VIDYAVARDHAKA COLLEGE OF ENGINEERING

(Autonomous, affiliated to VTU)

DEPARTMENT OF

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Mini Project Report

On

"SLEEPINESS DETECTION"

Submitted in partial fulfillment of the requirement for the completion of VI semester of

BACHELOR OF ENGINEERING

MOHAN K	**4VV21AI028**
HARISH GOWDA	**4VV21AI016**
KUNAL R	**4VV21AI025**
AP BHARATESH	**4VV21AI001**

Under the guidance of:

Dr. Sonali N M
Assistant Professor
Dept. of CSE(AI&ML)

CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

VIDYAVARDHAKA COLLEGE OF ENGINEERING

(Autonomous, affiliated to VTU)

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

CERTIFICATE

Certified that mini project work entitled "sleepiness detection", is a bona fide work carried out by

MOHAN K(4VV21AI028) , HARISH GOWDA H R(4VV21AI016) , KUNAL R()4VV21AI025 , A P BHARTESH(4VV21AI001) in partial fulfillment of the requirement for the completion of VI semester in CSE(AI&ML) of Vidyavardhaka College of Engineering during the year 2023-24. It is certified that all corrections/suggestions indicated for Internal Assessment has been incorporated in the report. The report has been approved as it satisfies the academic requirements with respect to Mini Project work.

Signature of the Guide	**Signature of the Incharge HOD**
Dr. Sonali N M	Dr. Chetan B K
Assistant Professor	Associate Professor

Vidyavardhaka Sangha®, Mysore
VIDYAVARDHAKA COLLEGE OF ENGINEERING
Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi
(Approved by AICTE, New Delhi & Government of Karnataka)
Accredited by NBA (CV, CS, EE, EC, IS & ME) | NAAC with 'A' Grade
P.B. No. 206, Gokulam III Stage, Mysuru-570 002, Karnataka, India
Phone: +91 821 4276201 /202 /225, Fax: +91 824 2510677
Web: http://www.vvce.ac.in @vvceofficial

VVCE

Table of Contents

Objectives of the Project

Objective:

The main goal of this project is to develop a reliable, real-time system to detect sleepiness by monitoring the state of a person's eyes using computer vision and deep learning. Specifically, it aims to:

- **Identify Eye States**: Accurately classify whether the eyes are open or closed using a deep learning model.
- **Monitor Blink Frequency**: Count the number of blinks within a specified time frame.
- **Determine Sleepiness**: Assess sleepiness based on the frequency of eye closures, alerting if the blink rate exceeds a certain threshold indicative of drowsiness.

The ultimate aim is to enhance safety by providing timely alerts to prevent accidents, particularly in scenarios like driving or operating machinery, where drowsiness can have severe consequences.

Introduction

Introduction:

Drowsiness detection systems have become increasingly important in preventing accidents and ensuring safety in various environments. The project's focus is to detect sleepiness through eye state monitoring. Sleepiness can significantly impair reaction times, decision-making, and overall alertness, posing a risk in activities requiring constant vigilance, such as driving.

The advancement in computer vision and deep learning provides the tools to create an efficient system capable of real-time analysis. By leveraging a convolutional neural network (CNN), this project aims to classify eye states and monitor blink frequency to determine sleepiness, providing an alert mechanism when necessary. This technology can be integrated into vehicles, workplaces, or even personal devices to enhance safety and productivity.

Vidyavardhaka Sangha®, Mysore
VIDYAVARDHAKA COLLEGE OF ENGINEERING
Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi
(Approved by AICTE, New Delhi & Government of Karnataka)
Accredited by NBA (CV, CS, EE, EC, IS & ME) | NAAC with 'A' Grade
P.B. No. 206, Gokulam III Stage, Mysuru-570 002, Karnataka, India
Phone: +91 821 4276201 /202 /225, Fax: +91 824 2510677
Web: http://www.vvce.ac.in
VVCE
@vvceofficial

Methodology of the Project

Methodology:

1. **Data Collection and Preprocessing**:
   - **Dataset**: Utilize an existing dataset of images labeled as 'eyes open' or 'eyes closed'.
   - **Image Augmentation**: Apply transformations like rescaling, rotation, and flipping to enhance the dataset and improve model robustness.
   - **Splitting Data**: Divide the dataset into training and validation subsets to evaluate model performance.
2. **Model Building**: o **CNN Architecture**: Construct a sequential model with convolutional layers to extract features, pooling layers for down-sampling, and dense layers for classification.
   - **Activation Functions**: Use 'ReLU' activation for hidden layers and 'sigmoid' activation for the output layer to predict binary outcomes (open or closed).
3. **Training**:
   - **Compilation**: Compile the model with the Adam optimizer and binary cross-entropy loss function.
   - **Training**: Fit the model on the training data, validating with the validation set over multiple epochs to optimize performance.
4. **Real-Time Detection**: ○ **Webcam Integration**: Capture live video feed from a webcam. ○ **Frame Processing**: Preprocess each frame to match the input size expected by the model (64x64 pixels). ○ **Prediction**: Use the trained model to predict eye states. ○ **Blink Counting and Sleepiness Detection**: Count the number of blinks and assess sleepiness based on blink frequency over a fixed period.

Code Snippets

Code Snippets:

1. **Data Loading and Preprocessing**:

```
import tensorflow as tf from tensorflow.keras.preprocessing.image
import ImageDataGenerator


datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)


train_generator = datagen.flow_from_directory(
'./input/eyes-open-or-closed/dataset/train',
target_size=(64, 64),     batch_size=32,
class_mode='binary',     subset='training'
)


validation_generator = datagen.flow_from_directory(
'./input/eyes-open-or-closed/dataset/test',
target_size=(64, 64),     batch_size=32,
class_mode='binary',     subset='validation'
)
```

2. **Model Building**: model =

```
tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
tf.keras.layers.MaxPooling2D((2, 2)),
```

```
   tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
tf.keras.layers.MaxPooling2D((2, 2)),
   tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
tf.keras.layers.MaxPooling2D((2, 2)),
   tf.keras.layers.Flatten(),
tf.keras.layers.Dense(512, activation='relu'),
tf.keras.layers.Dense(1, activation='sigmoid') ])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

3. **Training**:
```
model.fit(train_generator, validation_data=validation_generator, epochs=10)
```

4. **Real-Time Detection**: import cv2 import numpy as np
```
import time

cap = cv2.VideoCapture(0)

blink_count = 0 start_time
= time.time()
sleepiness = 'Not Sleepy'

while True:
   ret, frame = cap.read()
if not ret:        break

   resized_frame = cv2.resize(frame, (64, 64))
normalized_frame = resized_frame / 255.0
   reshaped_frame = np.reshape(normalized_frame, (1, 64, 64, 3))

   prediction        =        model.predict(reshaped_frame)
eye_state = 'Open' if prediction > 0.5 else 'Closed'   if
eye_state == 'Closed':        blink_count += 1

   elapsed_time = time.time() - start_time
   remaining_time = max(0, 10 - int(elapsed_time))

   if elapsed_time > 10:        sleepiness = 'Sleepy' if
blink_count > 10 else 'Not Sleepy'        blink_count = 0
      start_time = time.time()
```

```
    cv2.putText(frame, f'Eye State: {eye_state}', (50, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2, cv2.LINE_AA)
cv2.putText(frame, f'Sleepiness: {sleepiness}', (50, 100),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0) if sleepiness == 'Not Sleepy' else (0,
0,
255), 2, cv2.LINE_AA)     cv2.putText(frame, f'Time Remaining:
{remaining_time}', (50, 150),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE_AA)
cv2.imshow('Eye State Detection', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
break

cap.release()
cv2.destroyAllWindows()
```

Result

Results:

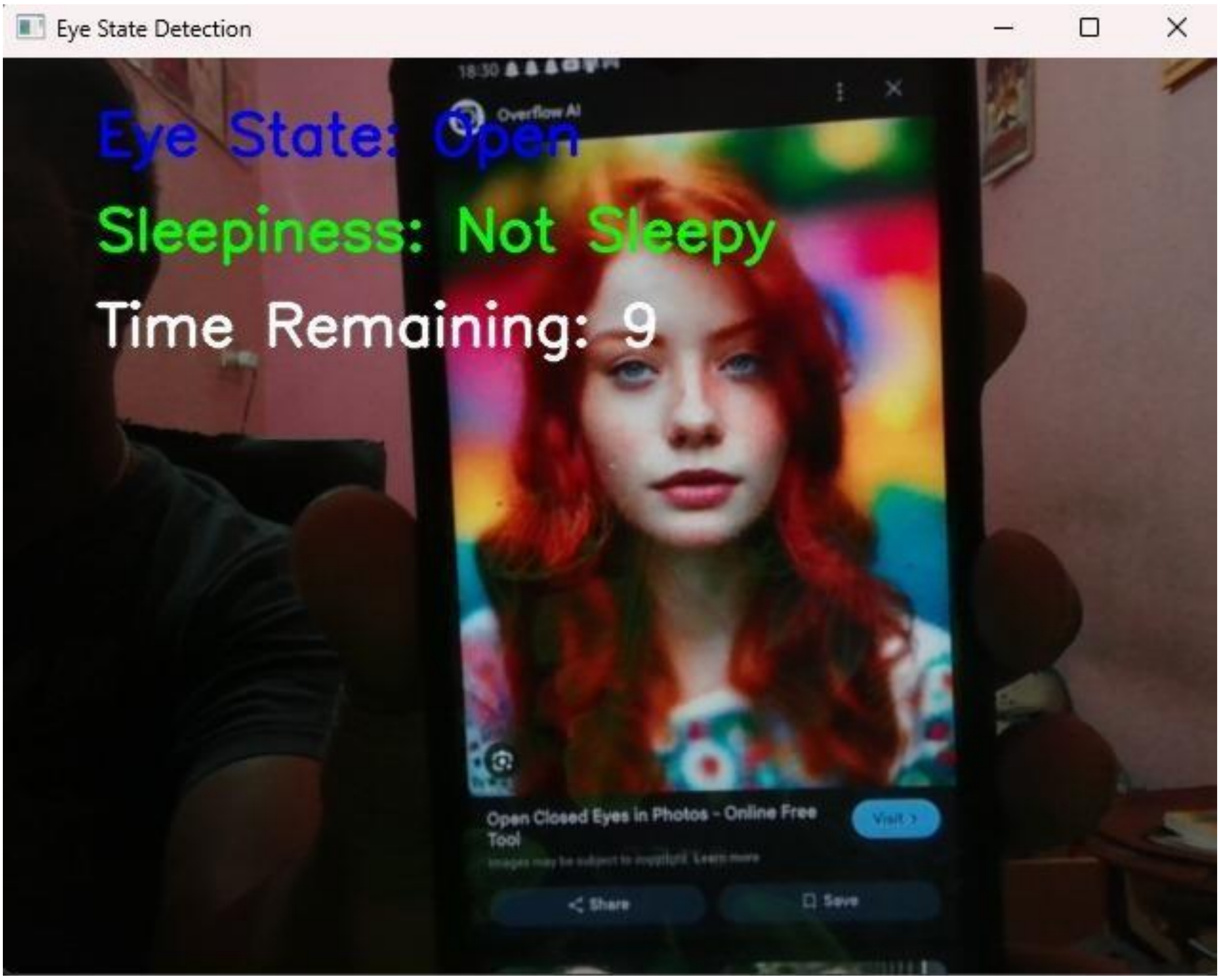
The model's performance can be evaluated based on its accuracy and loss on the validation dataset. During realtime detection, the system successfully:

- **Classified Eye States**: Accurately predicted whether the eyes are open or closed.
- **Counted Blinks**: Kept track of the number of blinks within each 10-second interval.
- **Assessed Sleepiness**: Determined sleepiness based on blink frequency, displaying a 'Sleepy' or 'Not Sleepy' status.

The system's output is displayed in real-time on the video feed, showing the current eye state, sleepiness status, and the remaining time in the current interval.

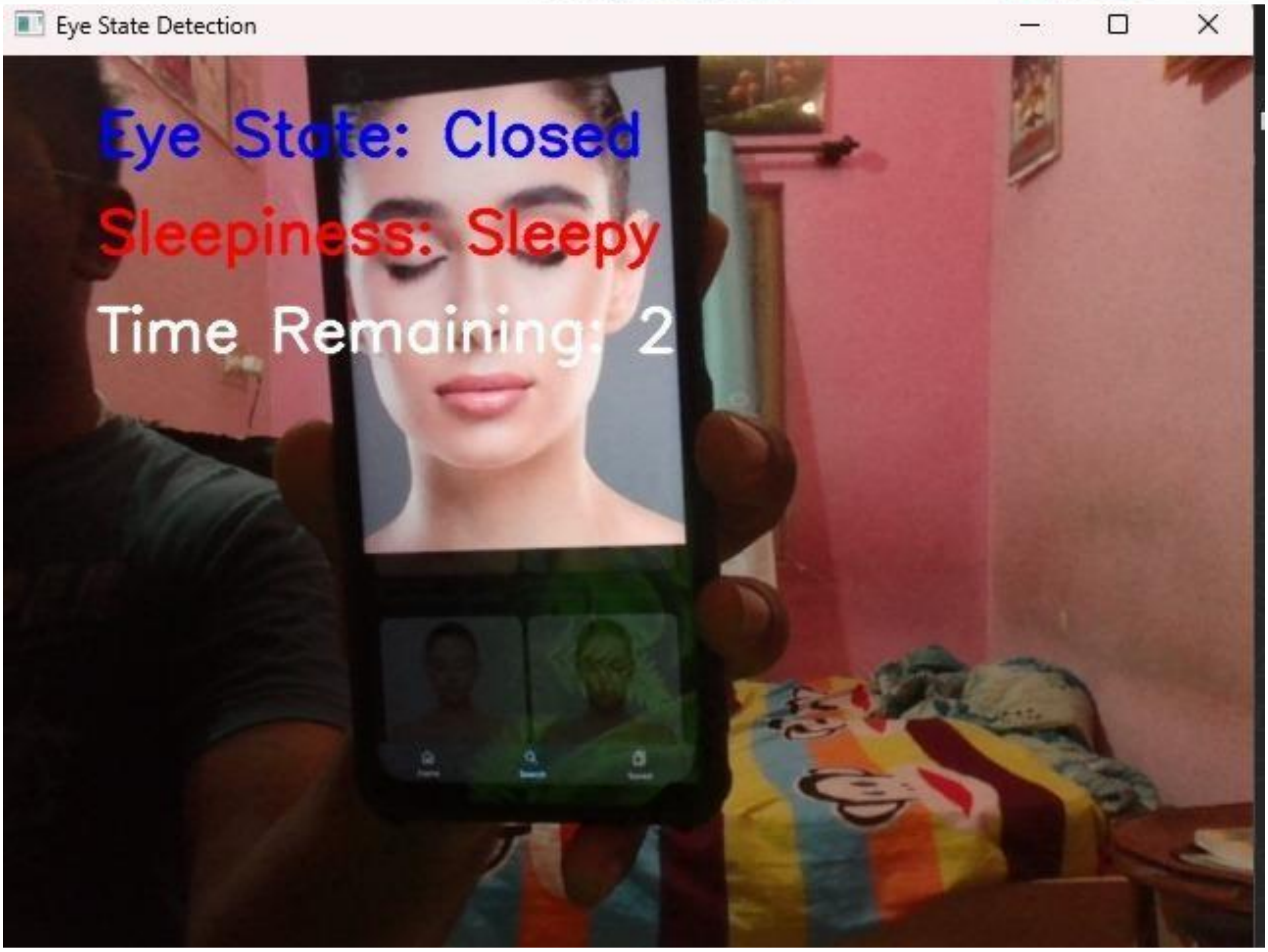
Output 1: EYE STATE: TRUE/OPEN

SLEEPY: FALSE



Output 1: EYE STATE: FASLE/CLOSE

SLEEPY: TRUE


Eye State Detection
Eye State: Closed
Sleepiness: Sleepy
Time Remaining: 2

Conclusion

Conclusion:

The sleepiness detection system effectively demonstrates the application of deep learning in real-time safety monitoring. By accurately classifying eye states and assessing sleepiness based on blink frequency, the system can alert users to potential drowsiness, helping to prevent accidents. Future improvements could include enhancing model accuracy with a larger dataset, optimizing performance for different lighting conditions, and integrating the system into vehicles or other environments where drowsiness poses a risk. The success of this project highlights the potential of AI-driven solutions in promoting safety and well-being.