CAPSTONE PROJECT:

🧠 Distraction Monitor – AI-Powered Attention Tracker

PRESENTED BY

STUDENT NAME: PRABHU SIDDARTH A V

COLLEGE NAME: SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT: INFORMATION TECHNOLOGY

EMAIL ID:
PRABHUSIDDARTHAV24IT@SRISHAKTHI.AC.IN

AICTE STUDENT ID:
STU681b56d5480b11746622165

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

- In today’s digital era, users often spend excessive time watching short-form videos like YouTube Shorts and Instagram Reels, unknowingly reducing their productivity. **Manual tracking is inefficient** and lacks **real-time control**. A system is needed to monitor these distractions and alert users when limits are exceeded.

PROPOSED SOLUTION

- **Overview:**
  An AI-based system that detects and limits exposure to distracting content like YouTube Shorts and Instagram Reels using real-time screen monitoring and deep learning.

- 📸 **Data Collection:**
  Screenshots manually categorized into:

  Short
  reel
  non_short

- 🧪 **Preprocessing:**
  Images resized, normalized, split into training/validation sets, and augmented.

- 🤖 **CNN Model:**
  Trained using TensorFlow/Keras

  Architecture: Conv2D → MaxPooling → Dense
  Achieved 100% validation accuracy

- 🖥️ **Real-Time Monitoring:**

  Screen captured via mss
  Images sent for prediction
  Counter tracks distractions

- 🔔 **Alerts:**

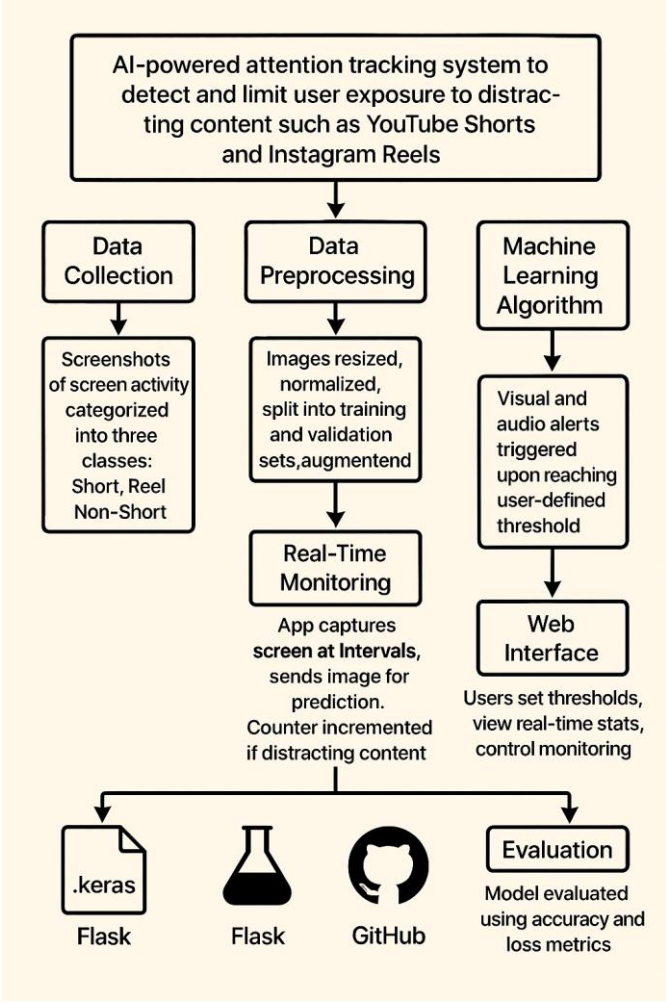
  Threshold-based visual/audio notifications

- 🌐 **Web Interface:**

  Built with HTML, CSS, JS
  User controls: threshold, stats, monitoring

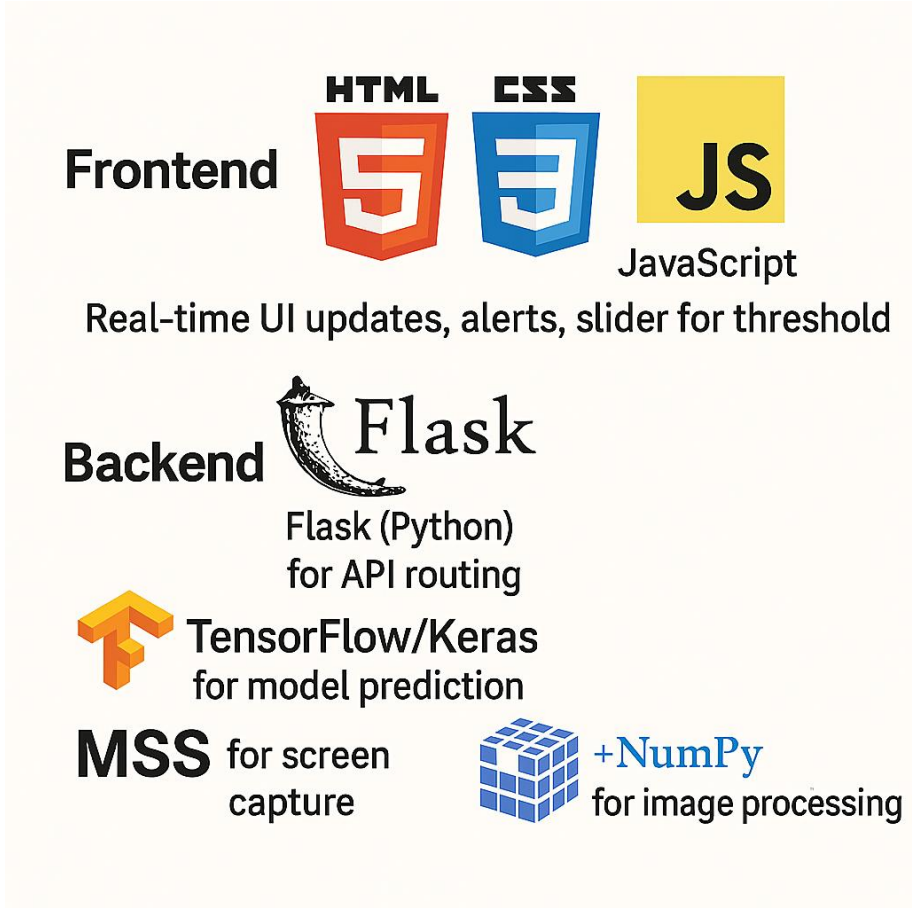
- 📈 **Evaluation:**

  Metrics: Accuracy &Loss
  Reliable real-world performance



SYSTEM APPROACH

- **Frontend:**
  - HTML, CSS, JavaScript
  - Real-time UI updates, alerts, slider for threshold
- **Backend:**
  - Flask (Python) for API routing
  - TensorFlow/Keras for model prediction
  - MSS for screen capture
  - PIL + NumPy for image processing



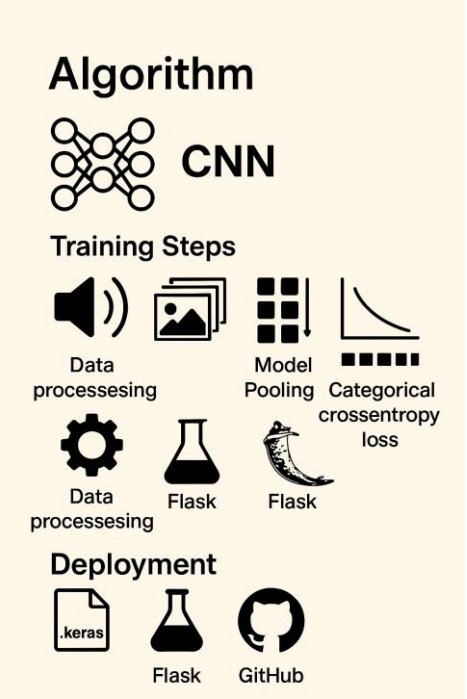
ALGORITHM & DEPLOYMENT

- **Algorithm:**

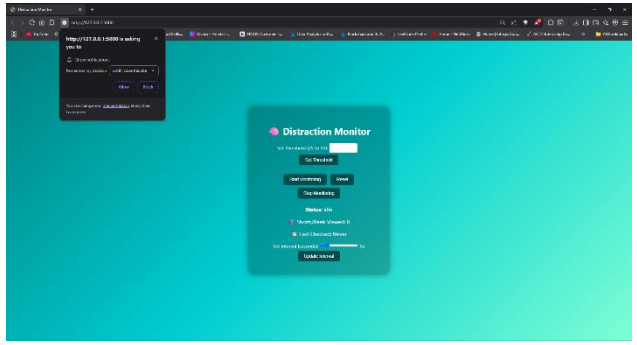
  A Convolutional Neural Network (CNN) trained on a custom dataset (short, reel, non_short)using image classification.

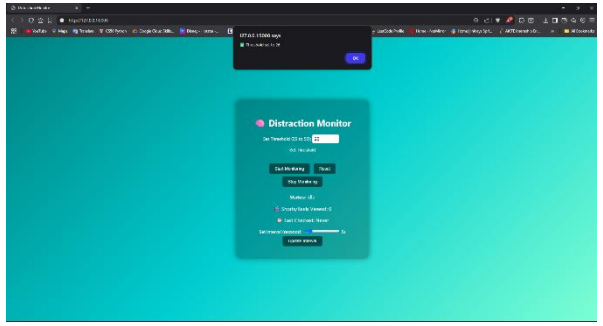
- **Training Steps:**
  - Data Preprocessing with “Image Data Generator”
  - Model: Conv2D → MaxPooling → Dense (Softmax)
  - Categorical Crossentropy Loss
  - 10 Epochs, Accuracy ~100% on Validation

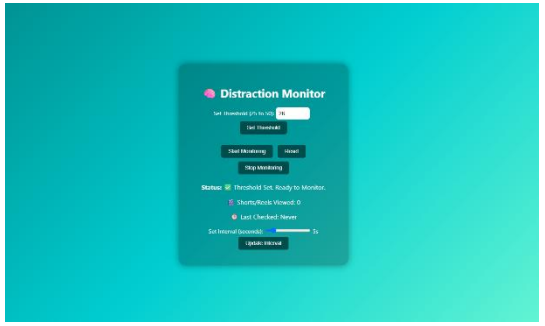
- **Deployment:**
  - Model saved as .keras
  - Flask serves as backend for monitoring
  - GitHub repo hosts open-source version

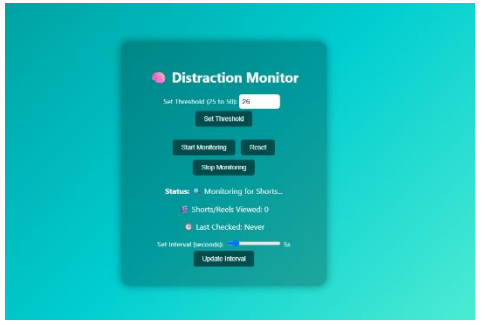


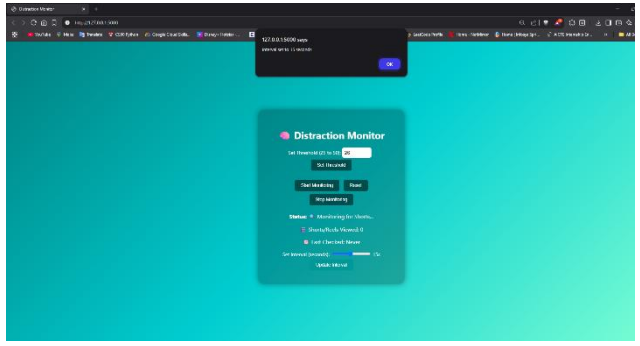
RESULT

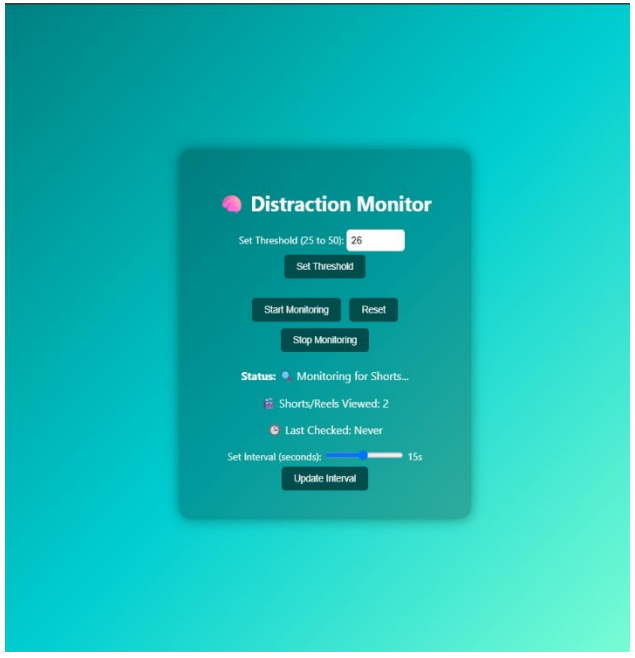


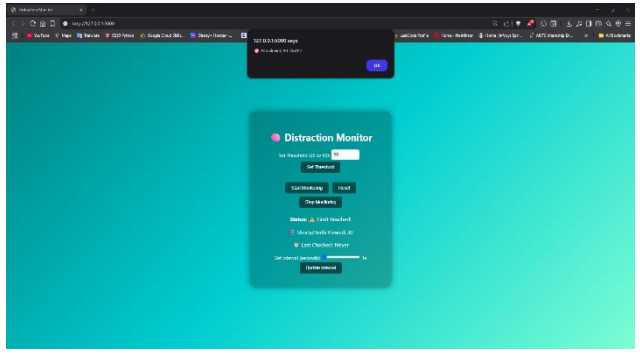













```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

127.0.0.1 - - [13/Jul/2025 00:49:27] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:49:28] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:49:29] "GET /start_monitoring HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:49:29] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:49:30] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:49:31] "GET /shorts_count HTTP/1.1" 200 -
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
✅ Non-short content detected.
127.0.0.1 - - [13/Jul/2025 00:49:32] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:49:33] "POST /set_interval HTTP/1.1" 200 -
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
✅ Non-short content detected.
127.0.0.1 - - [13/Jul/2025 00:49:46] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:49:46] "GET /start_monitoring HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:49:46] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:49:47] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:49:48] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:49:49] "GET /start_monitoring HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:49:49] "GET /shorts_count HTTP/1.1" 200 -
```

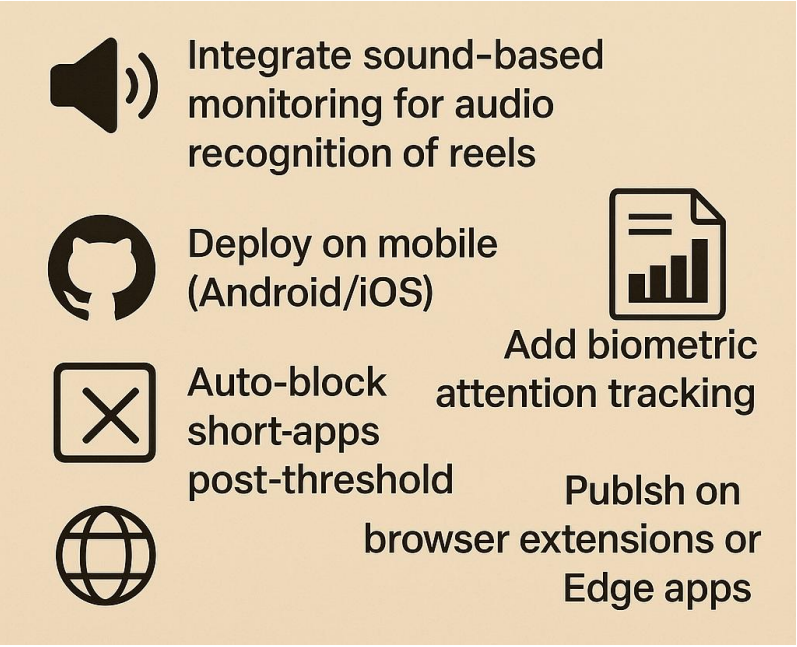
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

127.0.0.1 - - [13/Jul/2025 00:51:11] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:12] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:13] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:14] "GET /start_monitoring HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:14] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:15] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:16] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:17] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:18] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:19] "GET /start_monitoring HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:19] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:20] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:21] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:22] "GET /shorts_count HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:23] "GET /shorts_count HTTP/1.1" 200 -
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step
🔴 short detected — Counter: 2
127.0.0.1 - - [13/Jul/2025 00:51:24] "GET /start_monitoring HTTP/1.1" 200 -
127.0.0.1 - - [13/Jul/2025 00:51:24] "GET /shorts_count HTTP/1.1" 200 -
```

CONCLUSION

- Short-form video platforms like YouTube Shorts and Instagram Reels often lead to excessive screen time and reduced productivity. Distraction Monitor tackles this issue using a custom-trained CNN model to classify screen content and alert users when their viewing exceeds a set threshold. With real-time monitoring, visual and auditory cues, and customizable settings, the system helps users regain control over their digital habits.
- Built with Flask and JavaScript, the lightweight web-based interface ensures easy deployment and user-friendly interaction. The project blends deep learning with practical design, offering a promising tool for digital wellbeing. Future upgrades like cross-platform support and behavioral analytics could make it a widely adopted productivity solution.

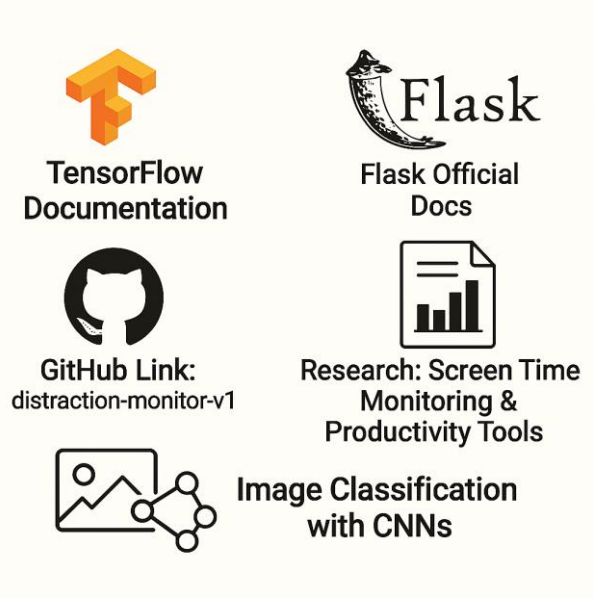
FUTURE SCOPE

- Integrate sound-based monitoring for audio recognition of reels
- Deploy on mobile (Android/iOS)
- Add biometric attention tracking
- Auto-block short apps post-threshold
- Publish on browser extensions or Edge apps



REFERENCES

- TensorFlow Documentation
- Flask Official Docs
- GitHub Link: distraction-monitor-v1
- Research: Screen Time Monitoring & Productivity Tools
- Image Classification with CNNs



Thank you