DRIVER DROWSINESS DETECTION USING DEEP LEARNING BRANCHED ENSEMBLE LEARNING

WONG WEI SOONG[1]* & AZIZI ABDULLAH[2]

[1,2]*Fakulti Teknologi & Sains Maklumat, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor Darul Ehsan, Malaysia*

Abstract

Driver drowsiness detection system is a car safety technology which helps prevent accidents caused by the driver getting drowsy. The basic concept to detect the drowsiness of driver is through the changes in the driver's facial organs such as eyes or mouth. This is because facial organ movements provide information about a person’s status like drowsy or fatigue. Driver drowsiness detection is very useful in ensure safety of a driver with apply this system in the smart car will able to detect and alert the drowsy driver, so it is able to reduce the risk for car accident. The objectives of this study are to improve the accuracy from the baseline model and develop a system utilizes ensemble branch CNN that is able to detect the driver drowsiness and able to alarm the driver when they are drowsy. One problem in unbranched CNN is it reduces image resolution and causes important spatial image information and local patterns to be lost after many times of subsampling. Therefore, this study proposes to develop a CNN model that is divided into various blocks. Each block has a classifier and analyzes which block has the classifier with the highest accuracy. This idea is come from branch convolutional neural networks (B-CNN) with implemented the branch classifiers in specific block with hierarchical concept. Classifiers with high accuracy will be combined using ensemble learning. In addition, this study has a second model which is Multilayer Perceptron (MLP). The MLP model aims to retrain the results from the first model in order to improve the accuracy of the first model. The performance of the proposed method is evaluated using the drowsiness dataset. Both models are used to develop a mobile application for a driver drowsiness detection system. The accuracy of the proposed model for the Drowsiness dataset is 97.93%.

Keywords: Drowsiness, Branch CNN, Ensemble Learning.

Introduction

Drowsy driver detection is a car safety technology that helps prevent accidents caused by drowsy drivers. The basic concept for detecting driver drowsiness is through the driver's facial expression. This is because facial expressions provide some information about a person's status such as sleepiness or fatigue.

In addition, car accidents are one of the main causes of death in our country. This is due to the negligence of the driver or the driver feeling sleepy while driving. Driver drowsiness is one of the factors that will affect the driver's driving performance and it is closely related to road accidents. The number of annual road traffic deaths has reached 1.35 million. Road traffic injuries are now the leading killer of people aged 5-29 (WHO, 2018). Among these events, the inability to control sleepiness has been considered as one of the critical factors that lower driving safety (Li, G, 2020). Therefore, drowsy driver detection is used to detect whether drivers are drowsy and alert them by warning them to recover their attention (Gabhane, 2018).

Although traditional facial expression detection methods through manually extracted features have been successful over the past decade, researchers have directed towards deep learning approaches due to their high automatic recognition capacity (Mellouk, Wafa, 2020). In many studies, researchers use Convolutional Neural Networks (CNN) for image classification. This is because CNN is the most popular way to analyze images (Mehendale, N., 2020). In this study, the branch classifiers are implemented in a model where the idea is taken from the Branch Convolutional Neural Network (B-CNN) from (Xinqi Zhu, 2017). The idea is to classify objects from a coarse level to a fine level.

The problem statement is the increase in the world's population has led to higher car ownership rates. As a result, traffic jams occur frequently, especially in cities like Kuala Lumpur, Malaysia. Driving for extended periods makes people feel tired and drowsy. Uncontrollable drowsiness can cause drivers to lose focus on the road, leading to road accidents.

Convolutional Neural Networks (CNN) are fundamental or simple architectures that extract features from facial images and classify them into correct labels. However, CNN have limitations in improving the model's accuracy for classifying drowsy images. One problem with unbranched CNN is that they reduce image resolution and cause the loss of important spatial image information and local patterns after multiple subsamplings. Low image resolution can significantly decrease CNN's classification performance, especially when fine details become crucial. When these details are lost, the detection rate will significantly decrease.

Therefore, There are three objectives discussed in this study:

(a) To study branch convolutional neural network (BCNN) for a drowsy driver detection system.

(b) To develop a model by implementing several branch classifiers in the identified block and develop a second model to retrain the output of the first model.

(c) To evaluate model performance using accuracy metrics.

Furthermore, the Agile model was used in this study. The Agile model was chosen as the research model development because it prioritizes fast delivery and adaptability to changes. Additionally, in the Agile process, there are opportunities to improve or adapt when issues arise. As a result, project risks are also reduced.

Literature Review

The Branch Convolutional Neural Network (BCNN) is also a variant of the traditional CNN model that produces multiple ordered predictions from coarse to fine along the convolutional layers, corresponding to the hierarchical structure of the target classes, which can be considered as a precursor form. BCNN is designed for hierarchical classification tasks where classes can be organized into hierarchical categories (Zhu, X., 2017).

Based on Figure 1, part (1) aims to make predictions for the input image at the coarse level. Furthermore, in part (2), it combines the layers or blocks from part (1) and makes predictions for the input image at the coarse level for the second time. Part (3) also combines the blocks from part (1) and (2) to make predictions for the input image at the fine level. Each of these predictions has an associated loss, which measures the difference between the predicted output and the actual output for that particular abstraction level. Finally, for part (4), the final loss function in the B-CNN model is a balanced summation of all these coarse losses. This means that each coarse loss is multiplied by a weight, and these weight losses are then added together to produce the final loss value. The weight of the loss determines the contribution of each coarse loss to the final loss value. This allows controlling the extent to which each abstraction level contributes to the final loss value.

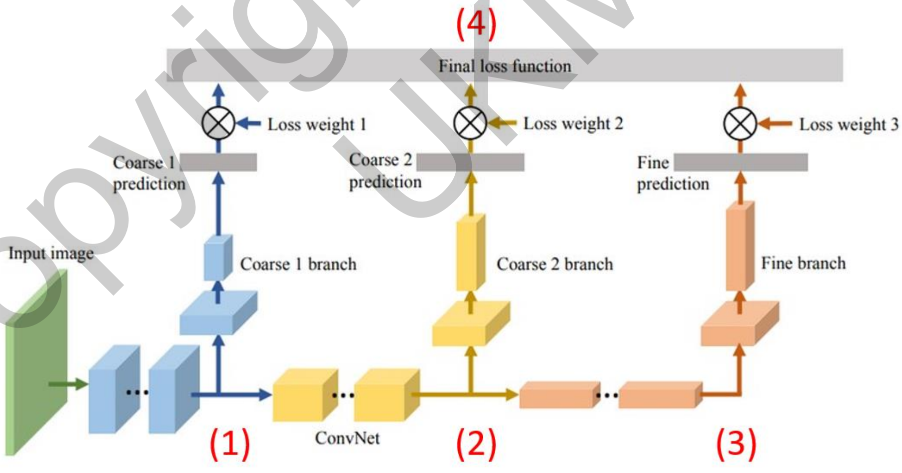

Figure 1 Branch Convolutional Neural Network (B-CNN) architecture (Source: Zhu, X., 2017).

Another research (Haq, et al., 2022), proposed a novel deep convolutional neural network (DCNN) approach based on feature fusion and ensemble learning strategies to enhance the detection and classification of abnormalities in mammography scans.

Based on Figure 2, the proposed CNN consists of four main blocks, three within section A and one within section B. Each block in section A comprises convolutional layers, MaxPooling, and dropout layers for feature extraction, while section B has a Flatten layer connected to three different blocks, namely sigmoid, Support Vector Machine (SVM), and Random Forest (RF). The combination of section A and individual blocks in section B will be considered as sub-networks. Consequently, this research has three sub-networks, with each sub-network having a different final block. The first sub-network consists of two fully connected layers and one sigmoid layer for classification. The second sub-network consists of Support Vector Machine (SVM), while the third sub-network involves Random Forest (RF) for classification. In the proposed model, features extracted from all three classifiers are combined and used for image classification.

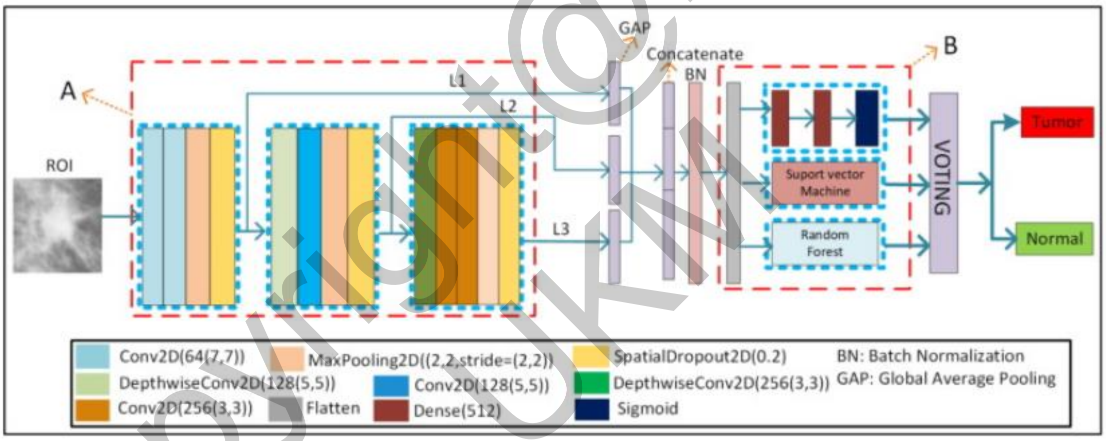

Figure 2 Architecture for CNN Ensemble (Source: Haq, et al., 2022).

Finally, the CNN Ensemble was evaluated using the MIAS dataset, and it achieved satisfactory results with high sensitivity, specificity, and accuracy of 0.995, 0.994, and 0.994, respectively, on the MIAS dataset.

Methodology

Before starting to train the model, each data set needs to be processed. After that, the datasets will be used to train the VGG-16, Branch VGG-16, Branch VGG-16 Modification, Ensemble Branch VGG-16, and the combination of Branch VGG-16 Modification with MLP.

There are some preprocessing step before training the models. This study uses three different datasets to validate the proposed model. The drowsiness dataset contains 2900 images and includes four classes, closed eyes, open eyes, no_yawn, and yawn. The CK+ dataset consists of 981 images and comprises seven classes, anger, contempt, disgust, fear, happiness, sadness, and surprise. The SC6-Net dataset contains 4020 images and comprises two classes, drowsy and non-drowsy. Subsequently, these three datasets will be divided into 70% training data, 20% test data, and 10% validation data, respectively. Table 1 displays the number of images for training, validation, and testing data for each dataset.

Table 1 The number of images for training, validation, and testing data for each dataset.

Cawangan VGG-16	**Data Latihan**	**Data Pengesahan**	**Data Ujian**
CK+	686	98	197
Drowsiness	2030	290	581
SC6-Net	2814	402	804

Furthermore, each dataset will undergo data augmentation before being used to train the model. Data augmentation is employed to expand the size of the dataset and mitigate the risk of overfitting. Implemented data augmentation techniques include shear, zoom, and horizontal flip. Figure 3 illustrates examples of augmented images.

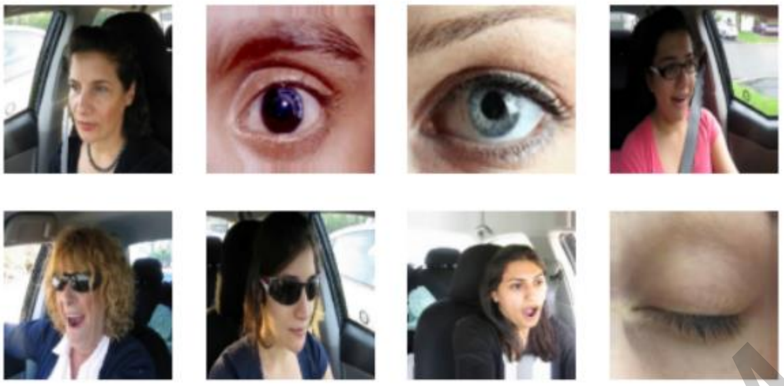

Figure 3 Examples of augmented images.

Next, the image size was also rescaled to 224 × 224 pixels to fit the study model. Normalization was also applied to the dataset by dividing each pixel value by 255. As a result, the image pixel values are scaled between 0 and 1. Normalization enables the model to effectively learn from the data. Figure 4 depicts the normalized image pixel values.

```
[[[[0.6686298  0.6313726  0.59607846]
   [0.6686399  0.6313726  0.59607846]
   [0.6686499  0.6313726  0.59607846]
   ...
   [0.5359263  0.50063217 0.44180864]
   [0.535876   0.50058186 0.4417583 ]
   [0.5358256  0.5005315  0.44170797]]

  [[0.66924506 0.6300294  0.59473526]
   [0.66923505 0.63001937 0.5947252 ]
   [0.669225   0.6300093  0.5947152 ]
   ...
   [0.51941806 0.48412395 0.42530042]
   [0.51936775 0.4840736  0.42525008]
   [0.5193174  0.4840233  0.42519978]]
```

Figure 4 Normalized image pixel values.

There are several model architectures that will be discussed in this study. The first network architecture is the VGG-16 pretrained network. Based on Figure 5, the input image used in this model is an RGB image with three color channels (C = 3). The VGG-16 model expects an RGB image of size 224×224 as input. However, the images in each dataset are not sized at 224×224. Therefore, the dataset images must be rescaled to the appropriate size of 224×224.

VGG-16 consists of 16 layers, organized into 5 sets of convolution layers, followed by a MaxPooling layer. Each convolutional layer contains 2 to 4 convolution operations with a kernel size of 3×3 and a step size of 1. The MaxPooling kernel size is 2×2 with a step size of 2. The activation function used is the "Rectified Linear Unit" (ReLU). Due to limited graphics memory on the researcher's laptop, the last three Fully Connected (FC) layers were discarded, and only the softmax layer remained for the classification process.

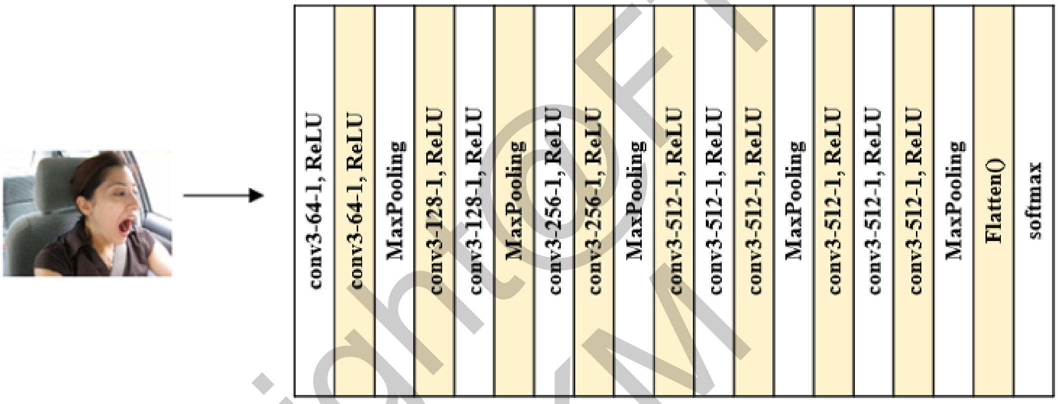

Figure 5 5 sets of convolutions in the VGG-16 network. Convolutional layer parameters are denoted as "conv-(kernel size)-(filter size)-(step)"

The second network architecture is the Branch VGG-16 network. Referring to Figure 6, for part (1), this model consists of 5 blocks, and each block has convolutional layers along with the RELU activation function, MaxPooling layer, and Flatten layer. For part (2), branch classifiers with the Softmax activation function will be formed in each block after the MaxPooling layer to perform image classification. The input data will also be classified from coarse to fine levels.

For part (3), each branch classifier will be evaluated using an accuracy matrix to make model selection. For part (4), and referring to Figure 7, model selection will be made through ensemble learning, which involves combining different branch classifiers. In each iteration, the branch classifier with the lowest accuracy in the combination will be discarded until only two branch classifiers remain. The combination of branch classifiers that provides the highest accuracy will be chosen as the study model. The classification process occurs after the flatten layer, which is followed by the softmax layer. This architectural approach is named Branch VGG-16.

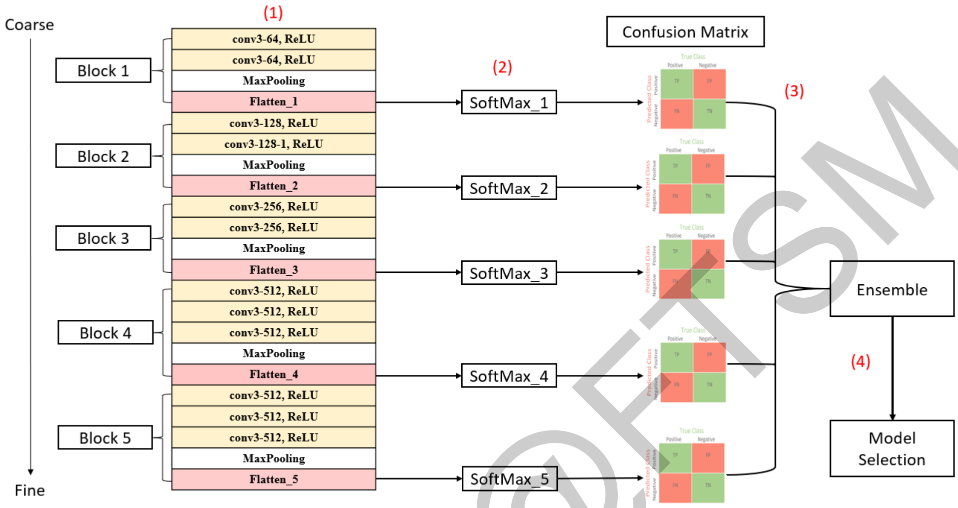

Figure 6 5 convolution blocks and 5 classifiers in the VGG-16 network. Convolutional layer parameters are denoted as "conv- (kernel size)- (sum of filters)"

Example: Ensemble using product rule.

Branch 1	Branch 2	Branch 3	Branch 4	Branch 5	Accuracy
✓	✓	✓	✓	✓	96.00%
	✓	✓	✓	✓	97.00%
		✓	✓	✓	98.00%
			✓	✓	$\boxed{99.00\%}$

Note: Assuming that in every round, the accuracy of the lowest branch will be discarded until only two branches are left.

Chosen Model

Figure 7 An example of making model selection using ensemble learning.

The third network architecture is the Branch VGG-16 Modification network. Based on Figure 8, the structure of the Branch VGG-16 model has been modified by adding a dense layer, a batch normalization layer, and a dropout layer to each block after the max pooling layer. The modifications are shown in the red region box. This new model structure is named Branch VGG-16 Modification.

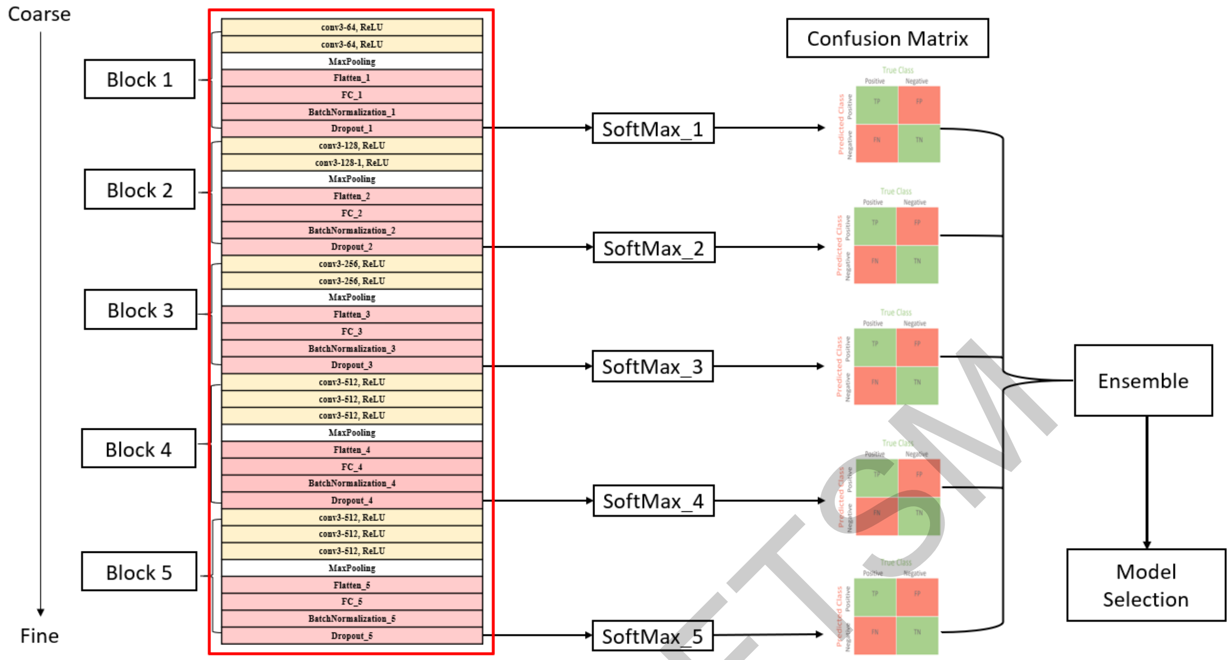

Figure 8 5 convolution blocks and 5 classifiers in VGG-16 network with dense layer, batch normalization layer and dropout layer. Convolutional layer parameters are denoted as "conv-(kernel size)- (filter size)"

The fourth network architecture is the Ensemble Branch VGG-16 network. Based on Figure 9, ensemble learning is employed to combine all outputs, which represent the probabilities of each branch classifier from the Branch VGG-16 Modification model, into a single model. The ensemble learning methods used are the product rule, the maximum rule, and the mean rule. These methods are utilized to determine the branch classifiers that should be combined in the next experiment.

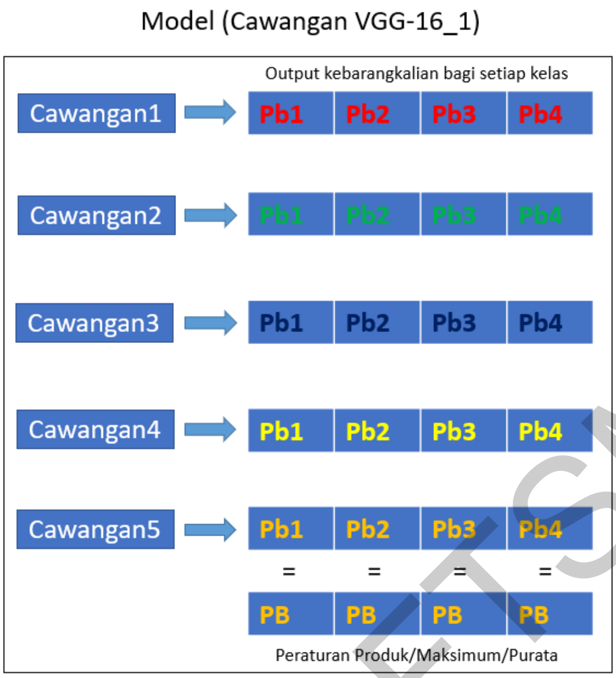

Figure 9 Ensemble learning construction.

The fifth network architecture is the combination of Branch VGG-16 Modification with MLP network. This study includes a second model called Multilayer Perceptron (MLP), which takes the output from Branch VGG-16 Modification as its input. The purpose is to further train the output from Branch VGG-16 Modification in order to enhance its accuracy. The output from Branch VGG-16 Modification represents the probability of each class in the dataset. Figure 10 illustrates the structure of the MLP used for retraining the output from Branch VGG-16 Modification. Figure 11 presents the steps involved in taking the output of Branch VGG-16 Modification as input to the MLP model for retraining.

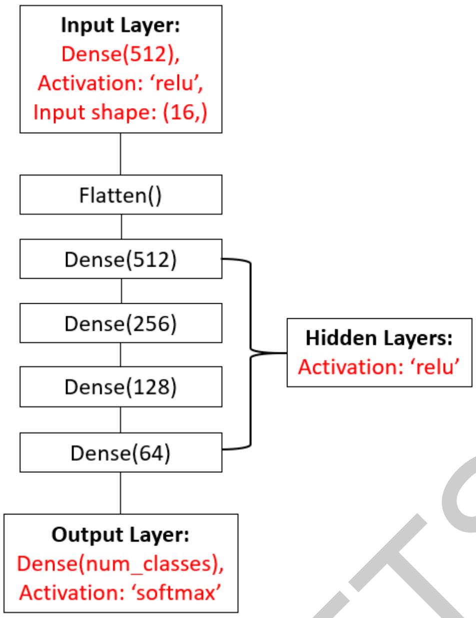

Figure 10 MLP Architecture.

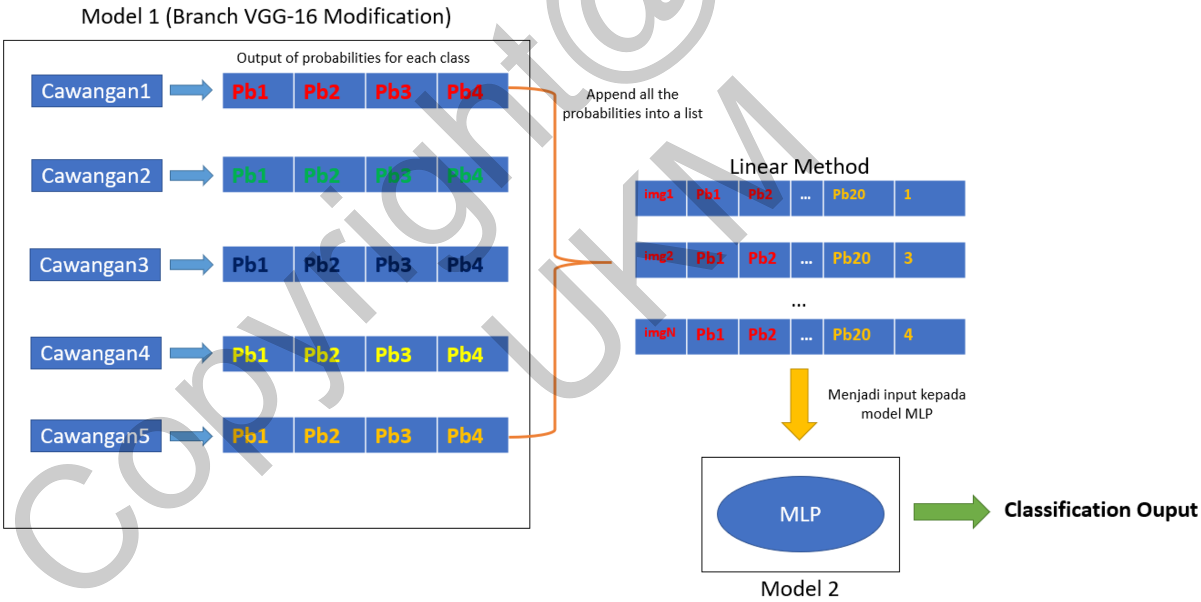

Figure 11 Steps to take the output of Branch VGG-16 Modification as input to the MLP model to retrain.

Each model was trained using Adam as the optimizer with a default learning rate of 0.001 and a batch size of 32. The training was conducted for 20 epochs for each dataset. The loss function used for each model is categorical cross-entropy. Based on training observations, all methods achieved convergence within the specified number of epochs. No additional training was performed during ensemble model training.

Results

Based on the equation below, TP (true positive) represents the number of predictions where the classifier correctly predicts the positive class as positive, TN (true negative) represents the number of predictions where the classifier correctly predicts the negative class as negative, FP (false positive) represents the number of predictions where the classifier mispredicts the negative class as positive, and FN (false negative) represents the number of predictions where the classifier mispredicts the positive class as negative. The following equation is used as an evaluation metric to assess the performance of the research models.

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

The quantitative results are shown below:

Table 2 Accuracy of all models for the drowsiness dataset. The best model scores are highlighted.

Test Model	Training (%)	Validation (%)	Testing (%)
VGG-16	98.57	96.90	95.17
Branch VGG-16	98.72	98.62	97.76
Branch VGG-16 Modification	98.23	96.90	97.59
Ensemble Branch VGG-16 (Product rule)	98.23	98.28	97.76
Branch VGG-16 Modification + MLP	**99.56**	**98.28**	**97.93**

Based on Table 2, VGG-16 is a pre-trained model and it gives lower accuracy in training, validation and testing data which is 98.57%, 96.90% and 95.17% respectively for drowsiness dataset. While Branch VGG-16 and Branch VGG-16 Modification have close accuracy, and their accuracy is also higher than VGG-16. In addition, the proposed method which is the combination of the Branch VGG-16 Modification model with MLP has given the highest accuracy in the training, validation and test data which is 99.56%, 98.28% and 97.93% respectively for the drowsiness dataset. This shows that the proposed method has successfully improved the accuracy of the model from the baseline model which is VGG-16.

Table 3 Accuracy of all models for the CK+ dataset. The best model scores are highlighted.

Test Model	**Training (%)**	**Validation (%)**	**Testing (%)**
VGG-16	99.85	97.96	98.48
Branch VGG-16	99.85	96.94	98.98
Branch VGG-16 Modification	99.27	97.96	99.49
Ensemble Branch VGG-16 (Product rule)	99.27	98.98	99.49
Branch VGG-16 Modification + MLP	**100.00**	**98.98**	**99.49**

Based on Table 3, the proposed method also gives the highest accuracy in training, validation and test data which are 100.00%, 98.98% and 99.49% respectively for the CK+ dataset. This CK+ dataset is only used to verify the performance of the study model. The results of the study model have shown high accuracy. Therefore, the researcher can conclude that this proposed model is excellent.

The models in this study include VGG-16, Ensemble VGG-16 Branch, and Branch VGG-16 Modification with MLP. In order to justify that the proposed model, Branch VGG-16 Modification with MLP is the best model among the models. All three models were trained using the SC6-Net, Drowsiness and CK+ dataset. The researcher's objective is to compare and validate the performance of the Branch VGG-16 Modification with MLP model. For this purpose, each dataset was randomized five times. This means that five different sets of data with different image arrangements for training, validation, and testing were generated for each dataset. The mean accuracy and standard deviation matrices are used to provide justification.

Based on Table 4, the Branch VGG-16 Modification with MLP model has the highest mean accuracy of 87.27% and the lowest standard deviation value of 1.13%. This indicates that this model exhibits more consistent performance compared to other models for SC6-Net dataset.

Table 4 Comparison of the accuracy of the proposed model for the SC6-Net dataset.

Proposed Model	**VGG-16 (%)**	**Ensemble Branch VGG-16 (%)**	**Branch VGG-16 Modification + MLP (%)**
Sample 1	70.40	87.44	88.06
Sample 2	83.96	85.20	86.32
Sample 3	83.83	87.31	87.44
Sample 4	80.97	85.32	85.70
Sample 5	84.83	88.81	88.81
Mean	**80.80**	**86.82**	**87.27**
Standard Deviation	**5.36**	**1.38**	**1.13**

Based on Table 5, the Branch VGG-16 Modification with MLP model has the highest mean accuracy of 97.07% and the lowest standard deviation value of 0.57%. This indicates that this model exhibits more consistent performance compared to other models for Drowsiness dataset.

Table 5 Comparison of the accuracy of the proposed model for the Drowsiness dataset.

Proposed Model	**VGG-16 (%)**	**Ensemble Branch VGG-16 (%)**	**Branch VGG-16 Modification + MLP (%)**
Sample 1	95.17	97.76	97.93
Sample 2	95.69	95.69	96.55
Sample 3	95.00	95.52	96.38

Sample 4	96.90	97.24	97.41
Sample 5	95.97	97.07	97.07
Mean	**95.97**	**96.66**	**97.07**
Standard Deviation	**0.86**	**0.89**	**0.57**

Based on Table 6, the Branch VGG-16 Modification with MLP model has the highest mean accuracy of 98.88% and the lowest standard deviation value of 0.74%. This indicates that this model exhibits more consistent performance compared to other models for CK+ dataset.

Table 6 Comparison of the accuracy of the proposed model for the CK+ dataset.

Proposed Model	VGG-16 (%)	Ensemble Branch VGG-16 (%)	Branch VGG-16 Modification + MLP (%)
Sample 1	98.48	99.49	99.49
Sample 2	93.91	98.48	98.48
Sample 3	96.95	96.95	97.97
Sample 4	94.42	100.00	100.00
Sample 5	96.45	98.48	98.48
Mean	**96.04**	**98.68**	**98.88**
Standard Deviation	**1.68**	**1.05**	**0.74**

From the above results from the three datasets, the mean accuracy and standard deviation calculations indicate that the proposed model significantly outperforms the baseline model, VGG-16. Therefore, we can justify that the Branch VGG-16 Modification with MLP has the best performance compared to the VGG-16 and Ensemble Branch VGG-16 models.

Furthermore, a confusion matrix for each dataset using the Branch VGG-16 Modification with MLP Model is generated. A sample dataset that achieved the highest accuracy in the test dataset using the proposed model, which is the Branch VGG-16 Modification with MLP, to generate the confusion matrix for all three datasets, namely Drowsiness, CK+, and SC6-Net. By applying the confusion matrix, researcher can calculate accuracy, precision, sensitivity, and F1-score using the following equations.

$$Precision = \frac{TP}{TP + FP} \quad \dots (2)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad \dots (3)$$

$$F1 - score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} \quad \dots (4)$$

Based on Table 7, the accuracy, precision, sensitivity, and F1-score are calculated and presented in Table 8. The results in Table 8 indicate that the proposed model is more accurate in classifying the Non_Drowsy class compared to the Drowsy class. This conclusion is drawn by analyzing the F1 score, where the Non_Drowsy class has an F1 score of 0.8916, which is higher than the "Drowsy" class's F1 score of 0.8843.

Table 7 Confusion matrix for the SC6-Net testing set.

	Actual Positive: Drowsy	**Actual Negative:** Non_Drowsy
Predicted Positive: Drowsy	344	58
Predicted Negatif: Non_Drowsy	32	370

Table 8 Accuracy, precision, sensitivity, and F1-score for the SC6-Net testing set.

Class	Accuracy	Precision	Sensitivity	F1-score
Drowsy	0.8881	0.8557	0.9149	0.8843
Non_Drowsy		0.9204	0.8645	0.8916

Based on Table 9, the accuracy, precision, sensitivity, and F1-score are calculated and presented in Table 10. The results in Table 10 indicate that the proposed model is more accurate in classifying the Closed and no_yawn classes compared to the other classes. This conclusion is drawn by analyzing the F1 score, where the Closed and no_yawn classes both have the highest F1 score of 0.9862 in the testing set.

Table 9 Confusion matrix for the Drowsiness testing set.

	Actual: Closed	**Actual:** no_yawn	**Actual:** open	**Actual:** yawn
Predicted: Closed	143	2	0	0
Predicted: no_yawn	2	143	0	0
Predicted: open	0	0	141	4
Predicted: yawn	0	0	4	141

Table 10 Accuracy, precision, sensitivity, and F1-score for the Drowsiness testing set.

Class	Accuracy	Precision	Sensitivity	F1-score
Closed	0.9793	0.9862	0.9862	0.9862
no_yawn		0.9862	0.9862	0.9862
Open		0.9724	0.9724	0.9724
yawn		0.9724	0.9724	0.9724

Based on Table 11, the accuracy, precision, sensitivity, and F1-score are calculated and presented in Table 12. The results in Table 12 indicate that the proposed model has classified all the classes in the testing set perfectly with a F1 score of 1.000.

Table 11 Confusion matrix for the CK+ testing set.

	Actual: anger	**Actual:** contempt	**Actual:** disgust	**Actual:** fear	**Actual:** happy	**Actual:** sadness	**Actual:** surprise
Predicted: anger	27	0	0	0	0	0	0
Predicted: contempt	0	11	0	0	0	0	0
Predicted: disgust	0	0	36	0	0	0	0
Predicted: fear	0	0	0	15	0	0	0
Predicted: happy	0	0	0	0	41	0	0
Predicted: sadness	0	0	0	0	0	17	0
Predicted: surprise	0	0	0	0	0	0	50

Table 12 Accuracy, precision, sensitivity, and F1-score for the CK+ testing set.

Class	**Accuracy**	**Precision**	**Sensitivity**	**F1-score**
anger		1.0000	1.0000	1.0000
contempt		1.0000	1.0000	1.0000
disgust		1.0000	1.0000	1.0000
fear	1.0000	1.0000	1.0000	1.0000
happy		1.0000	1.0000	1.0000
sadness		1.0000	1.0000	1.0000
surprise		1.0000	1.0000	1.0000

As a result, Branch VGG-16 Modification with MLP model demonstrates very good performance for these three datasets as analyzed through the F1-score.

The development of a mobile application for a driver drowsiness detection system is the final process for this study. This is intended to test the system in a real-time scenario. Therefore, the researcher has built a mobile based system using Android Studio software.

The Branch VGG-16 Modification model and the MLP model developed using Jupyter Notebook software need to be changed to the tensorflow lite version. This is due to ensure that the models can be used and run smoothly in the Android Studio software. A problem occurred while uploading tensorflow lite version models into Android Studio. The problem is that Android Studio does not accept models larger than 200MB. Therefore, the researcher used a weighted quantification method to reduce the size of the model. Weighted quantification involves sacrificing some model accuracy in exchange for a reduction in model size and making it more suitable for use on devices with limited resources such as mobile devices. However, the accuracy of the model will also be affected and decreased. Figure 12 shows the coding of weight quantification. Figure 13 shows the steps to use a drowsy driver detection system.

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
tflite_quant_model = converter.convert()
```

Figure 12 Coding of weight quantification.

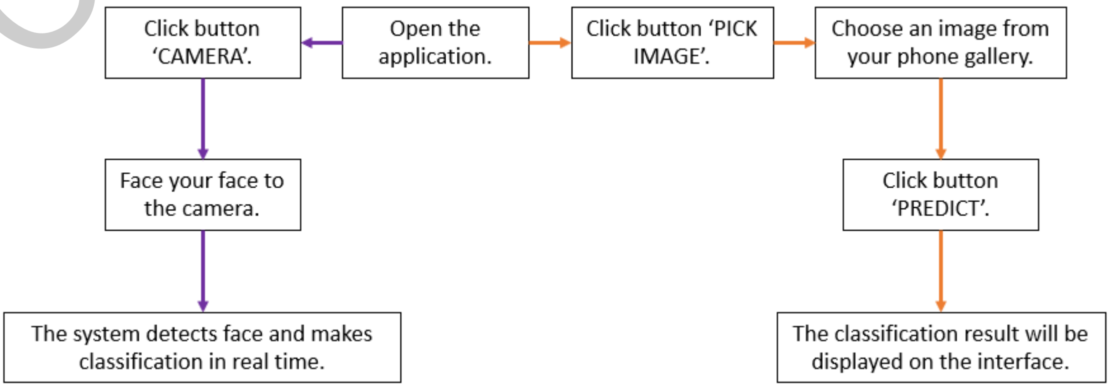

Figure 13 Steps to use a drowsy driver detection system.

Figure 14 shows the results from the application after making predictions. The interface on the left functions to classify images only and give outputs such as "Drowsy" and "Non Drowsy". It also gives the accuracy for each class in the drowsiness dataset. While the interface on the right serves to detect drowsy faces in real time. It also outputs “Drowsy” and “Non Drowsy” and makes an alert with a warning sound when it is detected as “Drowsy”.

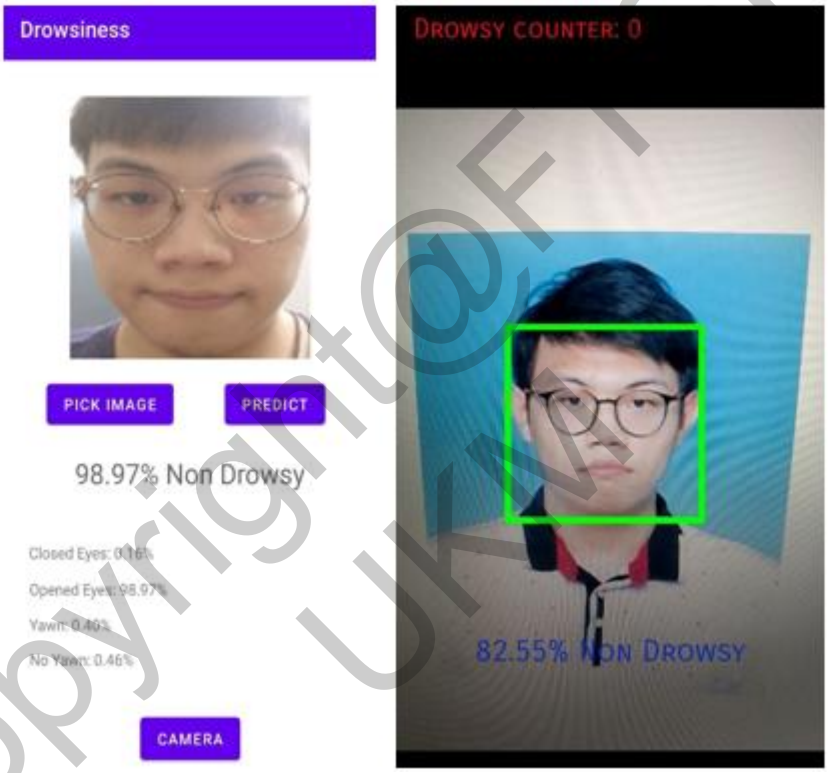

Figure 14 Results from the application after making predictions.

Discussion

Based on the results of the study, the model that has the lowest accuracy is the VGG-16 model which has given accuracy of 95.17% and 98.48% for Drowsiness and CK+ test data respectively. After studying the BCNN concept (Zhu, X., 2017), the VGG-16 Branch model was developed. This model has implemented the concept of multiple branches in the model structure in order to identify the best branch classifier in the model. The branch VGG-16 has given accuracy of 97.76% and 98.98% for drowsiness and CK+ test data respectively. The multi-branch concept successfully improves model accuracy because the Branch VGG-16 model gives higher accuracy compared to the VGG-16 model.

Next, to get the most optimal combination of branch classifiers, the Branch VGG-16 Modification model was developed. The branch classifiers output from this model will be used for ensemble learning. The method used is the product rule. This method is used to identify the most optimal combination of branch classifiers in the Branch VGG-16 Modification model.

Furthermore, the proposed model which is a combination of the Branch VGG-16 Modification model with MLP has shown the highest accuracy in all three main test models which are 97.93% and 99.49% for drowsiness and CK+ test data respectively. In addition, based on the justification using the SC6-Net dataset, the proposed model also achieved the highest mean accuracy of 87.27% compared to the other models.

One of the reasons why the proposed model gives the highest accuracy is that it contains more deeper layers. Therefore, the model can extract high-level features or more complex features from images. The next reason is that the proposed model has a second model. The second model retrains the output from the first model and refines the predictions further. This can lead to better accuracy and overall performance.

Lastly, the best performing model for the Drowsiness, CK+ and SC6-Net dataset is the proposed model, which is a combination of the Branch VGG-16 Modification and MLP models.

Conclusion

Overall, this study has tested three main models, namely VGG-16, Ensemble Branch VGG-16 and the combination of Branch VGG-16 Modification with MLP by using Drowsiness, CK+ and SC6-Net dataset. The proposed model is the combination of Branch VGG-16 Modification with MLP model gives higher accuracy score compared to the model without multiple branches and the single model.

In addition, this model has some limitations. One of the limitations is that it cannot always classify images accurately. This is due to various factors that can affect its accuracy. These factors include the distance between the camera and the driver's face and objects that may obstruct the face. The second limitation is related to the environmental conditions of the dataset. The model and system may be affected by environmental conditions such as low lighting, shadows, or light scattering. This can impact the system's ability to accurately detect signs of drowsiness.

The suggestion for improving the research model in the future is to train the model by creating a custom dataset. This self-built dataset should possess characteristics that are more realistic and suitable for the specific environment. Consequently, the model will be exposed to a broader range of real-world data, allowing it to learn from more authentic examples. Furthermore, another potential way for improvement is to experiment with other backbone models, such as ResNet. In deep learning, the backbone model serves as the core architecture that provides the foundation for the neural network. While VGG-16 has been utilized in the current study, trying different backbone model like ResNet may yield better results.

Finally, the proposed framework meets the objectives, and the objectives are achieved. The proposed model has implemented the branch idea from BCNN, and the results illustrate that the branch architecture has the ability to increase the model accuracy. Additionally, the proposed model has been successfully implemented into the mobile application.

Appreciation

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Assoc. Prof. Ts. Dr. Azizi Abdullah, for his invaluable advice, guidance, patience, and unwavering support throughout the course of this study.

Furthermore, I extend my thanks to the lecturers of FTSM for sharing their knowledge and expertise generously and for their endless assistance and guidance during my studies at UKM.

Additionally, I would like to extend my utmost appreciation to my parents and my family for their continuous moral support throughout the duration of this project.

Lastly, I would like to thank my friends who have always been there to lend a helping hand and provide guidance during the course of this study.

REFERENCE

Aytekin, A., & Mençik, V. (2022). Detection of Driver Dynamics with VGG16 Model. *Sciendo*.

Gabhane, & Jyotsna. (2018). Drowsiness Detection and Alert System. *International Journal for Research in Applied Science and Engineering Technology*.

Haq, I. U., Ali, H., Wang, H. Y., Lei, C., & Ali, H. (2022). Feature fusion and Ensemble learning-based CNN model for mammographic image classification. *Journal of King Saud University - Computer and Information Sciences*, 3310-3318.

Li, G., Lai, W., Sui, X., Li, X., Qu, X., Zhang, T., & Li, Y. (2020). Influence of traffic congestion on driver behavior in post-congestion driving. *Accid. Anal. Prev.*

Mehendale, N. (2020). Facial emotion recognition using convolutional neural networks (FERC). *SN Appl. Sci.*

Mellouk, W., & Wahida, H. (2020). Facial emotion recognition using deep learning: review and insights. *Procedia Computer Science*.

Narkhede, S. (2018, May 9). *Understanding Confusion Matrix*. Didapatkan dari Towards Data Science: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Ognjanovski, G. (2019, Jan 15). *Everything you need to know about Neural Networks and Backpropagation — Machine Learning Easy and Fun*. Didapatkan dari Towards Data

Science: https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a

Shah, S. (2021, Feb 25). *Cost Function is No Rocket Science!* Didapatkan dari Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/02/cost-function-is-no-rocket-science/

Tammina, S. (2019). Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. *International Journal of Scientific and Research Publications (IJSRP)*.

WHO. (2018). *Global status report on road safety 2018* . Geneva: World Health Organization.

Zhu, X., & Bain, M. (2017). *B-CNN: Branch Convolutional Neural Network for Hierarchical Classification*. Didapatkan dari https://arxiv.org/pdf/1709.09890.pdf

Wong Wei Soong (A180106)
Assoc. Prof. Ts. Dr. Azizi Abdullah
Fakulti Teknologi & Sains Maklumat,
Universiti Kebangsaan Malaysia