Predge

White Paper 2025: 02

Automated Detection of Foreign Objects in Bulk Material Transportation Using Image Thresholding and Transfer Learning with a Convolutional Neural Network from a Video Stream

Vanessa Meulenberg, Johan Öhman, Wolfgang Birk (Predge);Rune Nilsen (LKAB)

AUTHORED BY PREDGE AB

Automated Detection of Foreign Objects in Bulk Material Transportation Using Image Thresholding and Transfer Learning with a Convolutional Neural Network from a Video Stream

Vanessa Meulenberg
Predge AB
Luleå, Sweden
vanessa.meulenberg@predge.se

Johan Öhman
Predge AB
Luleå, Sweden
johan.ohman@predge.se

Wolfgang Birk
Predge AB
Luleå, Sweden
wolfgang.birk@predge.se

Rune Nilsen
LKAB
Narvik, Norway
rune.nilsen@lkab.com

ABSTRACT

Belt conveyor systems are widely used in industrial settings for bulk material transport, valued for their high capacity and minimal reliance on manual intervention. However, foreign objects on the conveyor can pose significant risks to the system's operation and downstream equipment. Given the critical role these systems play in maintaining production efficiency, along with their high level of automation and limited opportunities for manual inspection, there is a clear need for continuous, fully automated monitoring, which this paper aims to address.

A multi-step solution has been developed which utilizes video stream data from a camera positioned above a conveyor belt. Frames are pre-processed to determine whether the belt is operating, and the edges of the bulk material are detected to establish a region of interest (ROI) for object detection. Each frame is then passed through an object detection pipeline, incorporating several image processing techniques. Edge detection of the ore to establish an ROI is occasionally inconsistent, leading to false detections at the ore edges due to dark thresholding. To mitigate these false detections, a convolutional neural network (CNN) with pre-trained weights from the MobileNetV2 model was implemented as the final step in the object detection process.

This object detection system is deployed on an edge device at LKAB Narvik and connected to a SaaS platform that provides predictive maintenance and decision support. The system offers direct insights into whether operations should be halted based on the size and composition of detected objects. Results show a significant reduction in false detections, particularly at the ore edges, and the combination of light and dark thresholding allows for the detection of both high- and low-intensity objects.

Keywords

Computer vision, image analysis, convolutional neural networks, transfer learning, autonomous decision making, object detection

1 INTRODUCTION

Belt conveyor systems are widely used in industrial settings for bulk material transport, valued for their high capacity and minimal reliance on manual intervention. In Narvik harbour, LKAB has a dense system of conveyor belts which transports iron ore from the trains that come in from the LKAB mines in Kiruna, to the ships that distribute the ore across the globe. The conveyor systems, key logistical elements in the production chain, are exposed to harsh environments. Freezing temperatures, dust, humidity and excessive loading are some examples of what the conveyor belts must endure [1][2].

In addition to the harsh environmental conditions, foreign objects on the conveyor can pose significant risks to the system's operation and downstream equipment. Belts can be ripped by sharp objects, for example, or large concrete blocks can get trapped in the housing of smaller conveyor belts downstream, causing rips, material build up and damage to surrounding components. When damages occur in the downstream transportation where the conveyor belts become less accessible, the effects are usually realised too late, and the devastating consequences are a stop in production and the repair and replacement of parts, especially costly belts [3]. On top of that, objects need to be detected and reacted upon in a timely manner. Previous work done consisted of a model that depended on brightness and contour detection using colour channel weighing and processing by a Convolutional Neural Network (CNN) [4][5]. A limitation of the current approach is that dark objects are rarely detected. The algorithms usually consider those as part of the ore background. Currently, dark object detection relies on chance sightings through existing surveillance or manually inspecting sieves for trapped objects. This approach is time-consuming, may be too late to prevent damage, and is unreliable since objects can bypass the sieving system.

Object detection using classical methods, such as brightness and contour detection algorithms, is commonly applied in static, unchanging environments. However, detecting objects in complex environments where conditions constantly change can be challenging. As a result, a combination of methods is often required to address

various scenarios effectively.

The first step in the proposed methodology is to determine whether the conveyor belt is in operation. If the belt is stationary and unloaded, running the algorithm becomes unnecessary. When the belt is in operation, the Hough Line Transformation is employed to define a region of interest (ROI), as detection outside the ore edges is ineffective and may lead to false detections. A similar method is described by Sun and Sun [6], where several machine vision techniques are used to detect the edges of a conveyor belt.

Subsequently, frames are preprocessed by applying Gaussian blurring for noise reduction, followed by Contrast Limited Adaptive Histogram Equalization (CLAHE) to enhance image contrast. To detect both dark and light objects, binary thresholding is applied, followed by morphological operations such as erosion and dilation to remove noise. Contour detection is then performed. To mitigate false detections, a convolutional neural network (CNN) with transfer learning, leveraging the pre-trained MobileNetV2 model, is implemented as the final detection step.

While transfer learning has been extensively applied to image classification tasks in fields such as medicine and agriculture [7], its application in industrial scenarios, such as iron ore mining and conveyor systems, remains limited. MobileNetV2 is widely adopted for image classification due to its efficiency and suitability for mobile and edge devices [8]. Zhang et al. [9] successfully combined the YoloV4 and MobileNet networks to detect conveyor belt damages in the mining industry, achieving an accuracy of 93.22%. Similarly, Meng et al. [10] utilized a combination of ECANet and ShuffleNetv2 to detect foreign objects on coal mine conveyor belts, achieving an accuracy of 97.62%.

The methods described above form the basis of the object detection methodology used to alert operators within a Software-as-a-Service (SaaS) platform. This paper outlines the results of the CNN model and evaluates the performance of the entire solution in a production environment.

2 METHODS

Above one of the conveyor belts at LKAB Narvik's boat loading sites, a video camera (Axis P1455-LE) was installed. The camera is connected to the local area network, which also hosts an edge device running the object detection algorithm. The architecture of this setup is illustrated in Figure 1. Frames captured by the camera are stored in a queue until a worker thread becomes available to process them. The frame processing and detection scheme comprises several steps, which are detailed in the following sections. Many of the methods described in this section were implemented using OpenCV [11], with the underlying theory provided by Bradski and Kaehler [12]. The CNN was implemented using the Tensorflow library [13].

2.1 Belt Movement Detector

The belt movement detector captures two frames from the queue every five seconds and compares them by calculating the absolute difference of each corresponding element in the image matrices. These differences are then averaged, and if the average exceeds a predefined threshold, the belt is considered to be moving. Only frames from a moving belt are processed further.

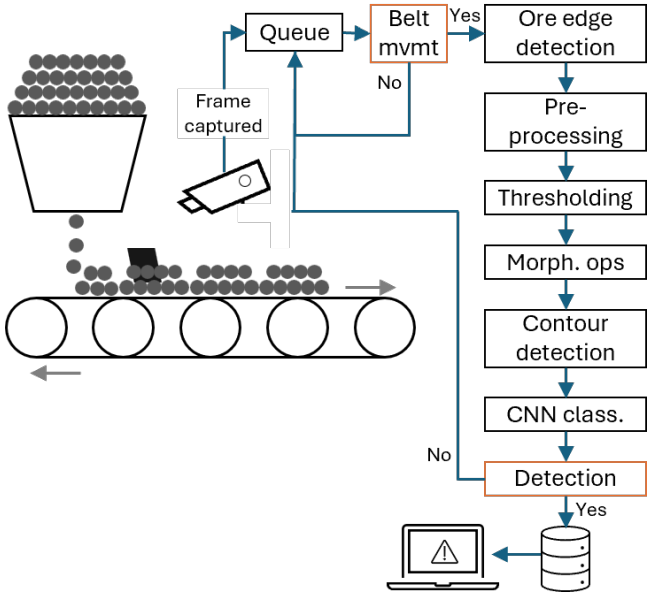


Figure 1: Architecture of the object detection scheme.

2.2 Ore Edge Detection

The ore edges (see the leftmost and rightmost images in Figure 2a for an example of the ore edges on the conveyor belt) are identified to determine a detection region within the frame (an ROI). This approach eliminates noise from areas surrounding the ore bed. Before detecting the ore edges, a Gaussian denoising filter is applied to the image, and Canny Edge Detection [14] is performed to identify all edges in the image.

The edges in the Canny image were then used to perform a Hough Line Transform [15], which detects straight lines within the frame. For each white point in the Canny image, a family of lines is generated that pass through that point by varying the angle θ. Each line is represented in polar coordinates as $\rho = x\cos\theta + y\sin\theta$. When multiple lines from different points intersect at the same (θ, ρ) coordinates in the parameter space, it indicates the presence of a straight line in the original image. A threshold is applied to determine the minimum number of intersecting points required to confirm the existence of a line. The ore edges are often distinct and can be effectively detected as lines using this method. Among the detected lines, the innermost ones are selected since the belt edges are also straight lines identified by the Hough Line Transform. The coordinates of the detected ore edges are used in the next detection steps as the region which to consider for the object detection.

2.3 Image Pre-Processing

Using the coordinates from the previous steps, the surrounding areas of the frames were masked out (their pixel values set to zero) and the image was converted to grayscale. Next, the CLAHE technique was applied to improve the contrast of the frames while reducing the risk of amplifying noise and artifacts. CLAHE builds on adaptive histogram equalization (AHE) and standard histogram equalization (HE) techniques. In standard HE, the intensity histogram of each frame is spread out by computing the cumulative

distribution function and then mapping the original pixel intensities accordingly. Because standard HE uses a global histogram, local bright or dark spots might be washed out. AHE addresses this issue by dividing the image into blocks and performing histogram equalization on each block independently, then blending block boundaries. However, this block-by-block approach can over-amplify noise, producing unrealistic contrast. CLAHE limits the contrast in each block using a "clip limit." Any bin in the block's histogram that exceeds this clip limit is truncated, and the excess is redistributed among other bins. This prevents a single intensity from dominating the histogram and mitigates noise amplification. Finally, to further reduce the graininess of the iron ore, the frames were denoised using a Gaussian blurring filter.

2.4 Object Detection

After pre-processing, the frames are passed through the object detection scheme which involves thresholding, morphological operations, contour detection and the final CNN classification step.

2.4.1 Thresholding, Morphological Operations, and Contour Detection

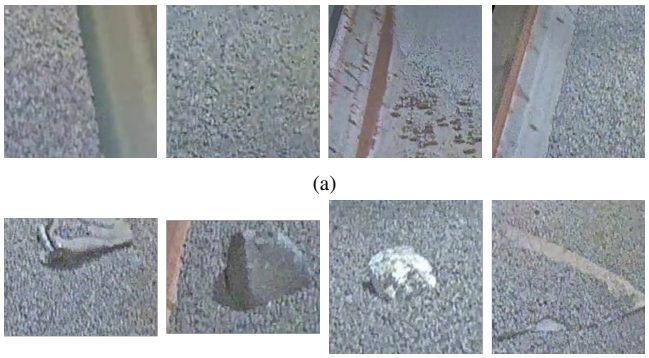
Thresholding is a simple technique in which pixel values are set to zero if they are smaller than or equal to a predefined threshold. In this work, each frame was thresholded twice: once to detect dark objects and once to detect bright objects. The two resulting thresholded images were then combined into a single binary mask, such that any detected object (dark or bright) appears white against a black background.

Next, the combined binary mask underwent several morphological transformations: Erosion (removal of small artifacts an noise), Dilation (compensation for the removed boundaries of larger objects during erosion) and Closing (filling in black holes that are entirely surrounded by white pixels). After these morphological operations, contour detection was performed on the resulting binary image. This process locates transitions from black to white in the image and designates them as object boundaries.

2.4.2 CNN Classification

Upon testing the described thresholding techniques, it became apparent that the ore edge detection did not flawlessly detect ore edges, resulting in several false detections due to belt edges. Consequently, a more advanced approach was required. The task of the CNN was, therefore, to classify between objects and non-objects detected in the previous methods. The detected contour coordinates (as a result from thresholding) allowed for the creation of a bounding rectangle, which was then used to crop the detections. These cropped detections were employed to train a CNN via transfer learning, as there was insufficient data to train a full network from scratch.

The cropped images were resized to 256x256 to produce a dataset of 670 images. One class contained 161 images of actual objects (true detections), whereas the other class contained 509 images false detections. Examples of these can be seen in Figure 2. In Figure 2a (from left to right), the ore edge is visible on the right side of the belt, followed by an image segment containing only ore, a half-loaded belt segment, and ore edges on the left side of the belt. In Figure 2b, several objects can be observed on the belt: a partially cropped concrete block, a concrete block near the ore edge,



(a)

(b)

Figure 2: Examples of the training data in the classes: (a) False detections and (b) True detections.

a brighter rock, and a metal plate partially covered by ore. Most objects in the dataset resemble stones, which can get stuck in narrower chutes and cause belt or equipment damage. The plate can rip the belt, and other items such as metal pipes or wooden pallets were also included, each posing potential hazards to the operation.

For training, the lightweight yet relatively accurate MobileNetV2 model (74.7% top-1 accuracy on the ImageNet dataset) developed by Sandler et al.[16] was chosen as the base for feature extraction. The original classification layer was removed, and a lightweight dense head was added to adapt the extracted features for this specific binary classification task. More complex models, such as ResNet-152, and VGG16, show a higher complexity and latency in general [17]. As shown in Figure 3, MobileNetV2 begins with a 3x3 convolution, followed by batch normalization and ReLU activation. It employs a series of inverted residual (bottleneck) blocks for feature extraction, with each block consisting of three stages: expansion (a 1x1 pointwise convolution to increase the number of channels), depthwise convolution (a 3x3 channel-wise operation, crucial for MobileNetV2's efficiency), and projection (a 1x1 pointwise convolution to reduce the number of channels). The structure concludes with a 1x1 convolution to enhance the feature map, followed by batch normalization and ReLU activation. These layers from the MobileNetV2 network were frozen. After the backbone, a global average pooling layer reduces each feature map to a single averaged value, transforming the spatial dimensions into a single vector per channel. This is followed by a dense layer that learns patterns specific to the provided dataset and a final dense layer for binary classification.

The Adam optimizer [18] with its explicit default settings was used, chosen for its convergence capabilities in practice. Since there were two classes, categorical crossentropy was used as the loss function, along with accuracy as a straightforward performance metric. Given the relatively limited training data, data augmentation was crucial to improve the model's adaptability to unseen data. In particular, random transformations such as slight rotations ($\pm 20^{\circ}$), translations (width and height shifts up to 20%), shear transformations, and zooming were applied. Horizontal flipping was also enabled, and all images were rescaled to a range of [0,1] by dividing by 255. The training–validation split was set to 80–20, and a batch size of 32 was used. Ten epochs were chosen for training, based on

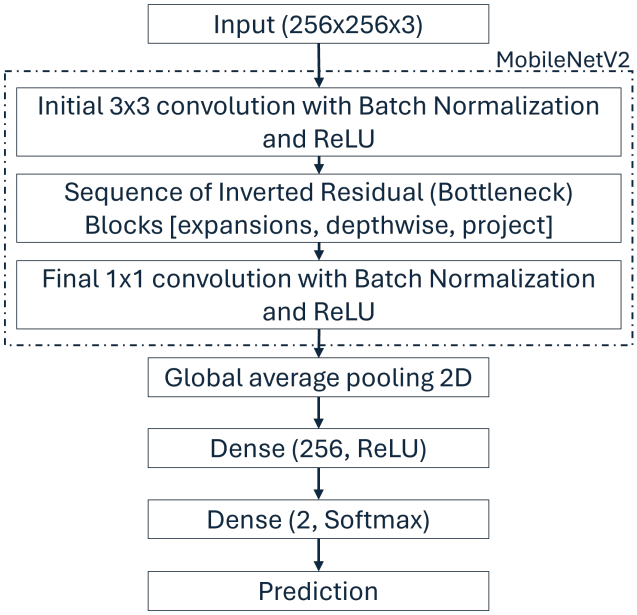


Figure 3: CNN model architecture diagram

preliminary observations of convergence to avoid overfitting while still providing sufficient iterations for stable learning.

The results were validated on an unseen test set of 77 images using recall, precision, accuracy, and F1-score, Equations (1)–(4). In these equations, TP (true positives) refers to objects correctly classified as objects, TN (true negatives) refers to non-objects correctly classified as non-objects, FP (false positives) refers to non-objects misclassified as objects, and FN (false negatives) refers to objects misclassified as non-objects.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

The macro average and weighted average of these metrics were also determined. The macro average is the arithmetic mean of the metrics across all classes, treating each class equally, and is determined to get a good sense of how well the model performs across all classes, without regard to class imbalance within the dataset. The weighted average considers the number of images in each class, and therefore the larger classes will have a greater impact on the metric. In addition to these metrics, the training and validation accuracies and losses were analyzed and the inference time was calculated.

3 RESULTS & DISCUSSION

The model's training and validation accuracies (left) and losses (right) over ten epochs are shown in Figure 4. The training accuracy increases rapidly from about 80% to nearly 95%, while the validation accuracy fluctuates between 90% and 95%. Both remain consistently above 90%, indicating that the model generalizes well to unseen data without severe overfitting. In addition, rapid convergence is observed as the model quickly learns discriminative features within the first few epochs. The training loss decreases from around 0.5 to below 0.1 – the low final losses suggest that the network has effectively minimized classification errors on both sets, and the fluctuations may be due to the modest dataset size. The consistently high validation accuracy and low loss confirm that the model learned robust features for the classification of objects and non-objects.

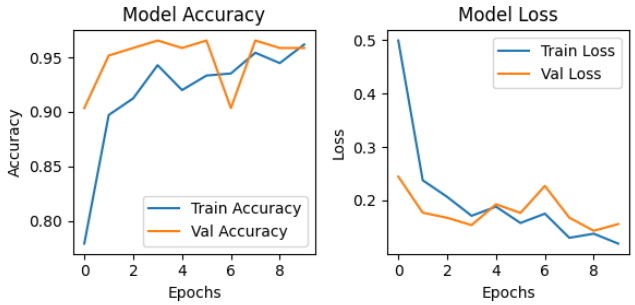


Figure 4: Training and validation accuracy and loss of the CNN model

The model's performance metrics (Table 1) for classifying an unseen test dataset of 77 images demonstrate that the model generalizes well beyond the training and validation sets. The model achieves a high accuracy of 96%, with the non-objects class having a recall of 1 and the objects class having a precision of 1. The slightly lower recall for objects, reflected in the F1-score (0.90), indicates that some objects are classified as non-objects. This is also evident in the confusion matrix (Figure 5), with one example of such a misclassification shown in the middle of Figure 6a.

The macro average and weighted average metrics indicate strong and consistent performance across both classes, suggesting that operators will not need to react to false alarms and that few objects will be missed. However, the small size of the testing dataset should be considered, and further testing of these metrics will benefit the validation of the model.

Table 1: Classification report for the unseen test set (77 images)

Class	Precision	Recall	F1-Score	# Images
Non-objects	0.95	1.00	0.98	60
Objects	1.00	0.82	0.90	17
Accuracy			0.96	77
Macro avg	0.98	0.91	0.94	77
Weighted avg	0.96	0.96	0.96	77

Examples of scored images from the test dataset are shown in Figure 6. After analyzing the score distributions, a threshold of 0.3 was implemented in the algorithm to distinguish between objects and false detections.

The CNN model has an inference time of 0.0558 seconds per image,

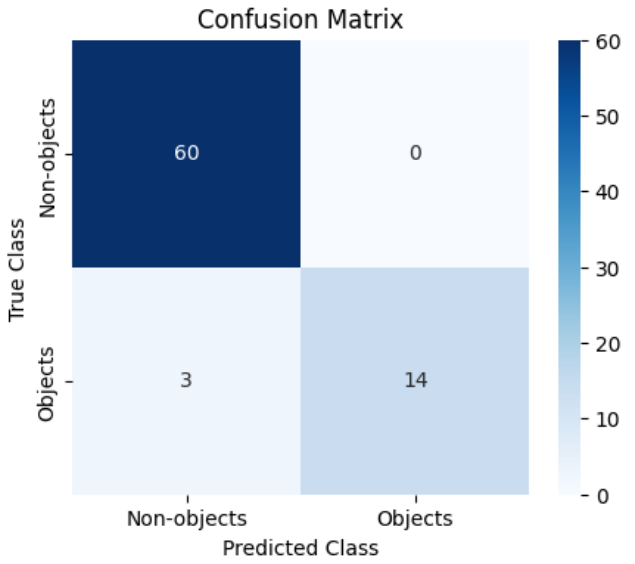


Figure 5: Confusion matrix for the unseen test set (77 images)

while the complete object detection algorithm processes a frame in 5.2 seconds—from fetching the frame to displaying it on the SaaS platform, when detecting an object. This includes frame acquisition, pre-processing, post-processing, and communication with the SaaS platform. Operators have approximately 50 seconds to react to detected objects before they move to the next conveyor belt.

The object detector described in this paper has been implemented for three months, during which the number of objects detected has increased significantly, with a total of 93 objects detected, zero reported missed objects, and 25 false detections. **This resulted in a precision of 79% and a recall of 100%. By adding the CNN to the pipeline, the number of false detections was reduced by 92%**.

4 CONCLUSIONS

This paper presented a system for detecting hazardous objects on an iron ore conveyor belt using a video camera. The approach integrates belt movement detection, region-of-interest identification via the Hough Line Transform, dual-threshold object detection (light and dark), and a final classification step using MobileNetV2.

By combining light and dark thresholding, the system efficiently identifies both bright and dark objects of interest, while the CNN filter effectively eliminates false alarms caused by inaccurate edge detection. The MobileNetV2 model demonstrated high accuracy on an unseen dataset, indicating strong generalization beyond the initial training set.

Continued monitoring of the deployed algorithm is recommended to identify any missed objects and track ongoing false positives. As the system remains in operation, gathering more data and incorporating operator feedback will enable iterative retraining of the CNN model, ensuring adaptability to unforeseen conditions and enhancing its overall reliability in harsh operational environments.

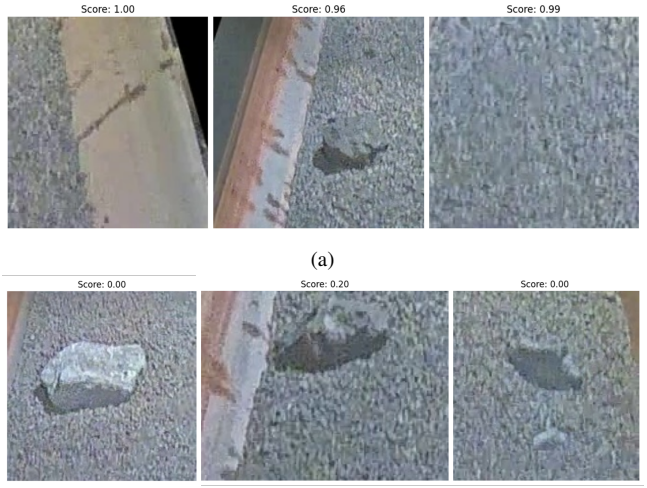


Figure 6: Example images and scores from the model classified as: (a) non-objects and (b) objects.

ACKNOWLEDGEMENTS

The authors of this paper would like to thank LKAB Narvik for their contributions, insights and help with testing.

REFERENCES

[1] M. Zhang, H. Shi, Y. Zhang, Y. Yu, and M. Zhou, "Deep learning-based damage detection of mining conveyor belt," *Measurement*, vol. 175, p. 109130, 2021.

[2] R. Yao, P. Qi, D. Hua, X. Zhang, H. Lu, and X. Liu, "A foreign object detection method for belt conveyors based on an improved yolox model," *Technologies*, vol. 11, no. 5, 2023. [Online]. Available: https://www.mdpi.com/2227-7080/11/5/114

[3] Conveyor Equipment Manufacturers Association, "Belt selection," in *Belt Conveyors for Bulk Materials*. United States of America: CEMA, 2002, ch. 7, pp. 197–224.

[4] V. Meulenberg, K. Moloukbashi Al-Kahwati, J. Öhman, W. Birk, and R. Nilsen, "Hazardous object detection in bulk material transport using video stream processing," in *International Congress and Workshop on Industrial AI and eMaintenance 2023*, U. Kumar, R. Karim, D. Galar, and R. Kour, Eds. Cham: Springer Nature Switzerland, 2024, pp. 531–543.

[5] K. Al-Kahwati, W. Birk, E. F. Nilsfors, and R. Nilsen, "Experiences of a digital twin based predictive maintenance solution for belt conveyor systems," *PHM Society European Conference*, vol. 7, no. 1, pp. 1–8, 2022. [Online]. Available: https://doi.org/10.36001/phme.2022.v7i1.3355

[6] L. Sun and X. Sun, "An intelligent detection method for conveyor belt deviation state based on machine vision," *Mathematical Modelling of Engineering Problems*, vol. 11, no. 5, pp. 1257–1264, 2024. [Online]. Available: https://doi.org/10.18280/mmep.110514

[7] A. Hassan, M. Refaat, and A. Hemeida, "Image classification based deep learning: A review," *Aswan University Journal of Sciences and Technology*, vol. 2, no. 1, pp. 11–35, 2022.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[9] M. Zhang, Y. Zhang, M. Zhou, K. Jiang, H. Shi, Y. Yu, and N. Hao, "Application of lightweight convolutional neural network for damage detection of conveyor belt," *Applied Sciences*, vol. 11, no. 16, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/16/7282

[10] J. Meng, Q. Wu, and B. Zhang, "Foreign object recognition method of coal mine conveyor belt based on lightweight network," in *2023 4th International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, 2023, pp. 119–123.

[11] Itseez, "Open source computer vision library," https://github.com/itseez/opencv, 2015.

[12] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 2008. [Online]. Available: https://books.google.se/books?id=seAgiOfu2EIC

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[14] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[15] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, p. 11–15, Jan. 1972. [Online]. Available: https://doi.org/10.1145/361237.361242

[16] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," *CoRR*, vol. abs/1801.04381, 2018. [Online]. Available: http://arxiv.org/abs/1801.04381

[17] R. Desislavov, F. Martínez-Plumed, and J. Hernández-Orallo, "Trends in ai inference energy consumption: Beyond the performance-vs-parameter laws of deep learning," *Sustainable Computing: Informatics and Systems*, vol. 38, p. 100857, Apr. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.suscom.2023.100857

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. [Online]. Available: https://arxiv.org/abs/1412.6980