong Research Grants Council under the General Research Fund (project no.: 15200023).

References

Bao, Y.; Song, K.; Liu, J.; Wang, Y.; Yan, Y.; Yu, H.; and Li, X. 2021. Triplet-Graph Reasoning Network for Few-Shot Metal Generic Surface Defect Segmentation. *IEEE Trans. Instrum. Meas.*, 70: 1–11.

Bergmann, P.; Fauser, M.; Sattlegger, D.; and Steger, C. 2019. MVTec AD - A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *Proc. of CVPR*, 9592–9600. Computer Vision Foundation / IEEE.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Proc. of NeurIPS*.

Denkowski, M. J.; and Lavie, A. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proc. of WMT@ACL*, 376–380.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proc. of NAACL*, 4171–4186. Association for Computational Linguistics.

Di, H.; Ke, X.; Peng, Z.; and Dongdong, Z. 2019. Surface defect classification of steels with a new semi-supervised learning method. *Optics and Lasers in Engineering*, 117: 40–48.

Gong, D.; Liu, L.; Le, V.; Saha, B.; Mansour, M. R.; Venkatesh, S.; and van den Hengel, A. 2019. Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection. In *Proc. of ICCV*, 1705–1714. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proc. of CVPR*, 770–778. IEEE Computer Society.

He, Y.; Song, K.; Dong, H.; and Yan, Y. 2019. Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network. *Optics and Lasers in Engineering*, 122: 294–302.

Hirota, Y.; Nakashima, Y.; and Garcia, N. 2023. Model-Agnostic Gender Debiased Image Captioning. In *Proc. of CVPR*, 15191–15200.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*, 9(8): 1735–1780.

Hou, J.; Zhang, Y.; Zhong, Q.; Xie, D.; Pu, S.; and Zhou, H. 2021. Divide-and-Assemble: Learning Block-wise Memory for Unsupervised Anomaly Detection. In *Proc. of ICCV*, 8771–8780. IEEE.

Hu, B.; and Wang, J. 2020. Detection of PCB Surface Defects With Improved Faster-RCNN and Feature Pyramid Network. *IEEE Access*, 8: 108335–108345.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR*.

Kisilev, P.; Walach, E.; Barkan, E.; Ophir, B.; Alpert, S.; and Hashoul, S. Y. 2015. From medical image to automatic medical report generation. *IBM Journal of Research and Development*, 59(2/3): 2–1.

Lan, Y.; and Huang, C. 2023. A Deep Learning Based End-to-end Surface Defect Detection Method for Industrial Scenes. In *Proc. of BIC*, 45–49. ACM.

Li, L. H.; You, H.; Wang, Z.; Zareian, A.; Chang, S.; and Chang, K. 2020a. Weakly-supervised VisualBERT: Pre-training without Parallel Images and Captions. *CoRR*, abs/2010.12831.

Li, X.; Yin, X.; Li, C.; Zhang, P.; Hu, X.; Zhang, L.; Wang, L.; Hu, H.; Dong, L.; Wei, F.; Choi, Y.; and Gao, J. 2020b. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *Proc. of ECCV*, volume 12375 of *Lecture Notes in Computer Science*, 121–137. Springer.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Liu, F.; You, C.; Wu, X.; Ge, S.; Sun, X.; et al. 2021a. Auto-encoding knowledge graph for unsupervised medical report generation. *Advances in Neural Information Processing Systems*, 34: 16266–16279.

Liu, W.; Chen, S.; Guo, L.; Zhu, X.; and Liu, J. 2021b. CPTR: Full Transformer Network for Image Captioning. *CoRR*, abs/2101.10804.

Luo, J.; Li, Y.; Pan, Y.; Yao, T.; Feng, J.; Chao, H.; and Mei, T. 2023. Semantic-conditional diffusion networks for image captioning. In *Proc. of CVPR*, 23359–23368.

Ma, S.; Sun, X.; Li, W.; Li, S.; Li, W.; and Ren, X. 2018. Query and output: Generating words by querying distributed word representations for paraphrase generation. In *Proc. of NAACL-HLT*, 196–206.

Mostafazadeh, N.; Misra, I.; Devlin, J.; Mitchell, M.; He, X.; and Vanderwende, L. 2016. Generating Natural Questions About an Image. In *Proc. of ACL*, 1802–1813.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL*, 311–318.

Qiu, L.; Wu, X.; and Yu, Z. 2019. A High-Efficiency Fully Convolutional Networks for Pixel-Wise Surface Defect Detection. *IEEE Access*, 7: 15884–15893.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proc. of ICML*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.

Shang, H.; Sun, C.; Liu, J.; Chen, X.; and Yan, R. 2023. Defect-aware transformer network for intelligent visual surface defect detection. *Advanced Engineering Informatics*, 55: 101882.

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In Singh, S.; and Markovitch, S., eds., *Proc. of AAAI*, 4444–4451. AAAI Press.

Vedantam, R.; Zitnick, C. L.; and Parikh, D. 2015. CIDEr: Consensus-based image description evaluation. In *Proc. of CVPR*, 4566–4575.

Wang, N.; Xie, J.; Wu, J.; Jia, M.; and Li, L. 2023. Controllable image captioning via prompting. In *Proc. of AAAI*, volume 37, 2617–2625.

Xing, C.; Wu, W.; Wu, Y.; Liu, J.; Huang, Y.; Zhou, M.; and Ma, W. 2017. Topic Aware Neural Response Generation. In *Proc. of AAAI*, 3351–3357.

Yang, L.; Xu, S.; Fan, J.; Li, E.; and Liu, Y. 2023. A pixel-level deep segmentation network for automatic defect detection. *Expert Syst. Appl.*, 215: 119388.

Yang, X.; Tang, K.; Zhang, H.; and Cai, J. 2019. Auto-Encoding Scene Graphs for Image Captioning. In *Proc. of CVPR*, 10685–10694. Computer Vision Foundation / IEEE.

Yang, X.; Zhang, H.; and Cai, J. 2019. Learning to collocate neural modules for image captioning. In *Proc. of CVPR*, 4250–4260.

Yao, T.; Pan, Y.; Li, Y.; and Mei, T. 2018. Exploring Visual Relationship for Image Captioning. In *Proc. of ECCV*, volume 11218 of *Lecture Notes in Computer Science*, 711–727. Springer.

You, Z.; Cui, L.; Shen, Y.; Yang, K.; Lu, X.; Zheng, Y.; and Le, X. 2022. A Unified Model for Multi-class Anomaly Detection. In *Proc. of NeurIPS*.

Zhang, D.; Hao, X.; Wang, D.; Qin, C.; Zhao, B.; Liang, L.; and Liu, W. 2023a. An efficient lightweight convolutional neural network for industrial surface defect detection. *Artif. Intell. Rev.*, 56(9): 10651–10677.

Zhang, H.; Wang, D.; Chen, Z.; and Pan, R. 2023b. Adaptive visual detection of industrial product defects. *PeerJ Comput. Sci.*, 9: e1264.

Zhou, L.; Palangi, H.; Zhang, L.; Hu, H.; Corso, J. J.; and Gao, J. 2020. Unified Vision-Language Pre-Training for Image Captioning and VQA. In *Proc. of AAAI*, 13041–13049. AAAI Press.