Generating Synthetic Training Images to Detect Split Defects in Stamped Components

Aru Ranjan Singh, Thomas Bashfod-Rogers, Sumit Hazra, and Kurt Debattista.

***Abstract*—Detecting rare and costly defects, such as necks and splits in sheet metal stamping, remains challenging for deep learning models due to low failure rates entailing few available samples to train on. Synthetic images provide a simulated alternative; however, the two main current approaches have limitations for generating split defect images. Image synthesis-based models generate implausible training data, while physics-based models are computationally expensive and lack the diversity required. To address this, we present a novel method combining the advantages of physics-based simulation with synthetic-based defect generation. The method first generates deformed 3D geometry through finite element simulation with plausible split locations determined using a forming limit curve. Subsequently, the fine details of captured real splits are mapped to the identified locations to generate realistic defect features. Our results show that training a deep neural network with the addition of synthetic images improves the performance significantly.**

***Index Terms*—Defect detection, Synthetic data, Finite element simulation, Industrial inspection, Sheet metal stamping.**

I. INTRODUCTION

The manufacturing industry constantly attempts to improve production rates without compromising product quality. Sheet metal stamping is a mass-production process widely used for a range of products, from white goods to automotive and aerospace body manufacturing. It can produce complex shapes at tens of parts per minute. However, like all manufacturing processes, it is susceptible to surface defects that reduce the quality of the components. Ghosh [1] provided a comprehensive list of defects that can occur during the stamping process. Necking and splitting are the most critical defects in stamped parts and are caused when the plastic deformation in the material exceeds its forming limits, resulting in a local thickness reduction or a through-thickness fracture. These defects are critical as they represent one of the most common types of defects encountered in stamping operations. Moreover, these defects cannot be reworked and must be scrapped, further highlighting the significance of their detection in sheet metal stamping. Since the visual characteristics of the neck and small split defects are similar, for simplicity, we shall refer to both defects as split defects for the remainder of the text. Human visual examination is currently the prevalent method for identifying these defects in stamped parts. However, splits are subtle in appearance, so there is a high risk that human inspectors pass them on for further processing. This increases the overall expense associated with rejecting these components at later stages.

Recently, machine vision-based approaches have shown potential for automating and replacing human inspection in several industrial applications [2]–[4]. In particular, these techniques work well when processed using *deep learning* (DL) algorithms. For example, Mangat et al. [2] automated human and robot picking and placing tasks, Cannizzaro et al. [3] developed an in-situ camera-based defect detection and monitoring system in additive manufacturing, Yu and Liu [5] detect wafer map defects in integrated circuits and [4] implemented DL for 6D pose estimation. Such machine vision-based detection systems are flexible, non-contact and can achieve high detection accuracy.

A crucial requirement for utilising DL to process image data is the generation of a sufficiently large training dataset. This is particularly problematic for defects which do not occur frequently but have a high impact on the manufacturing process when they occur. For example, splits in sheet metal stamping are rare because production processes are designed to avoid their occurrence. This happens when there is an unexpected variation in the material properties or processing conditions (e.g. changes to lubrication). As a result, they may affect between 1-5 per cent of components, but they can not be repaired and thus scraped. As a result, not sufficient split defect components for DL model training. To address this issue, few-shot training methods are usually used. Few-shot learning refers to approaches that train models with limited examples. Two main approaches to few-shot training include pretraining a model on similar data, and using data augmentation [6].

Pretraining involves training a model on a larger dataset that contains related or similar objects. By learning from this broader dataset, the model can capture general features and patterns that are applicable to the target defect, in this case, split defects. The pre-trained model can then be fine-tuned with the limited available data of split defects, enhancing its ability to classify them. However, most existing defect datasets do not adequately represent split defects, stamping environments' lighting and reflection conditions.

Another approach is to utilise data augmentations. Given the scarcity of split defect samples, traditional data augmentation methods are restricted to limited training data. In comparison, learning-based synthetic images fail to generate noise-less images or replicate the training data. Moreover, the generated synthetic images are restricted to the training data domain [7]. In contrast, computer graphics-based synthetic images allow control over the class specification, objects' location, lighting and synthetic image style.

Our work proposes a solution to this manufacturing problem through a novel image synthesis process which combines the strengths of physically principled defect location estimation and computer graphics-based photorealistic image generation of split defect for sheet metal stamping. In particular, a deformed 3D geometry and element-wise strain distribution are estimated from finite element method (FEM) simulation, from which defect locations are computed using *forming limit curve* (FLC). This leads to 3D geometries with plausible yet randomly distributed split defect locations. Finer details are then applied to the geometry using a novel mapping function combined with a dictionary of crack displacement images to generate a dataset of plausible splits on parts. This is then used to create photorealistic training data for split defect detection.

The proposed method is evaluated for split defect detection using several DL-based detectors. Our results indicate that using synthetic training images improves the model performance across all DL-based detectors.

Manuscript received July 16, 2023; revised September 22, 2023; accepted October 19, 2023. Date of publication xxx xx, xxxx; This work was supported in part by the WMG-IIT PhD programme. Paper no. TII-23-2655. (Corresponding author: Aru Ranjan Singh)

The authors are with the Visualisation Lab, Warwick Manufacturing Group, University of Warwick, CV4 7AL Coventry, U.K. (e-mail: aru-ranjan.singh@warwick.ac.uk; thomas.bashford-rogers@warwick.ac.uk; sumit.hazra@warwick.ac.uk; k.debattista@warwick.ac.uk).

Additionally, the proposed framework is compared with few-shot methods. Since no work in literature used diffusion model based method for defect image generation this study finetune stable diffusion model on our dataset.

To summarise, the main contributions of this work are:

- A framework for generating the datasets of rare and costly defects essential for the automated inspection of stamped metal parts. This novel approach leverages computer graphics techniques to create physically accurate and photorealistic split defect datasets in sheet metal stamping components.
- A novel approach combines FEM and FLC to accurately compute split defect locations. This approach enables us to generate diverse split defect locations while maintaining physical correctness, thereby enhancing the realism of the synthetic dataset.
- A method for creating diverse split defects using a limited set of captured defect textures, enhancing the visual variety of the split defects in the synthetic dataset.
- An extensive evaluation using several deep learning-based detectors. Additionally, a comparative study of our approach against other few-shot methods. This comparison highlights the superior performance of our method in detecting rare defects on stamped metal parts.

II. Related Work

This section presents the related work, including DL-based defect detection in sheet metal stamping, synthetic training images in industrial inspection and computer graphics-based synthetic crack image generation.

A. DL in Stamping Defect Detection

DL models have been widely used in industrial defect detection tasks, such as detecting rail surface defects [8], [9], weld defects [10], and surface defects on metal parts [11]. However, the majority of prior studies on metal parts have utilised flat-rolled steel components, and there has been limited research on the use of non-uniform components, such as sheet metal stamping [12]–[14]. Block et al. [12] demonstrated the detection and tracking of imprint defects on stamped components. Singh et al. [13], [14] used a dataset containing representative components for split defect detection and used CNN based models to detect defects. These studies highlight that producing defective stamping components for DL model training is challenging and expensive. For DL-based defect detection in industrial applications, it is necessary to collect data that may appear in the entire lifecycle of a manufacturing product [15]. This indicates the importance of computer graphics-based synthetic data generation for DL defect detection, where the defect type and class can be controlled and generated efficiently.

B. Synthetic Training Data in Industrial Inspection

The studies on image synthesis methods used to train DL models can primarily be divided into two groups: learning-based models [3], [16], and classical computer graphics-based methods [17]. Although modern state-of-the-art learning-based models can create convincing synthetic images, the models require significant training data. Siu et al. [17] argued that generative models are susceptible to convergence problems and require additional training data. Additionally, Zhao et al. [7] discussed that learning-based models tend to replicate the training data when the dataset size is small, leading to overfitting of models. Moreover, these models generate images with similar characteristics to the training images resulting in rare improvement in the diversity [15].

In contrast, computer graphics-based synthetic images allow control over the class specification, objects' location, and synthetic image style. Studies have used computer graphics to produce synthetic training images for object detection tasks [18], including industrial object detection [2], [19]. Mangat et al. [2] generated synthetic images for an object-picking task without considering realistic appearance, whereas, Li et al. [19] used synthetic images for industrial bin packing from small cluttered parts. However, relatively little prior research has employed synthetic datasets for surface defect detection on metal parts, such as castings [20], boiler pipes [21], and flat steel slabs [22]. Moreover, no prior study has attempted to employ synthetic datasets in stamping defect detection.

C. Computer Graphics-based Synthetic Training Data for Crack Defect Detection

Two primary approaches for crack modelling are physics-based and synthetic-based. Physics-based models involve simulating physical phenomena in a virtual environment based on known material properties and process parameters. FEM is the most common approach for realistic crack synthesis in computer graphics [23] and is extensively used in sheet metal stamping during product development and process optimisation [24] together with forming limit curves [25]. FLC is a ubiquitous split defect criterion in the sheet metal stamping process [25]. However, physics-based models lack in generating diverse defects required in the training dataset, as replicating the noise in material and stamping processes is difficult. Moreover, physics-based models are computationally expensive.

In contrast, synthetic-based models map a procedurally created or example-based image of a split known as a texture onto a 3D geometry to change the appearance of the surface without considering physical factors. This process of mapping a 2D image texture to a 3D surface is broadly known as texture mapping.

All the existing studies used synthetic-based models to create synthetic training data for split defects. Kondarattsev et al. [21] and Siu et al. [17] apply real defect textures to a cylindrical 3D CAD model for realistic defect generation on boiler pipes and sewer pipes, similarly, Zhai et al. [10], and Boikov et al. [22] procedurally generated defect textures on random locations of flat 3D models. These approaches focus on simplified geometries, such as planes or cylinders, however, real parts have more complicated geometries and therefore exhibit a non-uniform distribution of cracks over the surface which existing methods cannot represent. In contrast, DL models perform greatly well when the test and train samples have the same distribution [9]. Therefore, combining the advantages of physics-based and synthetic-based approaches, our method overcomes this limitation.

III. Motivation

The motivation for our hybrid method, which combines image texturing-based approaches and physics-based approaches, is rooted in removing the limitations of existing techniques for generating computer graphics-based synthetic datasets for split defect detection. Previous studies have employed two primary approaches for split/crack defect modelling: physics-based models and texturing methods (discussed in Section II).

Physics-based models provide a high level of physical accuracy in simulating defects. However, no prior study has used physics-based models for generating split defect training datasets, primarily due to the following reasons: 1) The resolution of the mesh in the FEM simulation needs to be extremely high to compute the required details, leading to high memory and computational costs, 2) Replicating noise

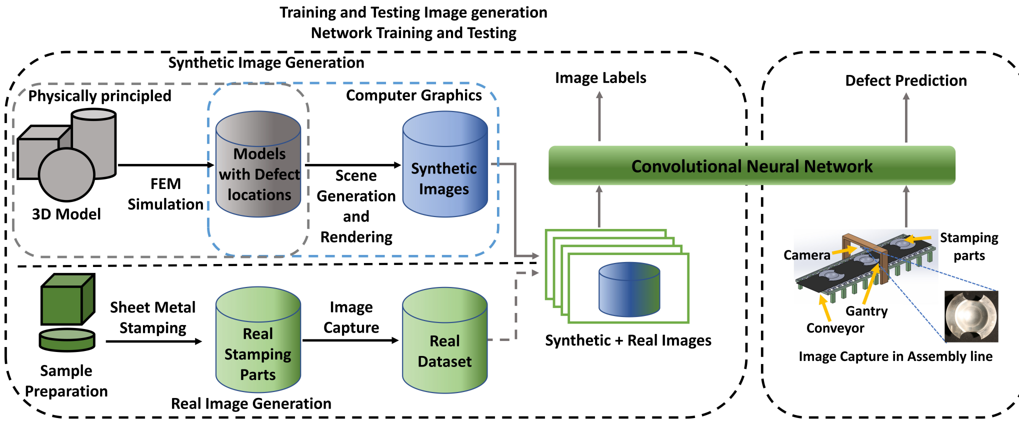

Fig. 1: Pipeline for stamping defect detection using synthetic images. The left box shows model training and implementation in the production line shown on the right. During training the model fuse synthetic data (above the dotted horizontal line) and real data (below the dotted horizontal line) for optimal training depending on real data availability. The synthetic data are generated in two stages: physically principled 3D models and defect location generation (grey dotted box) and photorealistic image generation using computer graphics (blue dotted box).

in real-world processes and material parameters is difficult, resulting in limited variations of defects in the datasets.

On the other hand, texturing methods have been widely used to generate synthetic training datasets for defect detection. These methods are computationally efficient and produce visually realistic defects. However, they often overlook the physical accuracy of the components and fail to capture the non-uniform distribution of defects observed in real parts. As a result, the generated datasets do not fully represent the complexity and diversity of real-world processes in sheet metal stamping.

To overcome these limitations, our study proposes a novel hybrid technique that combines physics-based and texturing methods. By integrating the strengths of both methods, we aim to generate a synthetic dataset that exhibits physically accurate defect features, visually realistic images, and a diverse representation of defect locations. This approach enhances the realism and accuracy of the generated dataset and improves the performance of deep learning models in split defect detection on sheet metal stamping components.

The challenge of producing real defective stamping components for validation requires a component that visually represents splits in real-world stamping parts and can be easily produced in a controlled environment. To meet this requirement, we validated our proposed method using Nakajima samples [26]. These samples are utilised in the limiting dome height test, which is a standardised test (ISO112004-2:2008) designed to evaluate the formability of sheet metal materials and simulate the complex loading conditions that occur during stamping. The open die tool used in the test allows for the monitoring of crack appearance and severity during the stamping process, unlike the fully closed tooling used in production. Because of the simulative nature of the limiting dome height process, the appearance of the splits and necks in the Nakajima sample is visually representative of those in production components, making it a suitable tool for validating the proposed method.

IV. Synthesising Stamping Images

In this section, the proposed image synthesis pipeline is introduced in detail. The overall pipeline of the proposed method is shown in Fig. 1. Depending on the availability of real images, the pipeline can fuse synthetic and real data for optimal model training. The synthetic image generation pipeline is divided into two stages. Stage 1, the physically principled method used to obtain plausible defect location, is shown in the grey dotted box in Fig. 1. Stage 2, the computer graphics method used to generate a photorealistic defective dataset, is shown in blue dotted box in Fig. 1. The details of stages 1 and 2 are described in Sections IV-A and IV-B, respectively.

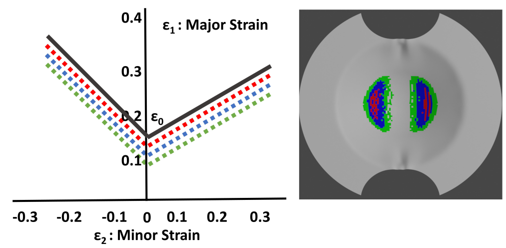

Fig. 2: Illustrates the parameterised FLC and corresponding defective elements on a simulated Nakajima sample. The solid black line represents the standard FLC $(a, p = 1)$. When a is set to the safety margin and the weight parameter $p = 1$, the FLC and corresponding defective elements are highlighted in red. As p is decreased, such that $p_{min} < p < 1$, the FLC and the newly assigned defective elements are shown in blue, and for $p = p_{min}$ highlighted in green.

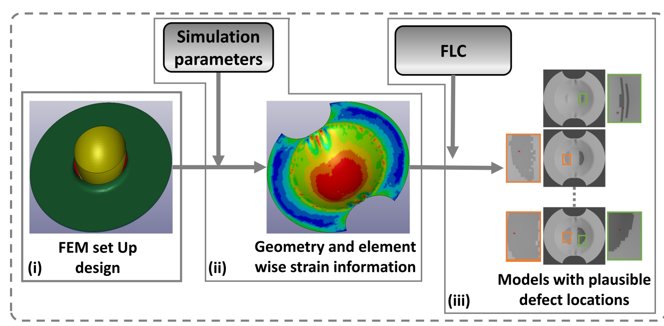

Fig. 3: Illustrates the steps of physical principled models and defect locations generation. (i) Creating CAD models and arranging for the stamping simulation, (ii) FEM simulation, and (iii) the FLC-based algorithm used to generate 3D models with diverse yet random defect locations.

A. A Novel Application of FEM and FLC for 3D Model Generation with Physically Principled Defect Locations

In sheet metal stamping, the components are designed to undergo deformation within a permissible threshold to avoid defects. FLC is a common defect criterion used during the stamping process design, which is a map of two principal strains that indicates the threshold

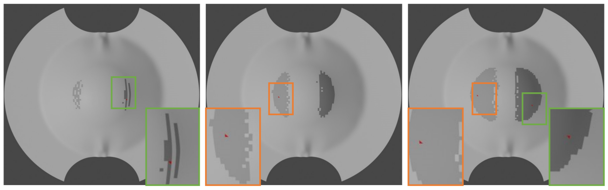

Fig. 4: Shows the deformed geometry with highlighted defective elements for three different values of parameter p: $p = 1$, $p_{min} < p < 1$, and $p = p_{min}$, in reading order (line 13). The shades of grey represent different defect clusters (line 14). The boxes show the clusters selected for defect appearance in the final image (line 15). Finally, the enlarged box highlights the randomly picked element per cluster that serves as the origin of the split (line 16).

TABLE I: Algorithm to determine the defect locations.

Input:	3D geometry and principal strains $\epsilon(\epsilon_1, \epsilon_2)$ for all elements E from FEM simulation
Results:	Geometries with Random defect locations
For every synthetic image	
1:	$p \leftarrow U(b, 1)$
2:	Initialise a dictionary for defective elements D_d
3:	**for** *each element* e **in** E **do**
4:	**if** $\epsilon_2 > 0$ **do**
5:	**if** $\epsilon_1 \geq a * p * \epsilon_0 + s_r * \epsilon_2$ **do**
6:	$D_d \leftarrow e$
7:	**end**
8:	**else**
9:	**if** $\epsilon_1 \geq a * p * \epsilon_0 - s_l * \epsilon_2$ **do**
10:	$D_d \leftarrow e$
11:	**end**
12:	**end**
13:	**end**
14:	Cluster D_d using *DBSCAN* and remove outliers
15:	One or more cluster picked randomly
16:	Randomly pick an element as the start of defect

for splitting. The FLC failure condition can be expressed as follow:

$$\epsilon_1 > flc(\epsilon_2) \tag{1}$$

Where $(\epsilon_1, \epsilon_2) \in \mathbb{R}^2$ are the major and minor strains, respectively and $flc(\cdot)$ is the function representing the FLC of the material. A typical FLC is shown in a solid black line in Fig. 2.

The FLC is a function of material properties such as strain hardening coefficient, strain path and thickness. However, uncertainties in the material and stamping processes cause the splits to originate during the stamping at various locations below the threshold and propagate, resulting in a variety of split defects at various locations in different samples for the same component. To replicate it in our synthetic dataset, this subsection aims to create multiple 3D models for a single component that feature diverse yet realistic defect locations, while utilising minimal computation cost. To achieve this, we propose a series of steps, as depicted in Fig. 4. The first step involves creating CAD models of the components and the stamping tools. These models are then used in a FEM simulation, which is performed on a per-part basis. The outcome of the simulation is a deformed 3D model with element-wise strain distribution. Finally, using an FLC-based algorithm, multiple 3D models with diverse and realistic defect locations are generated from a single FEM simulation. The algorithm is described in detail in Table I.

The proposed algorithm assumes that for numerous stamping of the same part, the uncertainty is uniformly distributed throughout the material. Hence the likelihood of a defect appearing in an element increases depending on how close the strain distribution is to the threshold of the defect criteria. Given the above assumption, we modified the FLC function by introducing extra parameters as $flc^p(a, p, \epsilon_0, s_l, s_r, \epsilon_2)$ to incorporate the uncertainty into the simulation. The detailed implementation of the FLC is outlined in lines 3-13 in the algorithm. Here, ϵ_0 is defined as the y-intercept of FLC and is calculated from strain hardening coefficient n and thickness of material t using formula $\epsilon_0 = \frac{n}{0.21}(23.3 + 14.3t)$ proposed by Keller and Brazier [27]. s_l and s_r are the slopes of the threshold lines on the left and right sides from the y-intercept. Finally, a is a hyperparameter that accounts for the safety factor, and random variable p, sampled from a uniform distribution $p \sim U(p_{min}, 1)$, is used to weight the elements being defined as defective based on their proximity to the threshold. For example, at a equal to the safety factor and $p = 1$ (i.e p_1) the elements assigned as defective visually can be seen in red colour in Fig. 2. However, as p reduced to $p_{min} < p < 1$, both red and blue elements are assigned as defective, as depicted in Fig. 2. Thus, elements that are defined as defective at higher values of p or with a strain distribution closer to the threshold have a higher probability of being assigned as the origin of the defect. The extreme uncertainty that can appear in the material is represented by p_{min}, which can be obtained either from pilot runs or by approximating an experimental FLC.

Further in the algorithm DBSCAN clustering was employed to cluster the defect regions (line 14), as defects in stamping can appear in multiple locations on a sample. These defect clusters are visualised in different shades of grey in Fig. 4. To account for the possibility of defects not appearing in every defect region, random clusters are selected for further processing in line 15, as depicted in the highlighted box in Fig. 4. Finally, a random element is chosen as the origin of the split defect for each selected cluster in line 16 of the algorithm, as shown in the enlarged portion of Fig. 4.

The proposed method has two distinct benefits. Firstly, it produces geometries with random, yet physically plausible, defect locations without producing fictitious defects. Secondly, the probability distribution of defects is similar to the real defect probability observed in the literature, for example [25].

B. Realistic Defect Synthesis

To generate photorealistic imagery of these defects, finer defect details need to be applied to the surface. The fine details can be achieved by increasing the resolution of the FEM simulation, but this incurs increasing computational costs and is limited to the physics-based nature of an FEM solver. Therefore, we add this type of detail by capturing images of the real defects on a small number of parts, then apply this to the crack cluster, thereby adding the finer details. More specifically, we capture bump maps which contain offsets of heights from the base surface and procedurally apply these to the clusters before rendering a photorealistic training sample for defect detection. A comparative image of bump mapping and traditional texturing with real crack is shown in Fig. 5.

The three steps required to move from a crack origin element from the FEM simulation to a photorealistic training image are: aligning the bump maps, defining surface appearance, and rendering the image. These are outlined in Fig. 6 and described in more detail in the following subsections.

1) Aligning displacement maps: In sheet metal stamping, the uncertainty associated with the material (internal defect, grain structure) and stamping process (wear and tear in machine parts, change in lubricants) allow defects to originate at random locations and propagate, resulting in a variety of split defect features at multiple locations. To capture this, we created split defect geometry in two steps. First, we used the elements selected in Section IV-A as the centroid of the defect and then projected the shape of the defect region onto the 3D mesh. The size of this defect region was sampled from a dictionary of captured defects from real samples; this dictionary also includes associated textures described later in this section.

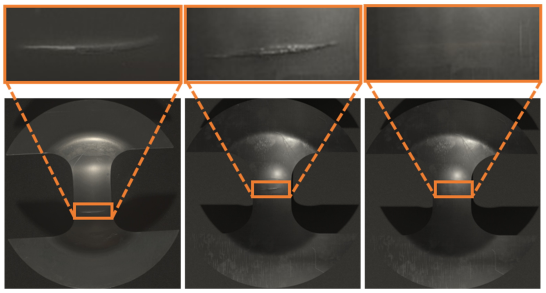

Fig. 5: Compare real crack (left) with defect mapping using Bump mapping (middle) and standard texturing (right).

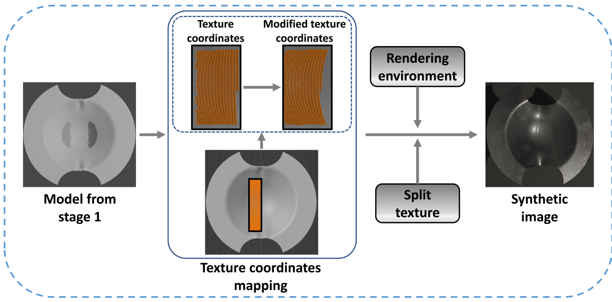

Fig. 6: Illustrates the computer graphics-based synthetic image generation. The process begins with the models from stage 1 as input (depicted in the left panel), followed by texture coordinate mapping (represented in the middle panel). The final output is a generated synthetic image, displayed in the right panel.

Given a coordinate $X \in \mathbb{R}^3$ lying on the surface of the mesh and texture coordinates $(u, v) \in \mathbb{R}^2$, we need a map $f : X \mapsto (u, v)$ to map 3D positions on the surface of the mesh to a texture coordinate on the bump map. This mapping needs to be aligned with the regions of the defective mesh resulting from the previous step (Section IV-A) but also needs to be further parameterised to ensure random mapping variants can be generated to produce multiple random samples for creating the synthetic data set. Therefore, we decompose this mapping into two steps: $g : X \mapsto (u, v)'$ which maps a point to a base texture coordinate space $(u, v)' \in \mathbb{R}^2$, then a second mapping $h : (u, v)' \mapsto (u, v)$ which distorts the texture coordinates to produce the random samples. The following sections describes each of the mappings.

To compute g, we analyse the geometry of the cluster to automatically, and optimally (in the ℓ_2 sense), produce a mapping which aligns the crack region on the mesh with a plane containing the bump map. Firstly, to ensure the texture is mapped to a contiguous region of the mesh, we first expand the cluster to include all the vertices of the mesh contained in the bounding box of the original vertices. We then extract the plane from these vertices by performing an Eigendecomposition of the covariance matrix of the vertices from the expanded cluster. The largest two Eigenvectors from the Eigendecomposition define a plane onto which the vertices from the cluster are projected, and based on these projected positions, the initial texture coordinates $(u, v)'$ are assigned. These have an origin centered on the element chosen to be the start of the defect (see Algorithm I, Line 16). This approach has the advantage of computing a low distortion mapping of a crack displacement map onto the model while considering the local geometry around the crack region.

For synthetic generation purposes, multiple random variations of the orientation of this crack displacement are required. Our method allows this via modification of the texture coordinates resulting from the map g. This modification leads to the second map h, which displaces the texture coordinates in a controlled manner. To ensure controlled and realistic variations, we propose to use a Bezier curve aligned with the direction of the crack to distort the texture coordinates. Without loss of generality, we assume that the v' coordinate of the i'th vertex is being deformed through the mapping $h : (u_i, v_i)' \mapsto (u_i, C(v_i))$ where

$$C(v_i') = \sum_{j=0}^{M} P_j \binom{M}{j} (1 - v_i')^{M-j} v_i'^{j}, \tag{2}$$

describes a Bezier curve with M control points P_j. Large modifications would lead to visible distortions, and too small modifications would lead to insufficient variation in the synthetic generation process. Therefore, these values were empirically chosen to lock the curve displacement map in place through the control points while providing sufficient variation in the synthetic image generation process.

This finally results in texture coordinates (u_i, v_i), which are suitable for the application of a displacement map which provides the high frequency details required for realistic crack synthesis. Each of these displacement maps is created from applying photogrammetry techniques to capture height variations from cracks on real parts. Precisely, for each crack, we capture an image of the crack where the camera is positioned perpendicular to the crack. This image of the crack is then converted to a displacement map using a technique such as [28]. We repeated this for several samples containing cracks, leading to a dictionary of displacement maps containing cracks. Then, for each synthetic part we wish to generate, we sample a displacement map randomly from this dictionary and apply it to the surface.

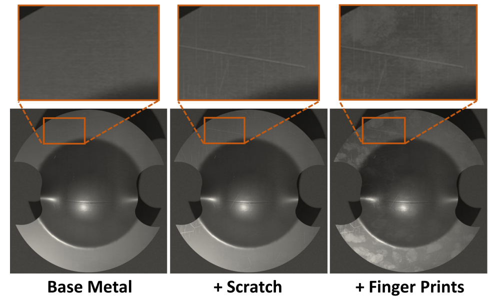

Fig. 7: Shows synthetic images at various steps of the pipeline.

2) Surface Appearance: As this work aims to produce realistic synthetic data to train defect detection algorithms, we also need to consider the appearance of the surface of the part, specifically how the part reflects light. This can be described by a Bidirectional Reflectance Distribution Function (BRDF) which describes how much incoming light from any direction is reflected into another direction. While many models have been proposed in the literature [29], we use a common microfacet-based model. This describes the microsurface as a distribution of oriented specular facets described by a roughness parameter and can simulate the appearance of a wide range of materials, from rough to highly polished surfaces. We use the principled BRDF [30] as the reflectance model, and we model the base reflectance using the index of refraction of the metal part and combine this with a measured roughness value of the part. We also modulate this with a procedurally created impurities texture, including real-world details that may be found on parts, such as fingerprints and scratches. Fig. 7 shows the appearance of the base part and the impact and increased realism of adding the impurities to the part.

3) Rendering: We render defective sheet metal stamping parts using path tracing [31], a Monte Carlo method for light transport

Fig. 8: The dictionary of textures used in the experiments

simulation which produces an unbiased estimate of the scene illumination by tracing light paths from the virtual camera into the scene and connecting these paths to the light sources. This allows for accurate simulation of all geometric optics based light transport phenomena that are needed for this work, including colour bleeding where the colour of one surface is reflected onto other visible surfaces and caustics which are the patterns formed by light focused through specular surfaces.

V. Experimental setup

This section presents the implementation details of a defect detection task used to evaluate the performance of the proposed image synthesis method. We use a dataset of real stamping samples produced in-house and a state-of-the-art CNN model for the experiments.

A. Dataset

As mentioned in Section I, obtaining stamped components which contain defects for training purposes is difficult because these defects rarely occur during serial production. Additionally, no open-source datasets for stamping defects are available. Therefore, we use defective (120) parts produced in-house based on *Nakajima* geometry [13]. The parts are produced using a 100 mm punch-to-punch material of varying widths. The test procedure followed (ISO 12004-2:2008). Although the stamping part has a straightforward hemisphere shape, the split defects in the parts are representations of split defects observed in real-world complex parts as described in Section III. Moreover, the Nakajima parts consist of additional real-world characteristics such as sharp edges, scratches and lubricants that share similar features with split defects, which can cause false predictions. Finally, the reflective nature of metal parts occasionally saturates defect regions [13], therefore this study used high dynamic range (HDR) images.

B. Synthetic Dataset

The main objective of our study was to create a realistic synthetic dataset for training DL models to detect split defects. To achieve this, we carefully tuned the hyperparameters of the proposed method. As a reference, we used the experimental FLC developed by Small et al. [25], considering that both their method and ours utilise the same aluminium material. The parameters used in our method, denoted as flc^p, were set as follows: the slopes s_l and s_r were assigned values of 45° and 30°, respectively. Furthermore, based on the observed variation in ϵ_0 reported in [25], we set p_{min} to 0.66. It is important to note that while the initial values were sourced from the literature [25] as a reference, we performed additional experimentation to refine and validate these parameter choices.

We implemented a Bezier curve-based distortion technique to introduce variations in the synthetic images. Considering that the observed defects in the components have a quadratic curve shape, we utilised a quadratic Bezier curve with three control points. The selection of control point values was performed experimentally. The

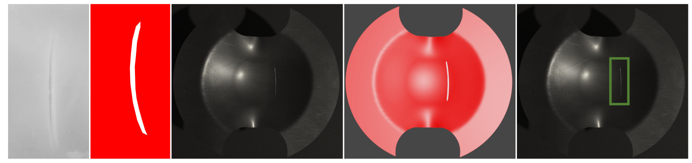

Fig. 9: Shows from left to right, an example defect texture, corresponding annotated mask for the defect texture, rendered image, mask and rendered image including the defect annotation.

extreme points were initially determined through experimentation, and a degree of randomness was achieved via a Gaussian distribution. The first and last control points of the quadratic Bezier curve were set as $P_0 = (0, c_0)$ and $P_2 = (0, c_2)$ respectively, with $c_0 \sim \mathcal{N}(0, 0.05)$ and $c_2 \sim \mathcal{N}(0, 0.05)$. The middle point was sampled from a 2D Gaussian with a mean of $(0.5, 0.2)$ and covariance $\begin{bmatrix} 0.07 & 0 \\ 0 & 0.05 \end{bmatrix}$.

Additionally, we created a split texture dictionary using ten randomly selected real samples (refer to Fig. 8). Since our study focuses on defect detection, in addition to rendering images, we generate mask images for the automatic defect annotation, as shown in Fig. 9. The resulting synthetic images were then compared with real images, and the comparison is presented in Fig. 10.

In synthetic training images, minor systematic deviations between automatic synthetic labels and human-annotated labels are inevitable. For instance, the point where the defect ends is not well-defined. In contrast, labels generated from synthetic pipeline produces tight annotations. Our work implemented a simple label randomisation by enlarging the labels based on sampling a fitted distribution of real annotated labels.

C. Network Training

As mentioned earlier in the literature, few-shot object detection models are studied extensively [2], [32]. However, Chen et al. [33] compared methods from the literature and found that most studies under-estimated the baseline (e.g. no data augmentation), and the gap between existing methods is reduced when deeper networks are used. Therefore a deeper state-of-the-art general-purpose CNN architecture Yolov5 [34] with data augmentation, particularly mosaic, translation, rotation, shearing, scaling and flipping, was used in the study. Since this study uses HDR images the model implemented in [13] was used. Furthermore, to make the comparison fair, the best-performing parameters were saved for validation by training all the models for a prolonged period (2000 epochs). Additionally, the study experiments were carried out with five-fold cross-validation by keeping five-fold test sets the same for all combinations of experiments.

VI. Results

This section evaluates the proposed framework for a real defect detection task. Initially, we present results for a fully factorial experiment that was conducted using various sets of real and synthetic images (see Table II). Qualitative and ablation studies follow. We show that the models trained by adding synthetic images outperform the state-of-the-art CNN models exclusively trained on real images. The method was evaluated using the dataset described in Section V with the same test set for all experiments.

A. Influence of Synthetic data on DL-based object detectors

Mean average precision (mAP), evaluated at 0.5 to 0.95 with an interval of 0.05 intersection over union (IOU), and mAP at 0.5 IOU (mAP50) are used as the evaluation metrics.

We evaluated our proposed method using two additional state-of-the-art object-detection DL models from different categories: Faster

TABLE II: Shows mAP50 (mAP) for various training sets. The underlines indicate the best performance in each row and bold indicates the best performance among all the experiments.

Detectors	Real \ Synth	Baseline	+10	+20	+40	+80
Yolov5 (No Aug)			0.050 (0.011)	0.167 (0.048)	0.353 (0.106)	0.530 (0.192)
	10	0.121 (0.033)	0.292 (0.083)	0.318 (0.090)	0.543 (0.185)	0.570 (0.229)
	20	0.265 (0.092)	0.382 (0.126)	0.547 (0.202)	0.536 (0.189)	0.682 (0.284)
	40	0.624 (0.253)	0.739 (0.315)	0.526 (0.200)	0.715 (0.349)	0.772 (0.366)
	80	0.783 (0.363)	0.849 (0.407)	0.877 (0.439)	0.879 (0.468)	0.879 (0.471)
FRCNN			0.123 (0.034)	0.343 (0.128)	0.413 (0.151)	0.550 (0.199)
	10	0.123 (0.031)	0.553 (0.165)	0.645 (0.206)	0.639 (0.208)	0.609 (0.212)
	20	0.664 (0.201)	0.655 (0.191)	0.675 (0.208)	0.620 (0.186)	0.729 (0.250)
	40	0.769 (0.266)	0.767 (0.269)	0.779 (0.265)	0.797 (0.291)	0.761 (0.276)
	80	0.855 (0.294)	0.852 (0.314)	0.868 (0.302)	0.847 (0.288)	0.866 (0.297)
DTER			0.038 (0.012)	0.081 (0.027)	0.167 (0.049)	0.434 (0.128)
	10	0.062 (0.023)	0.082 (0.03)	0.120 (0.048)	0.374 (0.153)	0.533 (0.209)
	20	0.192 (0.081)	0.217 (0.097)	0.281 (0.129)	0.368 (0.163)	0.672 (0.311)
	40	0.274 (0.11)	0.391 (0.184)	0.436 (0.202)	0.583 (0.287)	0.788 (0.390)
	80	0.791 (0.412)	0.762 (0.384)	0.748 (0.391)	0.835 (0.446)	0.879 (0.459)
Yolov5			0.658 (0.264)	0.765 (0.339)	0.796 (0.399)	0.822 (0.456)
	10	0.857 (0.452)	0.898 (0.497)	0.893 (0.500)	0.921 (0.531)	0.891 (0.548)
	20	0.940 (0.555)	0.937 (0.574)	0.939 (0.584)	0.953 (0.596)	0.944 (0.599)
	40	0.969 (0.631)	0.968 (0.631)	0.969 (0.632)	0.977 (0.642)	**0.981** (0.645)
	80	0.978 (0.665)	0.979 (0.683)	0.976 (0.676)	0.979 (0.684)	**0.981 (0.688)**

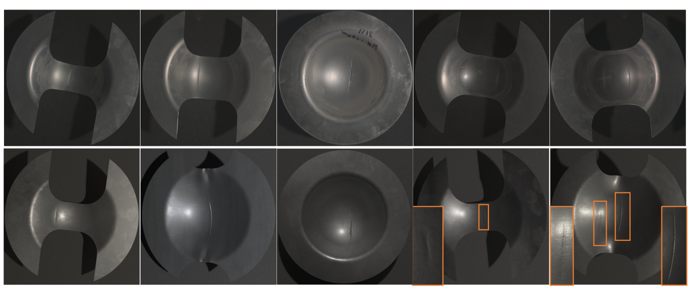

Fig. 10: Comparison of Real Images (Row 1) and Synthetic Images (Row 2), highlighting minor domain variations and realistic synthetic defects. The synthetic images, from left to right, illustrate a clear split defect, a curved split defect, a straight brittle fracture commonly found in circular samples, a small neck defect, and multiple defects within a single sample.

R-CNN [35], which represents a two-stage CNN model, and DETR [36], which represents a transformer-based model. As mentioned in Section V-C, Yolov5 used several traditional and advanced data augmentations. Therefore, additionally, we compared our method using a yolov5 model without data augmentation (Yolov5 (No Aug)) as a defect detector. Table II presents the quantitative results obtained with various deep learning-based object detectors. The table shows the mAP50 (mAP) values for different training sets under varying amounts of real and synthetic images. The rows represent the number of real images used, and the columns represent the number of synthetic images used. The underlined values in each row indicate the best performance, while the bold value in Table II represents the overall best performance among all the experiments. Incorporating synthetic images into the training process resulted in significant performance improvements for all the models, as measured by both mAP50 and mAP. This highlights the effectiveness of using synthetic data as a valuable resource for training object detectors, particularly in scenarios where real training data is limited.

The study used ten real stamping components to generate the synthetic images. Therefore, the model trained on the same ten real images is used as the baseline. The performance for Faster-RCNN, DTER and Yolov5 (No Aug) was poor. However, the addition of synthetic images led to 0.645, 0.533 and 0.570 mAP50 for Faster-RCNN, DETR and Yolov5 (No Aug), respectively, compared to their respective baselines.

The best mAP50 of 0.857 and mAP of 0.452 were achieved using the Yolov5 model for ten real samples. Using only synthetic images achieved comparable results of 0.822 and 0.456 mAP50 and mAP, thereby proving the quality of generated images. Adding synthetic images to the ten real training images improved 6.4% in mAP50 and 9.9% in mAP compared to the baseline. This indicates that incorporating synthetic images can effectively enhance the performance of the Yolov5 model even when limited real training data is available.

Furthermore, the results indicate that 80 real samples were sufficient to achieve a satisfactory performance of a mAP50 of 0.978 with the Yolov5 model. This indicates that the 80 real samples provided adequate data representation for the dataset used in the study. Thus, the performance achieved with 80 real samples using the Yolov5 model serves as a benchmark to demonstrate the superiority of our method in achieving comparable performance while using fewer real samples. It is important to note that creating a large enough dataset to adequately capture the data representation for small prototypes is feasible; however, the associated cost becomes significantly expensive when dealing with larger and more complex real stamping parts.

Our proposed method achieves comparable performance while utilising fewer real samples. For instance, combining 40 real images with 40 synthetic images yielded a mAP50 of 0.977, comparable to the mAP50 of 0.978 achieved using 80 real images. Moreover, augmenting the dataset with an additional 40 synthetic images (totalling 40 real and 80 synthetic images) resulted in an even higher mAP of 0.981. This finding showcases the effectiveness of our approach in achieving comparable performance while utilising a reduced number of real samples. The results highlight the potential of our method to address the limitations of cost and availability associated with real training data, particularly in scenarios where larger real stamping parts are involved.

B. Comparison with Few shot methods

In this section, we compare our proposed method with the two main categories of few-shot methods, namely the pre-training-based method and synthetic images from generative models. The Yolov5 detector was used for training for these experiments since it was the best-performing model in the previous section.

From the data augmentation category, we compared Defect-aware Feature Manipulation GAN (DFMGAN) [37] and a fine-tuned stable diffusion model to generate synthetic images. DFMGAN is a state-of-the-art few-shot model for generating defect images and consists of two training stages: first a backbone is trained on defect-free images, and then the model is fine-tuned on defect images with an additional defect-aware residual block. Conversely, there is no prior study that applies diffusion-based models for defect image generation tasks. Therefore, we propose to fine tune an stable diffusion model [38] as a further comparison. It should be noted that although the stable diffusion model was initially trained on extensive datasets, it encountered challenges in generating rare, application-specific images which is the focus of our work. To ensure a fair comparison with the DFMGAN model, we fine-tuned the diffusion model using 30 defect-free images in addition to defective images.

Fig. 11 presents a visual comparison of synthetically generated images generated by the aforementioned generative models and our proposed framework. As observed in Fig 11 row 2, DFMGAN exhibited difficulties in producing realistic defects, particularly within regions of bright reflections. Conversely, the diffusion model generated images with noticeable noise artefacts for a few training images (10 defective and 30 non-defective images). Although the quality of the generated images is aesthetically satisfactory when training with 40 or 80 defective images, the diversity of samples produced for training a defect detection method is insufficient leading to lower detection accuracy.

In the context of the pre-training method, we first trained the yolov5 model using 1470 open-source metal crack defect images collected from literature [39]. Then the model was fine-tuned using our dataset.

Results comparing DFMGAN [37], fine-tuned stable diffusion [38] and the pre-trained models with our proposed method are shown in Table III. The first row shows the number of real samples used during the training stage. As it can be observed our proposed framework consistently outperforms existing few-shot methods. This superiority over the pre-trained model can be attributed to two primary factors: firstly, the dataset employed for pre-training is inadequately representative of the stamping dataset, particularly with regard to neck defects and defects obscured in bright reflection areas. Secondly, our framework capitalises on HDR imaging, enabling the capture of intricate details, especially in challenging lighting conditions.

One reason our proposed framework surpasses the generative models is due to its capacity for conditional image generation. For example, our method provide control over defect class, location, light and other environmental conditions. Whereas, these generative models failed to provide additional information to the detection models due to their inability to generate a large diversity of defects or sample due to the limited representation in the training set. Moreover, these generative models are incompatible with HDR imaging, further underscoring the significant improvement offered by our framework over existing generative methods.

It is important to note that the inference time remains same 0.006 seconds while using Yolov5 model for our framework, traditional and other compared few shot methods. The primary distinction lies in the use of HDR images in our method compared to LDR images in other compared methods. This distinction does not affect computational time since the normalisation step is common to both approaches.

TABLE III: Comparison of defect detection performance between the few-shot models, alongside the performance of the Yolov5 model trained on LDR and HDR images. The performance is shown in the format mAP50 (mAP).

Methods	**10**	**20**	**40**	**80**
Pre-Trained	0.814 (0.420)	0.863 (0.498)	0.914 (0.561)	0.928 (0.613)
Stable Diffusion [38]	0.834 (0.472)	0.880 (0.511)	0.892 (0.549)	0.933 (0.605)
DFMGAN [37]	0.837 (0.455)	0.879 (0.518)	0.926 (0.588)	0.956 (0.623)
Ours	**0.921 (0.548)**	**0.953 (0.599)**	**0.981 (0.645)**	**0.981 (0.688)**

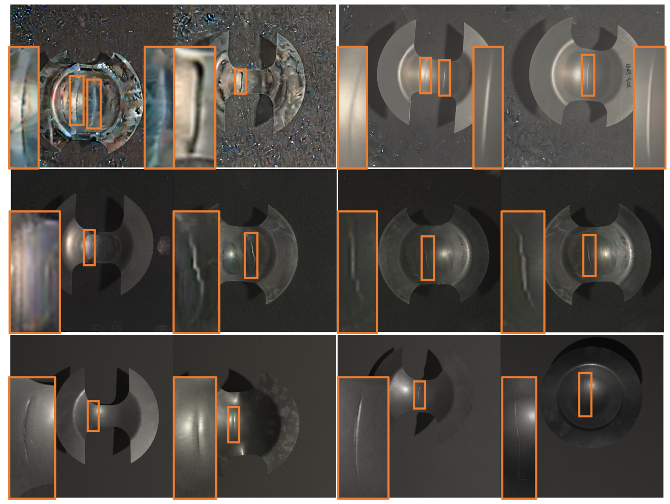

Fig. 11: Comparison of synthetic images from diffusion model (Row 1), DFMGAN (Row 2) and our method (Row 3). The left column shows for model trained using 10 real images and the right shows 80 real images.

Overall, our results reaffirm the efficacy of our proposed method in addressing the challenges associated with defect detection in sheet metal stamping. Furthermore, they highlight the potential of our approach for real-world industrial applications, showcasing its ability to overcome the limitations of few-shot methods and generate accurate defect detection outcomes.

C. Qualitative Results

The proposed framework generates defective stamping images with a minimum domain gap, as shown in Fig. 10. Fig. 12 shows qualitative results for the defect detection experiments. Fig. 12 shows the ground truth in "green" and prediction in "orange" with an assigned confidence score. The columns represent the number of real images used to train the CNN model, row 1 shows the results from exclusively using real images, and row 2 shows the improved results after combining synthetic training images with real training images. From Fig. 12, it can be observed that the model trained on mixed data not only increases the number of correct predictions but also reduces the false predictions. Additionally, using synthetic images improves the confidence score of correct predictions.

D. Ablation Studies

This section presents an ablation study that evaluates the influence of various stages of synthetic dataset generation, such as label randomisation and impurities. To study three datasets were prepared. First, dataset "C1: Basic" includes synthetic images without impurities and with tight annotations created directly from the synthetic pipeline. Second, dataset "C2: Basic + LR" includes C1 datasets with label randomisation (LR). The third dataset, "C3: Basic + LR + IM", includes C2 added with procedurally generated surface impurities (IM) such as scratches and fingerprints. The Yolov5 model was trained with the three above mentioned datasets and a mixture of C2 and C1 datasets with equal proportions.

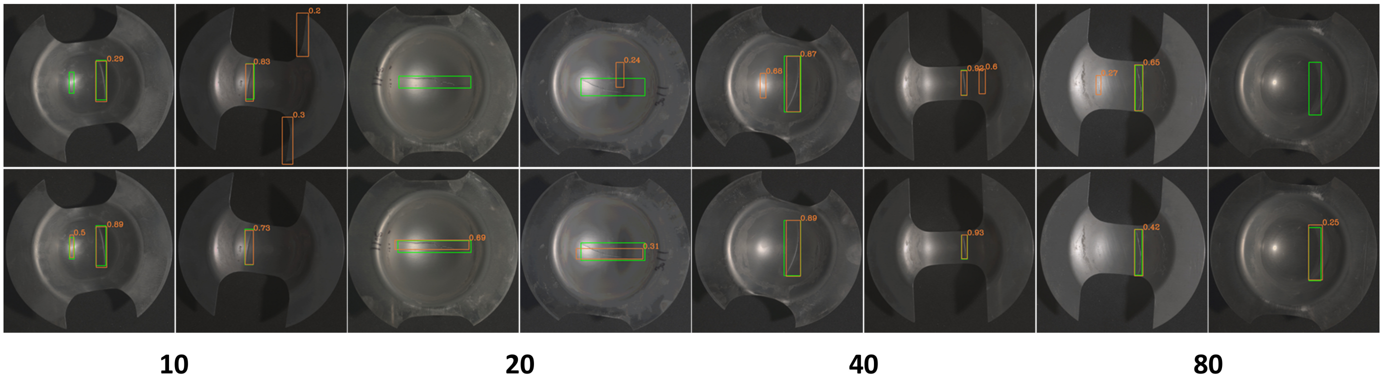

Fig. 12: Compare qualitative results for models trained with only real data (row 1) and combined real and synthetic data (row 2). The ground truth is shown in green and the prediction with confidence score is shown in orange.

TABLE IV: Presents mAP50 (mAP) for synthetic training datasets.

Datasets	Synthetic	Real
C1: Basic	0.886 (0.120)	0.574 (0.195)
C2: Basic + LR	0.951 (0.603)	0.688 (0.347)
C3: Basic + LR + IM	0.987 (0.616)	0.796 (0.422)
C2 + C3	0.990 (0.756)	0.822 (0.456)

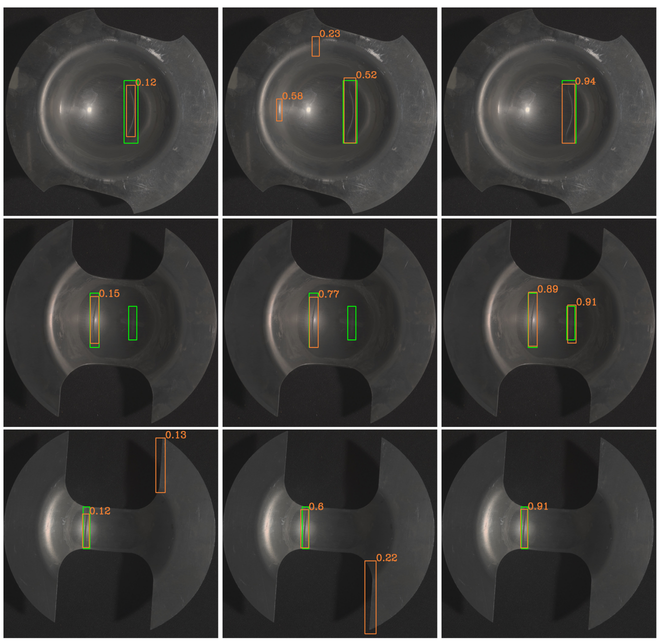

Fig. 13: Shows ground truth in green and prediction in orange for datasets C1, C2, and C3 in columns 1, 2, and 3 respectively.

The quantitative results evaluated on synthetic and real datasets, including mAP50 and mAP, are presented in Table. IV. The eighty synthetic test examples are taken from dataset C3. We can observe higher performance for the synthetic test set as the dataset comes from the same domain. However, the differences are substantial when the datasets are tested on real examples. For example, label randomisation improves mAP50 by 10.8% and mAP by 12.5%, and the inclusion of impurities as discussed in Section IV improves the mAP50 and mAP by 10% and 7%, respectively. Finally, combining data from C2 and C3 gave a further 3% improvement in both mAP50 and mAP.

Qualitative results for the experiments are shown in Fig. 13. The first, second, and third columns of Fig. 13 show the results for datasets C1, C2, and C3, respectively. Although the same number of defects are predicted correctly for datasets C1 and C2, the confidence and IOU of prediction are lower in the case of dataset C1 (see Fig. 13 columns 1 and 2). Furthermore, since the synthetic datasets (C1 and C2) have ideal surfaces without impurities, the models trained using these datasets are not only missing defects but also showing false predictions for edges and impurities (see columns 1 and 2 in Fig. 13). Finally, adding impurities to the synthetic images reduce false predictions (see Fig. 13 column 3). Additionally, label randomisation and impurities show a prediction comparable to the ground truth with high confidence.

VII. Discussion and Conclusion

This study proposed a framework that generates photorealistic imagery for training machine learning split defect detection classifiers for stamped metal parts. This work proposes a new, principled approach, which overcomes the low resolution and computational cost limitations of physical simulation-based methods, and the unrealistic defect placement from synthesis-based approaches. The proposed framework leverages both approaches to generate physically accurate and visually diverse photorealistic defects. The framework uses a novel application of FEM and FLC to find plausible defect locations and dimensionality reduction and Bezier curve-based texture mapping to make a wide range of split defects from a limited dictionary of captured crack displacements. The results show that the framework outperforms the model trained exclusively on real images, even when only half of the real examples are used. This indicates its potential for automating split defect inspection, particularly in the common case where capturing a real training dataset is challenging to achieve.

The framework can also be extended to other stamping components by tuning the hyperparameters. This allows split inspection on a wide range of stamping components. This study approximates the parameters $p, s_l,$, and s_r from an experimental FLC, which most stamping manufacturers generate for component design. However, these parameters can also be achieved easily from pilot runs. The core concept of the study, which uses both physics-based simulation and computer graphics to generate defect datasets for DL model training, opens up the potential for further research to develop similar frameworks for other types of defects and manufacturing processes, leading to a fully automated stamping defect inspection process. This framework has the potential for future improvements in other areas of manufacturing, where gathering datasets are expensive and automatic defect detection is essential.

Although the results indicate the feasibility of training a DL model using synthetic images generated with our framework, we acknowledge certain limitations in our study. Currently, our focus is primarily on split defects and representative parts. As future work, we intend to expand the number of parts in our dataset. This will need to involve collaborating with a stamping manufacturer to obtain a wide range of complex components with the necessary part design and material properties. This collaboration would allow us to validate the effectiveness of our approach on a broader range of stamping parts.

Moreover, for the successful deployment of a DL-based model for defect detection in stamping components, it is crucial to detect all types of defects encountered in practice. Therefore, an important future direction is to extend our framework to encompass other types of defects. One such defect commonly found in stamping components is wrinkles. To incorporate physical accuracy into the synthetic generation of wrinkle defects, we propose leveraging multiple FEM simulations with modified material and process parameters. This approach will allow us to explore the extreme range of wrinkle locations and capture the realistic variations in their occurrence. Then, following the framework proposed in this paper, we can use real wrinkle textures to generate realistic synthetic images of wrinkle defects.

In conclusion, this study presents a novel framework that combines physics-based simulation and computer graphics methods to generate photorealistic images of split defects in sheet metal stamping for training split defect detection classifiers. The proposed method generates both accurate locations and visual diversity of defects, making it an important step towards fully automated defect inspection in the manufacturing industry.

References

[1] S. Ghosh, "Principally on sheet metal forming defects as described in the eleventh biennial congress of the international deep drawing research group (iddrg)," *International Journal of Mechanical Sciences*, vol. 23, no. 4, pp. 195–211, 1981.

[2] A. S. Mangat, J. Mangler, and S. Rinderle-Ma, "Interactive process automation based on lightweight object detection in manufacturing processes," *Computers in Industry*, vol. 130, p. 103482, 2021.

[3] D. Cannizzaro, A. G. Varrella, S. Paradiso, R. Sampieri, Y. Chen, A. Macii, E. Patti, and S. Di Cataldo, "In-situ defect detection of metal additive manufacturing: an integrated framework," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 1, pp. 74–86, 2021.

[4] M. Sundermeyer, Z.-C. Marton, M. Durner, and R. Triebel, "Augmented autoencoders: Implicit 3d orientation learning for 6d object detection," *International Journal of Computer Vision*, vol. 128, no. 3, 2020.

[5] J. Yu and J. Liu, "Multiple granularities generative adversarial network for recognition of wafer map defects," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1674–1683, 2021.

[6] X. Zhang, C. Wang, Y. Tang, Z. Zhou, and X. Lu, "A survey of few-shot learning and its application in industrial object detection tasks," in *International Workshop of Advanced Manufacturing and Automation*. Springer, 2021, pp. 637–647.

[7] Y. Zhao, H. Ding, H. Huang, and N.-M. Cheung, "A closer look at few-shot image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9140–9150.

[8] X. Ni, Z. Ma, J. Liu, B. Shi, and H. Liu, "Attention network for rail surface defect detection via consistency of intersection-over-union(iou)-guided center-point estimation," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1694–1705, 2022.

[9] S. Ma, K. Song, M. Niu, H. Tian, Y. Wang, and Y. Yan, "Shape consistent one-shot unsupervised domain adaptation for rail surface defect segmentation," *IEEE Transactions on Industrial Informatics*, 2023.

[10] G. Zhai, Y. Narazaki, S. Wang, S. A. V. Shajihan, and B. F. Spencer Jr, "Synthetic data augmentation for pixel-wise steel fatigue crack identification using fully convolutional networks," *Smart Struct Syst*, vol. 29, no. 1, pp. 237–250, 2022.

[11] R. Mordia and A. K. Verma, "Visual techniques for defects detection in steel products: A comparative study," *Engineering Failure Analysis*, p. 106047, 2022.

[12] S. B. Block, R. D. da Silva, L. B. Dorini, and R. Minetto, "Inspection of imprint defects in stamped metal surfaces using deep learning and tracking," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 5, pp. 4498–4507, 2020.

[13] A. R. Singh, T. Bashford-Rogers, D. Marnerides, K. Debattista, and S. Hazra, "Hdr image-based deep learning approach for automatic detection of split defects on sheet metal stamping parts," *The International Journal of Advanced Manufacturing Technology*, vol. 125, no. 5-6, pp. 2393–2408, 2023.

[14] A. R. Singh, T. Bashford-Rogers, S. Hazra, and K. Debattista, "Deep learning-based defect inspection in sheet metal stamping parts," in *NUMISHEET 2022*. Springer, 2022, pp. 411–419.

[15] S. Niu, B. Li, X. Wang, and Y. Peng, "Region-and strength-controllable gan for defect generation and segmentation in industrial images," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4531–4541, 2021.

[16] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4367–4375.

[17] C. Siu, M. Wang, and J. C. Cheng, "A framework for synthetic image generation and augmentation for improving automatic sewer pipe defect detection," *Automation in Construction*, vol. 137, p. 104213, 2022.

[18] E. Martinson, B. Furlong, and A. Gillies, "Training rare object detection in satellite imagery with synthetic gan images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2769–2776.

[19] X. Li, R. Cao, Y. Feng, K. Chen, B. Yang, C.-W. Fu, Y. Li, Q. Dou, Y.-H. Liu, and P.-A. Heng, "A sim-to-real object recognition and localization framework for industrial robotic bin picking," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3961–3968, 2022.

[20] O. Schmedemann, M. Baaß, D. Schoepflin, and T. Schüppstuhl, "Procedural synthetic training data generation for ai-based defect detection in industrial surface inspection," *Procedia CIRP*, vol. 107.

[21] V. L. Kondarattsev, S. S. Krylov, and N. P. Anosova, "Creating a synthetic data generator for solving industrial flaw detection problems using deep learning methods," in *Advances in Theory and Practice of Computational Mechanics*. Springer, 2022, pp. 377–390.

[22] A. Boikov, V. Payor, R. Savelev, and A. Kolesnikov, "Synthetic data generation for steel defect detection and classification using deep learning," *Symmetry*, vol. 13, no. 7, p. 1176, 2021.

[23] J. F. O'brien, A. W. Bargteil, and J. K. Hodgins, "Graphical modeling and animation of ductile fracture," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 2002.

[24] A. Govik, "Finite element analysis of sheet metal assemblies: Prediction of product performance considering the manufacturing process," Ph.D. dissertation, Linköping University Electronic Press, 2014.

[25] N. Small, S. Hazra, D. Williams, R. Roy, T. Nichols, R. Aylmore, and D. Hollingdale, "Determining a probabilistic forming limit curve," in *AIP Conference Proceedings*, vol. 1567, no. 1. American Institute of Physics, 2013, pp. 398–401.

[26] K. Nakajima, T. Kikuuma, and K. Hasuka, "Yawata technical report no. 284," *Yawata, Japan*, pp. 678–90, 1971.

[27] S. Keeler and W. Brazier, "Relationship between laboratory material characterization and press-shop formability," 1977.

[28] X. Zhou, G. Zhong, L. Qi, J. Dong, T. D. Pham, and J. Mao, "Surface height map estimation from a single image using convolutional neural networks," in *Eighth International Conference on Graphic and Image Processing (ICGIP 2016)*, vol. 10225. SPIE, 2017, pp. 427–432.

[29] R. Montes and C. Ureña, "An overview of brdf models," *University of Grenada, Technical Report LSI-2012-001*, 2012.

[30] B. Burley and W. D. A. Studios, "Physically-based shading at disney," in *ACM SIGGRAPH*, vol. 2012. vol. 2012, 2012, pp. 1–7.

[31] J. T. Kajiya, "The rendering equation," in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 1986.

[32] S. Meister, N. Möller, J. Stüve, and R. M. Groves, "Synthetic image data augmentation for fibre layup inspection processes: Techniques to enhance the data set," *Journal of Intelligent Manufacturing*, vol. 32, no. 6, pp. 1767–1789, 2021.

[33] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HkxLXnAcFQ

[34] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomammana, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Y. , changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4154370

[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[36] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[37] Y. Duan, Y. Hong, L. Niu, and L. Zhang, "Few-shot defect image generation via defect-aware feature manipulation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, 2023, pp. 571–578.

[38] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[39] "Metal crack dataset," https://universe.roboflow.com/, 2022, visited on 2023-06-13. [Online]. Available: https://universe.roboflow.com/