Fine-Tuning YOLO-v12 for Efficient Halal-Logo Detection

Kok Swee Sim1, 2 a), Kai Liang Lew2, b), Hao Xian Cheong3, c), Chia Ling Hi2, d), Fazly Salleh Abas1, 2, e) and Nor Hidayati Abdul Aziz1, 2, f)

1Centre for Advanced Analytics, COE for Artificial Intelligence, Multimedia University, Jalan Ayer Keroh Lama, Melaka, 75450 Bukit Beruang, Malaysia

2Faculty Of Engineering & Technology, Multimedia University, Jalan Ayer Keroh Lama, Melaka, 75450 Bukit Beruang, Malaysia

3Faculty of Information Science & Technology, Multimedia University, Jalan Ayer Keroh Lama, Melaka, 75450 Bukit Beruang, Malaysia

*a)Corresponding author: sksbg2022@gmail.com  
b)1132703002@student.mmu.edu.my*

*c) imchaoxian321@gmail.com*

*d) chialing239@gmail.com*

*e) fazly.salleh.abas@mmu.edu.my*

*f) hidayati.aziz@mmu.edu.my*

**Abstract.** Counterfeit or missing Halal certification marks threaten consumer trust and impede regulatory oversight, yet manual logo inspection remains slow, error-prone, and unsuited to high-volume workflows. We present a lightweight, edge-deployable Halal-logo detection system that fine-tunes two recent YOLO-v12 backbones, nano and small on a 50-class public dataset which has 1,292 images with 640 × 640 resolution. Two training strategies are applied to the models, train from scratch and transfer learning. The transfer learning is fine tine the model with COCO-pre-trained weights while freezing the first three layers. Fine-tuned YOLO-v12 small lifts accuracy from 0.919 to 0.945 mAP@0.5:0.95 and halves GPU training time, whereas YOLO-v12 nano attains 0.947 mAP with only 2.6 M parameters, 4.7 ms single-image latency, and 509 MB peak VRAM. Transfer learning trims training energy from 0.14 kWh to 0.04 kWh and reduces estimated CO₂ emissions to 0.02 kg, underscoring its sustainability advantage. The study provides the first systematic evaluation of YOLO-v12 nano versus small for Halal-logo detection, evidence that lightweight backbones can match larger models when fine-tuned, and an open-source, reproducible training and inference pipeline. These results demonstrate that modern nano detectors deliver high accuracy with one-third of the energy footprint, paving the way for scalable, real-time Halal-logo verification in production lines, retail checkpoints, and mobile auditing applications. The code is published in Github (https://github.com/lewbei/Fine-Tuning-YOLO-v12-for-Efficient-Halal-Logo-Detection)

# Introduction

Halal is important for Muslims as it verifies that the food is clean for them [1]. It is also a rule for the manufacturers to follow. It can only be certified by an authorized Halal logo on the packaging. Halal logo applications should be strict because around 1.8 billion Muslim consumers rely on the logo for instant trust [2]. Therefore, the global halal trade is approximately 2 trillion USD. The certified halal logo product must go through several processes and be certified by agencies. After the product is certified, each has unique logos that manufacturers must print and make visible.

Regulators, retailers, and customs officers manually inspect millions of packages daily to detect counterfeit or missing Halal logos. However, manual inspection is inefficient, highly time-consuming, prone to human error, and increasingly unreliable under worker fatigue, especially as counterfeit logos can closely resemble authentic ones. Therefore, manual inspection is one of the challenges faced by regulators, retailers and customs officers [3]. Modern automation systems have demonstrated significant potential in addressing such inspection challenges across various manufacturing and quality control applications [4].

With the rise of deep learning technology, the gap can be bridged because the convolution neural network (CNN) can self-learn based on the provided dataset. This ability has shown its capability to automate the tasks. Thus, many tasks have been implemented with deep learning techniques. Among CNN-based methods, "You Only Look Once" (YOLO), particularly its latest lightweight version, YOLO-v12, stands out as highly efficient and accurate for real-time object detection tasks.

This paper proposes a lightweight Halal-logo detection system built by fine-tuning two YOLO-v12 backbones, specifically YOLO-v12 nano and small models, aiming to identify and classify up to 50 Halal logos accurately. The main research question in this paper is how effective a lightweight YOLO-v12 model is in accurately detecting and classifying Halal logos on product packaging.

The primary contribution is the transfer learning study for 50-class Halal logo detection. A model with COCO weights can gain more accuracy by fine-tuning than training from scratch. The second contribution is comparing the YOLO-v12 nano against the YOLO-v12 small model. Both models were trained under similar conditions to clarify the accuracy and latency trade-off for real-time deployment. The third contribution in this paper is the reproducible training and evaluation pipeline code release in GitHub.

# lITERATURE rEVIEW

## Halal Logo Detection and Classification

Before adopting deep learning, Halal logo detection relied primarily on traditional image processing techniques to classify and detect logos on product packaging.

Saipullah and Ismail [5] proposed a method based on the fractionalized principle magnitude (FPM) algorithm using the magnitude of the 1D Fourier transform to classify the logo. The proposed method was compared with other logo detection methods such as Histogram of Oriented Gradients (HOG), Hu Moments, Zernike Moments and Wavelet Color Histogram (WCH). The results show that the method outperforms the other methods.

Saipullah et al. [6] presented Fractionalized Principle Magnitude (FPM) for halal logo classification to extract the global and local features. The proposed method was compared with HOG, Hu moment, Zernike moment and WCH in classifying 50 approved Halal logos. The proposed method achieved the highest accuracy, with 90.4% and outperformed all the methods.

Razali et al. [7] developed a detection system to classify the authentic Jabatan Kemajuan Islam Malaysia (JAKIM) halal logo from unauthentic ones. The system consists of image acquisition, image preprocessing, detection and recognition. They used SURF, Scale-Invariant Feature Transform (SIFT), Generalized Search Trees (GIST), and K-means throughout the system. The system achieved 86.67% accuracy in classifying the authentic and unauthentic logos.

With the advancement of deep learning, recent studies have transitioned towards convolutional neural network (CNN)-based methods for more robust and accurate Halal logo detection. Deep learning approaches have demonstrated effectiveness across various signal processing and classification tasks, establishing their versatility in handling complex pattern recognition problems [8].

Hendrick et al. [9] used a deep learning model, GoogleLeNet, as a backbone to classify halal logos and product logos from different countries. They trained their model on NVIDIA DIGITS, a platform to train and manage the model and create an application to use it. The experiment showed that their model has 81.73% correctly classifying the halal logo.

Hasan et al. [10] introduced an automated system to detect and verify the logo by using the Speeded Up Robust Features (SURF) algorithm. It detected the Halal logo and matched it with reference images of the certified Halal logo by JAKIM. They tested their method with 100 images of certified and fake Halal logos. The results showed promising performance with 85.71% accuracy and on par with other existing methods.

Chew & Mohd-Mokhtar [11] developed a vision-based halal logo verification system using Visual Geometry Group (VGG)-16. The system consists of two stages: developing an image comparison algorithm and creating a graphical user interface (GUI). They used 51 unique Halal logos from 33 countries. The VVG-16 model performed well with 91.67% accuracy in the test sample.

## Object Detection for Small Objects and Logos

The object detection framework is widely used beyond Halal logo recognition, especially for tasks involving small objects and logos. Various versions of YOLO, such as YOLO-v5, YOLO-v7, and the most recent YOLO-v12, have shown significant performance and efficiency for small-object detection tasks.

Lee et al. [12] proposed a crop disease diagnosis solution using the YOLO-v5 model to perform object detection tasks. The solution also included an image captioning model using the Inception V3 and Transformer models. The average BLEU score of the image captioning model is 64.96%, and the mAP50 for YOLO-v5 is 0.382, which needs more improvement.

RetinaNet and EfficientDet models are also used in object detection. They are the improved model in accuracy with focal loss and multi-scale feature fusion. ERetinaNet is an efficient RetinaNet proposed to improve mammographic breast mass detection accuracy and inference speed [13].

Wang et al. [14] introduced a large logo detection dataset with complete annotation named LogoDet-3K. It consists of 3,000 logo categories with 158,652 images. They also proposed Logo-Yolo to solve the problems of multi-scale objects, imbalanced classes and inconsistent bounding-box regression. It showed that it has a 4% improvement compared with YOLO-v3.

Despite these advances, current literature primarily focuses on general or commercial brand logos. Certification marks such as Halal logos, characterized by strict graphical regulations and constraints, remain underexplored, especially with the latest YOLO-v12 backbone architectures.

## Transfer Learning in Object Detection

Transfer learning is one of the deep learning methods that uses the weight of the pre-trained model to fine-tune the detection accuracy. Most of the pre-trained models are trained with ImageNet or COCO large datasets. The model's early backbone layers are commonly frozen when performing transfer learning to preserve general image features and accelerate convergence.

Siddiqi [15] presented the effectiveness of transfer learning and fine-tuning in improving fruit image classification accuracy. They fine-tuned two models, Inception V3 and VGG-16, with the Fruits 360 dataset comprising 72 classes and 48,249 images. Their results showed that VGG-16 achieved the highest accuracy of 99.27%.

Tang et al. [16] conducted a study exploring transfer learning by fine-tuning models to detect human presence. They used 1463 images and split the images into train, validation and test subsets with a ratio of 70:20:10. They fine-tuned three RetinaNet with different backbones, such as ResNet50, ResNet101 and ResNet152, with the dataset. Their experiment showed that RetinaNet with ResNet152 backbone has the highest AP of 74.4% with an inference speed of 13.09 FPS.

Most studies focused primarily on large backbones such as ResNet, Inception and VGG models. Deep learning approaches have shown effectiveness across various signal processing and pattern recognition applications, including EEG signal classification [17], image quality assessment [18] and malware detection [19], demonstrating the versatility of these architectures beyond traditional computer vision tasks. On the other hand, smaller backbone architectures, such as YOLOv12-nano and YOLOv12-small, have rarely been explored for fine-tuning in the existing literature. This indicates a gap in exploring and understanding the effectiveness and potential advantages of fine-tuning lightweight models for specific detection tasks.

## Research Gap

Halal logo detection and classification without deep learning rely on handcraft features, and it is hard to scale the project to many logos. Early deep-learning approaches involved a small dataset with less than 10 Halal logos. No studies have used YOLO models to detect the Halal logo. Moreover, there is not much accuracy and computation efficiency benchmarking across different YOLO backbone sizes, nano and small size models. This paper fills these critical gaps by benchmarking YOLO-v12 nano and YOLO-v12 small models and quantifying the performance benefits of transfer learning on the Halal logo dataset.

# Methodology

The methodology introduces the latest YOLO-v12 architecture that is used in this study. YOLO-v12 is also known for its real-time performance and improved performance detecting small objects. YOLO-v12 integrates a CSPDarknet backbone with an enhanced Path Aggregation Network (PANet) and a decoupled detection head designed to handle it effectively. YOLO-v12 nano and YOLO-v12 small are used in this study.

YOLO-v12 nano is a compact variant with fewer parameters than YOLO-v12 small. It is optimized with a memory-constrained environment suitable for edge devices to maintain near real-time inference capability. YOLO-v12 small is a balanced model with a strong trade-off between accuracy and inference time. It is suitable for general object detection tasks.

## Transfer Learning and Training Procedure

The training procedure involves two primary strategies: from-scratch training and transfer learning for the backbones. The from-scratch training is training the model from scratch without using transfer learning. The model is trained without base COCO knowledge. YOLO-v12 small is trained on 100 epochs with a batch size of 16. The learning rate is 0.01.

Transfer learning is training the model with its COCO knowledge with the dataset. YOLO-v12 nano and YOLO-v12 small are trained on 50 epochs with a batch size 16. Both model's first three backbone layers froze to retain low-level general image features. The learning rate for this training is 0.001.

The default setting is used for both strategies. The input size is 640 x 640 resolution. The optimizer is Stochastic Gradient Descent (SGD) with momentum, and the loss function is composed of Binary Cross-Entropy (BCE) losses for class prediction and objectness scores, combined with Complete Intersection-over-Union (CIoU) loss for bounding box regression. The data augmentation includes the mosaic, hue, saturation, and value (HSV), translation, and scaling augmentation. Table 1 shows the overall setting for both training.

**TABLE 1.** YOLO-v12 Training Setting

|  |  |  |
| --- | --- | --- |
| **Hyperparameter** | **YOLO-v12 (scratch)** | **YOLO-v12 (fine-tune)** |
| Hyperparameter | Training from Scratch | Fine-tuning (Transfer Learning) |
| Epochs | 100 | 50 |
| Initial Learning Rate | 0.01 (default) | 0.001 (default) |
| Optimizer | SGD with Momentum | SGD with Momentum |
| Momentum | 0.937 | 0.937 |
| Weight Decay | 0.0005 | 0.0005 |
| Batch Size | 16 | 16 |
| Input Image Size | 640 × 640 | 640 × 640 |
| Backbone Layers Frozen | None | First three layers |
| Data Augmentation | Mosaic, HSV, Scaling, Translation (Ultralytics defaults) | Mosaic, HSV, Scaling, Translation (Ultralytics defaults) |
| Early Stopping (Patience) | 15 epochs | 15 epochs |
| GPU Hardware | NVIDIA RTX 2080 Ti | NVIDIA RTX 2080 Ti |

The dataset [20] is published in Roboflow. The dataset consists of 50 classes and 1292 images. The dataset is split into train, validation and test with a ratio of 72:26:2, which the author sets.

The evaluation metrics used in this study are mean average precision (mAP), average precision (AP), latency, electrical energy consumed () and carbon emissions (). mAP is averaged over intersection over Union (IoU) thresholds from 0.5 to 0.95 in 0.05 steps. It is the primary metric used to evaluate the detector quality. AP is the average precision at a single IoU threshold of 0.50. Latency estimates the end-to-end inference time for a single 640 x 640 image on the GPU. It indicates real-time capability. Energy used is the electrical energy consumed during the training while is the estimated carbon emissions associated with the energy. The model is trained on an RTX 2080 Ti and assumes the average power consumption in watts is 250 watts. The equation of is shown in Equation (1), and the carbon emissions equation is shown in Equation (2).

(1)

(2)

where H is the total GPU training time in hours, P is the average GPU power draw in watts (W), and EF is the grid-specific emission factor in kg . The EF is 0.4 kg per kWh in Malaysia.

# rESULTS AND dISCUSSION

## Results

The models have done the training. Table 2 shows the accuracy of the split dataset. Fine-tuning boosts the standard backbone from 0.919 to 0.945 mAP@50:95. The nano backbone edges it at 0.947 mAP, confirming that a much smaller model can match large-backbone accuracy on this task.

**TABLE 2.** The accuracy of the split dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Regime** | **mAP@0.5:0.95** | **AP@0.5** | **Precision** | **Recall** |
| YOLO-v12 small | Scratch | 0.919 | 0.995 | 0.917 | 0.902 |
| YOLO-v12 small | Fine-tune | 0.945 | 0.995 | 0.984 | 1 |
| YOLO-v12 nano | Fine-tune | 0.947 | 0.995 | 0.981 | 1 |

Table 3 shows the inference speed and memory of the three models. YOLO-v12 nano runs three times faster than scratch v12-s (4.7 ms vs 15.8 ms) and uses 50 % less VRAM (509 MB). The fine-tuned YOLO-v12 small is still real-time with 12ms, but the cost is around 904MB VRAM.

**TABLE 3.** Inference Speed and Memory

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Parameter | Latency (ms) | Peak VRAM (MB) |
| YOLO-v12 small Scratch | 9,250,230 | 15.8 | 975 |
| YOLO-v12 small Fine-tune | 9,250,230 | 12.3 | 904 |
| YOLO-v12 nano Fine-tune | 2,566,478 | 4.7 | 509 |

Table 4 shows the training energy and carbon footprint for the three models. The YOLO-v-12 small train from scratch produced the most . Furthermore, it takes up the longest training time compared to fine-tuning. Fine-tuned YOLO-v12 nano needs only 0.18 h of training time and 0.04 kWh, one-third the energy of scratch training, while matching and slightly surpassing the larger backbone's accuracy. In short, transfer learning can deliver models with a three times reduction in energy and carbon compared with training a large model from scratch.

**TABLE 4.** Training energy and carbon footprint

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **GPU (hours)** | **Energy (kWh)** | **(kg)** |
| YOLO-v12 small Scratch | 0.56 | 0.14 | 0.06 |
| YOLO-v12 small Fine-tune | 0.26 | 0.06 | 0.02 |
| YOLO-v12 nano Fine-tune | 0.18 | 0.04 | 0.02 |

## Discussion

The effect of fine-tuning benefits the model to improve its accuracy. It has raised YOLO-v12 small accuracy from 0.919 to 0.945 mAP@50:95 while halving GPU time from 0.56 to 0.26 hours. This is because the precision climbs to 0.984, and recall reaches 1.000. All the gain comes from eliminating residual false negatives that the scratch model missed. This confirms that low-level features learned on natural-image datasets remain useful for highly stylized certification logos and that only shallow layers need re-training. The YOLO-v12 nano has fewer parameters than the YOLO-v12 small, but it can outperform the YOLO-v12 small with 0.947 mAP. The inference latency drops from 12 ms to 4.7 ms and peak VRAM from 904 MB to 509 MB, making the nano model deployable on 4 GB edge devices without batching. These results show modern nano backbones provide high quality and speed. They deliver equal performance with 40 to 50% reduced memory.

The scratch training consumes 0.14 kWh and produces 0.06 kg CO₂, while the transfer learning of the same backbone cuts energy to 0.06 kWh a reduction of −57 % and carbon to 0.02 kg. Switching to YOLO-v12 nano further lowers training energy to 0.04 kWh, delivering a three-fold carbon reduction versus scratch while retaining accuracy.

A fine-tuned YOLO-v12 nano checkpoint offers 0.95 mAP, sub-5 ms latency, and < 600 MB VRAM. This shows that the YOLO-v12 nano's performance sufficient for in-line packaging inspection or mobile auditing apps.

# CONCLUSION

This study presents a lightweight Halal-logo detection system based on fine-tuning two YOLO-v12 backbones. Experiments on a 50-class public dataset demonstrate that transfer learning boosts accuracy from 0.919 to 0.945 mAP@50:95 while halving training time and energy and that the compact YOLO-v12-nano attains 0.947 mAP with only 4 ms latency and 509 MB VRAM which the performance suitable for real-time, edge deployment. Energy use drops from 0.14 kWh (scratch) to 0.04 kWh (nano fine-tune), reducing estimated CO₂ emissions to 0.02 kg. All training scripts and checkpoints are released to facilitate immediate adoption and reproducibility.

# Acknowledgments

# FRDGS support this research. No funds were received to cover article-processing or publication charges.

# References

1. O.A. Al-Mahmood, and A.M. Fraser, “Perceived challenges in implementing halal standards by halal certifying bodies in the United States,” PLoS ONE **18**(8), e0290774 (2023).
2. H.A. Usmani, and M. Ayaz, “Regulatory Framework of Halal Industry in Pakistan: Structural Issues and Challenges,” COMSATS Journal of Islamic Finance **9**(1), 1–12 (2024).
3. R.M. Ellahi, L.C. Wood, M. Khan, and A.E.-D.A. Bekhit, “Integrity Challenges in Halal Meat Supply Chain: Potential Industry 4.0 Technologies as Catalysts for Resolution,” Foods **14**(7), 1135 (2025).
4. Q.X. Pang, K.J. Low, K.L. Lew, Y.H. Choo, S.A. Babale, A.P. Yunus, and C.S. Lee, “A Review on Mechanical Fuzzy Logic Control Cutters for Latex Glove,” IJORAS **6**(2), 76–83 (2024).
5. K.M. Saipullah, and N.A. Ismail, “Determining Halal Product Using Automated Recognition of Product Logo,” Journal of Theoretical and Applied Information Technology **77**(2), 190–198 (2005).
6. K.M. Saipullah, N.A. Ismail, and Y. Soo, “Feature Extraction method for Classification of Approved Halal Logo in Malaysia using Fractionalized Principle Magnitude,” Engineering Management Reviews **2**(2), (2013).
7. S.M. Razali, N.F. Isa, Z.Z. Htike, and W.Y.N. Naing, “Vision-Based Verification of Authentic Jakim Halal Logo,” **10**(21), (2015).
8. X. Xu, H. Zhang, Y. Ma, K. Liu, H. Bao, and X. Qian, “TranSDet: Toward Effective Transfer Learning for Small-Object Detection,” Remote Sensing **15**(14), 3525 (2023).
9. Hendrick, C.-M. Wang, Aripriharta, C.-G. Jhe, P.-C. Tsu, and G.-J. Jong, “The Halal Logo Classification by Using NVIDIA DIGITS,” in *2018 International Conference on Applied Information Technology and Innovation (ICAITI)*, (IEEE, Padang, Indonesia, 2018), pp. 162–165.
10. N. Hasan, N. Awang, and F.N. Jamrus, “An Application of SURF Algorithm on JAKIM’s Halal Logo Detection,” Gjat **SI**(1), 18–26 (2023).
11. W.J. Chew, and R. Mohd-Mokhtar, “Real-Time Vision-Based Halal Logo Verification System,” in *Proceedings of the 13th National Technical Seminar on Unmanned System Technology 2023—Volume 1*, edited by Z. Md. Zain, Z.H. Ismail, H. Li, X. Xiang, and R.R. Karri, (Springer Nature Singapore, Singapore, 2024), pp. 201–215.
12. D.I. Lee, J.H. Lee, S.H. Jang, S.J. Oh, and I.C. Doo, “Crop Disease Diagnosis with Deep Learning-Based Image Captioning and Object Detection,” Applied Sciences **13**(5), 3148 (2023).
13. L. Chen, Y. Zhou, and S. Xu, “ERetinaNet: An Efficient Neural Network Based on RetinaNet for Mammographic Breast Mass Detection,” IEEE J. Biomed. Health Inform. **28**(5), 2866–2878 (2024).
14. J. Wang, W. Min, S. Hou, S. Ma, Y. Zheng, and S. Jiang, “LogoDet-3K: A Large-Scale Image Dataset for Logo Detection,” (2020).
15. R. Siddiqi, “Effectiveness of Transfer Learning and Fine Tuning in Automated Fruit Image Classification,” in *Proceedings of the 2019 3rd International Conference on Deep Learning Technologies*, (Association for Computing Machinery, New York, NY, USA, 2019), pp. 91–100.
16. J.C. Tang, A.F.B. Ab. Nasir, A. P. P. Abdul Majeed, L.L. Thai, M.A. Mohd Razman, and I. Mohd Khairuddin, “Fine-tuned RetinaNet models for Vision-based Human Presence Detection,” MEKATRONIKA **4**(2), 16–23 (2022).
17. K.L. Lew, K.S. Sim, and Z. Ting, “Deep Learning Approach EEG Signal Classification,” International Journal on Informatics Visualization **8**(3–2), 1693–1702 (2024).
18. K.S. Sim, M.E. Nia, C.P. Tso, and W.K. Lim, “Performance of new signal-to-noise ratio estimation for SEM images based on single image noise cross-correlation,” Journal of Microscopy **248**(2), 120–128 (2012).
19. M.N. Al-Andoli, K.S. Sim, S.C. Tan, P.Y. Goh, and C.P. Lim, “An Ensemble-Based Parallel Deep Learning Classifier With PSO-BP Optimization for Malware Detection,” IEEE Access 11, 76330–76346 (2023).
20. I. Zulkeefli, "FYP Dataset," Roboflow Universe (2022), available at https://universe.roboflow.com/izzatizulkeefli27-gmail-com/fyp-dataset.