Al-Khwarizmi Engineering Journal

Al-Khwarizmi Engineering Journal
ISSN (printed): 1818 – 1171, ISSN (online): 2312 – 0789
Vol. 22, No. 3, September, (2026), pp. 63–74

Conveyor Belt Speed Control Based on Object Classes Using a Convolutional Neural Network

Murtadha Shukur Mahmood [1*], Israa Rafie Shareef [2], Furat I. Hussein[3] and Muhannad Ahmed Obeidi [4]

[1,2,3]*Department of Mechatronics Engineering, Al-Khawarizmi College of Engineering, University of Baghdad, Baghdad, Iraq*
[4]*School of Mechanical Engineering, Technological University Dublin, Ireland*
*Corresponding Author's E-mail: mortada.shokr2202m@kecbu.uobaghdad.edu.iq

(Received 4 December 2024; Revised 30 May 2025; Accepted 26 May 2025; Published 1 September 2026)
https://doi.org/10.22153/kej.2026.05.002

Abstract

Conveyor belt systems facilitate the movement of goods and materials, making them an essential part of various fields. Deep learning control methods are therefore indispensable to assure their energy efficiency and optimal motor speed. Here, training is conducted via a convolutional neural network with transfer learning using VGG16 base net. Feature extraction layers are fixed, whilst fully connected layers are replaced. Thirty-three appropriate object classes are considered for classification, and images are acquired using an overhead camera. The obtained weight values are transmitted to Arduino that uses these data to control the stepper motor speed for optimal performance and minimum energy consumption, achieving approximate estimates of weight and efficient control of the motor. This work integrates the following hardware components: an Arduino microcontroller, a NEMA17 stepper motor, a GY-219 digital current sensor (INA219), an L298N motor driver and a high-resolution camera that processes images using Python with TensorFlow and OpenCV. These components enable the conveyor belt system to realise precise classification and instantaneous regulation of objects. Experimental results show the effectiveness of the proposed system in energising and managing the conveyor belt with an accuracy of 99.47% and a loss of 0.04874, highlighting how we can use the developed system to maximise energy economy and classification accuracy. For academics, engineers and business leaders, this study provides important new perspectives on the current scene and expected future developments in this expanding field.

Keywords: *Conveyor belt system; convolutional neural network (CNN); deep learning (DL); transfer learning; VGG16*

1. Introduction

Belt conveyors are continuous conveying equipment in modern production with the advantages of large capacity, suitability for long distances, low freight, high efficiency, stable operation, convenient loading and unloading, etc. [1]. They are amongst the most important equipment in the field of bulk material transportation. Intelligent transportation systems are safe, efficient, and unmanned transportation systems that integrate advanced technologies such as intelligent driving, control, operation and maintenance [2]. Currently, conveyor belts are widely used for transporting materials over short and medium distances because their conveying capacity is very efficient compared with that of other conveying methods [3].

Deep learning techniques have emerged as powerful tools in a wide range of applications. Deep learning models consist of multiple processing layers that learn different representations of data to solve various problems [4]. A deep neural network processes data across multiple layers, enabling it to perform complex tasks. It comprises multiple layers of neurons, enabling it to perform advanced tasks such as image classification and object detection [5]. Object detection is one of the fundamental tasks in the field of computer vision. It deals with the detection of states and locations in an image of certain objects, and the detected objects can then be classified into different categories. In recent years, deep learning, described as representation learning involving a hierarchy of features, has made rapid progress in object detection and has been implemented in many fields, such as computer vision, voice recognition, natural language processing, remote sensing, human motion tracking and industrial applications. Deep learning has a wide range of algorithms and methods for solving different tasks. For example, convolutional neural networks (CNNs) are commonly applied to image analysis. The classification accuracy of CNNs depends on several factors, such as the dataset, optimisation method and neural network architecture. In theory, currently developed deep learning methods allow us to detect any object in an image. Deep learning methods are applied to real-time object detection to integrate fault detection into several stages of the manufacturing process, such as component assembly, quality control, manufacturing and production calculation. These real-life object detection applications differ in their methodology, including the training dataset used, training parameters and object detection criteria [6]. The results of running the neural network are displayed on the basis of the learning loss function of the network and the effectiveness of the algorithm, then the error matrix is determined [7].

In this work, we implement and examine the capabilities of object detection methods in a real-time system to compute moving components on a conveyor belt. In addition, we provide a detailed explanation of the deep learning architecture and modification method, including experimental results for applying deep learning architectures to real-time applications. The related work will be presented in the next section. Section III describes the research method. Section IV presents the experiments and results, and the last section provides the conclusion.

The main contributions of this study are as follows:

1. Implementation of an object detection system for real-time conveyor belt monitoring using deep learning
2. Modification of the VGG16 model by replacing fully connected layers with three dense layers (512, 256 and 33) and using global average pooling instead of flattening
3. Development of a dataset consisting of 33 classes with 2,831 images (80% for training, 20% for testing)
4. Integration of a CNN-based classification system with an Arduino-controlled stepper motor to adjust conveyor belt speed dynamically
5. Achieving a high classification accuracy (99.47%) and low validation loss (0.04874), which indicates good model training and generalisation performance
6. Employing the system in industry for optimisation works by expediting operations, diminishing energy usage and optimising conveyor belt usage

2. Related Work

Deep learning-based conveyor systems can advance the processes of global conveyors. These systems offer control and optimisation capabilities that are highly developed. Using CNN algorithms, the system continuously observes and alters conveyor parameters. Merits include predictive maintenance, defect spotting and adaptive speed control. The conveyor, in other words, relies on machine learning to assess sensor data to ensure that operation and performance are smooth and optimal. The use of deep learning improves the system's reaction time, allowing for quick decisions to optimise efficiency and minimise downtime. CNN conveyor belt systems have been researched extensively.

The work in [8] used the VGG16 model with transfer learning to find flaws in printed circuit boards (PCBs). With a 1,245-defect image dataset, the model achieved an outstanding 97% accuracy in testing. The study showed how well CNNs, such as VGG16, improved object detection and quality assurance in manufacturing environments. This finding emphasises how important deep learning is to improving business processes.

In [9], the VGG16 model was also used in an application of surface defect detection on steel strip, which is important for quality assurance during the industrial production process. Using transfer learning, the model exhibited an excellent 99.44% classification accuracy, confirming its capacity to distinguish between closely related and nonrelated faults. The study underlined that transfer learning improves production standards and provides the industry efficient alternatives for fault classification, thus reducing the need for large labelled datasets. The work in [10] focused on changing the VGG16 model for small datasets, which usually pose challenges for standard machine learning models. Through pretrained model fine-tuning, an accuracy of 92% was achieved, exhibiting high precision and recall metrics. This result showed how well the model generalised over several object categories and reduced false positives and negatives. The impact of several hyperparameters was investigated to provide understanding of how transfer learning approaches might be optimised for maximum performance.

With an accuracy of 98%, the VGG16 model was upgraded in [11] for object recognition challenges. The study adjusted the fully connected layers to accommodate new datasets, leaving the convolutional layers for feature extraction. This method led to fewer training challenges and a better model, which solved the difficulties with different lighting, posture of the subject and background. These results reinforced the architectural strength of VGG16, particularly in its ability to extract hierarchical traits that are crucial for distinguishing between similar classes.

The work in [12] demonstrated the versatility of VGG16 across multiple object recognition tasks. The restricted model reached an accuracy of 86.3% and the enlarged one of 98.1%. This outcome highlighted the benefits of transfer learning, and VGG16 outperformed traditional techniques. This versatility is beneficial for tasks requiring high precision but with limited resources.

In [13], the VGG16 model was trained through data augmentation. The model achieved a validation accuracy of 95.4% after hyperparameter tuning, which demonstrated that smaller datasets could use pretrained weights from much larger datasets in the task of transfer learning efficiently. The study showed how transfer learning improved generalisation and performance in object recognition tests, proving its great utility in many different fields.

Pretrained on the extensive ImageNet dataset, the VGG16 model was fine-tuned for object classes in a new dataset in [14]. It achieved 92% accuracy in object classification, stressing the effectiveness of transfer learning in reducing the demand for labelled data. The results confirmed the computing efficiency and robustness of the model; hence, it can be optimised for valuable applications in real-world environments.

Using VGG16, the study in [15] classified antiloosening coated bolts with a 97% accuracy. The loss of spatial information caused the researchers to realise limitations in spotting flaws at borders or in complex areas. They improved the interpretability of the model's predictions and optimised the training process to focus on critical areas such as the bolt thread area through the Grad-CAM visualisation technique. This optimisation enhanced the applicability in industrial settings and the accuracy of the model.

The work in [16] investigated the automated classification of MR brain images using a pretrained VGG16 model applied under transfer learning. The study included 160 axial T2-weighted images from a Harvard Medical School repository. The model obtained remarkable sensitivity, specificity and accuracy of 100% by using exact preprocessing and a 10-fold cross-validation technique. These findings confirmed the superiority of transfer learning in medical image classification and exceeded present state-of-the-art methods.

3. System Architecture Design

The device comprises a sophisticated mechanical and control mechanism requisite to coordinate sensitive motion with pioneering capacity for object identification. Its structure consists of three rollers, namely, two upper horizontal rollers that align on the same axis whilst maintaining a fixed distance from each other and one roller in the middle at the bottom of these two entries. Every roller extends its shaft from the two ends and is fixed on two bearings, which makes the rotation of rollers smooth and frictionless. These bearings are fixed to the device's structure for stability. One shaft of the first upper roller (Roller 1) is longer on one side, allowing a pulley to be mounted on it. The pulley is connected via a belt to a stepper motor, which drives the system by rotating the belt, as shown in Figure 1. The stepper motor operates under precise control from a driver circuit interfaced with an Arduino microcontroller.

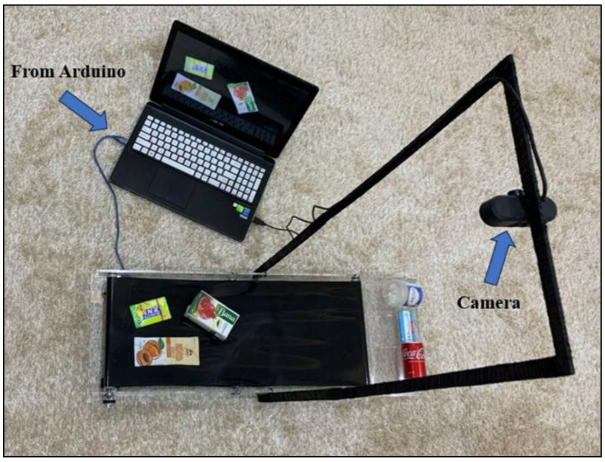

Fig. 1. System Architecture

The Arduino microcontroller receives commands from a laptop that processes real-time visual data captured by a camera mounted above the conveyor belt. This camera continuously monitors objects on the moving belt and sends images to the laptop, where a pretrained VGG16 deep learning model analyses the data to recognise and classify objects. Once the object is identified, the laptop chooses the appropriate system response for setting the belt speed. Arduino uses these commands to dynamically raise, lower, maintain or drive the belt speed in response to the detected object.

The system is built with great accuracy to offer stability and efficiency under several loads. Whilst the belt and pulley configuration effectively transfers motion from the stepper motor to the rollers, the bearings lower friction and extend the life of the rollers. The combination of mechanical stability and intelligent processing will effectively enhance object recognition and dynamic motion control, enabling versatile automation applications in a similar alignment with artificial intelligence-based data analysis (sorting, inspection) related tasks and utilisation in industrial conveyer systems.

4. Methodology and System Design

After the presentation of the system architecture, this section describes the methods applied in this work.

4.1 Preprocessing Steps for Image Training

Before conducting training, the collected images are preprocessed through various procedures to ensure that the model performs well and trained on consistent data:

1. Resizing: All photos are resized to (224 × 224) pixel in accordance with the VGG16 input requirement.
2. Colour Format: Images are converted into RGB (three channels) given that we will use the pretrained VGG16 model for colour images.
3. Normalisation: Pixel values are adjusted to the range [0, 1] using division by 255, enhancing training stability.
4. Data Augmentation: To increase the heterogeneity of the dataset and prevent overfitting, we use techniques such as rotation, flipping, zooming and shifting.
5. Standardisation: Images are standardised using the mean and standard deviation of ImageNet data.
6. Transformation to Tensor: The images are loaded in the form of tensor to work with deep learning libraries.
7. Batch Processing: Data are ingested in minibatches to utilise GPU.
   The dataset is divided into two subsets: training (70%–80%) and test (20%–30%).
8. Pretrained Model Adaptation: The input images are aligned to the VGG16 component for feature extraction without modifying any of the convolutional layers.

These preprocessing techniques ensure that the model is learnt with maximum efficiency whilst maintaining a balanced dataset for classification.

4.2 Data Collection

The VGG16 model is trained for object recognition on 33 classes, from 0 to 32. Each class corresponds to a particular object category utilised during classification. The dataset comprises multiple and diverse class distributions to achieve optimal model performance. Filters, which represent edges and other low-level patterns, are applied through convolutional layers to the input image and extracted and progressively aggregated into higher-level features in two higher layers. Rectified linear unit (ReLU) is a nonlinear function which allows the network to learn complex patterns and prevent vanishing gradients by controlling neuron activation. The pooling layer extracts the essential features by downsampling through max pooling and other approaches, which helps in reducing spatial variance, thus decreasing the numbers of trainable parameters and preventing overfitting [17]. Figure 2 shows the 33 object categories used in the classification task.

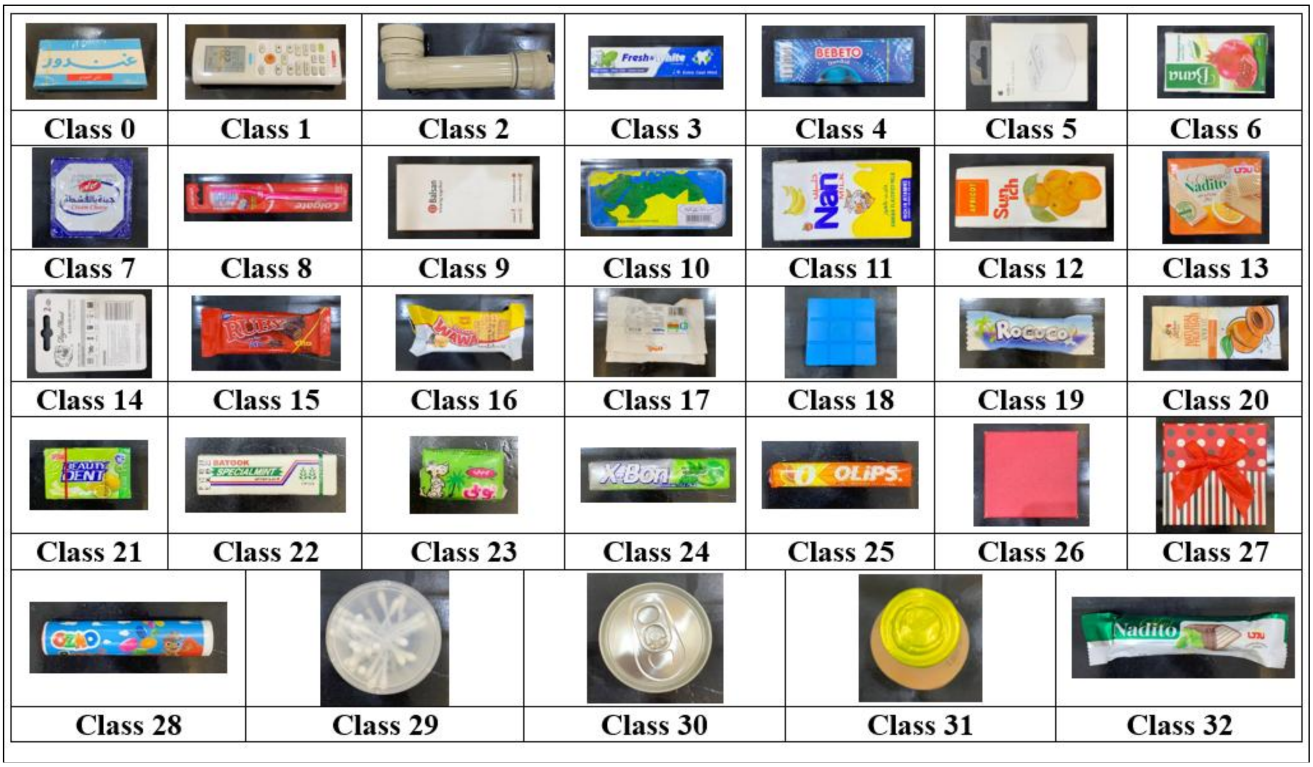

Fig. 2. Visualisation of the 33 Classes Used for Object Classification

A total of 8,231 images are collected and distributed across 33 classes for training a deep learning model. The dataset is divided into a training set containing 2,264 images and a test set with 567 images to ensure accurate model evaluation. Table 1 presents the distribution of images per class, in which the number of images varies depending on the nature of the available data. Approximately 70%–80% of the images in each class are allocated for training, whilst the remaining portion is used for testing. This distribution ensures a balanced approach to improving model accuracy and effective evaluation.

Table 1,
Dataset Distribution for Image Classification

Class	Total Images	Training Images	Test Images
Class 0	81	63	18
Class 1	82	72	10
Class 2	79	68	11
Class 3	91	74	17
Class 4	87	66	21
Class 5	76	65	11
Class 6	95	74	21
Class 7	71	60	11
Class 8	82	63	19
Class 9	82	67	15
Class 10	88	65	23
Class 11	66	51	15
Class 12	98	84	14
Class 13	98	87	11
Class 14	87	68	19
Class 15	129	102	27
Class 16	75	58	17
Class 17	77	63	14
Class 18	80	65	15
Class 19	76	59	17
Class 20	127	106	21
Class 21	91	71	20
Class 22	103	78	25
Class 23	74	49	25
Class 24	104	83	21
Class 25	88	71	17
Class 26	70	53	17
Class 27	95	73	22
Class 28	91	74	17
Class 29	71	56	15
Class 30	71	55	16
Class 31	62	49	13
Class 32	84	72	12
Total	S8231	S2264	S567

4.3 Pretrained VGG-16 Model

CNN is a function of associating input data (in our case, the input data are images) with an output, and it generally consists of convolutional layers, max pooling layers, ReLU activation layers and a softmax layer that generates a well-formed probability distribution for classification on the output. The first convolutional layer extracts edges, and the subsequent convolutional layers act as higher-level feature mappers. The pooling layer is used to compute the maximum value for each patch of the feature map from the previous layer. The role of the activation layer, such as ReLUs, is to address the saturation problem. After training, CNN can be used as a classifier or as a feature extractor in the case of transfer learning. Transfer learning is a technique used to improve the performance of machine learning by harnessing the knowledge gained from another task [18]. The feature extraction strategy applied in this study is freezing all parameters in the network before the fully connected layers, and the tensors are extracted from the last layer before the fully connected layers and normalised as features, which are trained by classifiers such as a multilayer perceptron. Transfer learning is typically applied when a new dataset that is smaller than the original dataset used to train the pretrained model exists [19]. In this work, the pretrained CNN model is VGG-16. It contains 13 convolutional layers, followed by rectification and pooling layers, and 3 fully connected layers. The network trained on the ImageNet database is retrained through the following steps, as shown in Figure 3:

1. The database is replaced with new data collection for 33 categories through the camera with a total of (2,831) input images with sizes of (224 × 224), of which (2264) are used for training at 80% rate and (567) for testing at 20% rate.
2. Global average pooling is used instead of flattening after the feature extraction layers take the average of each of the output channels. Instead of converting them into a one-dimensional matrix, this layer calculates the average of each output from the previous layer. If the output is (7, 7, 512), global average pooling will convert it to (512). We will obtain a one-dimensional vector of size 512, in which each element in this vector represents the average of all the values in that channel. This method helps reduce the dimensions effectively whilst retaining the essential information in the features.
3. The fully connected layers are removed, and three layers, Dense 512, Dense 256 and Dense 33, are created.

After these layers, softmax activation is used to classify through them. All convolutional layers use small 3 × 3 filters and 2 × 2 network pools.

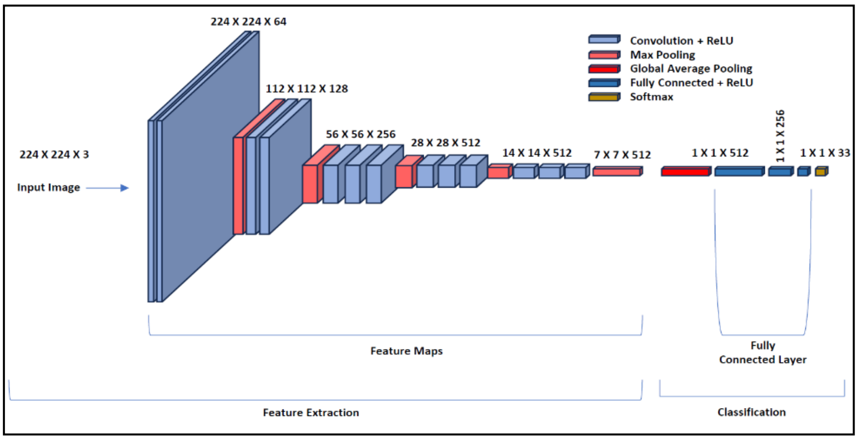

Fig. 3. Pretrained VGG-16 Model

4.4 Overall Flow of the Proposed Work

The block diagram of the system is shown in Figure 4. Firstly, frames are captured from a real-time video using the mounted camera, on which the rest of the processing steps are based. Then, binarisation is applied, in which the RGB values are combined into only two values, black and white, to smooth the boundaries of objects and simplify the processing. Afterwards, noise and unwanted parts are removed through the morphology open process. Next, the objects are contoured to separate them from one another and distinguish each object accurately. The number of detected objects is calculated by finding ‘count’. If no object is found, the process returns to the first step.

The regions of interest (ROIs) are extracted, that is, the image is cropped to contain only the desired objects, and the rest of the image is ignored. These regions are fed into the classification process. CNN is used to analyse and determine the shape and pattern of the image on the basis of the pretrained data. In this step, the total weight on the conveyor belt is determined given that each object has a known weight. Finally, the total weight is sent to Arduino, which controls the speed of the stepper motor in accordance with the total weight. The process continues in the ‘find count’ step. If no object is found, the motor is stopped.

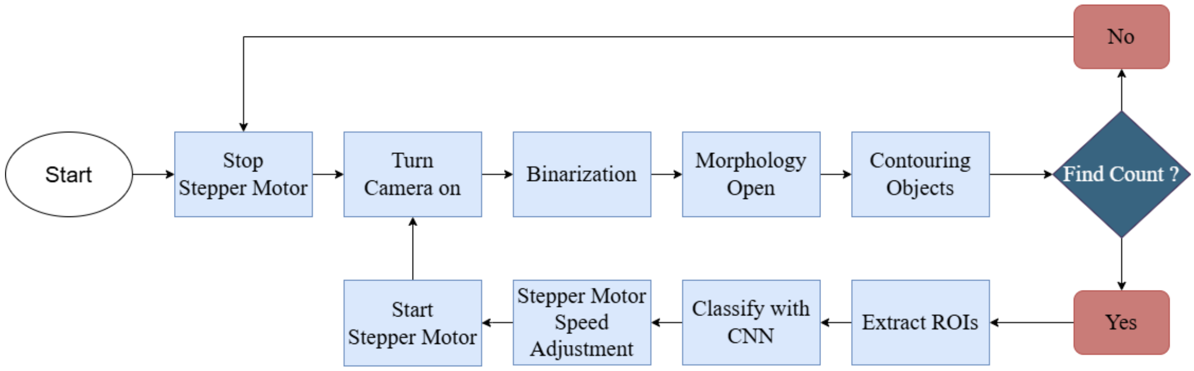

Fig. 4. Block Diagram of the System

5. Experiments and Results

Table 2 summarises the architectural design of the neural network model developed during this study. The model consists of gigantic 14,738,491 trainable parameters that are fine-tuned to optimise two tasks: feature extraction and classification. The convolutional layers (Conv2D) do the heavy lifting of feature extraction and are designed with an increasing number of filters per layer to perform two things: to compensate for the coarseness of extraction in the previous layer (if any) and to capture spatial and more complex patterns in the images. Max pooling layers (MaxPooling2D) are added to reduce spatial dimensions, compute efficiently and reduce overfitting. To compress the feature map output of the convolutional layers into a single vector whilst maintaining a pathway leading into the dense layers that is not too complex in terms of the number of parameters, we use a global average pooling layer (GlobalAveragePooling2D). This modified VGG16 model essentially gives us the best of both worlds: pretrained weights that act like a supersmart filter for the kinds of things we want to see in the images and a pathway leading into the dense layers that will not bog down the computer and will enable the efficient use of the parameters without overfitting.

Table 2,
Model Architecture for Transfer Learning with a VGG16-based Neural Network

Layer No.	Layer Type	Output Shape	Parameters	Description
1	Input Layer	(None, 224, 224, 3)	0	Input layer; accepts images of size 224 × 224 with three colour channels (RGB)
2–19	VGG16 Convolutional and Pooling Layers	Varies	Pretrained	Feature extraction layers from VGG16
20	GlobalAveragePooling2D	(None, 512)	0	Reduces dimensionality whilst retaining features
21	Dense	(None, 512)	262,656	Fully connected layer for feature learning
22	Dense	(None, 512)	131,328	Further abstraction of features
23	Dense (SoftMax)	(None, 33)	8,491	Final classification layer for 33 categories

Figure 5 shows the model's loss curves (the amount of error in predictions) for all training and validation data over the epochs. The training loss curve shows a very high value in the first epoch, a significant decrease during the initial epochs and a low and stable value (around 0.09091) at the end of training. As for the validation loss curve, it follows the same trend and starts with a high value and decreases rapidly until it reaches a low level close to the training loss curve (around 0.04874). Slight fluctuations occur in the loss curves, but after the 10th epoch, the curve remains very low and almost stable. The decrease in the curves indicates that the model is performing well and it reduces prediction errors. The validation data loss is low and close to the training loss, which means that the model is well generalised on new data. We observe the convergence of the two curves, implying that the model does not suffer from overfitting or underfitting and learns in a balanced manner.

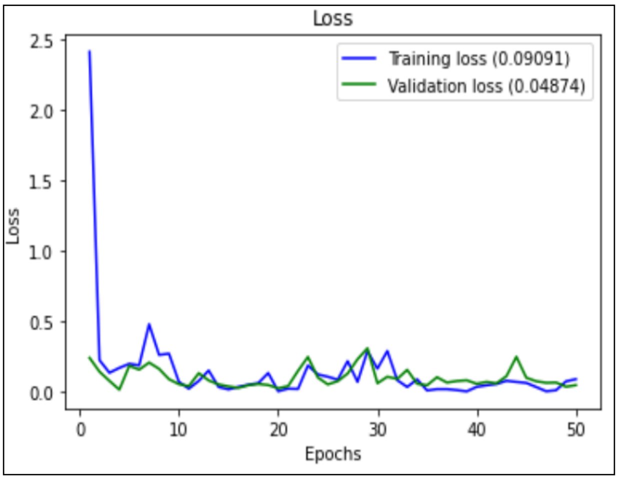

Fig. 5. Model Loss on Training and Validation Data over Epochs

Figure 6 shows the model performance accuracy curves for training and validation data over the epochs. The training accuracy curve initially appears at a low value, quickly attains a high level close to 100% during the first 10 epochs and reaches a high and stable level (about 0.99552) at the end of training. The validation accuracy curve follows the same pattern and starts to improve quickly, approaching a level between 99% and 100%, and reaches a high level close to the training accuracy curve (about 0.99471). The model has excellent performance and good generalisation capability owing to the convergence of the training and validation accuracy curves. That is, no significant difference exists between them, which means that the model's performance on the validation data is very similar to its performance on the training data – it does not suffer from overfitting. The model has successfully learnt and has become able to classify the training and validation data with high accuracy, due to the stability of the accuracy in the last epochs.

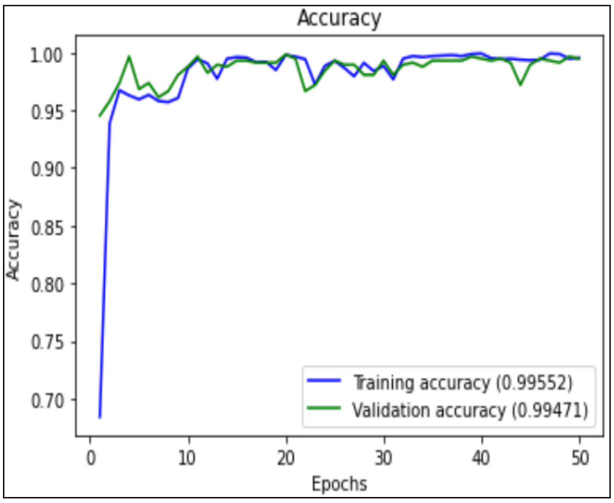

Fig. 6. Model Accuracy During Training and Validation over Epochs

The confusion matrix shown in Figure 7 evaluates the performance of a multiclass classification model. The (Predicted Label) horizontal axis represents the classes predicted by the model, whilst the (True Label) vertical axis represents the true classes to which the samples belong. The correctly classified cells are on the main diagonal of the matrix, which are the predicted classes that match the true classes. The high values on the diagonal indicate that the model classifies the classes correctly. The cells outside the main diagonal represent the samples that are incorrectly classified – the predicted classes that are different from the true classes. On the basis of the colour intensity, the colour gradient from light to dark reflects the number of samples from low to highest. This model classifies a higher number of samples correctly, which is evident in the main diagonal, where the colour is darker. The very low and almost nonexistent samples outside the diagonal indicate a low percentage of prediction errors and demonstrate the model's strength in differentiating classes.

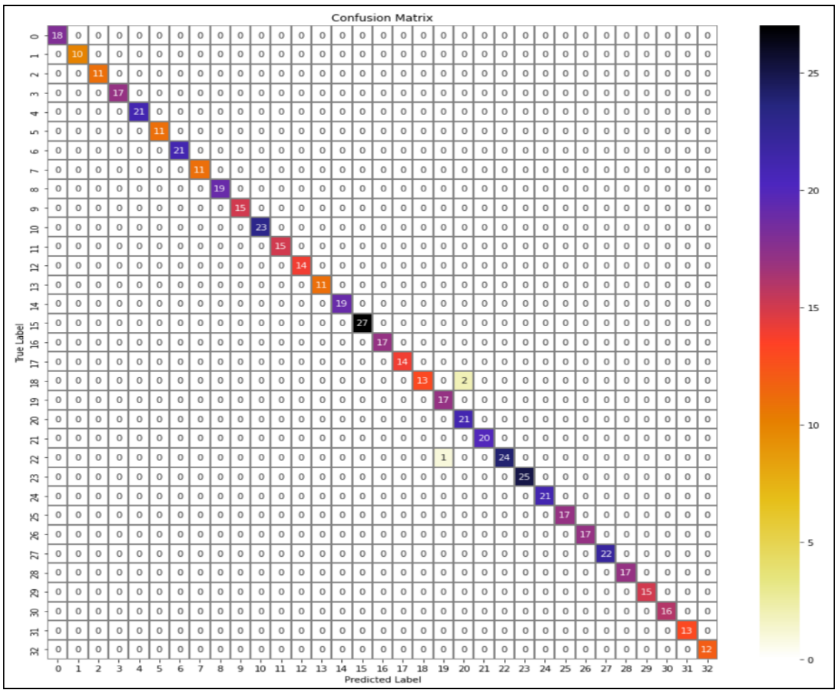

Fig. 7. Confusion Matrix of Model Performance Across Multiple Classes

6. Evaluation Metric Used in the Study

Accuracy is selected as the main evaluation criterion for assessing the object classification model for conveyor belts. The following formula is used to compute accuracy:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \ldots(1)$$

where:

- TP (True Positives): The number of correctly classified positive samples
- TN (True Negatives): The number of correctly classified negative samples
- FP (False Positives): The number of samples incorrectly classified as positive
- FN (False Negatives): The number of samples incorrectly classified as negative

Accuracy is chosen as the main assessment criterion for the following reasons:

1. Balanced Dataset: Accuracy is a valid metric for the 33 distinct object classes with an almost equal sample distribution in the dataset used in this work.
2. Suitability for Multiclass Classification: Accuracy offers a direct and complete assessment of the general performance of the model over all categories in multiclass classification problems.
3. Relevance in Industrial Applications: High classification accuracy guarantees exact speed adjustments in industrial systems, including conveyor belts, thus enabling effective operations and the best energy economy.
4. Minimal Impact of Class Imbalance: Accuracy is still a relevant metric even if the dataset does not show significant class imbalance because it is not dependent on any other metric, including precision, recall or F1-score.
5. Strong Generalisation Capability: With a low validation loss of 0.04874 and a great accuracy of 99.47%, the model indicates strong generalisation on fresh data, reinforcing accuracy as a useful assessment criterion.

7. Conclusion

The application of CNNs in deep learning to classify objects on a conveyor belt, estimate total weight and control the speed of a stepper motor using Arduino is presented in this paper. The fully connected layers were eliminated, three new layers (Dense 512, Dense 256 and Dense 33) were added, and transfer learning was used in the VGG16 model. Global average pooling was adopted instead of flattening after the feature extraction layers to compute the average of each output channel. A new dataset of 33 classes was collected using a camera, consisting of 2,831 images of size 224 × 224, with 80% employed for training and 20% for testing.

The model achieved very low loss, indicating minimal prediction error. Specifically, it exhibited a training loss of 0.09091 and a validation loss of 0.04874, reflecting stable performance. The model also proved its efficacy in classification tasks by showing exceptional classification accuracy, that is, 99.55% training accuracy and 99.47% validation accuracy. The confusion matrix findings validated the great precision of the model; hence, it is appropriate for industrial use.

The main advantages of the proposed system are implementation in industrial automation, excellent classification quality, real-time operation and reliable conveyor belt management. In addition, transfer learning through VGG16 has enhanced the robustness of the model and reduced the requirement for massive data acquisition followed by training from scratch. Motor control complements deep learning through a dynamic, feedback-driven system that optimises speed control and categorisation.

One of the main points of this study is the minimisation of energy consumption. To address this problem, the proposed system stops the conveyor belt if no weight has yet been detected. The VGG16 classifier also helps in detecting whether the weight of the belt is either less or more than the maximum limit by real-time image detection. Then, the system sends a command to the stepper motor to stop operating to prevent causing mechanical overload and to optimise energy efficiency. Once any weight is sensed on the conveyor belt, the stepper motor moves with speed in proportion to the current weight, enabling enough power and optimal material hauling.

A limitation of the system is dataset size (2,831 images), which is relatively small compared with that for large-scale deep learning applications. Although a good result has been achieved, further improvement can be attained, and data augmentation can be considered.

Unlike models in earlier studies that utilised VGG16 for defect detection in printed circuit boards, surface defect classification, object recognition and medical imaging with accuracies ranging from 92% to 100%, our model achieved a classification accuracy of 99.55%. Different from previous studies which were limited to classification problems only, our study combined real-time classification and conveyor belt speed control for future industrial automation applications in restaurants. These improvements lead to a much more robust model for real-world manufacturing and logistics applications – with the help of capturing new datasets and tweaks on the VGG16 model.

Conflicts of Interest

The authors declare no conflict of interest regarding the publication of this paper.

References

[1] Y. Liu, C. Miao, X. Li, and G. Xu, "Research on deviation detection of belt conveyor based on inspection robot and deep learning," *Complexity*, vol. 2021, Article ID 3734560, 2021. https://doi.org/10.1155/2021/3734560

[2] M. Zhang, Y. Zhang, Y. Wang, and H. Wang, "Application of lightweight convolutional neural network for damage detection of conveyor belt," *Applied Sciences*, vol. 11, no. 16, pp. 1–15, 2021. https://doi.org/10.3390/app11167282

[3] I. R. Shareef and H. K. Hussein, "Implementation of artificial neural network to achieve speed control and power saving of a belt conveyor system," *Eastern-European Journal of Enterprise Technologies*, vol. 2, no. 2(110), pp. 6–13, 2021. https://doi.org/10.15587/1729-4061.2021.224137

[4] N. M. Sulaiman, A. Al Tmeme, and M. N. Mahdi, "An overview of audio-visual source separation using deep learning," *Al-Khwarizmi Engineering Journal*, vol. 19, no. 4, pp. 42–55, 2023. https://doi.org/10.22153/kej.2023.06.003

[5] A. H. M. Yahya Ghufran Khidhir, "Real-time end-to-end self-driving car navigation," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 2S, pp. 366–372, 2023. https://doi.org/10.18201/ijisae.2732

[6] D. Varna and V. Abromavičius, "A system for a real-time electronic component detection and classification on a conveyor belt," *Applied Sciences*, vol. 12, no. 11, pp. 1–13, 2022. https://doi.org/10.3390/app12115608

[7] P. Bortnowski, H. Gondek, R. Król, D. Marasova, and M. Ozdoba, "Detection of blockages of the belt conveyor transfer point using an RGB camera and CNN autoencoder,"

Energies, vol. 16, no. 4, pp. 1–15, 2023. https://doi.org/10.3390/en16041666

[8] S. A. Althubiti, F. Alenezi, S. Shitharth, S. K, and C. V. S. Reddy, "Circuit manufacturing defect detection using VGG16 convolutional neural networks," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 1070405, 2022. https://doi.org/10.1155/2022/1070405

[9] A. A. M. S. Ibrahim and J. R. Tapamo, "Transfer learning-based approach using new convolutional neural network classifier for steel surface defects classification," *Scientific African*, vol. 23, p. e02066, 2024. https://doi.org/10.1016/j.sciaf.2024.e02066

[10] M. Shaha and M. Pawar, "Transfer learning for image classification," in *Proc. 2nd Int. Conf. Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, pp. 656–660, 2018. https://doi.org/10.1109/ICECA.2018.8474802

[11] D. I. Swasono, H. Tjandrasa, and C. Fathicah, "Classification of tobacco leaf pests using VGG16 transfer learning," in *Proc. 12th Int. Conf. Information & Communication Technology and System (ICTS)*, Surabaya, Indonesia, pp. 176–181, 2019. https://doi.org/10.1109/ICTS.2019.8850946

[12] A. A. Santos, F. A. S. Rocha, A. J. da R. Reis, and F. G. Guimarães, "Automatic system for visual detection of dirt buildup on conveyor belts using convolutional neural networks," *Sensors*, vol. 20, no. 20, pp. 1–16, 2020. https://doi.org/10.3390/s20205762

[13] S. Tammina, "Transfer learning using VGG-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, pp. 143–150, 2019. https://doi.org/10.29322/IJSRP.9.10.2019.p9420

[14] M. Alencastre-Miranda, R. M. Johnson, and H. I. Krebs, "Convolutional neural networks and transfer learning for quality inspection of different sugarcane varieties," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 787–794, 2021. https://doi.org/10.1109/TII.2020.2997474

[15] E. Noh and S. Hong, "Automatic screening of bolts with anti-loosening coating using grad-CAM and transfer learning with deep convolutional neural networks," *Applied Sciences*, vol. 12, no. 4, pp. 1–11, 2022. https://doi.org/10.3390/app12042029

[16] T. Kaur and T. K. Gandhi, "Automated brain image classification based on VGG-16 and transfer learning," in *Proc. Int. Conf. Information Technology (ICIT)*, Bhubaneswar, India, pp. 94–98, 2019. https://doi.org/10.1109/ICIT48102.2019.00023

[17] Y. G. Khidhir and A. H. Morad, "Comparative transfer learning models for end-to-end self-driving car," *Al-Khwarizmi Engineering Journal*, vol. 18, no. 4, pp. 45–59, 2022. https://doi.org/10.22153/kej.2022.09.003

[18] Z. Shi, Y. Zhang, Y. Zhang, and J. Zhang, "A deep CNN based transfer learning method for false positive reduction," *Multimedia Tools and Applications*, vol. 78, pp. 1017–1033, 2019. https://doi.org/10.1007/s11042-017-5370-6

[19] M. Hussain, J. J. Bird, and D. R. Faria, "A study on CNN transfer learning for image classification," in *Advances in Computational Intelligence Systems*, vol. 840, pp. 191–202, 2019. https://doi.org/10.1007/978-3-030-04239-4_15

نظام للتحكم في سرعة السيور الناقلة بناءً على تصنيف الأجسام باستخدام الشبكات العصبية الالتفافية

مرتضى شكر محمود[1]*، إسراء رافع شريف [1]، فرات ابراهيم حسين [3]
مهند احمد عبيد[4]
[1، 2، 3] قسم هندسة الميكاترونكس، كلية الهندسة الخوارزمي، جامعة بغداد، بغداد، العراق
[4] الهندسة الميكانيكية، جامعة دبلن، ايرلندا
*البريد الالكتروني: mortada.shokr2202m@kecbu.uobaghdad.edu.iq

المستخلص

تُعد أنظمة السيور الناقلة عنصرًا أساسيًا في مختلف الصناعات، حيث تضمن النقل الفعّال للمواد والسلع. تواجه الأساليب التقليدية للتحكم تحديات في تحسين سرعة المحرك واستهلاك الطاقة، مما يدفع نحو دمج تقنيات التّعلّم العميق. تعتمد هذه الدراسة على الشبكات العصبية التلافيفية (CNN) من خلال التّعلّم الانتقالي باستخدام النموذج المُدرّب مسبقًا VGG16، حيثُ يتم الاستفادة من طبقات استخراج الميزات مع تعديل الطبقات المتصلة بالكامل لتصنيف 33 فئة مختلفة من الأجسام وتحديد الوزن الإجمالي على السير الناقل بالاعتماد على الصور المُلتقطة بواسطة كاميرا مُثبتة أعلاه. ولتحقيق تقدير دقيق للوزن وضمان التحكم الفعّال في المحرك، يتم معالجة البيانات المُستخرجة وإرسالها إلى متحكم Arduino، والذي يقوم بضبط سرعة محرك Stepper NEMA17 للحفاظ على الأداء الأمثل مع تقليل استهلاك الطاقة. فتم تنفيذ النظام باستخدام Python، وTensorFlow، وOpenCV لمعالجة الصور، مع دمج مكوناتٍ إلكترونية تشمل المتحكم الدقيق Arduino، محرك NEMA17 الخطوي، مستشعر التيار الرقمي (INA219) GY-219، وحدة تشغيل المحرك L298N، وكاميرا عالية الدقة، حيث تعمل هذه المكونات معًا لضمان التصنيف الدقيق والتحكم في الزمن الحقيقي للنظام. تُظهر النتائج التجريبية فعالية النظام المقترح، حيثُ حقق دقة تصنيف بلغت 99.47% وخسارة بمقدار 0.04874، مما يؤكد قدرتهُ على تحسين دقة التصنيف، وتعزيز كفاءة استهلاك الطاقة، وتطوير إدارة أنظمة السيور الناقلة. تقدم هذه الدراسة رؤى قيّمة للباحثين والمهندسين والخبراء الصناعيين، حيثُ توفر نظرة شاملة حول الوضع الحالي لهذا المجال وآفاقه المستقبلية المحتملة.