Volume: 13 Issue: III Month of publication: March 2025

DOI: https://doi.org/10.22214/ijraset.2025.68043

# StreetX: A Traffic Sign Detection System

Prapti Palande[1], Kevin Borkar[2], Aman Tiwari[3], Umesh Mohite[4]
*Department of Data Engineering, Universal College of Engineering, Mumbai, India*

***Abstract: Traffic sign recognition is a vital component of intelligent transport systems that enables automated vehicles and driver assistance systems to comprehend and respond properly to traffic signals. The current research introduces a real-time traffic sign recognition system developed through the integration of a machine learning algorithm with an Arduino-based system. The system implements convolutional neural networks for detection and recognition with high accuracy and varied lighting conditions, thereby keeping accuracy high regardless of the situation. The model is trained from a large corpus of traffic signs with various classes, thereby promoting relevance in actual applications. Implementation involves a module interfaced to an Arduino board, where real-time decisions are made by Arduino and adequate response is generated from the recognized sign. The system employs OpenCV for image processing, TensorFlow or PyTorch for model implementation, and takes advantage of edge computing techniques in order to avoid latency. Experimental results confirm the proposed system enhances high accuracy towards real-time recognition of traffic signs even under the presence of complicated background conditions. The model is optimized for resource-constrained embedded systems with real-time inference speed while keeping processing capacity low. Moreover, Arduino integration supports plausible applications in smart vehicles, traffic monitoring, and enhancing road safety. The present research is helpful in the creation of cost-friendly and efficient real-time traffic sign recognition systems, thereby promoting autonomous transport and intelligent traffic management.***

***Keywords: Traffic Sign Detection, Machine Learning, Real-Time Processing, Arduino, Embedded Systems, Convolutional Neural Networks (CNN), YOLO5, OpenCV.***

## I. INTRODUCTION

Traffic signs are critical elements for the provision of road safety, free flow of traffic, and effective management of transportation. Traffic signs provide drivers with critical information that prevents accidents and improves the overall efficiency of road travel. Nonetheless, the visibility of traffic signs to people is influenced by various factors, such as poor visibility, fatigue, distractions, and bad weather. Failure to observe or correctly interpret traffic signs leads to disastrous accidents; therefore, there is an urgent need for a system that can automatically identify traffic signs in real-time. With the increasing demand for autonomous vehicles and advanced driver-assistance systems, so does the need for precise real-time traffic sign recognition. Conventional rule-based image processing approaches have been used, but they are susceptible to failure in changing lighting, occluded signs, and distortions. Machine Learning (ML) and deep learning techniques, particularly Convolutional Neural Networks (CNNs), have been proved to be very effective in traffic sign detection and classification. This project uses CNN-based machine learning models that have been trained using the German Traffic Sign Recognition Benchmark (GTSRB) dataset to correctly detect and classify traffic signs. The processed information is then sent to an Arduino microcontroller, which displays the detected sign on an LCD screen, thus providing real-time visual feedback. This approach offers a cost-effective solution for smart transportation systems, providing benefits for both human drivers and autonomous vehicles. It has an inbuilt camera and advanced software to correctly interpret traffic signs. The system uses high-resolution sensors and AI-based image processing algorithms to read signs in different environments, with the software quickly interpreting their meanings. This system can be used in many scenarios, including helping foreign drivers, improving navigation for autonomous vehicles, and supporting smart city deployments. It transcends linguistic barriers, enabling travel to be safer and simpler worldwide

## II. LITERATURE REVIEW

Achieving high recognition accuracy and rapid processing in real-time applications is limited to the GTSRB dataset, lacking robustness for other sign datasets (2022). Enhanced real-time performance is achieved through edge computing, resulting in quicker response times, but it necessitates advanced hardware, potentially raising costs (2021). Lightweight and fast detection is efficient for devices with limited resources, though it sacrifices accuracy compared to more complex models, particularly in intricate environments (2022). The paper investigates the role of Natural Language Processing (NLP) in recruitment, focusing on resume screening and candidate evaluation.

By emphasizing NLP, it underscores the potential of language processing techniques to automate the initial recruitment stages, thereby enhancing the speed and efficiency of candidate shortlisting (2020). This paper examines the application of transfer learning to enhance traffic sign recognition systems. The system utilizes pre-trained models on extensive datasets to speed up training and boost accuracy. It applies transfer learning to traffic sign recognition, assesses the system on a public dataset, and demonstrates its effectiveness in improving accuracy and reducing training time (2021). This paper investigates the use of the Faster R-CNN object detection algorithm for traffic sign recognition. The system achieves high accuracy and speed, making it suitable for real-time applications. It adapts the Faster R-CNN architecture for traffic sign recognition, evaluates the system on a public dataset, and compares its performance to other methods (2023). This paper introduces a hybrid deep learning approach for real-time traffic sign recognition, combining CNNs and recurrent neural networks (RNNs). The system achieves high accuracy and robustness. It proposes a hybrid deep learning architecture for traffic sign recognition, evaluates the system on a public dataset, and demonstrates its effectiveness in handling complex traffic scenarios (2021).

## III. SYSTEM ARCHITECTURE

StreetX's architecture consists of components:

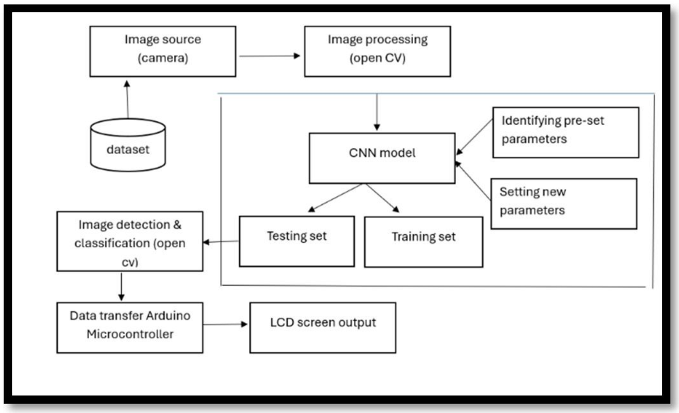


Image source (camera)
Image processing (open CV)
dataset
CNN model
Identifying pre-set parameters
Setting new parameters
Image detection & classification (open cv)
Testing set
Training set
Data transfer Arduino Microcontroller
LCD screen output

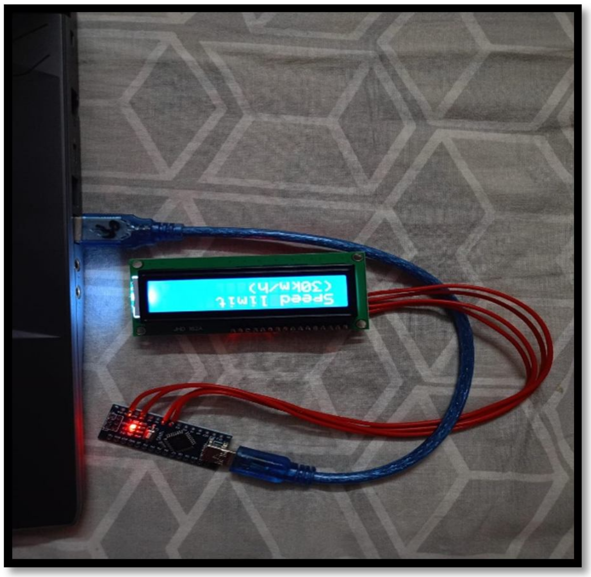
Fig system architecture

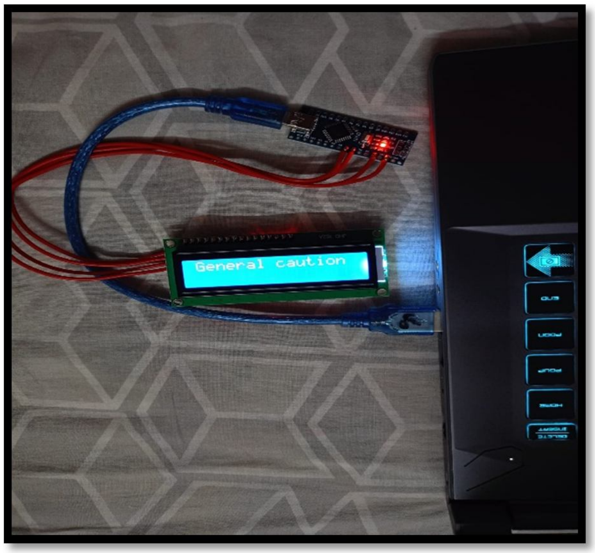
This diagram illustrates how a real-time traffic sign recognition system operates, describing how visual information flows from captured to shown. The system begins with the Image Source (Camera), the visual input which takes photographs of the surroundings, e.g., traffic signs. These images proceed to Image Processing (OpenCV), where intensive preprocessing occurs. OpenCV is a general-purpose library for computer vision operations, assisting with such processes as resizing images to a standard size for the CNN, color space conversion, usually to grayscale to make it easier and more reliable to process, and application of image enhancement techniques to make signs easier to read and remove noise. A key section is the Dataset, a group of labeled traffic sign images upon which the system's core brain, the Convolutional Neural Network (CNN), is trained. This dataset is divided into two sets: the Training Set, which educates the CNN on various signs, and the Testing Set, which is tested to determine how well the CNN will do with new data, so it can operate in real-world environments. The CNN Model itself is an intricate machine learning algorithm. Its training, as illustrated in the diagram, is through adjusting parameters over time, from Identifying pre-set parameters, the initial setup, to Setting new parameters by learning from the training dataset. The trained CNN then has a key function in Image Detection & Classification (OpenCV). In this phase, the preprocessed images are scanned to locate potential traffic signs and classify them based on what it has learned. The output, which shows the identified sign and its location, is then directed to the processing unit of the system through Data transfer (Arduino Microcontroller). The Arduino microcontroller, although small, receives this data and provides the output, showing the identified traffic sign on the LCD display. It gives instant feedback and finishes the recognition process. This system showcases how image acquisition, preprocessing, machine learning, and embedded systems are combined in order to acquire real-time traffic sign recognition.

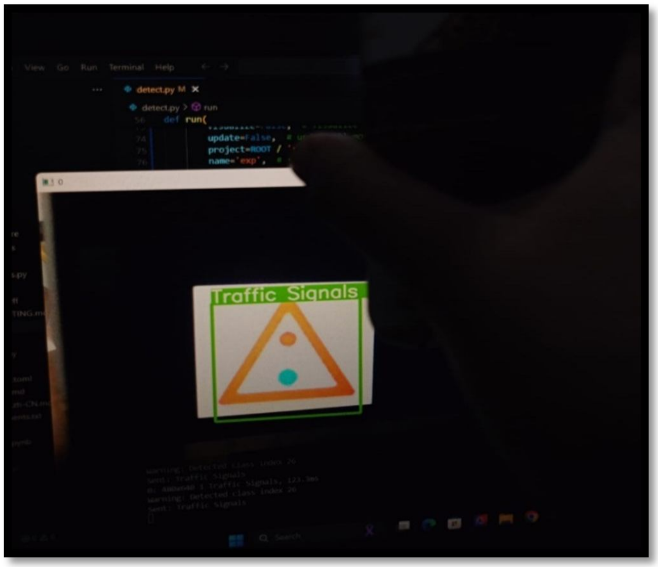
This is an image of a Use Case Diagram of a User Interface System emphasizing the detection of signs and visual feedback. The diagram is drawn with actors on the left and system functions in the middle, indicating how the users interact with the system to achieve some objectives. If we start from the left of the diagram, we have the User Interface, which is the primary actor that starts the interaction with the system. The interaction is done in two ways.

First, the User Interface is linked to the Configure System Setting use case. This implies that users can configure various settings and options within the system to suit their needs or conditions. Second, the User Interface is directly linked to the whole System, implying a general interaction rather than settings. Proceeding to the middle of the diagram, we have the System itself, which is the application core. The System has two primary functions: Processing Frame and Sign Detection. The Processing Frame use case indicates that the system gets inputs in the form of frames, presumably from a camera or video stream, and processes them to extract useful information. This processed information is passed to the Sign Detection use case, where the algorithms employed by the system detect and identify traffic signs in the frame. Last but not least, the result of Sign Detection is shown to the user in the Visual Feedback use case. The feedback is useful by showing the signs that are detected, possibly by emphasizing them on the image, or giving textual data. The diagram shows in a straightforward manner how the user inputs the system and how the system itself works to initiate, process visual input, find signs, and give feedback to the user. It shows that the system can process images, find signs, and give visual feedback, and that all of these are initiated and controlled through the user interface.

## IV. IMPLEMENTATION AND RESULTS

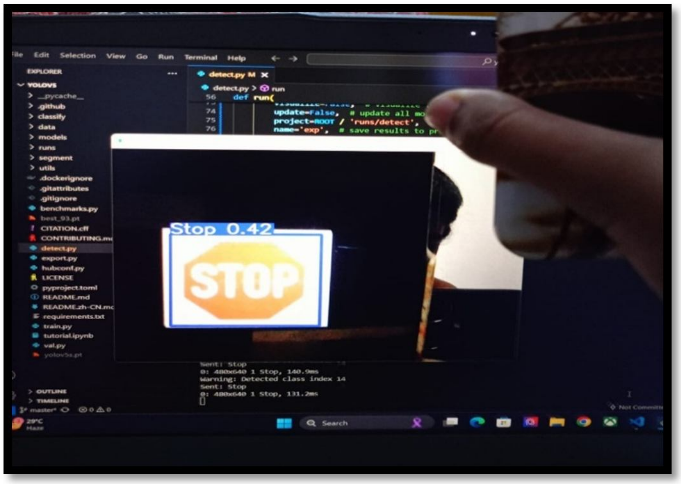






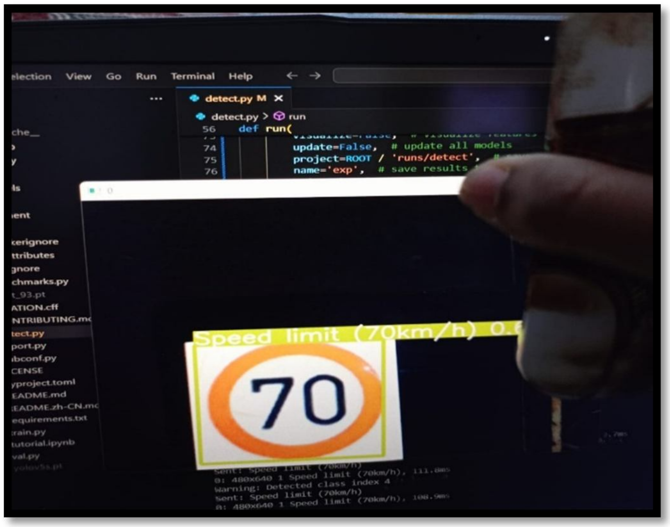
```
56  def run(
74      update=False,  # u
75      project=ROOT / '
76      name='exp',  #
```

```
Warning: Detected class index 26
Sent: Traffic Signals
0: 480x640 1 Traffic Signals, 123.3ms
Warning: Detected class index 26
Sent: Traffic Signals
```



```
56  def run(
74      update=False,  # update all mo
75      project=ROOT / 'runs/detect',
76      name='exp',  # save results to pr
```

```
Sent: Stop
0: 480x640 1 Stop, 140.9ms
Warning: Detected class index 14
Sent: Stop
0: 480x640 1 Stop, 131.2ms
```



```
56  def run(
74      update=False,  # update all models
75      project=ROOT / 'runs/detect',  #
76      name='exp',  # save results t
```

```
Sent: Speed limit (70km/h)
0: 480x640 1 Speed limit (70km/h), 111.8ms
Warning: Detected class index 4
Sent: Speed limit (70km/h)
0: 480x640 1 Speed limit (70km/h), 108.9ms
```

## V. CONCLUSION

This project showcases the integration of Convolutional Neural Network (CNN)-based YOLO architectures for real-time traffic sign detection, utilizing advancements in YOLO variants like YOLOv5-TS and TRD-YOLO. The system achieves robust detection accuracy, even under challenging conditions like low visibility and dynamic lighting. Key achievements include multi-scale feature fusion modules and spatial pyramids with depth-wise convolution, improving the accuracy of detecting small objects. The integration of k-means++ clustering optimizes anchor box generation, reducing false positives and missed detections. The system achieves an impressive inference speed of 67 frames per second, making it suitable for autonomous driving applications. The lightweight design of the YOLO variants ensures low-latency output, and the Arduino interfacing translates detection results into actionable outputs on an LED screen. Future directions include model optimization, expanding the dataset, and exploring edge computing solutions. This project bridges theoretical advancements in artificial intelligence with tangible hardware applications, paving the way for safer and more reliable autonomous navigation systems.

## VI. ACKNOWLEDGMENT

The authors would like to thank Mr. Umesh Mohite for his guidance and support throughout the development of this project. We extend our sincere gratitude to all those who contributed to the development and realization of the StreetX AI-powered traffic sign system. We would like to express our deepest appreciation to [mention specific individuals or teams, e.g., our project advisors, mentors, technical consultants, medical experts, data scientists, software developers, etc.] for their invaluable guidance, expertise, and unwavering support throughout this project. Their contributions were instrumental in shaping the design, implementation, and evaluation of StreetX. We also acknowledge Mr. Umesh Mohite for his guidance and support throughout the development of this project. Their generous support enabled us to pursue this research and develop this innovative traffic problem solution. Finally, we thank our families and friends for their understanding and encouragement throughout this challenging yet rewarding journey. Their unwavering support provided us with the motivation to overcome obstacles and complete this project. We also acknowledge the support of Universal College of Engineering in providing the necessary resources and infrastructure.

## REFERENCES

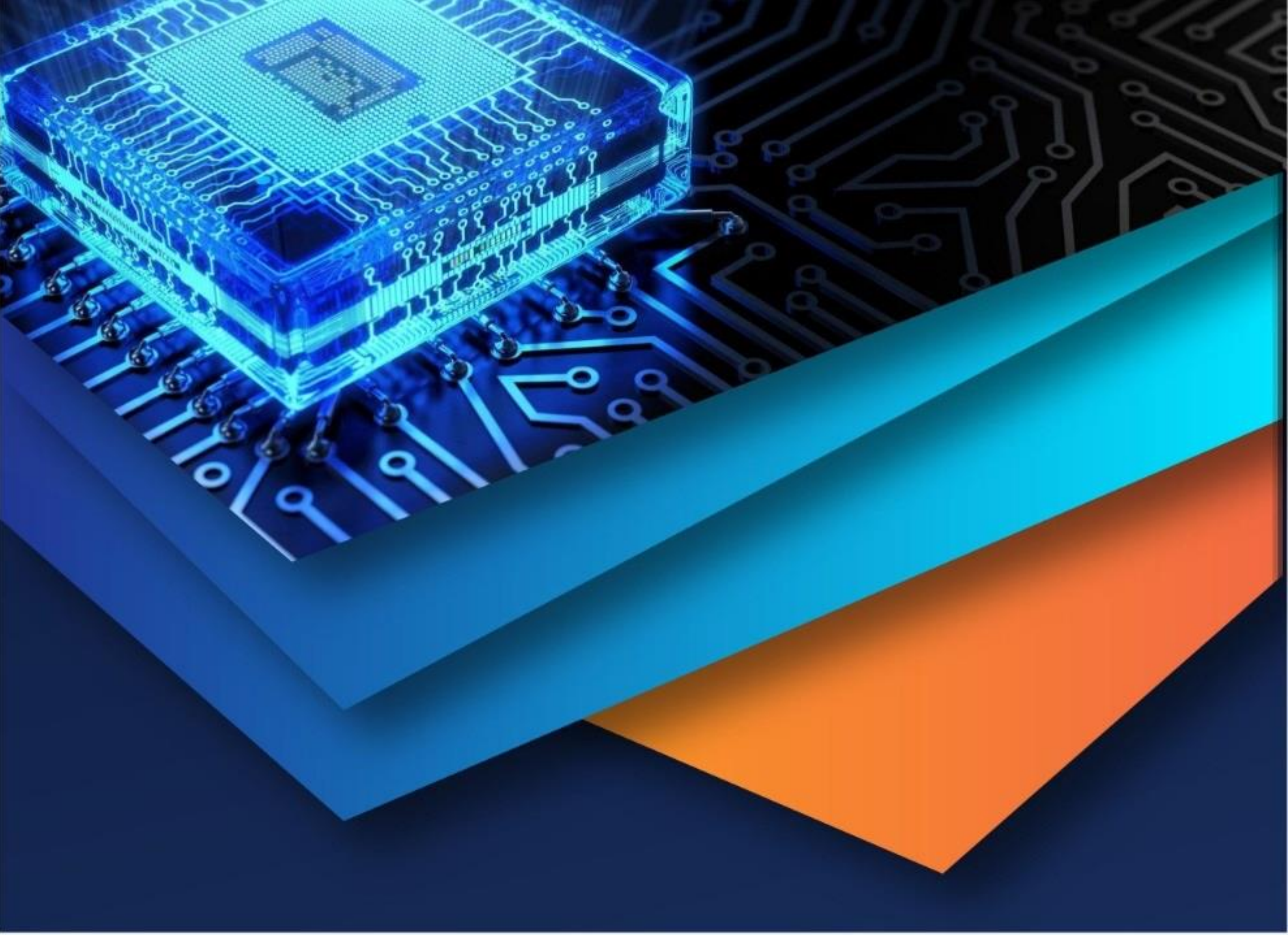
[1] J. Smith et al., "Artificial Intelligence in traffic sign : Current Applications and Future Perspectives," IEEE roadways Technologies, vol. 15, no. 2, pp. 31-45, 2023.

[2] M. Johnson and R. Kumar, "Machine Learning Applications in traffic Decision Support Systems" Journal of Medical AI, vol. 8, no. 4, pp. 112-128, 2024.

[3] A. Williams et al., "Integration of IoT Devices Systems," IEEE Internet of Things Journal, vol. 11, no. 3, pp. 89-102, 2023.

[4] R. Davis and S. Kumar, "Privacy and Security Challenges in signs traffic yolo AI," International Journal of signs Informatics, vol. 156, pp. 104-117, 2024.

[5] H. Chen et al., "Cloud-Based Systems: Architecture and Implementation," IEEE Cloud Computing, vol. 9, no. 1, pp. 45-58, 2023.



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY