DETECTION OF DROWSINESS OF DRIVER'S FACE THROUGH IOT

Dr. Nirupama MP, Nipun Rajput[1], Ketan Thakur[2], Godavarthi Sai Nikhil[3], Utkarsh Singh[4]

Department of Computer Science and Engineering School of Engineering and Technology, BML Munjal University, Kapriwas, Haryana, India

ABSTRACT: A real-time driver sleepiness detection system for driving safety is presented in this research. Detection of drowsiness through IoT, aims to support drivers and strengthen their cars not only to improve the driving experience, but also to be safe. Even if you look at the whole passenger safety, Driver drowsiness in recent years. It is one of the main causes of traffic accidents. It leads to serious financial loss. This paper, targets vision-based real-time driver warning system mainly to monitor driver drowsiness and distraction. The face detection algorithm starts with image frames that have been collected and then processes them. We have found a simple solution to detect Drowsiness with the help of this project. We can reduce the number of accidents and spread awareness about driving an adequate amount of rest. The driver's face is located from a color video shot in an automobile using computer vision techniques. Following that, face detection is performed to find the driver's eye areas, which are then used as templates for eye tracking in subsequent frames. Finally, the images from the tracked eye are used to identify drowsiness and create warning alarms. The purpose of image processing is to recognize the driver's face and then extract the image of the driver's eyes for sleepiness detection. On a Raspberry Pi 4 Model, the proposed system was tested. The test results look good and promising. On some test videos, the system could achieve more than 15 frames per second for face and eye tracking, with an average accuracy rate of 99.0 percent for eye placement and tracking. As a result, the proposed strategy can be inferred to be a low-cost and effective solution method for detecting driver drowsiness in real time.

Key Words: Eye Detection, Drowsiness, Facial landmark Recognition, Raspberry pi 4, Python.

INTRODUCTION:

When a person is fatigued due to prolonged driving will tend to sleep during driving which can cause an accident, which in result will harm himself or possibly people around him. Drowsiness is notoriously difficult to detect and track. In India, drunk driving is a serious problem. The risks of drowsy driving, as well as the often-fatal results, are frightening. Drowsy driving is the most dangerous sort of driving.

Driving while drowsy is a dangerous combination. Moreover, weariness When a driver loses control of his or her vehicle, this normally happens. If he hasn't gotten enough sleep, it could be due to a variety of things, including drugs, alcohol, or shift work.

However, the number of people drinking, and driving is on the rise. Every year, approximately 200 individuals are killed as a result of drinking and driving. It is not only the driver who suffers but also his passengers and other passengers. In 2016, it was estimated that 100 footsloggers and 390 car passengers were critically hurt or killed by drunk drivers.

Drunk drivers killed or seriously injured 40 children in that year. When our bodies go to sleep, no one knows exactly what happens.

It's obvious that falling asleep while driving is dangerous, but it also impacts the driver's ability to drive safely. Drowsy driving caused 72,000 collisions, 44,000 injuries, and 800 deaths in 2013, according to the National Highway Traffic Safety Administration.

OBJECTIVES:

Through this project, we are trying to reduce accidents happening due to drowsiness. In India, the ministry of road transportation revealed that 40% of accidents happen because of drowsiness. The project scope is good who are traveling at the night and can use this project, many car companies are trying to implement it during the manufacturing of cars to reduce accidents.

LITRATURE\REVIEW:

We found different ways to detect drowsiness to prevent road accidents. We have researched and concluded that 91 out of 100% of accidents are happening due to driver's error, 5% from Surrounding factors, and 4% due to vehicle technical problems. With the help of Artificial Intelligence, we are going to detect the driver's drowsiness [8].

These are some ways to measure drowsiness Vehicle-based, Physiological and Behavioral measures.

a) Vehicle-based is mainly used in abroad countries it detects the lane of the vehicle and also steering wheel position, to determine driver's weakness.
b) Physiological measure observes the physical condition of the driver's body to detect fatigue. It will observe the pulse rate and brain activities to prevent accidents. With the help of beta and gamma waves, they detect drowsiness.
c) Behavioral measures are one of the best and easiest ways to detect drowsiness, it continuously detects eye closure and eye blinking. We can do this method with the help of this project by simply placing a camera module in front of the driver. This device continuously captures the driver's eye position and further detects the drowsiness [4].

We are considering Behavioral measures in this project.

The main concept of this project is to utilize all the functioning sensors and devices we have chosen.

A. LCD(LM016L)

LCD stands for liquid crystal display, and it is a device that shows messages. We have connected the LCD to the Raspberry Pi and with the help of it the LCD can display a 'Welcome' message and also it will display a message when the driver is in Fatigue like 'Sleep Alert', 'Drowsy Alert', 'Driver Active' and 'Accident Happened'. It is connected to all

Figure 1. LCD

the components so that certain signals may be received, and messages can be displayed on the LCD Screen.
It also supplies 5V power to other modules.

B. GSM Module:

'Global System for Mobile Communications', with the help of this device the owner can get an audio warning and a message ('Accident Happened') in case of an accident, happened and it will active the buzzer in the device and also it sends the GPS location to the Owner and nearest police station.

Figure 2. GSM Module

C. Vibration Sensor:

It is connected to the Raspberry Pi and the function of this sensor is to detect any jerk delivered to the car (in other words, it measures the frequency of vibration of the particular vehicle), emulating the occurrence of an accident in real-time. The LCD's 5V port provides power to this device. The output generates and sends signals.

Figure 3. Vibration Sensor

D. Raspberry pi 4:

It's a single-board computer that's built to operate Internet of Things (IoT) applications. It will also have an SD card for installing the operating system. The Raspberry Pi is quicker and has more RAM than the Arduino board. We used it in our project because it allows us to Program and connects LCD, vibration sensors, and other devices using the Python computer language. We were using the Raspberry Pi 4 to develop a prototype for drowsy detection for vehicle drivers. The video of the driver is recorded and fed into a cascade classifier tool in our prototype. After the pictures have been detected, they are transmitted to a neural network that has been trained to determine the driver's level of fatigue. Our Raspberry Pi 4 prototype can be installed in a wide range of automobiles and can provide non-invasive warnings to drivers while maintaining a high level of accuracy. Due to its low cost and energy effectiveness, the Raspberry Pi 4 has the upper hand.

Figure 4. Raspberry Pi 4

E. Buzzer:

The use of a buzzer is to give a subsequent warning to the driver in a certain interval of time after the detection of drowsiness which is interrelated to the posture and eyes of the driver on the road.

Figure 5. Buzzer

F. Python:

We have chosen Python as the programming language because it is one of the easiest languages to understand the logical codes and is helpful for large-scale projects. We have used the 'IDLE Python 3.10' app to work on the code.

Figure 6. Python IDLE 3.10

DESIGN REQUIREMENT:

Hardware Components are

- Raspberry pi 4
- Pi Camera sensor
- 32 Gb Sd Card
- GPS
- LCD (Display Board)
- GSM Module
- Buzzer
- Vibration Sensor

Software Components are

- proteus 8 professional
- VSPE
- Python – python 3.10
- Libraries like NumPy, Dlib, imutils, OpenCV, SciPy. Etc.

Figure 7. Proteus 8 Professional

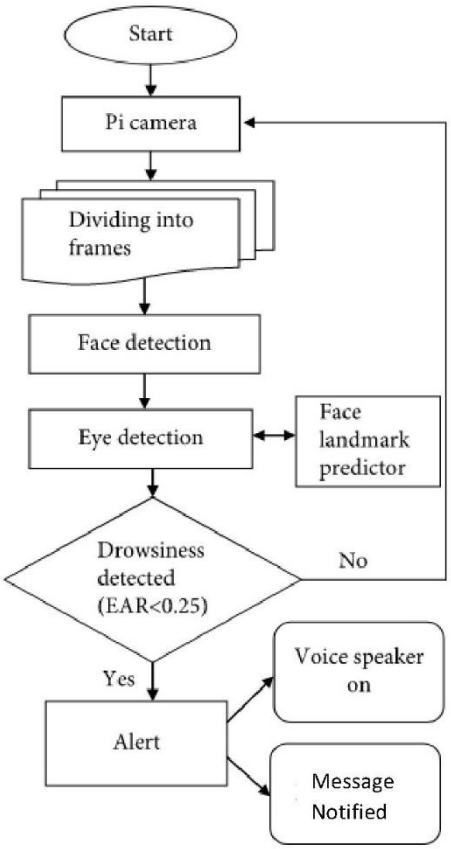
METHODOLOGY:

To avoid road accidents, we discovered many techniques to detect drowsiness. We had observed that 90% of all accidents are caused by driver error, 5% by external factors, and 4% by car technical faults. We will detect the driver's drowsiness with the help of this device [4].

Eye closure and blinking are continuously monitored, making behavioral measures among the fastest and easiest ways to detect drowsiness.

With the help of this project, we can implement this strategy by simply placing a camera module in front of the driver. This device monitors the driver's eye position in real-time and detects drowsiness.

The proposed methodology includes detecting a driver's eyes, analyzing the driver's inattentiveness using a frontal face angle-based facial attention angle and a drowsiness threshold based on the eyes aspect ratio, and alerting the driver with such a real-time alarm system.



Flow Chart of the process

As you can see from the flow chart above, we have detailed the workflow of our project.

Face, eye, and sleepiness detection are the three steps of the proposed method. The purpose of image processing is to recognize the driver's face and then extract the image of the driver's eyes for sleepiness detection. The face detection algorithm starts with image frames that have been collected and then processes them.

3.1 Eye Detection:

The image or video will be captured from the Pi camera which is placed on the dashboard of the car and the video will be converted into multiple frames [3] with the help of OpenCV.

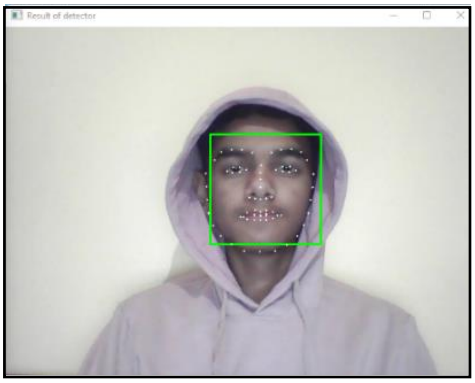


Figure 8. Eye Detection

Then, utilizing open cv and open cv Classifier (a rectangle is formed around the face) as seen in the above figure, we use feature face extraction to calculate the position of the eye, then compare it to an open CV to recognize the eyes.

We used facial landmarks to localize the areas of the face like Eyes, Eyebrows, Nose, Mouth, etc [3].

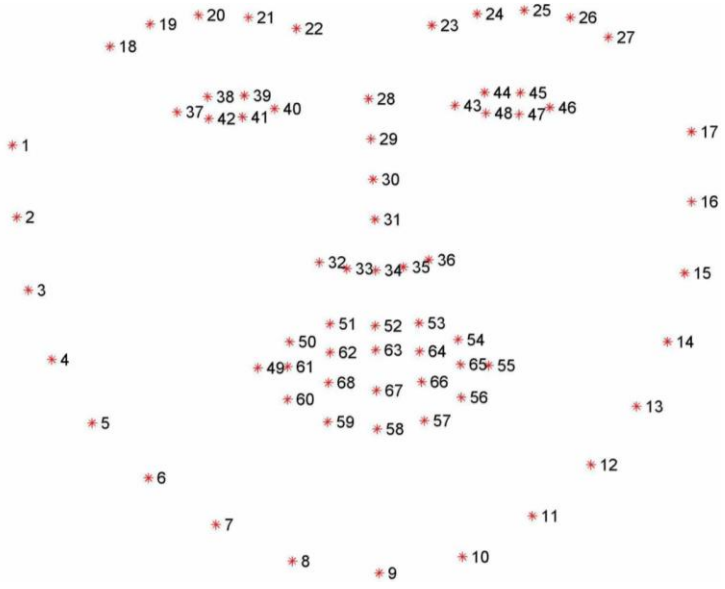


Figure 9. 68 Landmark Detection

First, the device will localize the image before detecting then we need to predict facial structures on the face then with the help of a 68 landmark detector which is included in the Dlib library we can implement the data [5].

The x-y coordinates of regions that surround the region of the face.

Then the facial landmark detector which is pre-installed in the Dlib library is used to judge the 68 coordinates of the face. With the help of it, we can detect eyes.

Left Eye points

Right Eye points

Euclidean is the distance between two locations. As you can see from the above two images, we have mentioned the points of both eyes. If (ratio>0.25) in the left eye from 43 to 48 and in the right eye from 37 to 42 the eye is opened and if (ratio>0.21 and ratio=0.25), the eye is closed.

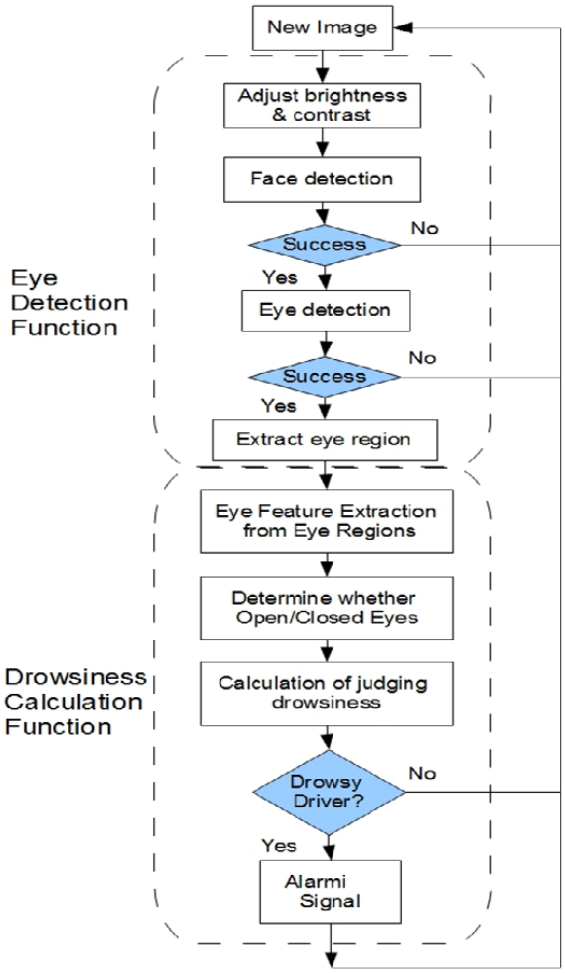
Based on Cartesian coordinates, the Pythagorean theorem is used to calculate the Euclidean distance. In various statistical and optimization applications, the square of the Euclidean distance is applied instead of the distance itself.

3.2 Recognition of eye's state:

This area is judged frame to frame and then the final decision is taken whether the eye is covered by eyelids or not. There is also a different method which is matching with the images of an open and closed eye but there are many drawbacks to these features such as illumination, motion dynamics, etc. So we went with the simple method to use an algorithm that detects the eye using facial landmark detector [1].

3.3 Eye state determination:

The decision for eye state is taken by the Euclidian distance, if the distance is close to zero then the eye is detected as closed, else it is detected as open. Frame by frame technique is used in this also and then the final decision is taken that whether the eye is open or covered by eyelids [5]. There is also a different approach that used images of open and closed eyes to detect.



Flow Chart of the detection of face

Looking at the flow chart above, we can see why the process of recognizing eye fatigue leads to drowsiness alarms.

3.4 Drowsiness detection:

It is the last step of the algorithm to determine the condition of the person. When the system detects either the driver getting tired or the driver is not paying attention while driving, then a warning will be triggered in this final stage. A buzzer within the Raspberry Pi is connected to the GPIO of the Raspberry Pi to generate a moderately loud beeping sound. This sound is meant to inform the driver if he or she is fatigued or inattentive. The duration of the average blink is about 100-400 milliseconds. The buzzer is triggered if a person is drowsy during a certain interval and the eye is closed for even more than 15 seconds.

EXPERIMENTAL RESULTS:

With the help of this device, it will capture the face of the drivers by the live video using the Dlib frame to detect both eyes. Now we will move further with the help of OpenCV to calculate the 'Euclidean function' (Vertical and horizontal distance of eye to calculate the ratio of eye aspect).

The python software and Dlib library help us to detect the eye moments whereas the face is detected by a green box, now by using 68 Face Landmark the system will detect the eye moments.

- These are some of the Screenshots of our Project:

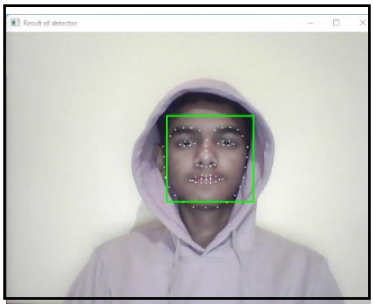


Figure (10.A). Open Eyes

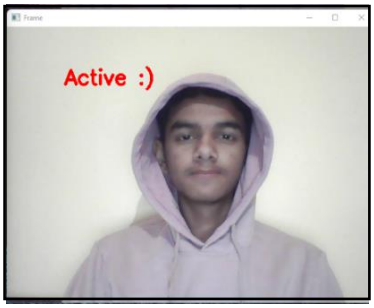


Figure (10. B). Result For (10.A)

Figure (10.A) shows that the driver's eyes are open more than 85% of the time, and the Euclidean distance ratio is more than 0.25, confirming that the driver is in the "Active State" which is stated in the Figure (10.B).

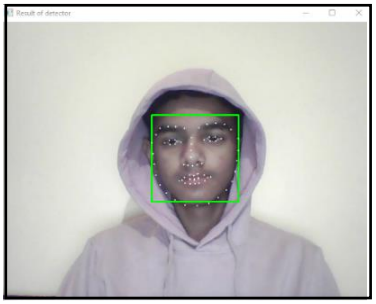


Figure (11. A). Drowsy Eyes

Figure (11. B). Result For (11. A)

Now that you can see in Figure (11.A) that the driver's eye is nearly 60% open and the Euclidean distance ratio is equal to 0.25, we can assume that the driver is in the "Drowsy State" described in Figure (11.B).

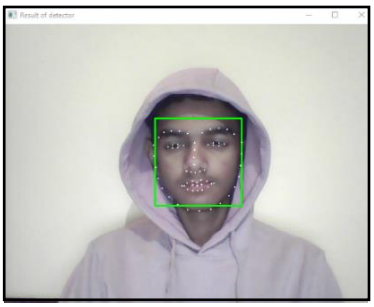


Figure (12. A). Close Eyes

Figure (12. B). Result For (12.A)

Figure (12. A) shows that the driver's eye is closed around 80% of the time, and the Euclidean distance ratio is less than 0.21 and equal to 0.25, indicating that the driver is in "Sleeping State", which is shown in Figure (12. B).

As you can see in Figure 13, this device can distinguish the difference between the eye blink and drowsiness with the help of movement of the eye. In case of an accident happened, the device will automatically send the 'Accident Happened' message and an audio warning to the mobile number and nearest police station with the help of a GPS module.

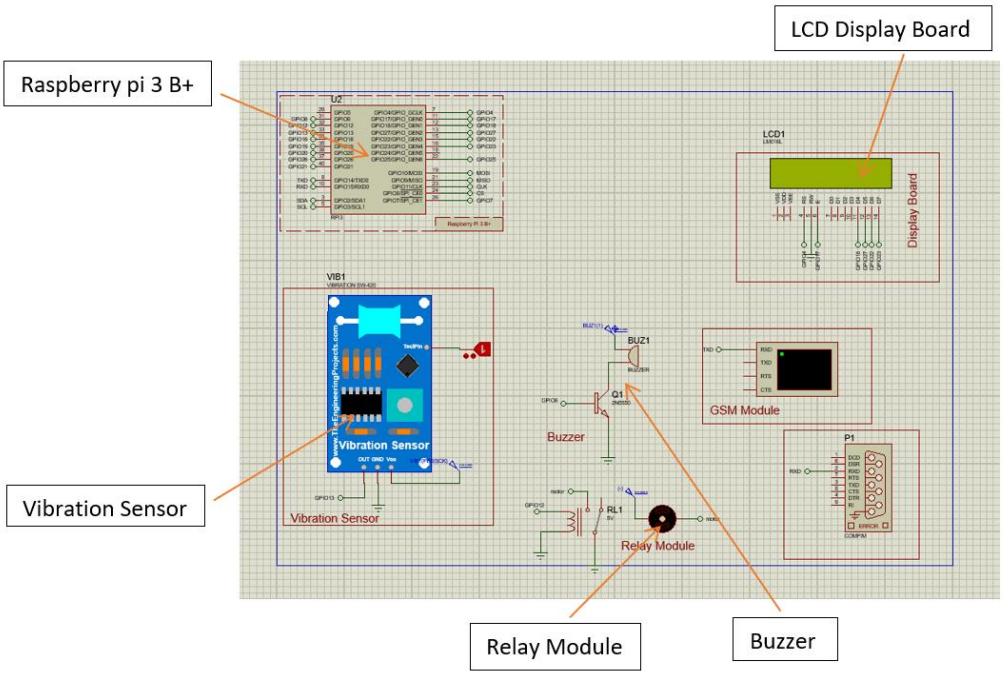
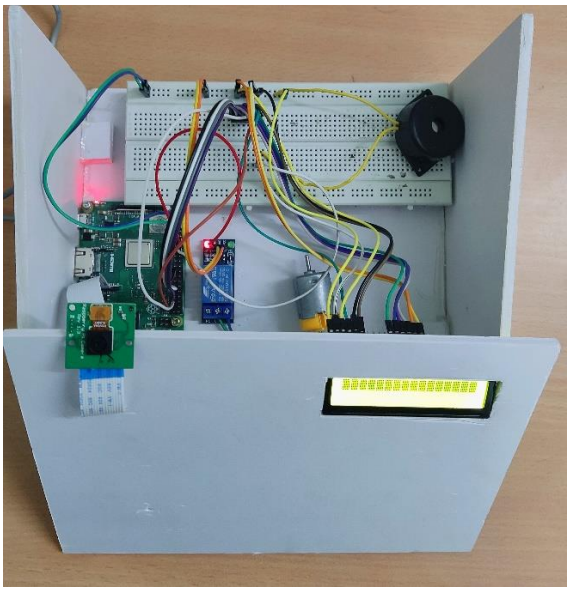


Figure 13. Software Simulation Using Proteus

The above Figure 13, is our working software project, we have created the circuit with the help of "Proteus".

These are some of our pictures of our Hardware project:



We developed an innovative device that detects driver drowsiness and can be installed in front of the driver in the car.

CONCLUSION AND FUTURE WORK:

CONCLUSION:

The Raspberry Pi 4 is the system's major component, which is used for the continuous capture of facial landmarks that are localized using facial points and then analyzed. Likewise, it gathers images from the video, which are rapid enough to recognize a driver's face. The framework makes use of open-Source software known as the "open CV" picture handling libraries. The general architecture is made out of a Raspberry Pi and an open CV to make it an easy-to-tired location framework. The buzzer goes off if the driver becomes drowsy and closes his eyes for more than a few seconds. As a result, the ratio of accidents drops. On the Raspberry Pi 4, we developed a prototype for drowsiness detection for car drivers. It totally fulfills the system's aims and objectives. This equipment can tell a difference between normal eye blinks and fatigue, which prevents the driver from falling asleep behind the wheel. Even when drivers are wearing glasses or driving in low-light conditions, the device will work. During monitoring, the equipment may detect if the eyes are open or closed. The buzzer beeps to inform the driver when he/she is in a state of drowsy (when the eyes are closed for around 15 seconds). If the device detects the drowsiness of the driver’s face, it will quickly display a drowsy message on the LCD and alert the driver through the buzzer’s sound to avoid accidents.

FUTURE SCOPE:

The blink rate and status of the vehicle factors can be used to improve this model. The device can also be attached to the accelerator to reduce the vehicle's speed and therefore the impact of the accident. The system can be improved even more by employing an infrared camera for better picture quality in a variety of lighting circumstances, as well as for persons with brown complexions and small eyes.

REFERENCES:

1. R. A. ZOROOFI & P.R TABIZI- 'FOR OPEN AND CLOSED EYE ANALYSIS TO DETECT DROWSINESS'.

2. R.L HSU AND A.K JAIN - 'FOR DETECTING THE COLOUR IMAGES OF FACE'.

3. P.E HART – 'FOR DETECTING LINES AND CURVES IN THE PICTURE'.

4. https://www.jotr.in/article.asp?issn=0975-7341;year=2013;volume=6;issue=1;spage=1;epage=6;aulast=Ruikar

5. K.C YOWAND – 'FOR DETECTING HUMAN FACE'. 'IMAGE VISION'.

6. N. AHUJA, M.H YANG – 'DETECTION OF FACES IN THE IMAGE'.

7. IOT Based Driver Drowsiness Detection and Health Monitoring System

8. MARIO I CHACON – 'DETECTION OF DRIVER DROWSINESS – SURVEY ON SYSTEM DESIGNS AND TECHNOLOGY' – IEEE Consumer Electronics Magazine.