Automating Image Quality Detection for Parcel Identification Systems: A Computer Vision Approach

Qiuyu Tian[1,*], Kun Wang[2], Hongwei Tang[1,2,3]
[1]Institute of Information Superbahn, Nanjing, China
[2]Institute of Computing Technology Chinese Academy of Sciences, Beijing, China
[3]University of Chinese Academy of Sciences, Nanjing, China
**Corresponding Author.*

Abstract: This paper presents an overview of the computer vision methodologies employed in an image quality detection project for parcel identification systems. The core objective of this project is to minimize erroneous chargebacks that occur when the system fails to capture accurate carton label information. Our developed defect detection framework replaces the labor-intensive and error-prone human validation process, thereby significantly reducing labor costs and improving the accuracy of defect identification in parcel identification images. The system uses a multi-sided vision tunnel to capture images of cartons from all angles. However, various issues such as dirty cameras, out-of-focus images, overly bright camera flashes, and partial images can impair image quality, leading to failures in automated parcel receiving. Each failure incurs additional costs for the logistics provider and may result in unwarranted chargebacks to partners, adversely affecting relationships. To address these challenges, we propose a computer vision-based approach to systematically identify and exclude poor-quality images from chargeback datasets, preventing erroneous chargebacks. This approach not only enhances the accuracy of automated receiving operations, but also supports downstream analyses to identify locations with frequent image quality issues. By partnering with these locations, preventive measures can be implemented to improve parcel image quality across the network. The outcomes of this project aim to streamline the automated receiving process, reduce operational errors, and foster better partner relationships by ensuring fair and accurate chargeback practices.

Keywords: Auto-receive; Image Quality; Vendor Chargeback; Computer Vision;

1. Introduction

The Parcel Identifier (PID) system is a sophisticated multi-sided vision tunnel initially developed and implemented in large-scale logistics operations, such as those used by companies like Amazon. This system captures images of each carton from all angles, which is crucial for predicting carton contents and facilitating the automated receiving process, as it is shown in Figure 1. Chargeback programs often leverage signals from such systems to penalize partners who fail to place valid carton labels. Precise carton label information is critical for automated processing of carton contents. However, inaccuracies in PID signals can lead to erroneous chargebacks, adversely impacting partner relationships and experiences.

Currently, the process of identifying the root causes of PID inaccuracies involves an exhaustive and meticulous manual image audit by the Chargeback Dispute Management team. This manual process is not only labor-intensive but also inherently prone to human error. When a parcel fails to be auto-received, a chargeback is automatically applied to the partner. If the partner disputes this chargeback, a manual review is undertaken to determine whether the issue was due to the absence of a valid label or poor image quality. Each instance of a carton failing the automated receiving process incurs additional costs. Common causes for the system's failure to accurately capture carton labels include dirty cameras, out-of-focus images, excessively bright camera flashes, and partial images.

The primary objective of this project is to replace the current manual validation process

with an automated, machine learning-based framework that enhances accuracy and reduces labor costs. By systematically identifying and excluding poor-quality images, we aim to prevent erroneous chargebacks and support downstream analyses to identify and collaborate with sites prone to image quality issues. This collaboration will enable the implementation of preventive measures to improve PID image quality across various regions and potentially extend to global retail and fulfillment entities.

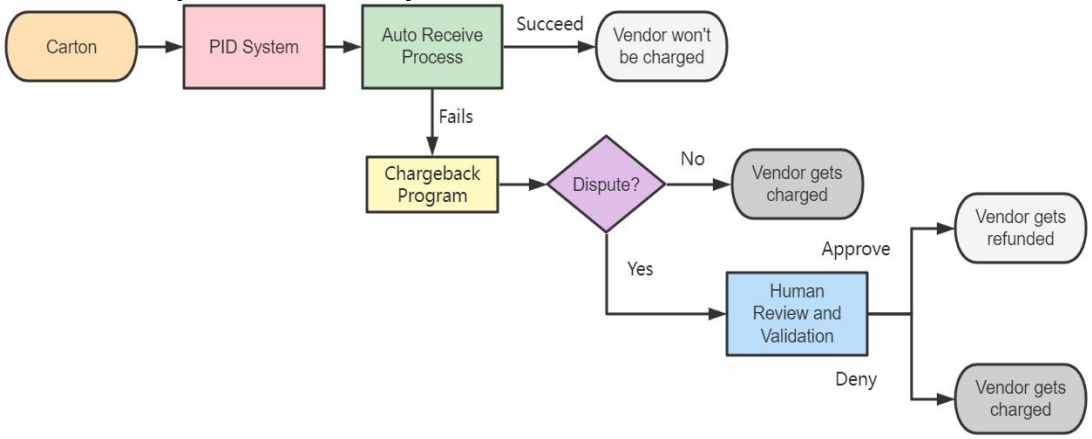

Figure 1. Flowchart of Current Chargeback Program and Dispute Process

To address these challenges, a comprehensive software system has been developed that leverages machine learning to systematically identify and exclude defective images from the chargeback process. This system categorizes defect types based on preprocessing steps and applies appropriate detection methods for each type. It sequentially processes each image by checking for bottom-side defects, generating brightness histograms for classification, and applying object detection techniques to extract and analyze carton and label areas. Among various evaluated models, we ultimately selected the Single Shot Multi-Box Detector (SSD) [1] for its balance of accuracy and response time, making it well-suited for real-time applications.

This project introduces an advanced software system that automates the validation process, significantly reducing the need for manual intervention. The system enhances accuracy in defect identification, thereby preventing erroneous chargebacks and improving partner relationships. Furthermore, it supports downstream analyses to identify sites with frequent image quality issues and implement preventive measures. By improving PID image quality, this project aims to streamline the automated receiving process, minimize operational errors, and contribute to a superior overall experience for partners. This system has been effectively implemented in Amazon's logistics centers, demonstrating its value in large-scale, real-world operations.

2. Problem Definition

2.1 Defect Types

The effectiveness of the Parcel Identifier (PID) system in accurately scanning carton labels is crucial for the automated receiving process. However, several factors can cause the PID system to fail in capturing clear and readable images of carton labels. These failures can be attributed to the following primary defect types:

- **Dirty Bottom Camera:** This occurs when the camera view is from the bottom side and the image appears blurry or fuzzy. This defect is often accompanied by dark or light vertical lines running throughout the image.
- **Blurry/Image Out-of-Focus:** In this case, the camera view is not from the bottom, but the label appears blurry, making the barcode and text unreadable.
- **Camera Flash too Bright:** This defect arises when the carton surface is highly reflective, causing the black text on the carton label to become excessively bright and unreadable due to the camera flash.
- **Partial/Cut-off Image:** This issue occurs when the image does not capture the entire carton (partial) or there exists a line cutting through the image, resulting in a dislocated or incomplete image (cut-off).
- **Black-out Image:** This defect is characterized by the camera being blacked out, displaying only the starting screen, or capturing an image where no carton is visible.

The above defect types are the most prominent reasons for the PID system's failure to scan carton labels effectively. Addressing these defects is essential to enhancing the accuracy and reliability of the automated receiving process, thereby reducing operational costs and improving partner relationships.

2.2 Challenges

2.2.1 Unreliable label quality

The training labels are generated by annotators using a data labeling tool that allows for the inclusion of human annotators from various sources. The images and label information are stored in a cloud storage solution. Each image is typically labeled by multiple annotators, leading to potential disagreements on whether an image is defective. Consequently, the label information is presented as a dictionary, with each key representing a defect type and the value being a confidence score (e.g., {Dirty Bottom Camera: 0.88, Blacked-out: 0.33, Partial/Cut-off Image: 0.22}).

Some images may be labeled both as defective and non-defective, such as {Dirty Bottom Camera: 0.34, No Defect: 0.91}. The confidence score indicates the proportion of annotators who consider an image to contain a specific defect type. Due to the varying nature of defects, some are more challenging for annotators to identify, resulting in lower average confidence scores. This variability in annotator judgment adds complexity to the training process and can affect the overall performance of the defect detection model. The confidence scores for different defect types are shown in Table 1.

Table 1. Confidence Scores for Different Defect Types

Defect Type	Average Score
Dirty Bottom Camera	0.805
Blurry/Out-of-Focus	0.472
Partial/Cut-off Image	0.539
Camera Flash too Bright	0.494
Black Out Image	0.778

The quality of our training labels depends on these confidence scores, which are subject to some level of uncertainty due to annotator disagreements. To maintain high label quality, we set a high threshold for the confidence score to consider an image as containing a certain defect. However, a higher threshold results in fewer positive instances (defective images), which can lead to model instability and over-fitting issues. Figure 2 and Table 2 below show the histogram of Blurry/Out-of-Focus defects and the breakdown relative to different thresholds.

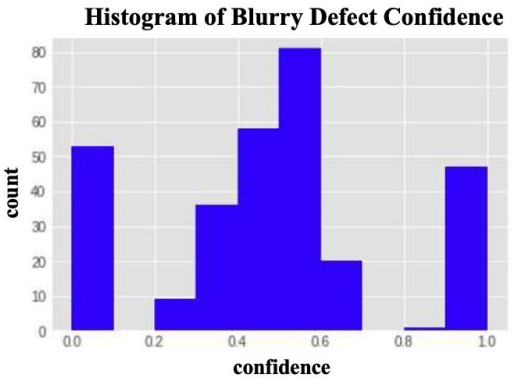

Figure 2. Histogram of Confidence Score Distribution for Blurry

Table 2. Confidence Scores Breakdown

Defect Type	Average Score
0-0.1	53
0.1-0.2	0
0.2-0.3	7
0.3-0.4	38
0.4-0.5	58
0.5-0.6	73
0.6-0.7	28
0.7-0.8	0
0.8-0.9	1
0.9-1.0	47

2.2.2 Insufficient training data

A total of 15,000 annotated images are available. Disregarding confidence scores and focusing solely on the appearance of defect types, the breakdown is as follows in Table 3:

Table 3. Defect Count Breakdown

Defect Type	Count
Dirty Bottom Camera	305
Blurry/Out-of-Focus	1255
Partial/Cut-off Image	95
Camera Flash too Bright	1311
Black Out Image	52

One obvious issue here is the uneven distribution among defect types, which means positive samples are not enough for some defect types. For example, there are only 52 images labelled as blackout image defect. This is primarily due to the difference in probabilities of the occurrence of those defects in our system. Based on the information we have, only Partial/Cutoff Image and Dirty Bottom Camera defects have decent number of

positive training samples.

However, collecting more annotated images may not resolve this issue, as most new images will likely be non-defective. Therefore, we need alternative methods to augment the existing images: Image Augmentation and Image Manipulation.

Image Augmentation. Image augmentation involves creating new training examples by slightly modifying the original images through transformations such as rotation, flipping, and color adjustments. Examples of commonly used image augmentation techniques in our scenario are shown in Figure 3.

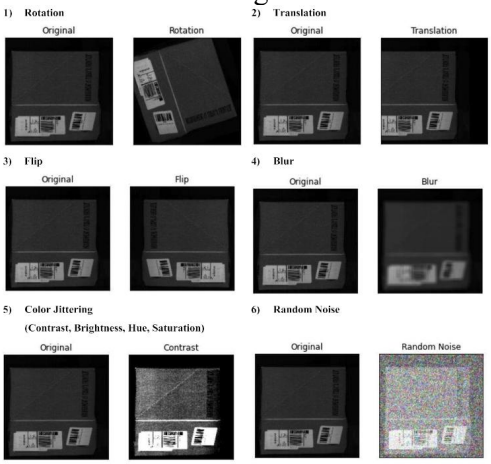

Figure 3. Examples of Image Transformations

Image Manipulation. Image manipulation aims to restore degraded image content or transform images to achieve desired outcomes. In the PID project, we implemented Deep Convolutional Generative Adversarial Networks (DCGANs) [2] to generate new positive instances for Black-out Image defects (Figure 4).

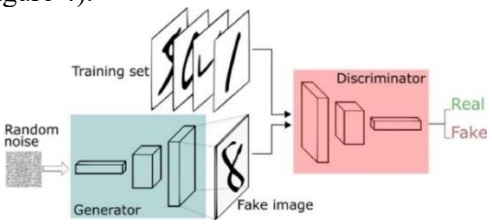

Figure 4. Deep Convolutional Generative Adversarial Networks (DCGANs)

2.2.3 Incorrect labeling logic

A Parcel Identifier (PID) image consists of three parts: 1) Carton, 2) Carton Label, and 3) Background (Figure 5). These parts have varying levels of significance in determining defect types and processes. For the automated receiving process, the priority is: Carton Label > Carton > Background. For defects such as Blurry/Out-of-Focus Image, Camera Flash too Bright, and Partial/Cutoff Image, considering the background when detecting these defects may result in an excessive number of unnecessary chargebacks (false positives), as the background does not contain any information relevant to the receiving process.

Figure 5. Different Parts of PID Image

However, a closer examination of the originally annotated images reveals that annotators often determine whether an image is defective based on the entire image, not just the carton part. This approach is misaligned with the problem definition and expectations. For instance, examples were found where images containing clear carton and carton label sections, but with blurry backgrounds, were labeled as Blurry/Out-of-Focus defects (Figure 6).

Figure 6. Example of Mislabeled Blurry Defect

To correct this labeling logic, additional labeling tasks were created with improved instructions to ensure that annotators focus on the relevant parts of the image. Annotators were instructed to relabel the affected images accordingly. Furthermore, an object detection model was developed to extract the carton and carton label sections, allowing for preprocessing of the images before training the defect prediction model. Details of the object detection model will be discussed in Section 4.

Moreover, Partial Image and Cut-off Image are inherently different defect types. In Cut-off defects, one or two horizontal lines cut through the image, while Partial defects occur when only part of the carton is visible, not the entire carton. Examples of these patterns are shown in Figure 7.

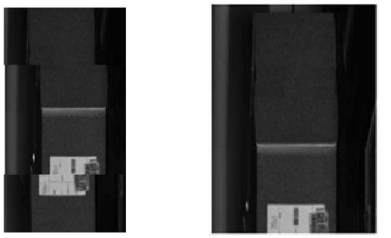

Figure 7. Examples of Different Patterns in Partial/Cut-off Image Defect

However, Partial Image and Cut-off Image defects consist of only 52 images in total, resulting in insufficient data, as described in Section 3.2. To address this issue, non-defective images were modified through random line-cutting and partial region extraction to generate additional training samples. By tackling these challenges in labeling logic, the goal is to improve the accuracy and reliability of the defect detection models, ultimately enhancing the overall performance of the PID system.

3. Methods

3.1 Overview

The identification and classification of image defects in the PID system involve a series of preprocessing steps tailored to the specific characteristics of each defect type. For instance, defects such as Blurry/Out-of-Focus and Partial-Cutoff Image focus primarily on the carton part of the image, necessitating the use of a carton detector (extractor) as a preprocessing step. The procedure for detecting each defect type is illustrated in Figure 8.

1) Initial Check: The system first determines if the image is from the bottom side. If it is, the image is processed using the Dirty Bottom Camera prediction model.

2) Brightness Analysis: If the image is not from the bottom side or does not exhibit a Dirty Bottom Camera defect, the system generates a histogram of the image's brightness distribution. This histogram is used to classify the image as Reflective/Flash too Bright, Blacked-out, or non-defective. Note that an image cannot be classified as both Reflective and Blacked-out, as these defects are mutually exclusive in terms of brightness distribution.

3) Object Detection: Images preliminarily classified as non-defective undergo further analysis using an object detector model to extract the carton and carton label areas.

4) Detailed Defect Analysis: For each carton area extracted, the system applies specific prediction models to check for Partial Image defects or Blurry defects. If neither defect is identified, the image is finally classified as non-defective.

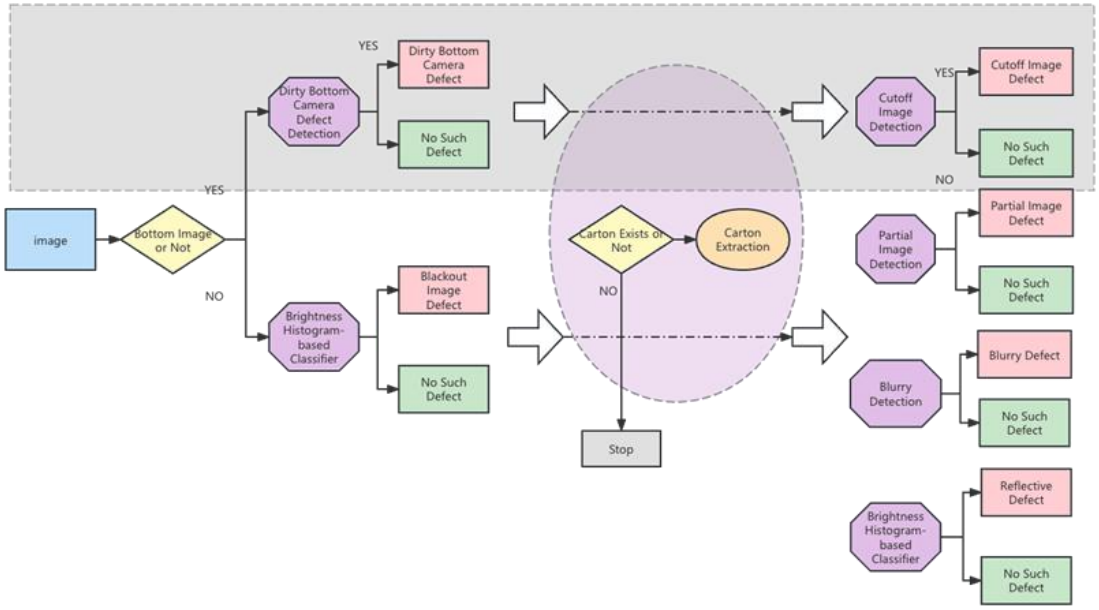

Figure 8. Designed PID Image Quality Detection Architecture

Based on the definitions of each defect type, the following potential solutions are proposed

in Table 4:

Table 4. Solution Overview

Defect Type	Solutions
Dirty Bottom Camera	CNN-based Classifier
Blurry/Out-of-Focus	Carton Label Extraction + CNN-based Classifier
Partial Image	Carton Extraction + CNN-based Classifier
Cut-off Image	Carton Extraction + CNN-based Classifier
Camera Flash too Bright	1) Brightness-based ML model 2) Carton Extraction + CNN-based Classifier
Black Out Image	1) Brightness-based ML model 2) CNN-based Classifier

This comprehensive detection architecture ensures that each defect type is accurately identified and classified, utilizing the most appropriate machine learning techniques. The integration of both traditional and advanced models, such as Convolutional Neural Networks (CNNs) [3] and brightness-based classifiers, enhances the robustness and precision of our defect detection system.

By systematically addressing the preprocessing needs and classification criteria for each defect type, our methodology improves the reliability of the PID system, ultimately reducing erroneous chargebacks and enhancing overall operational efficiency.

3.2 Object Detection Model

To enhance the accuracy of the defect detection system, an object detection model was implemented to identify and extract carton and carton label areas from the images. A random sample of 1,600 images was selected from a dataset of 15,000 images, and bounding box annotation tasks were created using a data labeling tool. An example of the annotation output and the expected outcome of the object detection model are illustrated in Figure 9.

The annotation results are provided in a dictionary format. Under the key "annotation", there are three dictionaries with different "class_id" values. The dictionary with "class_id: 0" represents the carton area, while the other two with "class_id: 1" represent the carton label parts. It is important to note that the label on the top-right part of the image is not a carton label, as it lacks a barcode.


```
{
  "source-ref": "s3://pid-image-defect/bounding_box/task-
1/DATALOGIC_TUNNEL_MDW2_PID-1_20210629_002654605_F7FB7D3E9FCFE773_155_TP.jpg",
  "BoundingBoxAnnotation-Part1": {
    "image_size": [{"width": 3360, "height": 6248, "depth": 3}],
    "annotations": [
      {"class_id": 0, "top": 320, "left": 49, "height": 5257, "width": 3311},
      {"class_id": 1, "top": 867, "left": 433, "height": 1254, "width": 851},
      {"class_id": 1, "top": 3510, "left": 510, "height": 806, "width": 1206}
    ]
  },
  "BoundingBoxAnnotation-Part1-metadata": {
    "objects": [{"confidence": 0}, {"confidence": 0}]
  },
  "class-map": {"0": "Carton", "1": "CartonLabel"},
  "type": "groundtruth/object-detection",
  "human-annotated": "yes",
  "creation-date": "2022-04-25T20:24:04.881350",
  "job-name": "labeling-job/boundingboxannotation-part1"
}
```


Figure 9. Sample Annotating Result and Its Corresponding Image

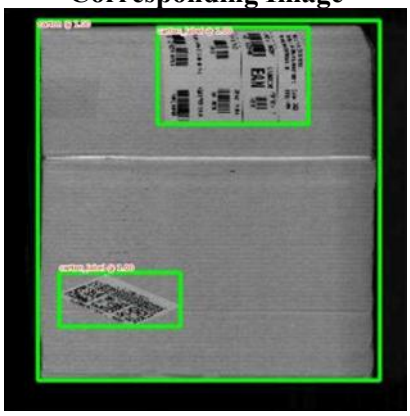

Figure 10. Illustration of Object Detection Outcome

Given that the resolution of images may change during the modeling step, we need to scale the bounding box coordinates to maintain consistency. The original format in the annotation job is (left, top, width, height), where (left, top) represents the coordinates of the upper-left corner of the carton bounding box. The following transformations are implemented to convert these coordinates into a model-ready format, expressed as the ratios of the upper-left corner$(X_1\%, Y_1\%)$ and lower-right corner $(X_2\%, Y_2\%)$ of the image:

$$X_1 = \frac{\text{left}}{\text{width}},$$
$$Y_1 = \frac{\text{top}}{\text{height}},$$
$$X_2 = \frac{\text{left} + \text{width}}{\text{width}},$$
$$Y_2 = \frac{\text{top+height}}{\text{height}} \quad (1)$$

The objective function for a typical object detection model aims to optimize both classification loss and regression loss.

1. **Classification Loss:** The classification loss

(L_{cls}) measures the error in predicting the class probabilities of objects. It is often computed using the Cross Entropy Loss (CE Loss), which is defined as:

$$L_{\text{cls}} = -\sum_i [p_i \log(\hat{p}_i) + (1 - p_i) \log(1 - \hat{p}_i)] \quad (2)$$

where p_i is the true probability of class i and $\hat{p}_i$ is the predicted probability.

2. **Regression Loss:** The regression loss (L_{reg}) measures the error in predicting the coordinates of the bounding boxes. It is typically computed using the Smooth L1 Loss (also known as Huber Loss), which is defined as:

$$L_{\text{reg}} = \sum_i \text{smooth}_{L1}(t_i - \hat{t}_i) \quad (3)$$

where t_i is the true bounding box coordinate and $\hat{t}_i$ is the predicted bounding box coordinate. The smooth L1 loss function is given by:

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & otherwise \end{cases} \# \quad (4)$$

The total loss (L) for the object detection model is the sum of the classification loss and the regression loss:

$$L = L_{\text{cls}} + L_{\text{reg}} \quad (5)$$

There are several object detection models available for implementation, including R-CNN based models (R-CNN[4], Fast R-CNN[5], and Faster R-CNN[5]), YOLO[6], and SSD. While these models generally perform similarly in terms of accuracy in simple contexts (with a small number of objects in the same background), their prediction response times vary significantly due to differences in model architectures.

Given our requirements for real-time defect detection and the comparative performance metrics, we selected the Single Shot Multi-Box Detector (SSD) model for our implementation. The SSD model provides an optimal balance between accuracy and computational efficiency, making it well-suited for applications requiring quick and reliable defect detection. Detailed performance comparisons of the selected models will be presented in Section 5.

3.3 Defect Prediction Model

To address the various defect types identified in the PID system, we employed two primary types of prediction models: deep learning-based CNN classifiers and traditional machine learning-based brightness histogram models. The CNN classifiers were implemented using both basic and transfer learning approaches to leverage the advantages of pre-trained models.

3.3.1 The CNN-based classifier

The CNN-based classifier is designed to automatically identify defects by learning feature representations from the input images. We configured the basic CNN classifier with multiple convolutional layers for feature extraction and fully connected layers for classification. In addition, we employed transfer learning techniques using pre-trained models such as ResNet18 [7] and VGG16 [8] to enhance the model's performance. Transfer learning significantly reduces training time and improves generalization, especially in scenarios with limited data.

3.3.2 Brightness-based machine learning models

Image data is represented as a matrix of brightness values ranging from 0 to 255, with a shape of $h \times w \times c$, where h is the height, w is the width, and c is the number of channels. Each element in this matrix is called a pixel. For colored images, there are three channels: red, green, and blue ($c = 3$), whereas for grayscale images, there is only one channel ($c = 1$). Channels function as color planes. The brightness-based machine learning (ML) classifier is utilized to detect defects such as Camera Flash too Bright/Reflective and Blacked Out Image. We expect the former to exhibit a large portion of high pixel values (as 255 represents white), and the latter to have a significant portion of low pixel values (0 represents black). Figure 10 illustrates the brightness distribution histogram for an image labeled as Camera Flash too Bright/Reflective compared to a non-defective one.

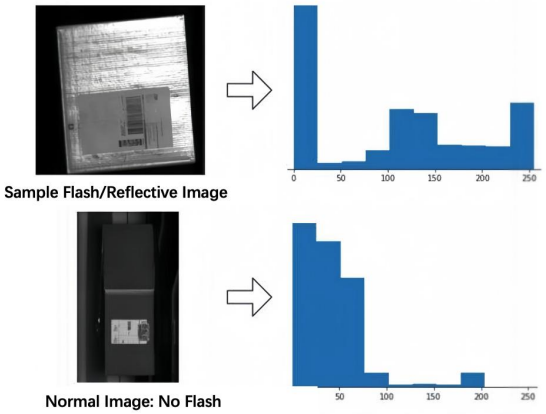

Figure 11. Brightness Histogram Comparison

Figure 11 shows the histograms of brightness

distribution for reflective, blacked-out, and non-defective images. The main differences among these histograms are observed in the regions of low and high brightness values. Additionally, brightness values in reflective and non-defective images are more mixed, while the distribution for blacked-out images is highly skewed, as it is shown in Figure 12.

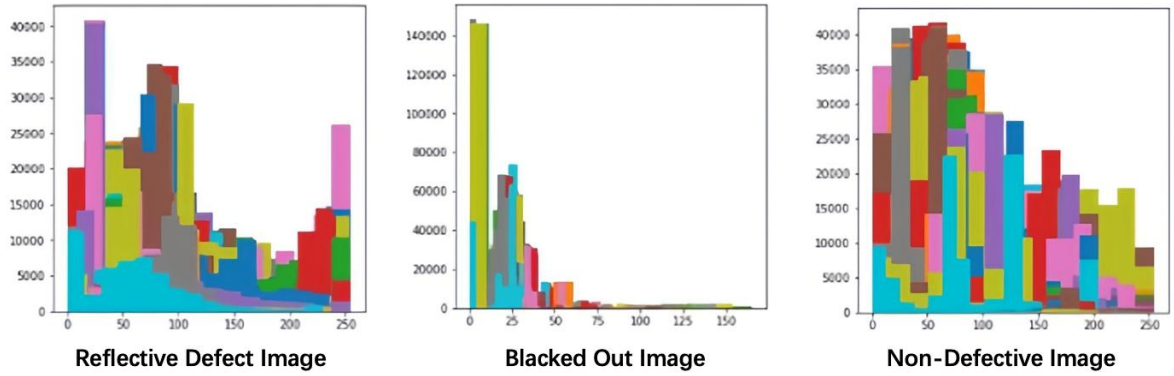

Figure 12. Concatenated Histograms of Brightness Distribution

To build an effective machine learning model, we need to perform feature engineering to extract representative features that distinguish different brightness distributions. We extract three types of features as shown in Table 5:

1) Histogram Bin Heights
2) Quantile Differences
3) General Statistics

Table 5. Selected Features

Features	Type	Description
B1, B2, …, B15	Histogram Bin Heights	The histogram of each image is set to have 15 bins, and B1~B15 represents the heights of those bins.
Bin Entropy	Quantile Difference	Entropy of bin values
IQR	Quantile Difference	Q 75% - Q 25%
Lower-QR	Quantile Difference	Q 25% - Minimum
Upper-QR	Quantile Difference	Maximum - 75%
Median	General Statistics	Q 50%
Q1	General Statistics	Q 25%
Q3	General Statistics	Q 75%

For the brightness-based approach, we considered several supervised machine learning models due to the continuous nature of the features. These models included: (1) Logistic Regression (Logit-R) [9], (2) Support Vector Machine (SVM) [10], (3) K-Nearest Neighbors (KNN) [11], and (4) Naïve Bayes (NB) [12]. After evaluating the performance of these models, Gaussian Naïve Bayes was ultimately selected for its superior performance in this specific application. Detailed comparisons of model performance will be presented in Section 5.

4. Experiments and Results

4.1 Model Evaluation Metrics

In evaluating the performance of the models, Recall, Precision, and the area under the Receiver Operating Characteristics curve (ROC AUC) are utilized as primary metrics [13]. The ROC AUC is a comprehensive measure of a model's ability to distinguish between classes. The following terminologies are essential for understanding these metrics:

1) **True Positive (TP):** An image correctly predicted as defective.
2) **True Negative (TN):** An image correctly predicted as non-defective.
3) **False Positive (FP):** An image incorrectly predicted as defective.
4) **False Negative (FN):** An image incorrectly predicted as non-defective.

The Recall and Precision are defined below:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (6)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (7)$$

The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR). The TPR is equivalent to Recall, while the FPR is defined as:

$$\text{FPR} = \frac{\text{FP}}{\text{FP+TN}} \quad (8)$$

The area under the ROC curve (AUC) provides a single value to compare classifier performance. An ideal classifier achieves a

TPR of 1.0 and an FPR of 0.0, resulting in an AUC of 1.0. Conversely, random guessing yields an AUC of 0.5.

4.2 Model Performance

4.2.1 Object detection model

Four object detection models were implemented and evaluated: R-CNN, Faster R-CNN, YOLO, and SSD. Table 6 summarizes their performance metrics, including Mean Prediction Time (in seconds), Precision, Recall, and F1-Score.

Table 6. Comparison of Different Object Detection Models

Model	Mean Pred Time	Precision	Recall	F1
R-CNN	1.88	0.86	0.82	0.84
Faster R-CNN	0.127	0.90	0.85	0.93
YOLO	0.041	0.95	0.93	0.95
SSD	0.034	0.96	0.93	0.95

4.2.2 Defect classification model

The performance of the defect classification models was evaluated based on Recall, Precision, and ROC AUC. Additionally, the mean prediction time was considered, as quick result generation is expected in production. The final model performances for each defect type are provided below:

1) Dirty bottom camera:

Table 7. Model Performance for Dirty Bottom Camera Defect

	Basic CNN	ResNet18	VGG16
Recall	0.69	0.77	0.87
Precision	0.92	0.88	0.77
ROC AUC	0.82	0.85	0.86
Mean Pred Time	0.26	0.43	0.44

Table 7 indicates the model performance for dirty bottom camera defect. Recall is prioritized over Precision, and the VGG16-based Transfer Learning model demonstrated the highest ROC AUC. Consequently, the VGG16 model was selected as the final model for this defect type. It is worth noting that some experiments included applying Gaussian High Pass filters as preprocessing steps to enhance model performance in predicting Dirty Bottom Camera defects.

2) Blurry/out-of-focus:

Table 8. Model Performance for Blurry/Out-of-Focus Defect

	Basic CNN	ResNet18	VGG16
Recall	0.94	0.94	0.95
Precision	0.91	0.92	0.93
ROC AUC	0.93	0.93	0.94
Mean Pred Time	0.24	0.41	0.43

Table 8 illustrates the model performances for blurry/out-of-focus defect. The three models demonstrate similar performance in terms of Recall, Precision, and ROC AUC. However, the basic CNN classifier's prediction time is significantly lower, averaging 0.24 seconds. Consequently, we selected the basic CNN model as the final model for this defect type.

3) Partial/cutoff image:

Two separate models were developed for this defect type: the partial defect model and the cutoff defect model. These models were trained with manipulated positive instances and achieved nearly perfect performance. To ensure a fair estimation, original images were used for testing; if either model predicted the image as defective, it was considered to contain a Partial/Cutoff defect. The aggregated model performances are as follows in Table 9:

Table 9. Model Performance for Partial/Cutoff Defect

	Basic CNN	ResNet18	VGG16
Recall	0.76	0.78	0.75
Precision	0.83	0.84	0.88
ROC AUC	0.80	0.82	0.83
Mean Pred Time	0.25	0.43	0.42

The ResNet18-based Transfer Learning model was selected as the final model for the Partial/Cutoff Image defect due to its superior performance.

4) Blacked out image:

Given the limited number of images (52) labeled as containing Blacked Out Image defects, we employed image manipulation techniques to generate additional training samples. Due to the nature of this defect type, it is relatively easier for models to detect, resulting in nearly perfect performance, as it is shown in Table 10. The comparisons are primarily among the basic CNN classifier and several classical machine learning models.

Table 10. Model Performance for Blacked Out Image Defect

	Basic CNN	SVM (rbf)	Logit-R	KNN	GNB
Recall	1.0	1.0	1.0	0.995	0.951
Precision	1.0	1.0	1.0	0.992	0.971
Mean Pred Time	0.23	0.18	0.03	0.32	0.14

5) Camera flash too bright/reflective:

The Camera Flash too Bright/Reflective defect is similar to the Blacked Out Image defect in that both can be detected using brightness-based machine learning models and a CNN classifier. The structures and feature engineering steps for these two defect types are identical, with the exception that carton extraction is applied as a preprocessing step for the Camera Flash too Bright/Reflective defect.

Table 11. Model Performance for Camera Flash too Bright/Reflective Defect

	Basic CNN	SVM (rbf)	Logit-R	KNN	GNB
Recall	0.88	0.73	0.79	0.84	0.72
Precision	0.85	0.70	0.81	0.82	0.71
Mean Pred Time	0.30	0.21	0.07	0.36	0.19

As it is shown in Table 11, unlike the Blacked Out Image defect, machine learning models do not perform as well overall for this defect type. The only model with both Precision and Recall above 0.8 is the K-Nearest Neighbors (KNN) classifier; however, its mean response time, along with Recall and Precision, is inferior to that of the basic CNN classifier. Therefore, the basic CNN classifier was selected as the final model for the Camera Flash too Bright/Reflective defect.

5. Conclusion and Future Steps

In this project, six models were developed (two specifically for Partial/Cutoff defects) to address five key image quality issues identified in the Parcel Identifier (PID) system. These models were integrated into a comprehensive prediction pipeline that sequentially checks for the five defect types upon receiving a PID image, returning a list of potential defects. The implemented models demonstrated strong performance, achieving ROC AUC scores above 0.8, indicating high accuracy in defect detection. However, there is significant potential for enhancement through more extensive image preprocessing, model tuning, and data collection efforts.

This initiative reflects a commitment to leveraging machine learning approaches to address image anomaly problems within parcel identification systems. The successful deployment of these models can significantly improve operational efficiency by reducing erroneous chargebacks and enhancing the accuracy of the automated receiving process.

Looking ahead, the project scope is expected to expand to include a wider range of use cases and business entities. The next phase will focus on more complex issues, such as one label covering another and masking essential information like barcodes. By exploring advanced solutions for these emerging challenges, the aim is to further refine the system and broaden its applicability.

Additionally, future work will involve continuous improvement of the current models through advanced techniques in image preprocessing and feature extraction. Integrating additional data sources will enhance the robustness of the models. Collaborations with other teams and stakeholders will be crucial in identifying new defect types and developing innovative solutions to maintain high standards of image quality and operational efficiency.

Acknowledgments

This research was supported by The Research of Key Technologies and Industrialization for Intelligent Computing in Large-Scale Video Scenarios under Digital Social Governance of Henan, China (grant number 241100210100).

Reference

[1] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.

[2] Pinheiro Cinelli, L., Araújo Marins, M., Barros da Silva, E. A., & Lima Netto, S. (2021). Variational autoencoder. In Variational Methods for Machine Learning with Applications to Deep Networks (pp. 111-149). Cham: Springer International Publishing.

[3] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12), 6999-7019.

[4] Ross, Girshick. "Rich feature hierarchies for accurate object detection and semantic segmentation". Proceedings of the IEEE Conference on Computer Vision and

Pattern Recognition. IEEE. pp. 580–587. arXiv: 1311.2524. doi: 10.1109/CVPR.2014.81. ISBN 978-1-4799-5118-5. S2CID 215827080 (2014).

[5] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6), 1137-1149.

[6] Redmon, J. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition.

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[8] Liu, S., & Deng, W. (2015, November). Very deep convolutional neural network based image classification using small training sample size. In 2015 3rd IAPR Asian conference on pattern recognition (ACPR) (pp. 730-734). IEEE.

[9] Hosmer, David W.; Lemeshow, Stanley. Applied Logistic Regression (2nd ed.). Wiley. ISBN 978-0-471-35632-5. (2000)

[10] Cortes, Corinna; Vapnik, Vladimir. "Support-vector networks" (PDF). Machine Learning. 20 (3): 273–297. CiteSeerX 10.1.1.15.9362. doi:10.1007/BF00994018. S2CID 206787478. (1995)

[11] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

[12] Webb, Geoffrey I., Eamonn Keogh, and Risto Miikkulainen. "Naïve Bayes." Encyclopedia of machine learning 15.1 (2010): 713-714.

[13] Powers, David M W. "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation" (PDF). Journal of Machine Learning Technologies. 2 (1): 37–63. (2011).

