**Related Works**

Driver monitoring using deep learning is common these days. Many different method haven proposed. \cite{e1} for instance use building block as describe in ``Fig. /ref{}” for their Driver monitoring system. Some blocks or modules depend on the output of others, for example, the input of the blink detection module depends on the output of the face detection module. The implementation of these building blocks makes use of CNNs, which has produced superior outcomes to more traditional image processing or computer vision-based approaches.

Timeline

Description automatically generated

A few steps have been taken to optimize the procedure because this model is intended to run on inexpensive embedded devices. The chosen platform is used for the training and inference phases. As a result, the platform can support the produced model's complexity while maintaining the appropriate level of accuracy.

Pruning and quantization are also used to optimize the selected model. Although post-training methods like weight quantization are likely to produce results that are less accurate than the full precision version, the accuracy can be increased by retraining the network using the full precision version's quantized weights as a starting point.

Utilizing more advanced deep learning techniques, such as replacing the convolution layer with a depth-wise convolution layer, is another way to reduce the model's complexity. Another method of optimizing for embedded platforms is choosing the appropriate deep learning technology. TensorFlow was chosen for this strategy because it supports remote computation and has effective graph visualization features, which are crucial, especially during training and It offers C++ and Python APIs and compiles more quickly than most other libraries. Additionally, it includes the (Tflite) inference library, which is lightweight and designed for embedded devices.

This approach also supports multithreading, which helps make the most of multicore processors. Since some of the modules rely on the results of the earlier modules, this must be done strategically. This device is offered for sale and goes by the moniker See 'n Sense for the ARM/iMX platform.

\cite{e2} proposes a further deep learning-based solution for driver monitoring systems that focuses on drowsiness detection. This technique utilized a customized YoloV5 pretrained model and Vision Transformers (ViT) to analyze the robust binary image classification model, proposing a behavioral method framework from face detection to drowsiness detection. ``Figure \ref{}” depicts the framework. The dataset was increased through image augmentation.

Diagram

Description automatically generated with medium confidence

After numerous thorough analyses, the YoloV5 obtained an accuracy rate of about 90%. The ViT model was evaluated, and the results showed that by using a custom dataset, the framework had achieved good average precision and sensitivity values.

The suggested architecture's drawback is that a substantial amount of data with labeled scene circumstances is needed in order to train the model. To reduce costs and increase computational efficiency without degrading, the design must be adapted for use in microcomputer systems. To improve the performance of the model, employ generative adversarial networks to expand the quantity of the training data.

In this paper, a method for drowsiness detection using deep learning is proposed that can improve some aspect from pervious method especially in terms of accuracy. For those reasons, the method is also implementing transfer learning to increase the accuracy as implemented in \cite{e3} and use module technique as proposed by \cite{e1}. The dataset that will be used will be expand using augmentation method as proposed by \cite{e2}. The detail about the method will be explain in proposed method section.