EMERGING TECH CONFERENCE – Edge Intelligence

Volume 03, 2024, Pages 127 – 132

Proceedings of Emerging Tech Conference: Edge Intelligence 2024

Computer vision-based identification of motorcycle helmets using a TPU-based platform

Sotiris Michalakeas and Christoforos Kachris
University of West Attica Department of Electrical and Electronics Engineering Athens, Greece

Abstract

The use of helmets can significantly reduce fatal injuries in motorcycle accidents. How ever, there are still many instances where helmets are not used. In this paper, we present a computer-vision system for the automatic identification of helmet use by motorbike riders, that can be used to encourage the use of helmets. The system is developed on a Google Coral Dev Board with an embedded EdgeTPU, connected to a display and a buzzer that are used to provide feedback to the people it detects. Firstly, we present the training of the specific model with several optimizations and then we evaluate two versions of the SSD-MobileNetV2 model in terms of performance, accuracy and energy consumption. The proposed system can achieve high accuracy, low-latency and it consumes less than 4.2 Watts on a TPU board.

1 Introduction

According to the latest report from the European Transport Safety Council, respectively 15.5% and 2.9% of all road fatalities in the EU in 2018 were motorcyclists and moped riders. In Greece, from the 621 fatal accidents that were recorded in 2023, 211 of them were motorbike riders, while the overall reduction of the fatal accidents compared to 2022 was just 5% [1].

According to the data, helmets can significantly reduce the amount of fatalities among motorcycle riders in road accidents. In this paper we develop a low-cost, low-latency embedded system that automatically detects helmet use by motorbike riders using computer vision and provides feedback to promote helmet usage.

The system consists of a camera that feeds images in real time to the Google Coral Dev Board, which runs an object detection model and displays its results in a monitor. At the same time it sends logic signals for each class to an external microcontroller. The microcontroller processes these signals and controls an LCD and a passive piezo buzzer to provide both visual and auditory feedback to the motorbike riders.

The main contributions of this paper are the followings:

- Acomputer-vision system based on a low-cost Google Coral TPU that utilizes the advan tages of the TPU accelerators.
- Some optimization options for the training data and the training process to increase the accuracy of the helmet identifications.
- Integration of the computer vision system with a display and an audio device for the encouragement of helmet use when it is not used and positive feedback for when it is used.

- Performance evaluation of the proposed system using the SSD-MobileNetV2 and the SSD MobileNetV2-FPNlite architectures in terms of accuracy, inference throughput, and power consumption.

2 Related work

In recent years, the application of computer vision for helmet recognition has gained significant traction, particularly in enhancing workplace safety across various industries. Several studies have explored different methodologies to effectively identify helmet usage among workers, contributing to safer working environments.

In [3], it is presented a survey of several efforts for the automatic detection of helmets using computer vision algorithms.

One notable approach is presented by Wu and Zhao [4], who developed an intelligent vision based system that integrates pedestrian detection with helmet identification. Their method employs image recognition technologies to ensure compliance with safety protocols in construction sites, demonstrating the potential of computer vision to reduce accident rates significantly.

Another significant contribution comes from a study that proposed a dual-function system for helmet and identity recognition [5]. This research highlights the integration of helmet detection with worker identification, which is crucial for monitoring compliance in real-time. The authors conducted extensive testing under varying visual conditions, indicating that such systems can effectively enhance safety management in construction environments.

Moreover, advancements in deep learning have led to the development of robust helmet detection models that can operate in real-time. For instance, a study utilizing convolutional neural networks achieved high accuracy in detecting safety helmets across different scenarios, showcasing the effectiveness of these technologies in practical applications [2].

These advancements illustrate the critical role of computer vision in promoting safety through effective helmet recognition, paving the way for further research and development in this vital area. However, in these cases the identification of the helmet is restricted to in dustrial environments with low processing requirements. In the case of helmet detection for motorcycle riders, the system needs to use hardware accelerators to achieve low latency and high throughput as the objects are moving in high speed.

3 Training of the Helmet Detector

The training of the helmet detector was initially performed using a dataset from Kaggle that consists of 774 images of people with and without helmets. After a careful analysis of the dataset, it was found that many images (249) were duplicates and were removed, making the next trained model very inaccurate. To improve it, we added 500 more images and relabeled the entire dataset with two classes, ”With Helmet” and ”Without Helmet” using labelImg. These images were carefully sourced from the internet and contained objects in different scenarios, such as varying lighting levels, distances and backgrounds. The new dataset contained a total of 1015 annotated images in Pascal VOC format, but some of them were removed, because of the lower data quality that they provided. For example, cases where people weren't facing the camera, or were wearing a headscarf or a burqa. The final dataset would contain 958 images. During preprocessing, a python script would randomly split the full dataset into a training subset (80%), a validation subset (10%) and a testing subset (10%). The total amount of images is quite small to fully train an object detection model from scratch. However, we would be using a method called transfer learning, in which a pre-trained model is re-trained with a new custom dataset. If done right, this method can achieve great results with much fewer data and training time.

For the training of the models we used the TensorFlow framework and the Tensorflow Object Detection API, running on Google Colab, so that we can utilize a CUDA enabled GPU to massively accelerate the training process. We split the dataset into training, validation and testing data. The xml files, that contain the annotation data for each image, were processed and converted into csv files for each subset. These files would later be converted to TFRecords to make them compatible with TensorFlow. The pre-trained models (SSD-MobileNetV2 and SSD-MobileNetV2-FPNlite) were imported from the Tensorflow 2 Detection Model Zoo.

The training of the models required around 38,000-40,000 steps, as after that, the total loss metric was not decreasing any more. The batch size was set to 16 because of memory limitations, the warmup learning rate was set to 0.0266 and the base learning rate to 0.08. These hyperparameters were applied to both of the architectures. After the completion of the training process, we converted the TensorFlow models to TFlite and we quantized the models to make them compatible with the Edge TPU board.

4 Integration and Inference on Coral TPU board

The Coral TPU Development Board uses an iMX SoC from NXP (with a Quad-core Arm Cortex-A53, plus a Cortex-M4F) integrated with a TPU engine specialized for edge applications. The board we used has 4GB LPDDR4. Two folders containing each model were compressed (zip files) and transferred to the Google Coral board via a microSD card. With the camera and the monitor connected to the board, the next step would be to set up the resolution and the minimum confidence threshold. The inputs of the models were 510x510 for the SSD MobileNetV2 and 320x320 for the SSD-MobileNetV2-fpnlite, so the resolution of the camera would be set close to these resolutions (640x480) to avoid any further distortion of the images. The framerate would be set to 30 frames per second, as it was the maximum the camera could achieve. After some testing, the optimal minimum confidence threshold was found at 30%.

Figure 1: The detection results from a camera test, in the monitor

After some changes to the inference python script, two logic signals that contain the de tection status of each of the two classes would be transferred via two GPIO (General-Purpose Input/Output) pins to the external microcontroller. If any object is detected as class 2 (With out Helmet) then the negative visual and auditory feedback would be played. Positive feedback will only be played if every detected object is classified as class 1 (With Helmet).

Figure 2: Back and Front panel of the Integrated device using the Coral TPU board, LCD screen and the display

5 Performance evaluation

To evaluate the models' performance we used two different sets of images. The first one was the testing subset from the custom dataset which was used to get a first estimate of its accuracy. Due to the fact that this set was randomly created each time a new model was trained, we couldn't use these results to directly compare them with each other. So we gathered 11 more images, containing 23 instances of each class (46 total), with varying distances, lighting levels and environments (backgrounds) to compare the two final models.

SSD-MobileNetV2 (45 FPS)

TARGET / OUTPUT	With Helmet	Without Helmet	SUM
With Helmet	12 35.29%	2 5.88%	14 85.71% 14.29%
Without Helmet	4 11.76%	16 47.06%	20 80.00% 20.00%
SUM	16 75.00% 25.00%	18 88.89% 11.11%	28 / 34 82.35% 17.65%

Figure 3: Confusion matrix for the SSD-MobileNetV2

SSD-MobileNetV2-FPNlite (10 FPS)			
TARGET / OUTPUT	With Helmet	Without Helmet	SUM
With Helmet	10 31.25%	0 0.00%	10 100.00% 0.00%
Without Helmet	7 21.88%	15 46.88%	22 68.18% 31.82%
SUM	17 58.82% 41.18%	15 100.00% 0.00%	25 / 32 78.13% 21.88%

Figure 4: Confusion matrix for the SSD-MobileNetV2-FPNlite

Fig. 1 and Fig. 2 show the confusion matrices of the two models using the SSD-MobileNetV2 and the SSD-MobileNetV2-FPNlite architectures. As it is shown in the figures, the SSD MobileNetV2 offers slightly better overall accuracy and a faster inference time, but the SSD MobileNetV2-FPNlite is better in terms of identifying the riders that do not wear helmets.

The following equations show the Mean Average Precision of the SSD-MobileNetv2 and SSD-MobileNetV2-FPNlite.

SSD-MobileNetV2: $mAP \ = \frac{1}{n}\sum_1^n \frac{TP}{TP+FP} = \frac{1}{2}\cdot(\frac{12}{12+2} + \frac{16}{16+4}) = \ 0,8286$

SSD-MobileNetV2-FPNlite: $mAP \ = \frac{1}{n}\sum_1^n \frac{TP}{TP+FP} = \frac{1}{2}\cdot\left(\frac{10}{10+0} + \frac{15}{15+7}\right) = \ 0,8409$

Based on the testing of the models, the SSD-MobileNetV2-FPNlite seems to work very well on good lighting and on relatively small distances (up to 7m). This model is quite conservative in the identification of the helmets and it tends to classify more objects as ”Without Helmet”. Due to its complex architecture, that embeds an FPN in the SSD-MobileNetV2 model, many operations could not be transferred to the TPU, resulting in slower inference times. Specifically, the SSD-MobileNetV2-FPNlite achieves up to 11 fps, meaning that it could be ideal for cases when the objects (motorcycles) are moving no more than 30km/h.

Due to its more simple architecture (nearly all of its operations are handled by the TPU), the SSD-MobileNetV2 can achieve up to 110 frames per second with its default input size of 300x300x3. Since the camera’s framerate cannot surpass 30 fps, it would be a good idea to increase the input size to get better detection accuracy. The reduction in performance doesn’t affect the system, as long as the inference time doesn’t drop below 33 ms. After increasing the input size to 510x510x3, the model achieved 40-45 fps and was better at identifying objects in short distances (up to 5 meters). Its low inference time makes it a great choice, when the system is installed in roads where the motorcycles can be moving in higher speeds.

Generally, both models had great accuracy in scenarios where the lighting was good and the distance of the object didn't exceed 5 meters. They can work well in small roads or parking lot entrances/exits, but will struggle in bigger roads and cannot perform at all during nighttime.

The device is portable and needs to be self powered (i.e. using batteries powered by solar power panels). Therefore the power consumption of the system needs to be low. To this end, we have measured the power consumption of the TPU board running the 2 models for computer vision.

Table 1 shows the power consumption of the TPU board (using a 5V power supply) for the 2 different object detection models. As it is shown, the SSD-MobileNetV2 consumes approxi mately 3,74% less power compared to the FPNlite model.

Model	**Power (W)**
SSD-MobileNetV2	3.876
SSD-MobileNetV2-FPNlite	4.021

Table 1: Comparison of power consumption for the 2 computer vision models

6 Conclusions

In this paper, we show an embedded system for the identification of helmet usage by motorbike riders, using computer vision, in a low-cost TPU board that can achieve the strict requirements in terms of high accuracy and low-latency. The performance evaluation shows that the TPU board consumes low energy, making it ideal for self-powered systems (e.g. using a solar panel). The models can perform very well in well lit areas with objects that are not further than 5-7 meters and can achieve low enough inference times to detect in real time. We have high hopes that a system like this can increase helmet usage across the country, by providing feedback to the motorbike riders to increase awareness and decrease the number of deaths or severe injuries.

Acknowledgments: The registration fees were totally covered by the University of West Attica.

References

[1] EU. European road safety observatory facts and figures– motorcyclists and moped riders- 2023, 2023.

[2] Yange Li, Wei Han, Han Zheng, Huang Jianling, and Wang Weidong. Deep learning-based safety helmet detection in engineering management based on convolutional neural networks. Advances in Civil Engineering, 2020:1–10, 2020.

[3] Arya K M and Ajith K K. A review on deep learning based helmet detection. Proceedings of the International Conference on Systems, Energy Environment (ICSEE) 2021, 2020.

[4] Hao Wu and Jinsong Zhao. An intelligent vision-based approach for helmet identification for work safety. Safety Science, 109:123–130, 2018.

[5] Wei Zhang et al. Worker's helmet recognition and identity recognition based on computer vision. Scientific Research Publishing, 12:1–10, 2024.