DIE-VIS: an Automated Visual Inspection System for Cardboard Box Manufacturing

Flavia Monti, Matteo Marinacci, Francesco Leotta, and Massimo Mecella

Sapienza Università di Roma, Rome, Italy
{monti,marinacci,leotta,mecella}@diag.uniroma1.it

Abstract. In recent decades, the industrial world has undergone a significant transformation through the inclusion of innovative technologies that enhance manufacturing processes. In this context, Machine Vision inspection systems play a key role in ensuring quality by identifying defects in production. Automated defect detection systems improve productivity by reducing manual interventions, which can be time-consuming and prone to errors. This paper presents DIE-VIS, a real-world implemented visual inspection system for detecting defects in cardboard box manufacturing using traditional Computer Vision techniques. We provide a comprehensive evaluation comparing it to the YOLOv8 state-of-the-art deep learning model, demonstrating how, in the specific application of cardboard manufacturing, customized solutions still offer fundamental advantages.

Keywords: Smart Manufacturing · Visual inspection system · Cardboard box production

1 Introduction

The modern industrial world is rapidly evolving, leading to a new era known as Industry 4.0 and, more recently, Industry 5.0 [39]. This transformation is driven by the integration of technologies such as Artificial Intelligence (AI), Internet-of-Things (IoT), Big Data, and Cloud Computing. These innovations have significantly improved manufacturing efficiency by enabling automated, adaptive, optimized, and precise decision-making [1, 29].

A key aspect of this advancement is represented by visual inspection systems, particularly defect detection systems, which are characterized by identifying defects from visual data [18]. Automated defect detection is crucial in manufacturing, where manual inspection can be time-consuming, labor-intensive, and prone to errors [16]. As a result, automated solutions are vital for maintaining product quality and efficiently identifying issues.

Machine Vision (MV) plays a pivotal role in this context, serving as the "eyes" of machines by capturing environmental data and providing insights through Computer Vision (CV). Many current solutions rely on feature-based techniques, involving extensive image processing techniques and algorithms. Recently, Deep Learning solutions have gained popularity due to their accuracy and versatility.

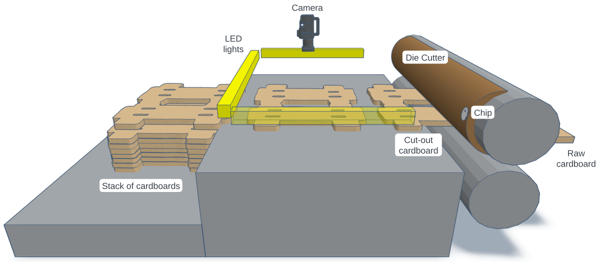


Fig. 1: Industrial setting

However, these methods require larger datasets and more expensive computational power to perform effectively, especially when high speed manufacturing is under monitoring. Despite this, MV has significantly improved inspection processes and product quality [3,14,28].

In this paper, we present a visual inspection system called DIE-VIS (DIE cutting VISion inspection), leveraging traditional CV techniques to detect defects in cardboard box manufacturing. Ensuring quality is crucial in this context, as non-compliant boxes leads to a loss of money and time for suppliers.

Cardboards are cut out by die cutting machines, which are composed of heavy rollers that move a single raw cardboard sheet through conveyor belts into two rollers. One roller is equipped with a die cutter, which is made of specialized blades producing cuts and creases. However, the blades deteriorating over time, together with an imprecise setup of the machine for a production batch, may unpredictably lead to defective boxes that do not meet requirements.

DIE-VIS inspects each produced box to detect defects in the form of missing holes and creases, ensuring the quality of production. Fig. 1 depicts a blueprint of the visual inspection setting. An industrial camera is installed above the die cutting machine, with LED light bars grazing the vertical and horizontal edges of the cardboard to highlight its surface characteristics. The die cutter features a chip that keeps track of different characteristics of production such as speed, temperature, and humidity. In addition, the chip leads the image acquisition process by detecting the beginning and the end of production sessions.

DIE-VIS is currently employed in a recent commercial solution for quality monitoring installed in several factories. However, due to confidentiality reasons, the names of the organizations involved and the source code cannot be disclosed.

The contribution of this paper is two-fold. First, it presents the DIE-VIS approach. Second, it compares it with the state-of-the-art YOLOv8 deep learning model, discussing how, despite the good performance, in real-world applications, traditional CV solutions like DIE-VIS still offer advantages with respect to modern deep learning techniques. In order to do this comparison, a dataset is provided to the research community to be used as a benchmark.

The paper is structured as follows. Sec. 2 presents related works in the literature. Sec. 3 describes the DIE-VIS solution. Sec. 4 discusses the evaluation results with YOLOv8 model. Finally, Sec. 5 concludes the paper.

2 Related Works

The literature presents a wide range of works related to vision-based systems for assessing the quality of production in manufacturing domains. These systems, known as vision-based measurement systems, typically use cameras and software applications to process images to detect anomalies [21].

The techniques for assessing the quality of products are divided into methods based on *designed features* and those exploiting *learned features* [13]. The former strongly rely on domain knowledge, whereas, the latter, usually based on machine and deep learning, loose this dependency being partially or fully knowledge free. DIE-VIS belongs to the first category. In the following, we focus on surface defects detection literature, which is relevant to our work.

Methods based on designed features detect defects relying on existing knowledge to extract relevant features from images. These methods often employ statistical approaches such as histogram analysis, differences, means, and standard deviations applied on the pixels of the images [9, 17, 24]. Filter-based methods using Sobel, Canny, Laplacian operator, wavelet, and Fourier transforms are also widely used to derive information from images [8, 19, 24, 25].

In the last decade, learned features methods have been introduced, mostly based on Convolutional Neural Networks (CNNs), autoencoder, and Recurrent Neural Networks (RNNs) for segmentation, classification, or detection tasks [38]. Such methods are commonly used to identify cracks, and holes [7,12,22,36,40,42].

In general, a limited number of works focus on cardboard box inspection. In [2], a vision-based system using traditional computer vision techniques was developed to count cardboard sheets by estimating stack height and sheet thickness. Similarly, authors in [33] analyze cardboard corrugation to count sheets in a stack. The work in [31] presents a CNN-based solution to classify and detect deformed corrugated cardboard. An interesting approach is presented in [26], which employs laser sensors for crease defect detection on cardboard surfaces.

The above mentioned works do not fit our application scenario. Cardboard box production has unique characteristics, in terms of types of defects, that differ from other materials or industries. Most existing research does not tackle these specific challenges, such as variations in superficial texture due to corrugation. The only comparable method, presented in [26], due to performance issues, is intended to be employed on production samples. DIE-VIS is instead designed with the idea of checking each single box on a high speed manufacturing line peaking up to 3 cardboards per second. Also, the focus of [26] is to show that geometry properties such as width and height, measured through a laser scanner, are useful to separate acceptable from defective creases. Finally, no extensive evaluation of the approach is provided thus making a fair comparison impossible.

3 Proposed solution: DIE-VIS

DIE-VIS consists of several components, which are depicted in Fig. 2. The system takes as input an RGB video stream and a Computer-Aided Design (CAD) model file, which represents the desired shape and features of the cardboard.

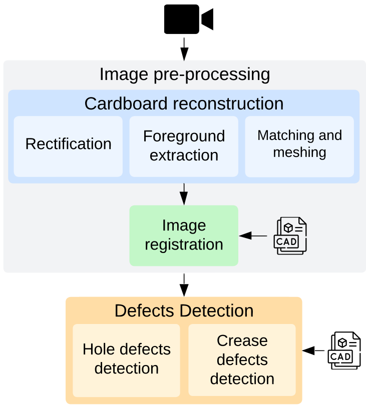


Fig. 2: DIE-VIS inspection solution

The CAD model is used in DIE-VIS both in the pre-processing and the detection of defects phases. CAD models, or engineering drawings, are an essential element for quality control of the manufactured product. In general, CAD models are representations of products that include geometric and descriptive information [32]. This information represents the gold standard that the manufactured product must meet. Specifically, the CAD model of a cardboard box[1] is essentially a list of geometrical entities (*i.e.*, lines, circles, arcs, polylines) representing the position and type of the blades the die cutter is equipped with. In particular, we consider two types of blades producing different effects: *(i)* cut blades, producing cuts, and *(ii)* crease blades, producing creases. The deterioration of such blades results in the production of defective cardboard boxes. Concerning cut blades, we only consider here those producing holes in the cardboard.

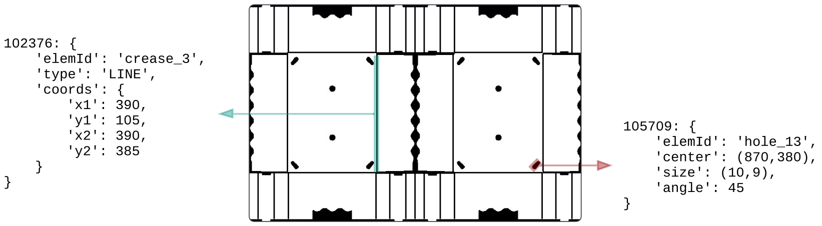
The entities of the CAD model are used in DIE-VIS to understand where creases and holes should be located. Fig. 3 depicts an example of a cardboard box CAD reference model with the geometrical information about a crease and a hole. Specifically, for each crease, we are interested in the coordinates of the line, while, for each hole, we are interested in the information of the box bounding it, *i.e.*, the coordinates of the center, the size of the box, and the rotation angle.

The remaining of this section is organized as follows. Sec. 3.1 describes the components involved in the *pre-processing* phase, i.e., the *cardboard reconstruction* and *image registration* components. Sec. 3.2 focuses instead on the actual *defects detection* module.

3.1 Image pre-processing

Die cutting machines come in different layouts. One of the consequences is that, in general, it is not granted that an appropriate camera location is available to

[1] In general, a CAD model includes multiple boxes to be produced out of a single raw cardboard sheet. Without loss of generality, we will ignore this detail.



```
102376: {
    'elemId': 'crease_3',
    'type': 'LINE',
    'coords': {
        'x1': 390,
        'y1': 105,
        'x2': 390,
        'y2': 385
    }
}
```

```
105709: {
    'elemId': 'hole_13',
    'center': (870,380),
    'size': (10,9),
    'angle': 45
}
```

Fig. 3: Cardboard CAD reference model

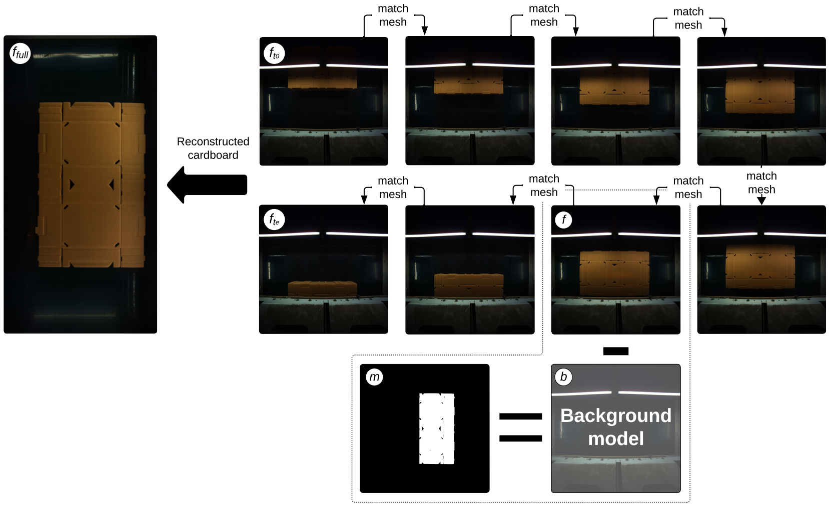


Fig. 4: Cardboard box reconstruction

frame an entire cardboard box (in the movement direction). This is particularly true in the case of large cardboard boxes. To overcome this issue, we developed a solution to reconstruct the entire cardboard box from a stream of video frames and apply an image registration algorithm to adjust the image to be analyzed.

Cardboard reconstruction. Fig. 4 shows the cardboard box reconstruction process from partial frames through an example. Our solution is based on template matching [27] applied to the sequence of consecutive rectified frames belonging to a single cardboard box. As a preliminary step, the video stream is rectified through a camera calibration step performed offline [23,41].

For each rectified video frame, a *cardboard box slice mask*, *i.e.*, a black and white image highlighting the visible cardboard slice in white and the background in black, is calculated by applying a classical foreground extraction technique [10]. In particular, referring to Fig. 4, the foreground extraction mechanism takes a new frame ⓕ and the current model of the background ⓑ, and compares them to extract the foreground, *i.e.*, the cardboard box slice mask ⓜ.

The background model ⓑ is represented in memory as a Mixture of Gaussians (MOGs) [43,44] and is updated over time only when the movement of the cardboard over the conveyor belt is not detected. This is obtained by computing the contours extracted from the cardboard box slice mask using a contour scanner [34], and updating the model only if no contours are identified. The update of ⓑ guarantees that artifacts including shadows, sudden illumination changes, light hotspots, and those deriving from background dynamicity, *i.e.*, those phenomena caused by the activities of the background such as moving conveyors or external objects in the scene, are integrated into the background model itself.

Each cardboard box slice mask is then classified, through an intuitive mechanism based on position thresholds, as being the beginning, the inside, or the end of the box being produced. This classification is needed to correctly identify the partial masks composing the specific box under verification. We will denote the single initial and final slice masks of a box as m_{t_0} and m_{t_e}. We will denote the full sequence of slice masks and camera frames belonging to a single cardboard box as $< m_{t_0}, \ldots, m_{t_i}, \ldots, m_{t_e} >$ and $< f_{t_0}, \ldots, f_{t_i}, \ldots, f_{t_e} >$ respectively.

The *matching and meshing* component iteratively reconstructs the entire image of the cardboard f_{full} – $\textcircled{\text{f}_{\text{full}}}$ in Fig. 4 – and the corresponding mask m_{full}. Initially $m_{full} = m_{t_0}$ and $f_{full} = f_{t_0}$. At each step i, a mask template $m^T_{t_i}$ is selected from the current slice mask m_{t_i} and is matched against the current m_{full}, computing a displacement value k and the width δ of the matching region in the movement direction (we here suppose, for simplicity, no displacement is detected in the orthogonal direction). This means that a pixel position (x, y) in $m^T_{t_i}$ corresponds to a pixel position $(x - k, y)$ in m_{full}. The same relationship holds between $f^T_{t_i}$ (the corresponding template from f_{t_i}) and f_{full}.

At this point, $f^T_{t_i}$ and f_{full} are meshed to ensure a seamless color transition. Pixels in the overlapping region are mixed by weighted averaging. Denoting with $f^T_{C,t_i}(x, y)$ the value of the pixel (x, y) in $f^T_{t_i}$ for the channel C (where C can be R, G or B), and with $f_{C,full}(x - k, y)$ the corresponding pixel channel value in f_{full}, we will update this latter as follows $f_{C,full}(x - k, y) = \alpha \times f^T_{C,t_i}(x, y) + (1 - \alpha) \times f_{C,full}(x - k, y)$, where $\alpha = \frac{x}{\delta}$. Intuitively, the farther we are from the beginning of the matching region, the more the color resembles the ones from f_{t_i}. Finally, f_{full} is extended with the pixels from f_{t_i} that are outside the template region and represent the part of the box previously unseen.

Meshing $m^T_{t_i}$ and m_{full} is instead simpler, with $m_{full}(x - k, y) = m_{full}(x - k, y) \wedge m^T_{t_i}(x, y)$.

Noteworthy the reconstruction process does not require as input the real speed using, for example, an encoder.

Image registration. Ideally, once m_{full} and f_{full} are computed, a simple resize operation would be enough to superimpose the CAD model to the cardboard box under analysis. Unfortunately though, they need to be refined due to possible *(i)* presence of cardboard residual on the border, *(ii)* rotation due to unbalanced pressure on the cylinder, *(iii)* stretching due to imperfect template matching, and *(iv)* imperfect foreground extraction. As a consequence, before

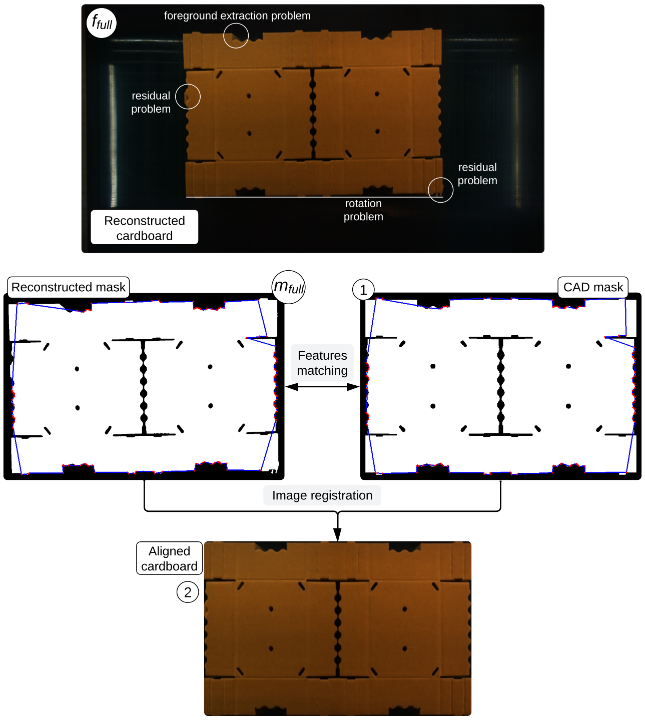


Fig. 5: Cardboard reconstruction

detecting defects, the reconstructed box needs to be further processed through *image registration* [35].

Fig. 5 depicts an example of such refinement. The reconstructed cardboard box suffers from the above mentioned problems (as highlighted in the $\textcircled{f_{full}}$ image). In order to apply image registration though, relevant features must be extracted from both the reconstructed cardboard box and the CAD model and a matching must be computed between the two sets.

In DIE-VIS, features are computed over a set of select points of the external contour. In particular, a contour scanner [34] is applied to both the cardboard box mask m_{full} – $\textcircled{m_{full}}$ in Fig. 5 – resulting from the reconstruction and the mask computed from the CAD model – $\textcircled{1}$ in Fig. 5 – to compute the external contours of both. For each CAD contour point, a set of m_{full} contour points is kept within a fixed distance value. Among these, the most similar is selected as matching – red points in $\textcircled{m_{full}}$ and $\textcircled{1}$. The similarity is computed as the distance between the Hu Moments of a CAD contour point and a m_{full} contour point. The Hu Moments vector is composed of seven values, where the first six values are invariant to translation, scaling, and rotation, while the seventh provides reflection invariance [15]. Once the matching features are extracted, the Homography matrix is computed to align the cardboard box image with the

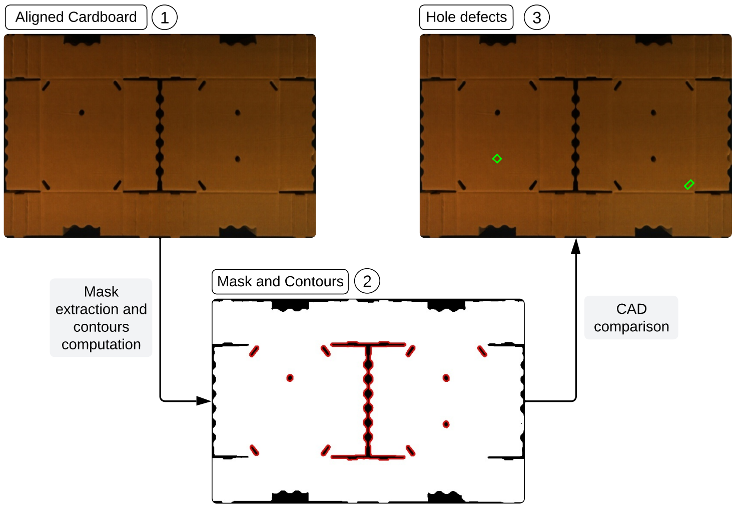


Fig. 6: Missing hole defect detection

CAD image through perspective correction. Result of the image registration is reported in ② in Fig. 5.

3.2 Defects detection

DIE-VIS detects two kinds of defects, namely missing holes, and creases. These are detected through two separate modules following a reference-based approach by leveraging the information coming from the CAD. Both modules take as input the aligned images resulting from the image registration phase.

Missing holes detection. Missing hole defects are identified by applying a contour detection algorithm [34] to the mask of the aligned cardboard, focusing on internal contours. For each contour, defined as a curve that connects all points along a boundary with the same color, we compute the relative bounding rectangle and compare its coordinates with the CAD reference information. Specifically, for each contour c, we compute the box, i.e., box_c, bounding it and the relative information including the center $c_{box_c} = (x, y)$, the dimension $d_{box_c} = (width, height)$, and the angle of rotation α_{box_c}. Notably, these can be easily compared with information extracted from the CAD (*cf*. Fig. 3). In particular, we classify a hole as present if the detected shape resembles, within a certain uncertainty interval, the original one in the CAD model.

Fig. 6 depicts the results of hole defect detection. The analyzed cardboard ① is of the same type as the CAD in Fig. 3, where two defects are detectable. Extracted contours of the holes are shown in red in ②. By computing the bounding boxes and comparing them with the CAD information, two missing holes are detected and relative boxes are depicted in green in ③.

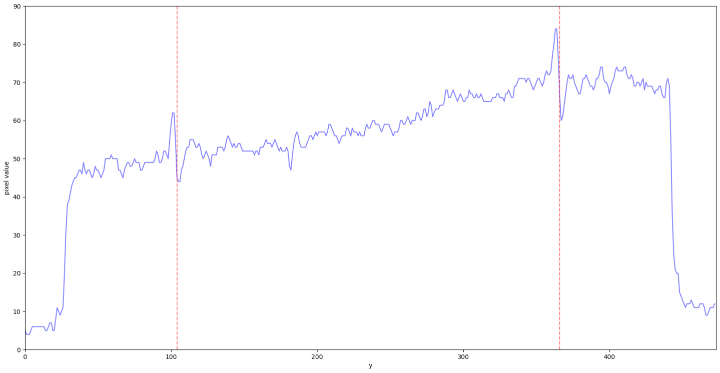


Fig. 7: Pixel values transitions over creases

Missing creases detection. To detect missing creases on cardboard, we developed an algorithm that processes the rows and columns of the image to identify relevant transitions of pixel values of a channel of the RGB-aligned cardboard box image. Fig. 7 shows an example of the phenomenon of interest. Relevant transitions of the pixel values are visible in correspondence to the crease positions (red-colored dashed lines). Traditional edge detection algorithms, such as the Canny [4] or Sobel filters, do not produce significant results due to their high sensitivity to the noise present on the cardboard surface irregularities due to corrugations.

Algorithm 1 reports the pseudocode of the algorithm. In particular, the pseudocode refers to the analysis of each row of the image to highlight the relevant pixels in correspondence to the transitions due to vertical creases. The corresponding algorithm for horizontal creases can be easily obtained by similarity.

The logic of the algorithm is as follows. After the state variables used to mark and track the pixel analysis are initialized, a double cycle analyses each pixel of the image (lines 2-32). Each pixel is compared to the next one to detect downward (lines 6-17) and upward (lines 18-29) transitions. For both kinds of transitions, we check whether the difference between the analyzed relevant pixels is within a certain threshold range and mark the pixel in the middle of the two if the difference falls within this range (lines 12-15 and 24-27).

After the execution of Algorithm 1 is completed, the relevant pixels computed along the rows and columns of the cardboard image are then processed to determine if the creases are present or not. In particular, the output of the rows scan is used to analyze vertical creases, while the columns scan is employed to analyze horizontal creases. In the case of oblique creases, the combined result of rows and columns scans is employed. The presence of a crease is determined based on the count of closely located pixels compared to the length of the real crease, which is extracted from the CAD model.

The result of the crease defect detection algorithm is shown in Fig. 8. The aligned cardboard box image ① is processed to analyze all rows and columns. ②

Algorithm 1 Relevant transitions detection and pixel marking (Rows scan)

```
Require: image $I$, low_thresh $l_t$, high_thresh $h_t$
Ensure: crease image $F$
$h, w \leftarrow$ image dimensions, $F \leftarrow$ zero matrix of size $(h, w)$
for each row $i$ in $I$ except last one do
    $start, down, up \leftarrow$ False, $pre \leftarrow 0$, $pre\_coord \leftarrow 0$
    for each column $j$ in $I$ except last one do
        $p_1 \leftarrow$ current pixel value $I(i, j)$, $p_2 \leftarrow$ next pixel value $I(i, j + 1)$
        if $p_1 > p_2$ then                                    ▷ Detecting downward transition
            if not $start$ then
                $start \leftarrow$ True, $down \leftarrow$ True, $pre \leftarrow p_1, pre\_coord \leftarrow j$
            end if
            if $up$ then                                       ▷ Downward transition update
                Calculate $\Delta \leftarrow |pre - p_1|$
                if $l_t \leq \Delta \leq h_t$ then
                    $middle\_j \leftarrow (j + pre\_coord)/2$
                    $F(i, middle\_j) \leftarrow 255$                     ▷ Marking pixel
                end if
                $up \leftarrow$ False, $down \leftarrow$ True, $pre \leftarrow p_1, pre\_coord \leftarrow j$
            end if
        else if $p_1 < p_2$ then                               ▷ Detecting upward transition
            if not $start$ then
                $start \leftarrow$ True, $up \leftarrow$ True, $pre \leftarrow p_1, pre\_coord \leftarrow j$
            end if
            if $down$ then                                     ▷ Upward transition update
                Calculate $\Delta \leftarrow |pre - p_1|$
                if $l_t \leq \Delta \leq h_t$ then
                    $middle\_j \leftarrow (j + pre\_coord)/2$
                    $F(i, middle\_j) \leftarrow 255$                     ▷ Marking pixel
                end if
                $down \leftarrow$ False, $up \leftarrow$ True, $pre \leftarrow p_1, pre\_coord \leftarrow j$
            end if
        end if
    end for
end for
```

and ③ feature the relevant pixels (in white) extracted by the algorithm. These images demonstrate that scanning the columns of the image (*cf.* ②) highlights horizontal creases, while scanning the rows (*cf.* ③) highlights vertical creases. Considering the CAD model of the cardboard (*cf.* Fig. 3), it is clear how pixels in correspondence of the creases are marked, almost forming lines. These pixels are evaluated to determine their presence. Defects are highlighted in green in ④. For example, the long vertical crease near the central hole, which is highlighted in green in ④, is not very visible in the aligned cardboard, resulting in a low number of highlighted pixels, thus marking it as a defect.

An important aspect to consider in this context, which is also visible in ③ in Fig. 8, is that creases along the flute direction of the corrugation are less

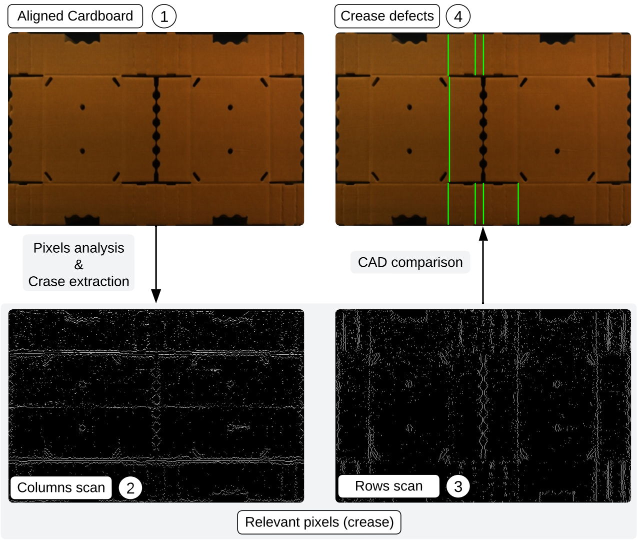


Fig. 8: Crease defects detection of cardboard

visible compared to transverse ones. This is because the width of vertical folds is smaller [26], making pixel transitions less evident.

4 Experiments

At the best of our knowledge, no other algorithm is available in the literature for the detection of cut and crease errors that can be compared with DIE-VIS. As already discussed in Sec. 2, while [26] share similar goals, the kind of sensors employed and the lack of extensive evaluation make a direct comparison unfeasible. For this reason, in this section, we compare the results obtained with our approach on a real manufacturing line with a baseline consisting of a fine-tuned YOLOv8 for different tasks. The source code employed to fine tune YOLOv8 and the evaluation results are available at `https://github.com/iaiamomo/cardboard_yolo`. For confidentiality reasons, we cannot instead disclose DIE-VIS source code.

From the software point of view, DIE-VIS has been implemented using Python and the OpenCV library. YOLOv8 validation has also been implemented in Python, leveraging the `ultralytics` library based on the Pytorch framework.

4.1 Baseline: YOLOv8

YOLO (You Only Look Once) [30] is a state-of-the-art learning-based model used in many domains for different applications, among which manufacturing and production processes for surface inspection to detect defects and anomalies to improve quality control [37]. The YOLO family includes several versions and variants that differ in characteristics such as network architecture, loss function, anchor box processing, and then input resolution scaling. Among them, we choose YOLOv8[2] as the baseline for our experiments as it is open source, well-documented, and well-integrated in the `ultralytics` Python library, allowing for easy training and validation. In addition, it supports different vision tasks, such as detection, segmentation, classification, pose estimation, and tracking. In our experiments, we consider the detection, segmentation, and classification tasks, evaluating the ability of YOLOv8 to detect, segment, and classify creases and holes in cardboard boxes.

To run the experiments, we fine-tuned the pre-trained YOLOv8 *detection*, *segmentation*, and *classification* models on a custom dataset [6]. Particularly, the *detection* and *segmentation* models are pre-trained on the MS COCO dataset [20], while the *classification* model is pre-trained on the ImageNet dataset [11]. The custom dataset is realized in a semiautomatic way using an algorithm that leverages CAD models to label a set of 1627 cardboard box images, which are then manually validated. The obtained dataset is detection-based however, the segmentation-based and classification-based are directly derived by converting the detection labels representing boxes into segments and cropped images respectively. The three datasets are also augmented with 90° clockwise and counterclockwise rotations and are balanced following 70%, 20%, and 10% allocation for the train, validation, and test sets.

The trainings are run on an NVIDIA DGX composed of NVIDIA A100 GPUs with 40GB of RAM. The training runs are characterized by 500 epochs with a patience parameter for early stopping set to 30 epochs. The image size is set to 640 and the batch size is set to 32. The learning rate strategy used is one cycle, with an initial learning rate $lr_0 = 0.01$ and a final learning rate of 0.001. The optimizer is SGD, with a momentum factor of 0.937 and weight decay of 0.0005. We train over the 5 different sizes of the YOLOv8 models, *i.e.*, `n` (nano), `s` (small), `m` (medium), `l` (large), and `x` (extra-large), which feature an increasing number of hyper-parameters.

4.2 Evaluation results

The evaluation is based on a dataset containing 250 images of defected and non-defected cardboard boxes [5] we retrieved from a real manufacturing line employing DIE-VIS and manually labeled.

For all approaches, we measure the precision P and recall R at recognizing the two classes, *i.e.*, `crease` and `hole`. A high precision value indicates a low rate of false positives, i.e., that most of the identified holes/creases are correct. A high recall value indicates a low rate of false negatives, i.e., that the identified holes/creases are most of the correct ones that should be identified.

Additionally, we measure mAP50-90 for the YOLOv8 detection and segmentation models and top1A for the YOLOv8 classification models. mAP50-90 is the

[2] `https://github.com/ultralytics/ultralytics`

Table 1: Experiments results

Approach	Task	$\mathrm{P}_{\mathrm{crease}}$	$\mathrm{R}_{\mathrm{crease}}$	$\mathrm{P}_{\mathrm{hole}}$	$\mathrm{R}_{\mathrm{hole}}$	$\mathrm{mAP50\text{-}95}_{\mathrm{all}}$	top1A
YOLOv8n	det	0.84	0.82	0.95	0.94	0.57	–
	cls	0.96	0.95	0.99	0.96	–	0.93
	seg	0.84	0.81	0.94	0.93	0.55	–
YOLOv8s	det	0.85	0.84	0.95	0.95	0.63	–
	cls	0.96	0.9	0.99	0.95	–	0.9
	seg	0.85	0.83	0.95	0.95	0.58	–
YOLOv8m	det	0.86	0.86	0.95	0.96	0.66	–
	cls	0.96	0.67	0.98	0.97	–	0.73
	seg	0.85	0.83	0.95	0.94	0.61	–
YOLOv8l	det	0.87	0.86	0.96	0.96	0.67	–
	cls	0.97	0.83	0.97	0.97	–	0.85
	seg	0.85	0.83	0.95	0.95	0.59	–
YOLOv8x	det	0.86	0.86	0.95	0.95	0.67	–
	cls	0.96	0.82	0.96	0.96	–	0.84
	seg	0.87	0.85	0.95	0.95	0.66	–
DIE-VIS	–	1.0	0.84	1.0	1.0	–	–

mean average precision averaged over Intersection over Union (IoU) threshold from 50% to 95% (at 5% steps). This metric provides a comprehensive evaluation by considering a wide range of IoU thresholds, capturing both high and low overlap between predicted and ground-truth boxes. Typically values of mAP50-95 above 0.3 are considered good, especially on complex datasets. In particular, top1A is computed only for classification models and refers to the conventional accuracy, *i.e.*, how much the model answers exactly as expected.

Tab. 1 reports the results of the experiments. We are interested in having a few false positives, *i.e.*, high precision, meaning we want a solution able to identify the presence of defects, *i.e.*, missing creases and holes.

DIE-VIS achieves perfect precision and recall for the hole class, and perfect precision with 0.84 recall for the crease class. This indicates that DIE-VIS excels in identifying all holes but misses some folds.

Among the different YOLOv8 versions, the classification models emerge as the best to be used in our application scenario. This is not surprising as in this case, YOLOv8 does not have to identify a region of interest. Classification works then in the most similar setting to DIE-VIS. Finally, interestingly, YOLOv8 models show similar performance across different sizes, indicating robustness regardless of model scale.

In the experiments, we are not providing the computational time. Provided that both approaches are suitable to work in real-time at the maximum speed of the cardboard manufacturing lines, which peaks up to 3 cardboard boxes per second, other aspects must be considered. The most important one is that the two approaches run on different types of processors: YOLOv8 leverages GPU, while DIE-VIS utilizes CPU. On the one hand, CPU-based solutions are more accessible in terms of cost and size. In contrast, high-performance GPU-based solutions are more difficult to find and often require more space and power consumption, which may not always be available in industrial settings. In practice, the employment DIE-VIS and YOLOv8 depends on the specific conditions of the industrial system to be monitored.

5 Discussion

In this work, we presented DIE-VIS, an automated vision inspection system based on designed features that leverages traditional CV techniques for detecting cardboard cut and crease defects in real-time. We evaluated it against a baseline consisting of different versions of the pre-trained YOLOv8 models for different tasks, namely detection, classification, and segmentation models. In particular, YOLOv8 models were fine-tuned on a custom cardboard-based dataset.

For the sake of space, this paper focuses only on the perception section of a wider industrial information currently deployed in several factories. The system includes Human-Machine Interaction component, session management, data management, and analysis components that are outside the scope of this paper.

To perform experiments, a freely available dataset has been constructed and labeled, which is, to the best of our knowledge, the only one available for the research community.

Evaluation shows how both our approach and the YOLOv8 based one reach good performance, with DIE-VIS outperforming YOLOv8 models. This is not surprising, as DIE-VIS was specifically designed with domain knowledge. Anyway, the obtained numbers need to be analyzed in the light of what is the intended application. In particular, from the point of view of the human operators working on the shop floor, the precision of the system is of the utmost importance. The presence of false positives may result in entire batches containing an undetected error leading to waste of material and money. The presence of false negatives is instead, even though annoying, acceptable, especially if the Human-Machine Interface provided allows to silent specific warnings. From this point of view, this paper proves how in many applications in the industrial scenario, application specific approaches, being able to reach precision close to or equal to 1.0, can be more easily introduced and accepted.

Acknowledgements

The DIE-VIS system is part of the TopKontrol quality monitoring system (see `https://ikarton.it/topkontrol/`). The work of Flavia Monti was supported by the agreement on “Agile&Secure Digital Twins (A&S-DT)”. Matteo Marinacci is supported by Thales Alenia Space and Regione Lazio, through the fellowship 35752-22066DP000000040-A0627S0030 *Utilizzo dell'Intelligenza Artificiale a supporto della progettazione e qualifica del prodotto spaziale*. The work of Francesco Leotta and Massimo Mecella is partially funded by MICS (Made in Italy—Circular and Sustainable) (PE00000004) Extended Partnership (CUP B53C22004130001) funded by the EU - NextGenerationEU PNRR MUR.

References

1. Alharbi, O.: Industry 4.0 operators: core knowledge and skills. Advances in Science, Technology and Engineering Systems Journal (2020)
2. Balestrino, A., Landi, A., Pacini, L.: Vision system for monitoring the production of corrugated cardboard. In: IEEE ICCAD. pp. 626–631. IEEE (2006)
3. Büchi, G., Cugno, M., Castagnoli, R.: Smart factory performance and Industry 4.0. Technological forecasting and social change **150**, 119790 (2020)
4. Canny, J.: A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence (6), 679–698 (1986)
5. cardspace: cardboard_testset_ dataset (2024), `https://universe.roboflow.com/cardspace/cardboard_testset_`
6. cardspace: hole_fold dataset (2024), `https://universe.roboflow.com/cardspace/hole_fold`
7. Chen, H., Pang, Y., Hu, Q., Liu, K.: Solar cell surface defect inspection based on multispectral convolutional neural network. Journal of Intelligent Manufacturing **31**(2), 453–468 (2020)
8. Chondronasios, A., Popov, I., Jordanov, I.: Feature selection for surface defect classification of extruded aluminum profiles. The International Journal of Advanced Manufacturing Technology **83**, 33–41 (2016)
9. Chu, M., Gong, R., Gao, S., Zhao, J.: Steel surface defects recognition based on multi-type statistical features and enhanced twin support vector machine. Chemometrics and Intelligent Laboratory Systems **171**, 140–150 (2017)
10. Cristani, M., Farenzena, M., Bloisi, D., Murino, V.: Background subtraction for automated multisensor surveillance: a comprehensive review. EURASIP Journal on Advances in signal Processing **2010**, 1–24 (2010)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNET: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
12. Ding, C., Pang, G., Shen, C.: Catching Both Gray and Black Swans: Open-set Supervised Anomaly Detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7378–7388 (2022)
13. Gao, Y., Li, X., Wang, X.V., Wang, L., Gao, L.: A Review on Recent Advances in Vision-based Defect Recognition towards Industrial Intelligence. Journal of Manufacturing Systems **62**, 753–766 (2022)
14. Hridoy, M.W., Rahman, M.M., Sakib, S.: A framework for industrial inspection system using deep learning. Annals of Data Science **11**(2), 445–478 (2024)
15. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Transactions on Information Theory **8**(2), 179–187 (1962)
16. Huang, S.H., Pan, Y.C.: Automated visual inspection in the semiconductor industry: A survey. Computers in industry **66**, 1–10 (2015)
17. Lee, S., Chang, L.M., Skibniewski, M.: Automated recognition of surface defects using digital color image processing. Automation in Construction **15**(4), 540–549 (2006)
18. Lee, X.Y., Vidyaratne, L., Alam, M., Farahat, A., Ghosh, D., Diaz, T.G., Gupta, C.: XDNet: A few-shot meta-learning approach for cross-domain visual inspection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4375–4384 (2023)
19. Li, W.C., Tsai, D.M.: Wavelet-based defect detection in solar wafer images with inhomogeneous texture. Pattern Recognition **45**(2), 742–756 (2012)

20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
21. Lins, R.G., Santos, R.E.d., Gaspar, R.: Vision-based measurement for quality control inspection in the context of Industry 4.0: a comprehensive review and design challenges. Journal of the Brazilian Society of Mechanical Sciences and Engineering **45**(4), 229 (2023)
22. Liu, Y., Xu, K., Xu, J.: Periodic surface defect detection in steel plates based on deep learning. Applied Sciences **9**(15), 3127 (2019)
23. Louhichi, H., Fournel, T., Lavest, J., Aissia, H.B.: Self-calibration of scheimpflug cameras: an easy protocol. Measurement Science and Technology **18**(8), 2616 (2007)
24. Manish, R., Venkatesh, A., Ashok, S.D.: Machine vision based image processing techniques for surface finish and defect inspection in a grinding process. Materials Today: Proceedings **5**(5), 12792–12802 (2018)
25. Nascimento, R., Martins, I., Dutra, T.A., Moreira, L.: Computer vision based quality control for additive manufacturing parts. The International Journal of Advanced Manufacturing Technology **124**(10), 3241–3256 (2023)
26. Ojer, M., Alvarez, H., Lajas, I., Larranaga, A., Amozarrain, L.: Automatic inspection of paperboard creases to improve the quality of the packaging process. The International Journal of Advanced Manufacturing Technology **125**(5), 2455–2466 (2023)
27. OpenCV: OpenCV: Template Matching. `https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html`, online; accessed July 2024
28. Osterrieder, P., Budde, L., Friedli, T.: The smart factory as a key construct of industry 4.0: A systematic literature review. International Journal of Production Economics **221**, 107476 (2020)
29. Pellicciari, M., Andrisano, A.O., Leali, F., Vergnano, A.: Engineering method for adaptive manufacturing systems design. IJIDeM (2009)
30. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
31. Rogalka, M., Grabski, J.K., Garbowski, T.: In-Situ Classification of Highly Deformed Corrugated Board Using Convolution Neural Networks. Sensors **24**(4), 1051 (2024)
32. Scheibel, B., Mangler, J., Rinderle-Ma, S.: Extraction of dimension requirements from engineering drawings for supporting quality control in production processes. Computers in Industry **129**, 103442 (2021)
33. Suppitaksakul, C., Rattakorn, M.: Machine vision system for counting the number of corrugated cardboard. In: 2014 International Electrical Engineering Congress (iEECON). pp. 1–4. IEEE (2014)
34. Suzuki, S., et al.: Topological structural analysis of digitized binary images by border following. Computer vision, graphics, and image processing **30**(1), 32–46 (1985)
35. Szeliski, R., et al.: Image alignment and stitching: A tutorial. Foundations and Trends® in Computer Graphics and Vision **2**(1), 1–104 (2007)
36. Tabernik, D., Šela, S., Skvarč, J., Skočaj, D.: Segmentation-based deep-learning approach for surface-defect detection. Journal of Intelligent Manufacturing **31**(3), 759–776 (2020)

37. Terven, J., Córdova-Esparza, D.M., Romero-González, J.A.: A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. Machine Learning and Knowledge Extraction **5**(4), 1680–1716 (2023)
38. Wang, J., Ma, Y., Zhang, L., Gao, R.X., Wu, D.: Deep learning for smart manufacturing: Methods and applications. Journal of manufacturing systems **48**, 144–156 (2018)
39. Xu, X., Lu, Y., Vogel-Heuser, B., Wang, L.: Industry 4.0 and Industry 5.0—Inception, conception and perception. Journal of manufacturing systems **61**, 530–535 (2021)
40. Youkachen, S., Ruchanurucks, M., Phatrapomnant, T., Kaneko, H.: Defect segmentation of hot-rolled steel strip surface by using convolutional auto-encoder and conventional image processing. In: 2019 10th International conference of information and communication technology for embedded systems (IC-ICTES). pp. 1–5. IEEE (2019)
41. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on pattern analysis and machine intelligence **22**(11), 1330–1334 (2000)
42. Zhao, Y.: OmniAL: A Unified CNN Framework for Unsupervised Anomaly Localization. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3924–3933 (2023)
43. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. vol. 2, pp. 28–31. IEEE (2004)
44. Zivkovic, Z., Van Der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern recognition letters **27**(7), 773–780 (2006)