Detecting Correct Helmet Use with Deep Learning

Abstract

In Taiwan, motorcycles and scooters are the primary mode of transportation, and improper helmet use remains a major contributor to severe road injuries and fatalities. This study investigates a computer-vision approach for automatically detecting correct and incorrect helmet wearing using the public HelmetML dataset (Patil et al., 2024), focusing on the half-face subset, the most common helmet type among Taiwanese riders.

We fine-tune a pretrained MobileNetV2 model and design a compact convolutional neural network trained from scratch, both evaluated under identical experimental conditions. MobileNetV2 achieves 99.27 percent test accuracy, while the refined CNN reaches 93.16 percent, quantifying the benefit of transfer learning over a lightweight task-specific model. These results indicate that deep-learning–based classification can provide reliable support for future road-safety applications such as automated helmet-monitoring and enforcement systems.

Introduction

This study examines two deep-learning approaches for detecting proper and improper helmet use in motorcycle traffic imagery, with a specific focus on half-face helmets, the type most commonly used in Taiwan. First, a pretrained MobileNetV2 model is fine-tuned to reproduce the behaviour reported in the original HelmetML study and to establish a strong performance baseline. Second, a custom convolutional neural network is designed and trained from scratch to assess whether a lightweight, task-specific architecture can achieve competitive accuracy without relying on external pretrained features.

Both models are trained and evaluated using identical dataset partitions and preprocessing procedures, enabling a controlled comparison between a state-of-the-art transfer-learning strategy and a compact model trained from scratch. This modelling framework aligns with the long-term goal of developing reliable automated helmet-monitoring systems that could be integrated into road-safety infrastructure.

Dataset

The experiments are based on the HelmetML dataset introduced by Patil et al. (2024). The full dataset comprises 28,736 labelled images across four helmet categories—full-face, half-face, modular and off-road. Each one of them represented in both correct and incorrect wearing conditions under diverse lighting environments. In this work, only the half-face subset is used. It contains 7,290 images in total, evenly split between correct and incorrect helmet wearing, ensuring a perfectly balanced binary classification setting. Restricting the dataset to half-face helmets reduces computational demands while remaining relevant for practical use in Taiwan, where this helmet type predominates.

Further details on dataset construction, augmentation ratios, environmental conditions and acquisition devices are extensively documented in the original HelmetML publication and are therefore not repeated here.

Method

Overview

Two models are developed in this study: a pretrained MobileNetV2 and a custom CNN. The methodological description is divided into: components shared by both models and components that differ between them.

Dataset Preparation

A custom PyTorch `Dataset` class was used to load each image, convert it from BGR to RGB and apply the appropriate transform before returning the image–label pair. Images were organized into two folders (“Correct Way” and “Incorrect Way”). Any unreadable files were discarded.

To preserve strict class balance, images of each class were shuffled once and split separately into 70 percent training, 15 percent validation and 15 percent test data. The class-wise partitions were then merged, resulting in stratified splits identical for all models. Labels were encoded as 1 for correct and 0 for incorrect. These fixed partitions were used for both MobileNetV2 and the custom CNN.

No additional augmentation

The HelmetML dataset already includes extensive augmentation: for every original image, three augmented versions are provided. Because these samples are integrated into the dataset, any additional augmentation during training would distort the distribution and artificially oversample some transformations. For this reason, no extra augmentation was applied.

Model-specific normalization

For MobileNetV2, images were resized to 224 × 224 pixels, converted to tensors and normalized using ImageNet mean and standard deviation to match the pretrained training conditions.

For the custom CNN, images were resized to the same resolution but kept within the native [0, 1] tensor range, which is appropriate for training from scratch.

Common Training Pipeline

Both the MobileNetV2 model and the custom CNN were trained using the same optimization strategy, data flow, and evaluation procedure to ensure a strictly controlled comparison. All computations were executed on the available device (GPU when available, otherwise CPU), and each model was moved to this device before training.

For both architectures, training and validation used CrossEntropyLoss as the objective function and the Adam optimizer with a fixed learning rate of 1×10^{-3}, without any scheduling or additional momentum settings. Each model was trained for

30 epochs with a batch size of 8, which provides a stable compromise between memory consumption and gradient noise.

Data were fed through identical `DataLoader` configurations. The training loader used `shuffle=True` to randomize the sample order at every epoch, whereas the validation and test loaders did not shuffle, ensuring reproducible evaluation. In each epoch, the training loop for both models followed the same sequence: forward pass, loss computation, backpropagation, parameter update, and accumulation of loss and accuracy metrics. Validation was then performed with the model in evaluation mode and with gradient computation disabled.

Model checkpointing was also identical. After every epoch, the validation loss was compared to the best value observed so far; if an improvement was detected, the model weights were saved. At the end of training, this best-performing version (according to validation loss) was reloaded and evaluated once on the held-out test set to obtain the final accuracy and confusion matrix. Aside from architectural differences and normalization choices described elsewhere, the entire training and evaluation pipeline was shared between both models.

Motivation for Model Selection and Experimental Strategy

In the original HelmetML study, the authors analysed four different helmet types, specifically full-face, open-face, modular and off-road helmets. They also evaluated several deep learning architectures, including VGG19, ResNet50 and MobileNetV2, both in their pretrained and fine-tuned variants. Their objective was to build a general model capable of identifying correct and incorrect helmet wearing across diverse visual conditions.

In the present work, a more focused and controlled methodology was adopted due to practical constraints related to dataset management and the available computational resources. The analysis was limited to a single helmet category, namely half-face helmets. MobileNetV2 was selected as the only pretrained architecture for experimentation. This choice was motivated by the computational limitations of the Google Colab environment. Larger models such as VGG19 or ResNet50 require significantly more GPU memory, longer training times and greater computational power, which makes them unsuitable in this context. MobileNetV2 offers a favourable balance between efficiency and accuracy and is therefore well aligned with the computational reality of the project.

The first step of the experimental design was to fine-tune MobileNetV2 on the half-face dataset to verify whether the performance described in the HelmetML paper could be reproduced. This step establishes a baseline that is directly comparable to the reference work and confirms the validity of the training setup.

Once this baseline was established, a second model was introduced. A compact convolutional neural network was designed and trained entirely from scratch. This decision was driven by two considerations. The original study did not explore non-pretrained architectures, which leaves open the question of how effectively a model can learn this task without relying on ImageNet features. Furthermore, comparing a pretrained MobileNetV2 to a lightweight custom CNN makes it possible to evaluate how much of the final performance depends on transfer learning and how much can be achieved through a task-specific architecture alone. This comparison provides insight into the necessity of pretrained features for detecting subtle cues such as helmet orientation, strap fastening or coverage.

For these reasons, the experimental strategy consists of two stages. The first stage examines the behaviour of a fine-tuned MobileNetV2 and assesses whether the results from the reference study are reproducible under the present constraints. The second stage evaluates the performance of a custom CNN trained from scratch and compares it with the pretrained model in order to understand the contributions of transfer learning versus direct task-specific learning.

MobileNetV2: Performance Analysis and Results

The training of the MobileNetV2 model showed stable and progressive behavior, demonstrating that the network effectively learned the features needed to distinguish between images where a helmet is correctly worn and those where it is not. From the very first epochs, a rapid increase in accuracy is observed on both the training set and the validation set. Validation accuracy surpasses 96% in the first epoch and remains consistently high throughout the process, fluctuating between 96% and 99%. This regular trend indicates not only that the model learns quickly, but also that it maintains a very high generalization capability, avoiding overfitting. The closeness between the training and validation curves, together with the progressive decrease in loss, confirms that the model is converging toward a solution representative of the data.

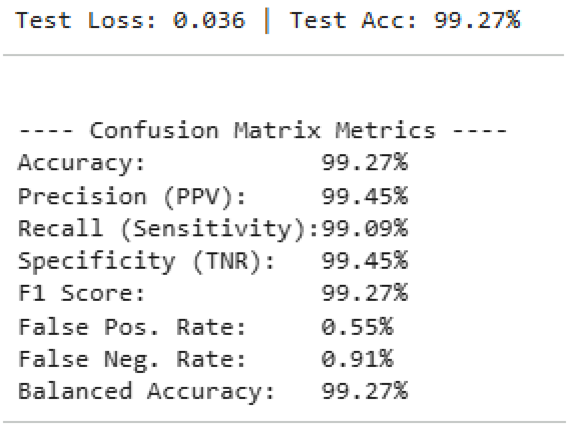
The model's behavior is further supported by the results of the confusion matrix. Out of a total of 1,096 samples, MobileNetV2 makes only eight errors: three false positives and five false negatives. The model correctly recognizes the presence of a helmet in nearly all cases, with only five instances where it fails to detect a helmet that is actually present.

The metrics derived from the confusion matrix reinforce this interpretation. The precision of 99.45% shows that positive predictions are almost always reliable, while the recall of 99.09% highlights the model's ability to correctly identify the vast majority of positive cases. Specificity also reaches very high values, indicating that the network is equally effective at recognizing negative examples.

A further supporting element emerges from the final evaluation on the test set. Test accuracy again reaches 99.27%, accompanied by a loss of just 0.036. This result shows that the model maintains the same performance level even on previously unseen data. If overfitting had occurred, we would observe a drop in accuracy or an increase in loss on the test set. On the contrary, the model's behavior is extremely consistent across training, validation, and test phases, confirming strong generalization capability.

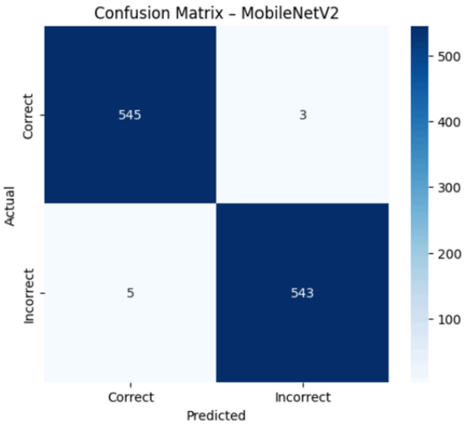
The network successfully extracted the relevant image features, accurately distinguishing the presence or absence of a helmet. It is likely that the few remaining errors are due to non-ideal visual conditions, such as poor lighting, unusual angles, partial occlusion of the helmet, or color similarities with other objects. Despite this, the extremely small number of errors demonstrates that the model is robust and reliable even when faced with such variability.

The obtained test accuracy of 99.27 percent is broadly consistent with the strong performance reported for MobileNetV2 in the HelmetML study. Although a direct numerical comparison is not possible because the original work evaluates all helmet types rather than only half-face helmets, our result still serves as a reliable benchmark demonstrating that pretrained architectures perform exceptionally well on this classification task.



```
Test Loss: 0.036 | Test Acc: 99.27%

---- Confusion Matrix Metrics ----
Accuracy:              99.27%
Precision (PPV):       99.45%
Recall (Sensitivity):99.09%
Specificity (TNR):     99.45%
F1 Score:              99.27%
False Pos. Rate:       0.55%
False Neg. Rate:       0.91%
Balanced Accuracy:     99.27%
```



Custom CNN Architecture

A baseline convolutional neural network was first implemented to examine how a simple, non-pretrained model would perform on the helmet-wearing classification task. This initial configuration consisted of three convolutional layers arranged in a sequential block structure. Each layer was followed by batch normalization and a ReLU activation, and max pooling was applied after every block, causing the spatial resolution to be reduced at each stage of the feature extraction process. After the third block, adaptive global average pooling compressed the feature map into a single feature vector, which was processed by a small two-layer classifier with dropout. The complete implementation of this baseline configuration is provided in the corresponding notebook included with this work.

This setup provided a computationally lightweight starting point, but the combination of only three convolutional stages and early repeated pooling limited the model's ability to retain fine-grained visual information such as strap details, partial coverage or subtle helmet misalignment.

To overcome these constraints, a refined architecture with greater representational capacity was designed. The improved model contains four convolutional layers arranged in four successive blocks, again each followed by batch normalization and a ReLU activation. In contrast to the baseline, the first block does not apply pooling, allowing the network to extract high-resolution features before downsampling begins. Max pooling is applied only after the second, third and fourth blocks, ensuring a more gradual and information-preserving reduction in spatial size. The deeper four-layer feature extractor enables the network to learn progressively richer and more discriminative patterns relevant to distinguishing correct from incorrect helmet use.

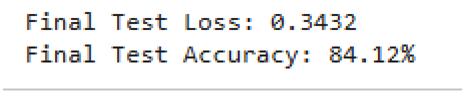
As in the baseline configuration, the output of the final convolutional block is reduced by adaptive global average pooling, but the resulting feature vector is now substantially more informative due to the deeper convolutional hierarchy. The classifier head is simplified to a single fully connected layer with dropout followed by the output layer, providing more stable regularization without excessively weakening the signal.

Overall, increasing the depth from three to four convolutional layers and postponing the first pooling operation resulted in a more effective architecture. The refined model preserves critical spatial detail in the early processing stages and develops richer, more discriminative features in the later blocks. These structural adjustments make the improved CNN a more reliable baseline for evaluating how much can be achieved through task-specific learning alone. This configuration provides a clearer point of comparison both against the initial CNN and against the fine-tuned MobileNetV2, and it sets the stage for the performance analysis presented in the following section.

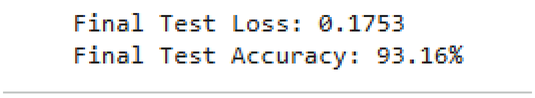
Comparative Evaluation of the Two CNN Configurations

CNN_baseline

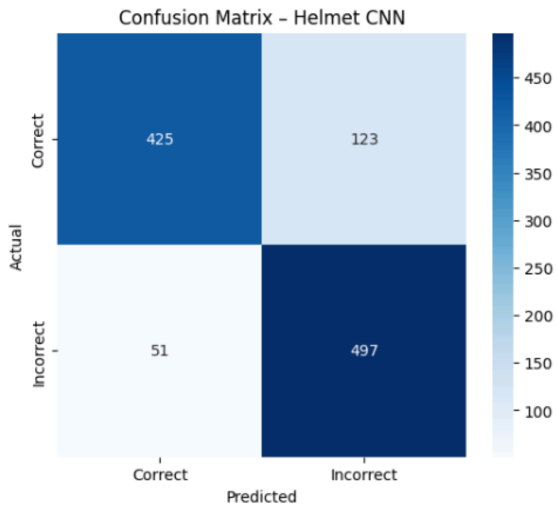
CNN_refined

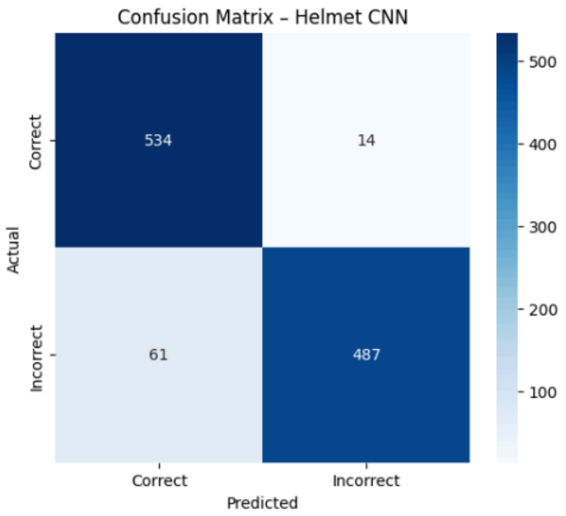


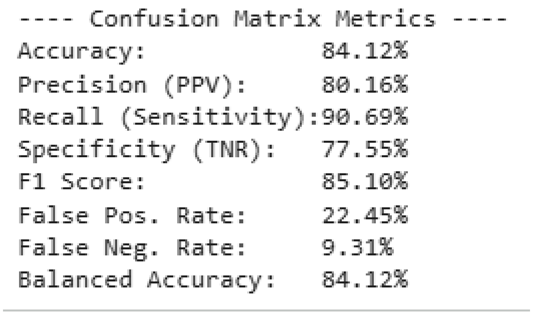
```
Final Test Loss: 0.3432
Final Test Accuracy: 84.12%
```



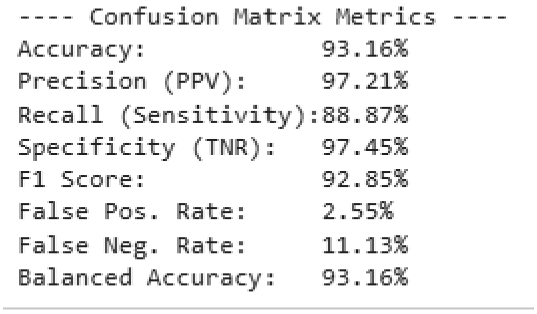
```
Final Test Loss: 0.1753
Final Test Accuracy: 93.16%
```







```
---- Confusion Matrix Metrics ----
Accuracy:             84.12%
Precision (PPV):      80.16%
Recall (Sensitivity):90.69%
Specificity (TNR):    77.55%
F1 Score:             85.10%
False Pos. Rate:      22.45%
False Neg. Rate:      9.31%
Balanced Accuracy:    84.12%
```



```
---- Confusion Matrix Metrics ----
Accuracy:             93.16%
Precision (PPV):      97.21%
Recall (Sensitivity):88.87%
Specificity (TNR):    97.45%
F1 Score:             92.85%
False Pos. Rate:      2.55%
False Neg. Rate:      11.13%
Balanced Accuracy:    93.16%
```

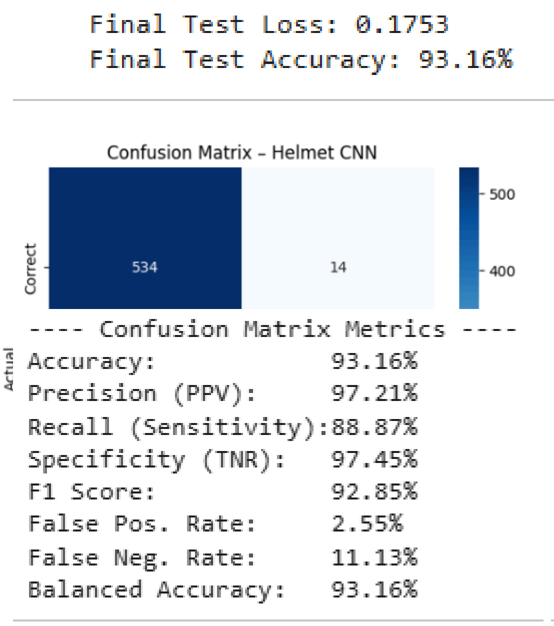
The comparison between the baseline and refined CNN models demonstrates that architectural adjustments had a substantial impact on performance. Increasing the depth of the feature extractor and delaying early spatial downsampling led to a marked improvement in accuracy, which increased from 84.12 percent to 93.16

percent. Precision and specificity rose sharply, indicating that the refined model is significantly more reliable in identifying incorrect helmet wearing and produces far fewer false alarms. Although recall decreased slightly, the overall balance of metrics, including a much lower false positive rate, shows that the revised architecture extracts more discriminative features and provides a considerably more robust classifier. These results confirm that a modest increase in model complexity can yield a clear performance gain while remaining far lighter than pretrained architectures such as MobileNetV2.

Performance Comparison Between MobileNetV2 and a Traditional CNN

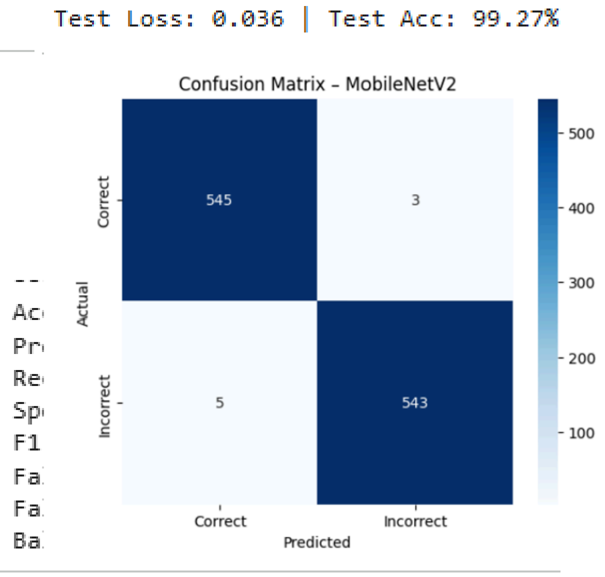
CNN_refined

MobileNetV2



```
Final Test Loss: 0.1753
Final Test Accuracy: 93.16%
```

```
---- Confusion Matrix Metrics ----
Accuracy:              93.16%
Precision (PPV):       97.21%
Recall (Sensitivity):88.87%
Specificity (TNR):     97.45%
F1 Score:              92.85%
False Pos. Rate:       2.55%
False Neg. Rate:       11.13%
Balanced Accuracy:     93.16%
```



```
Test Loss: 0.036 | Test Acc: 99.27%
```

```
--
Ac
Pr
Re
Sp
F1
Fa
Fa
Ba
```

The comparison between the refined CNN and the fine-tuned MobileNetV2 highlights the substantial impact of transfer learning on this classification task. Although the improved CNN achieves solid performance with an accuracy of 93.16 percent and a balanced set of precision and specificity values, its results remain clearly below those of MobileNetV2. The pretrained model reaches 99.27 percent accuracy and produces only a handful of misclassifications, reflected in extremely low false positive and false negative rates.

This difference is expected, since the CNN is trained entirely from scratch and must learn all relevant visual features solely from the task-specific dataset, whereas

MobileNetV2 benefits from large-scale ImageNet pretraining that already encodes rich and diverse feature representations. The refined CNN therefore demonstrates how far a compact, domain-specific architecture can go on its own, while the superior performance of MobileNetV2 confirms the effectiveness of transfer learning for extracting subtle cues such as strap positioning and helmet coverage.

Together, these results show that a well-designed scratch-trained CNN can provide competitive baseline performance, but pretrained architectures remain significantly more reliable for high-precision helmet-wearing detection.

Conclusion

This work examined deep learning approaches for detecting correct and incorrect helmet wearing using the half-face subset of the HelmetML dataset, the most common helmet type in Taiwan. Focusing on this specific category kept the computational demands manageable while preserving direct relevance for real-world applications.

Two modelling strategies were evaluated. A fine-tuned MobileNetV2 served as the transfer-learning benchmark, and two custom CNN architectures were trained from scratch to assess how much performance can be achieved without pretrained features. The refined CNN showed clear improvements over the baseline through increased depth and more gradual downsampling, demonstrating that carefully designed lightweight models can learn discriminative features effectively.

Nonetheless, the comparison revealed a substantial performance gap between the scratch-trained CNN and MobileNetV2. While the refined CNN achieved strong accuracy and balanced metrics, the pretrained MobileNetV2 reached near-perfect performance, confirming the advantages of transfer learning for capturing subtle cues related to helmet positioning and coverage.

Overall, these findings indicate that pretrained architectures are the most reliable option for high-precision helmet-use detection, while compact CNNs remain useful for scenarios with limited computational resources. Such models could form the basis for future governmental or municipal safety systems, including automated helmet-monitoring infrastructure. Future work may explore hybrid architectures that combine lightweight designs with pretrained components, as well as deployment strategies for real-time inference on embedded devices

Reference

Patil, K., Jadhav, R., Suryawanshi, Y., Chumchu, P., Khare, G., & Shinde, T. (2024). HelmetML: A dataset of helmet images for machine learning applications. *Data in Brief*, *56*, 110790. https://doi.org/10.1016/j.dib.2024.110790

Use of AI Tools and Assistance

This project was developed using standard deep-learning frameworks and programming tools. AI-based assistants, such as large language models, were used exclusively for code structuring, debugging suggestions, linguistic editing and text refinement. All modelling decisions, experimental design choices and result interpretations were made by the authors, and all model training and analysis were executed manually in Python notebooks.

Supplementary Material

The complete implementation of all models used in this study is included as part of the submission. This comprises the notebook for the fine-tuned MobileNetV2 model as well as the notebooks corresponding to the baseline CNN and the refined CNN architecture. Each notebook contains the full training pipeline, evaluation code and reproducible experimental setup used to obtain the reported results.