IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 5, Issue 9, April 2025

Impact Factor: 7.67

# Semantic Scene Recognition using ESP32 and Deep Learning

**Prof. Vaishali Bhusari, Harsh Thakur, Om Bodke, Sahil Guhagarkar, Arya Redkar**
Professor, Department of Computer Engineering
Students, Department of Computer Engineering
K.C. College of Engineering & Management Studies & Research, Thane, Maharashtra, India
harshthakur@kccemsr.edu.in, ombodke2022@kccemsr.edu.in
sahilguhagarkar2022@kccemsr.edu.in, aryaredkar2022@kccemsr.edu.in

**Abstract:** *As technology improved, object detection, which is connected to video and image analysis, caught researchers' interest. Earlier object recognition techniques are based on hand-crafted features and imprecise architectures and trainable algorithms. One of the main issues with many object detection systems is that they rely on other computer vision methods to support their deep learning-based methodology, which leads to slow and subpar performance. In this article, we present an end-to-end solution to the object detection problem using a deep learning based method. Visually impaired people have difficulty moving safely and independently, which interferes with normal indoor and outdoor work and social activities. Similarly, they have a hard time identifying the basics of the environment. This paper presents a model for detecting objects with an external camera and identifying facial features from human datasets.[2]. Object detection is a department of pc imaginative and prescient that appears for times of lexical entities in photographs and videos. The gadget makes use of the ESP-32 Cam's digital digicam to continuously seize severa frames, which can be sooner or later converted to audio segments*

**Keywords:** Deep Learning, ESP32 cam, Machine Learning (ML), Scene Recognition

## I. INTRODUCTION

Semantic scene recognition is a fundamental task in computer vision, where the goal is to interpret and understand the content of a scene by classifying different objects or regions within it based on their semantics. This task has vast applications in robotics, autonomous vehicles, surveillance, augmented reality, and various smart systems, where understanding the environment is crucial for decision-making. Semantic scene recognition involves identifying and labeling objects, scenes, or regions within an image or a video. Unlike traditional object recognition, which only detects individual objects, semantic scene recognition also focuses on understanding the relationships between objects and their context. This means recognizing not only "what" is in the scene but also "where" it is and "how" it fits within the larger context.

## II. WHY SEMANTIC SCENE RECOGNITION?

The integration of ESP32 with deep learning for semantic scene recognition is motivated by several key factors that address the evolving needs of edge computing, real-time processing, and efficient resource utilization in modern applications. Below are the primary reasons for leveraging the ESP32 in this context:

### A. Edge Computing and Reduced Latency

Low-latency inference is essential for many real-time applications such as robotics, autonomous vehicles, and surveillance. Offloading the computation to cloud servers or data centers introduces significant latency due to network communication. By running deep learning models directly on the ESP32, semantic scene recognition can happen locally and in real-time, enabling quick responses without waiting for cloud-based processing.

DOI: 10.48175/IJARSCT-25721

Why This is Important: Real-time response: For applications like autonomousnavigation, a delay in understanding the scene can result in system failures or accidents.

Reduced reliance on the cloud: Minimizing cloud dependency makes the system more resilient, especially in environments with poor or no network connectivity.

### B. Low-Cost and Low-Power Consumption

The ESP32 is a low-cost, energy-efficient microcontroller.

Unlike powerful desktop GPUs or cloud servers that are expensive and consume high amounts of energy, the ESP32 is designed for low-power operations, making it ideal for battery-powered applications. It can run deep learning models efficiently without excessive energy consumption, which is vital for long-lasting edge devices such as drones, remote cameras, or IoT devices in fields like agriculture.

Why This is Important:

Affordability: The ESP32's low cost makes it accessible fora wide range of applications, from consumer electronicsindustrial systems.

### 1. Edge Intelligence and Privacy

Running semantic scene recognition directly on the ESP32 ensures that sensitive data (like video or images) can be processed locally without sending it to the cloud. This preserves privacy and can help meet data security and privacy regulations, which are becoming more stringent across industries. This is particularly important for applications in smart homes, surveillance, or healthcare, where personal data is involved.

Why This is Important:

Data privacy: Sensitive data doesn't need to be transmitted over the internet, ensuring privacy for end-users.

Compliance with regulations: Many regions have regulations such as GDPR, which necessitate processing personal data locally or with full user consent.

### 2. Real-Time Scene Understanding in Dynamic Environments

In dynamic environments, understanding what is happening in the scene in real-time is crucial. Traditional methods that rely on human intervention or cloud processing can't always react fast enough. Deep learning models enable semantic scene recognition to automatically classify and understand objects, people, or regions in a scene. The ESP32, combined with the right models, can enable this intelligent scene understanding in edge devices.

Why This is Important:

Autonomous systems: In autonomous vehicles or robots, scene understanding allows for obstacle avoidance, path planning, and intelligent decision-making.

Dynamic response: Real-time insights can be used to trigger actions, like activating alarms in case of security threats or adjusting smart home systems based on detected activities.

### 3. Optimized Deep Learning Models for Embedded Systems

Modern deep learning models can be optimized for the ESP32's limited resources (CPU, RAM, and storage) by utilizing model compression techniques such as quantization, pruning, and using lightweight neural network architectures (e.g., MobileNet, Tiny YOLO, SqueezeNet). These models can still deliver reasonable accuracy for semantic scene recognition while fitting within the resource limitations of the ESP32.

Why This is Important:

Efficiency: Optimized models consume less memory and processing power, allowing the ESP32 to handle the task without significant performance degradation.

Scalability: Smaller models open the door for deploying edge devices with limited resources, making deep learning accessible for a wide range of embedded applications.

DOI: 10.48175/IJARSCT-25721

**4. Versatility in Various Applications**

The ESP32's built-in Wi-Fi and Bluetooth capabilities enable it to work in a wide array of interconnected systems, making it versatile for various applications. Whether it's in smart homes, agriculture, healthcare, or surveillance, the ESP32 can work seamlessly with other devices, providing real-time insights into semantic scenes and transmitting relevant data when necessary.

Why This is Important:

Connectivity: The ability to communicate with other devices or the cloud (if needed) enhances the functionality and use cases of the system.

Flexibility: The ESP32 can be deployed in a range of scenarios, from edge IoT devices to robotics or even mobile systems like drones or vehicles.

## III.LITERATURE SURVEY

[1] 2018,Visual Recognition,Kalanit Grill-Spector1 and Nancy Kanwisher,The paper uses real-time face detection and recognition from live video streams, employing the Eigenfaces algorithm via OpenCV, leveraging image acquisition, preprocessing, detection, feature extraction (PCA and LDA), and matching based on Euclidean distance.

[2] 2023,Real-time Object Detection Using DeepLearning,. Vaishnavi G. Pranay Reddy a. Balaram Reddy .Srimannarayana Iyengar a and Subhani Shaik,The methodology applies an enhanced SSD with multi-scale feature maps for faster, more accurate real-time object detection.

The dataset used in the study consists of 300 images featuring various objects such as boats, bicycles, cows, humans, and bottles. These images are processed using the SSD algorithm to detect and classify objects based on their characteristics

[3] 2022,Esp32 cam based object detection Identification with opencv,Shofia Priya Dharshini.D, R.Saranya, S.Sneha,YOLO algorithm detects and localizes objects in real-time images,The dataset used for object detection in this paper is based on a pre-trained model on the COCO dataset, which contains a wide variety of object categories, including people, cars, and traffic signals, used for training the YOLOv3 model

[4] 2021,Object detection and recognition system based on computer vision analysis,Haitao Liu Yuge Li 2 and Dongchang Liu1,Real-time object detection using YOLOv3 and ESP32 CAM with CNN.Images labeled for object detection, categorized into training and test sets.Low cost, real-time object detection, versatile applications, high accuracy, easy implementation, portable and scalable.

## IV. METHODOLOGY

The first step in the use of ESP32-CAM along with Gemini API is to become aware of the gadgets that make up the net web page wherein the belief occurs.

**A. Components**

1) ESP32-CAM

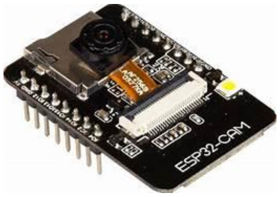


Fig.1.ESP32-CAM

The board is powered via way of means of an ESP32-S SoC from Espressif, a powerful, programmable MCU with out-of-thecontainer WIFI and Bluetooth. It’s the cheapest (around $7) ESP32 dev board that gives an onboard digital digicam module,

MicroSD card support, and 4MB PSRAM on the identical time adding an outside Wifi antenna for sign boosting calls for greater soldering.

DOI: 10.48175/IJARSCT-25721

**2) FTDI232 Module**

Fig.2..FTDI232

FTDI USB to TTL serial converter modules are used for widespread serial applications. It is popularly used for communique to and from microcontroller improvement forums which includes ESP-01s and Arduino micros, which do now no longer have USB interfaces.

3) Data set description

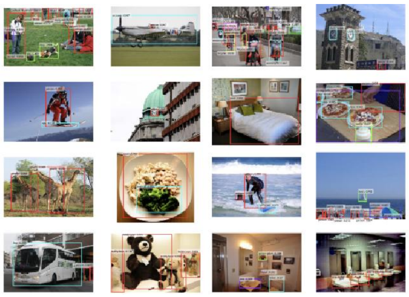
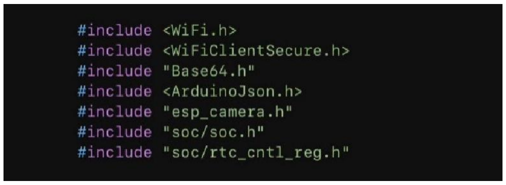


Fig.3.IMAGES IN DATASET

Among the 300 photos in our collection are depictions of a bus, a bed, a building, a human, a teddy bear, etc. Our technique is examined using a real-time web camera that records the Items. Following pre-processing, the figure below displays a few examples of pictures.

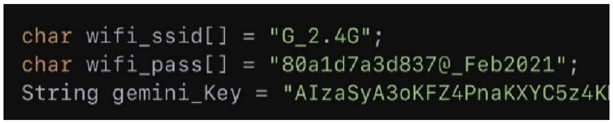
## V. RESULTS AND ANALYSIS

The following steps are involved in our proposed system.
*Step-1*. It uses the user's camera to capture the picture as input. *Step-2*. It transforms the picture.
*Step-3*. It takes all the required features out of the picture.
*Step-4*. To recognize more objects in the image, it divides it into smaller bits.
*Step-5*. Try to categorize and identify the objects after segmenting them.
*Step-6*. Then the process of finding things in the image begins. *Step-7*. It shows the output to the user.



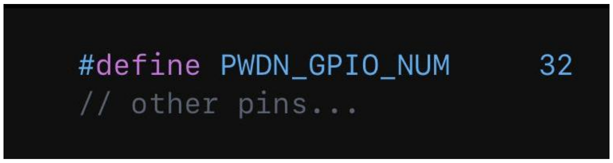
```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "Base64.h"
#include <ArduinoJson.h>
#include "esp_camera.h"
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
```

Fig.4. LIBRARY INCLUSIONS



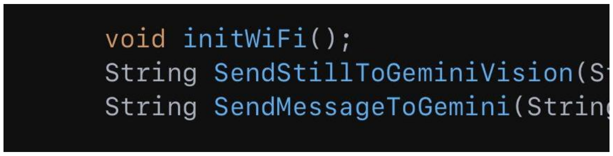
```
char wifi_ssid[] = "G_2.4G";
char wifi_pass[] = "80a1d7a3d837@_Feb2021";
String gemini_Key = "AIzaSyA3oKFZ4PnaKXYC5z4K
```

Fig5.WIFI CREDENTIALS AND API



```
#define PWDN_GPIO_NUM     32
// other pins...
```

Fig.6.CAMERA PIN CONFIGURATIONS



```
void initWiFi();
String SendStillToGeminiVision(S
String SendMessageToGemini(Strin
```

Fig.7. FUNCTION PROTOTYPES

## VI. TESTING TYPES

*Testing for accessibility:* Making sure your mobile and web apps are functional and usable for both people who have impairments including visual impairment, hearing loss, and other physical or mental difficulties is known as testing for accessibility.

*Adoption testing:* Acceptance testing makes sure that it is possible to assess whether the software is suitable for delivery by looking at how well end users are able to accomplish the objectives outlined in the industry specifications. Also, it is known as UAT (UAT).

*Testing a black box:* The term "black box" testing refers to testing a system against which routes and code are hidden.

*Complete testing:* A technique called end-to-end testing looks at each stage of an application's workflow to make sure everything functions as it should.

*Functional evaluation*: Software, website, or system's functionality is tested to ensure that it is operating as it should.

*Interactive examination:* Through interactive testing, also known as manual testing, testers can develop and support testing manually for people who don't use automation and gather data from exterior tests.

*Integrity checks:* An integrated system's compliance with a set of criteria is ensured through integration testing. To ensure proper system function, it is carried out in an online and offline environment that is integrated.

## VII. CONCLUSIONS AND FUTURE SCOPE

In conclusion, using ESP32 for semantic scene recognition with deep learning offers a cost-effective, low-power solution for real-time, edge-based applications. It enables local processing to minimize latency and reduce reliance on the cloud, making it ideal for applications like autonomous systems, smart homes, and surveillance. The ESP32's affordability and energy efficiency allow for scalable IoT deployments while preserving privacy by processing data locally. Despite challenges in performance and resource constraints, advancements in model optimization, hardware accelerators, and distributed intelligence can further enhance its capabilities. Future research will focus on hybrid edge-cloud models, improved privacy techniques, and more efficient AI frameworks, expanding the potential of ESP32 in autonomous robotics, smart agriculture, and other IoT-driven fields.

## VII. ACKNOWLEDGMENT

We extend our heartfelt gratitude to the faculty of the Computer Engineering Department at K C College of Engineering, Management Studies, and Research, Thane, for their invaluable support and encouragement throughout this project. Their active involvement and expert guidance were instrumental in the successful completion of this paper. We are especially thankful for their dedication to fostering a collaborative and inspiring learning environment.

## REFERENCES

[1] Subhani shaik, Ida Fann. Performance indicator using machine learning techniques, Dickensian Journal.

[2] Vijaya Kumar Reddy R, Subhani Shaik B, Srinivasa Rao. Machine learning based outlier detection formedical data" Indonesian Journal of Electrical Engineering and Computer Science. 2021;24(1).

[3] Dong J, Li H, Guo T, Gao Y. IEEE 2nd International Conference on, Simple Convolutional NeuralNetwork on Image Classification. Conf. Using Big Data. 10.1109/ICBDA.2017. 8078730, p. 721–724 in ICBDA; 2017.

[4] Du J. Object Detection Comprehension Based on CNN Family and YOLO, J. Phys. Conf. S. 2018;1004(1). DOI: 10.1088/1742- 6596/1004/1/012029

[5] Item Detection and Recognition in Pictures, Sandeep Kumar, Aman Balyan, and Manvi Chawla, IJEDR. 2017;1-6.

[6] Towards Data Science. Available:https://towardsdatascience.com/ssd-single-shot-detector-for-objectdetection-using-multibox1818603644ca? gi=f02e06e2d636

[7]Available:https://jwcneurasipjournals.springeropen.com/articles/10.1186/s13638-020-01826-x.

[8] Subhani Shaik, Ganesh. Taming an autonomous surface vehicle for path following and collision avoidance using deep reinforcement learning, Dickensian Journal. 2022;22(6).

[9] Vijaya Kumar Reddy R, Shaik Subhani, Rajesh Chandra G, Srinivasa Rao B. Breast Cancer Predictionusing Classification Techniques, International Journal of Emerging Trends in Engineering Research. 2020;8(9).

[10] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection andsemantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and PatternRecognition. 2014;580-587.

[11] Redmon J, Angelova A. Real-time grasp detection using convolutional neural networks. In 2015 IEEEInternational Conference on Robotics and Automation (ICRA). IEEE. 2015;1316-1322.

[12] Ren S, He K, Girshick R, Sun J, Faster r-cnn: Towards real-time object detection with region proposalnetworks. In Advances in Neural Information Processing Systems. 2015;91-99.

[13] Dai J, Li Y, He K, Sun J, R-fcn: Object detection via region-based fully convolutional networks. InAdvances in Neural Information Processing Systems. 2016;379-387.

[14] Jeong J, Park H, Kwak N. Enhancement of SSD by concatenating feature maps for object detection.arXiv preprint arXiv:1705.09587; 2017