iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79510

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

AR-Powered Retail Assistant Using YOLOv8 and LLM Integration

Roshan Raj N[1], Mahima Jane E[2]
[1, 2]*Department of Data Science, Madras Christian College, Chennai, India*

Abstract: In most physical retail settings, customers rely mainly on product labels or search on their phones to understand product details, which can interrupt the overall shopping flow. This project explores a simple approach to improving that experience by combining computer vision, augmented reality, and a basic conversational system. A lightweight YOLOv8 model is used to recognize selected retail products through a mobile camera. When a product is identified, key information such as price, features, and ingredients is shown directly on the screen as an overlay. The system also includes a small question–answer feature where users can ask about the detected product and receive responses from a locally running language model. Everything is designed to run in a browser, so there is no need for installation. To make the system feel complete, a basic cart and payment simulation is also included. During testing, the model was able to detect the selected products with good consistency and respond within a short time under normal conditions. While the system is limited to a small set of products, it shows how combining these technologies can make everyday shopping a bit more interactive and easier to navigate.
Keywords: Augmented Reality, YOLOv8, Computer Vision, Retail Assistant, Large Language Model, Smart Shopping

I. INTRODUCTION

These days, technology is improving a lot, especially in online shopping. When we shop online, it is very easy to see product details, compare items, and understand what we are buying. But when we go to a physical store, things are still mostly the same. We pick a product, read the label, and try to understand it. If something is not clear, we either ignore it or search on our phone.

Because of this, there is a difference between online and offline shopping. In stores, we can see the product directly, which is good, but we don't get extra information. Sometimes it takes time to understand a product, especially if it is new or if there are many similar items.

In many shops, the extra information is given through things like QR codes, stickers, or printed offer boards. But these need to be updated regularly. Whenever the offers change or when the paper gets damaged, they need to be replaced again. This creates extra work and sometimes outdated information may still be there.

So in this project, a simple idea is used to make shopping a bit easier. The system uses the camera to detect products and show some basic details on the screen itself. This way, the user doesn't have to search separately. Along with that, a small question–answer option is added, where the user can ask something about the product and get a reply.

The system architecture is simple. because It works in a web browser, we don't need to install anything. It combines product detection, a basic overlay display, and a simple response from llm. A small cart feature is also added just to show how the full shopping experience .

This is not a fully developed product. It is mainly developed to show how these technologies can be used together in a simple way to improve the shopping experience.

II. RELATED WORK

In this section, some existing ideas related to this project are discussed. This mainly includes augmented reality, object detection, and basic AI-based response systems, since these are the main parts used here.

A. Augmented Reality

Augmented reality has been used in different areas for some time now. In retail, it is slowly coming up, but still not very common in normal stores. Most people still follow the usual way of shopping, like picking a product and reading what is written on it. AR is mostly seen in some apps or larger brand systems, but not in everyday use.

Some research works mention that AR can improve user experience. For example, Javornik (2016) explained that AR can make shopping more interactive. Instead of just seeing the product, users can also get extra information in a better way. Poushneh and Vasquez-Parraga (2017) also mentioned that when users get more information easily, it helps them make decisions faster. Hilken et al. (2017) discussed that AR can connect online and offline shopping experiences to some extent.

Even though these works show good results, in real shops AR is still not used much. One reason is that many systems need separate apps or setup. This makes it less practical for normal use. So there is still a need for something simple that can work without much setup.

B. Object Detection Using Computer Vision

Object detection is another important part of this project. It is used to identify products through the camera. In simple terms, the system checks an image and tries to say what object is present.

There are different models for this, like YOLO, SSD, and Faster R-CNN. Some models are accurate but slow, and some are fast but not very accurate. For this type of system, speed matters more because it should work in near real time.

YOLO, introduced by Redmon et al. (2016), is commonly used because it detects objects in a single step. This makes it faster compared to older methods. Because of this, it is used in many real-time applications.

In retail, computer vision is already used in some systems like product tracking or automation. But most of those systems are not simple and need more setup. In this project, the idea is not to build something complex, but just to use detection in a basic way to recognize products and show details.

C. AI-Based Response Systems

AI-based response systems are also used in many places like websites and apps. These systems help users by answering questions. Earlier, they were very limited and worked only for fixed questions. If the input changed slightly, they would not give proper answers.

Now, newer systems are better and can handle different types of questions. They are commonly used in online shopping for support and product queries. But in physical stores, this type of system is not widely used.

In this project, the response system is kept very simple. It only answers questions related to the detected product. It is not designed as a full chatbot, just a small feature to help the user.

So overall, these technologies already exist, but they are mostly used separately. This project is trying to combine them in a simple way so that they can be used together in a real-world shopping experience.

III. METHODOLOGY

In this section, the working of the system is explained in a simple way.The system is built using a few main parts like product detection, display, and a simple response system. All these parts are connected so that the user can detect a product and see information directly.

A. Dataset Description

For this project, a small dataset was created using a few retail products. It mainly includes items like shampoo, biscuits, oats, and noodles. The images were taken from different angles and in different lighting conditions so that the model can learn how the products look in various situations.

The dataset is not very large, but it was enough for this setup. Each image was manually labeled to mark the product clearly. This helps the model understand where the object is present.

Some sample training images are shown in Fig. 1. These images show how the products were captured during the dataset preparation.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Fig. 1RetailTraining Images

Some sample images from the dataset are shown in Fig. 2. These images include different views of the products and also some real store conditions where multiple items are present. Bounding boxes are used to mark the products clearly. This helps the model learn how the objects look in different situations.

B. System Flow and User Interaction

This system starts with the camera. When the system is opened in the browser, it accesses the mobile camera and keeps capturing the scene continuously. These frames are then sent to the backend at regular intervals.

On the backend side, a YOLOv8n model is used for product detection. This model is trained using a small dataset containing a few selected retail products. When an image is received, the model checks it and tries to find if any known product is present. If a product is detected with enough confidence, the system takes that result and moves to the next step.

After detection, the system needs to show some details about the product. For this, product information is stored in a simple JSON file. Each product has details like name, price, and some basic information. When a product is detected, it is matched with this data, and the corresponding details are retrieved.

These details are then sent to the frontend. On the user side, the information is displayed directly on the screen as an overlay. A bounding box is also shown around the detected product so that the user can clearly see what is being identified. This gives a basic augmented view, even though it is not a full 3D AR system.

Along with this, a small question–answer feature is included. When the user enters a query, it is sent to the backend. A locally running language model (LLaMA 3.1 through Ollama) is used to generate a response. The system is kept controlled so that it mainly answers questions related to the detected product.

A simple cart feature is also added. When a product is detected, the user can add it to the cart. The cart keeps track of items, quantity, and total price. There is also a basic payment option, but it is only for demonstration and does not involve real transactions.

All these parts work in a continuous loop. The camera captures images, detection happens, details are shown, and the user can interact at the same time. The system is designed to be simple so that it can run completely in a browser without requiring installation.

Overall, the methodology is not very complex, but it shows how different components like detection, data handling, and user interaction can be combined into one working system.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

The overall working of the system is shown in Fig. 2.

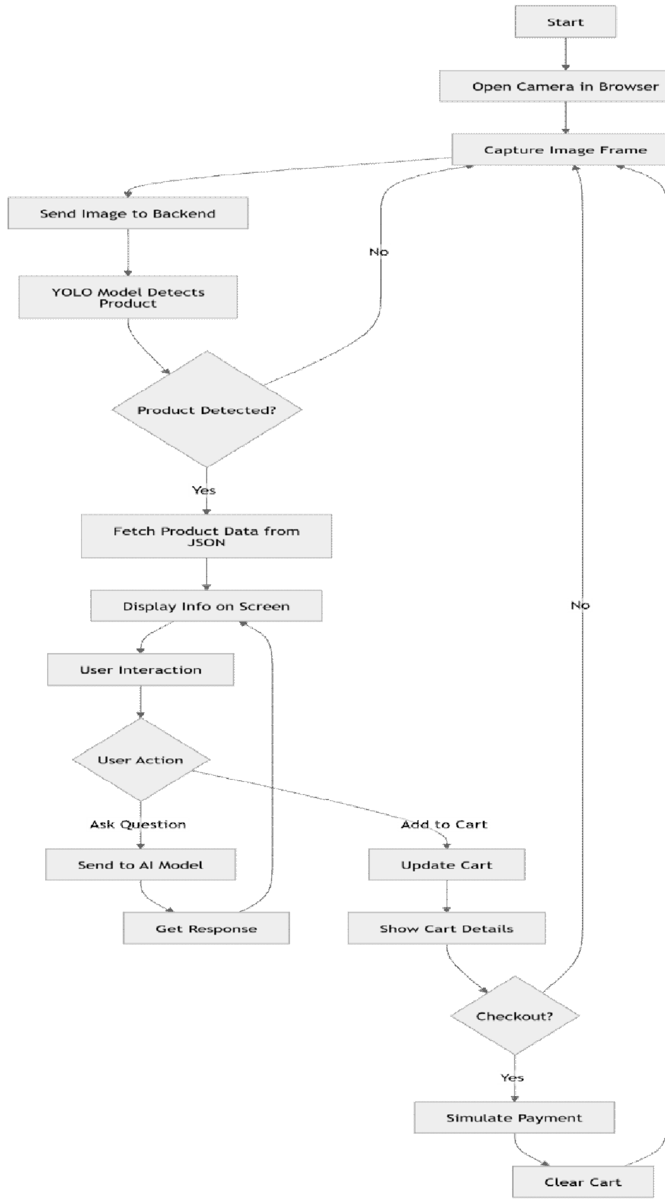


Fig. 2 System Workflow

The workflow in Fig. 1 shows how the system works step by step. First the camera opens and it keeps capturing images continuously. These images are then sent to the backend side after a small delay.

In the backend, the model checks the image and tries to detect if any product is there. If a product is detected, then the system will take that result and search for the product details.

Those details are already stored, so it just picks the correct one and sends it back. Then on the screen, the product is highlighted and some basic details like price are shown.

After this, the user can do some actions. The user can ask about the product or add it to the cart. If a question is given, the system generates a reply based on that product.

This process keeps running again and again while the camera is active. So it looks continuous from the user side.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

IV. RESULTS AND ANALYSIS

After completing the implementation, the system was tested using the trained product classes. The main goal during testing was just to see if the system can detect the products properly and show the correct details on the screen.

In most cases, the detection worked fine when the product was clearly visible in front of the camera. The system was able to identify the product and show basic information without much delay. The overall flow from detection to display was working as expected

A. Detection Performance

The detection speed was also checked during testing. When a product was shown to the camera, the system took a short time to process the image and display the result. It was not fully instant, but still fast enough for this setup.

After training, the model was giving pretty good results for the dataset we used. The mAP50 was around 0.96, so most of the products were getting detected correctly. Precision came around 0.94, which means most of the detections were right. Recall was around 0.87, so in some cases the model missed a few products, but overall it was working fine.

The precision–recall behaviour of the model is shown in Fig. 3.

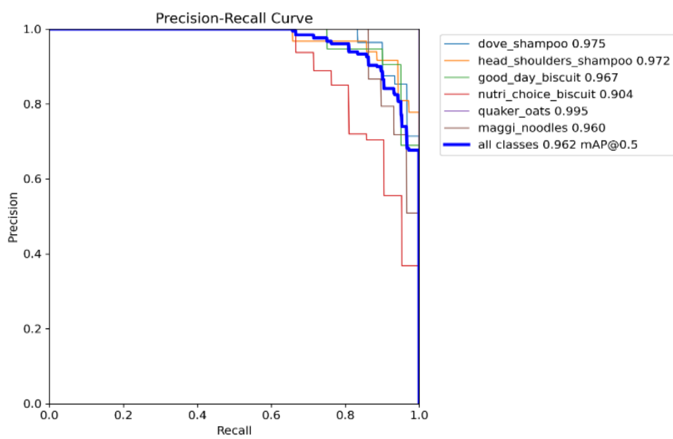


Fig. 3 Precision–Recall Curve

During testing, products like shampoo, biscuits, oats, and noodles were detected without much issue. The system was able to handle different angles and positions of the products.

In some cases, the detection was not that accurate. When the lighting was low or the image was slightly blurred, the model either gave wrong results or missed the product. It didn't happen too often, but we did notice it during testing.

The confusion matrix for the model is shown in Fig. 4.

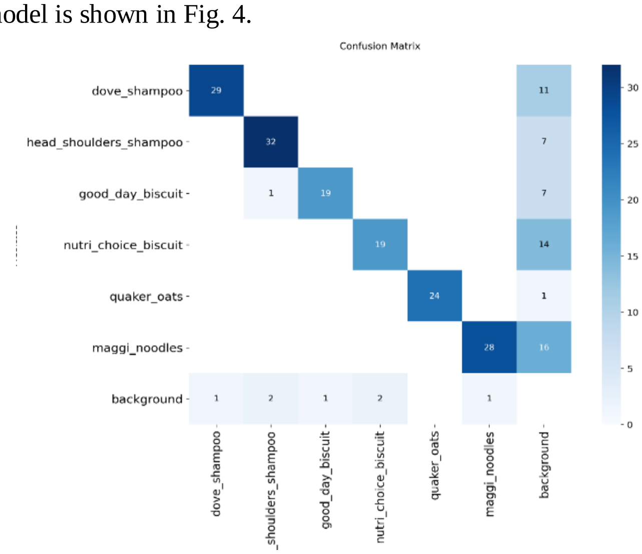


Fig. 4 Confusion Matrix

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

B. Model Comparison

For comparison, different pretrained models like YOLOv8, SSD, and EfficientDet were tested on the same dataset.

A comparison was also done using other models like SSD and EfficientDet to understand the performance difference. SSD was very fast during testing, but it was not able to detect products properly in many cases. EfficientDet was able to detect some products, but the results were not very consistent.

Compared to these models, YOLO gave more stable results for this setup. Even though it was slightly slower than SSD, the detections were more reliable. Because of this, YOLO was considered more suitable for this system.

The comparison between models is shown in Fig. 5.

image	yolo_time	yolo_det	yolo_succe	ssd_time	ssd_det	ssd_succe	eff_time	eff_det	eff_success
dove_shampoo_01.jpg	0.643875	4	1	0.058129	0	0	0.170099	1	1
dove_shampoo_02.jpg	0.404386	4	1	0.060442	0	0	0.148048	1	1
dove_shampoo_03.jpg	0.380352	2	1	0.055837	0	0	0.135838	1	1
dove_shampoo_04.jpg	0.374574	0	0	0.054172	0	0	0.159284	1	1
dove_shampoo_05.jpg	0.373283	2	1	0.04973	0	0	0.147061	1	1
dove_shampoo_06.jpg	0.383084	1	1	0.057379	0	0	0.166896	1	1
dove_shampoo_07.jpg	0.370346	1	1	0.046456	0	0	0.134276	1	1
dove_shampoo_08.jpg	0.362227	0	0	0.054088	0	0	0.131199	1	1
dove_shampoo_09.jpg	0.358001	3	1	0.052678	0	0	0.108572	1	1
dove_shampoo_10.jpg	0.398294	1	1	0.050873	0	0	0.135077	1	1

Fig. 5 Model Comparison

Table 1 Comparison of Detection Models

Model	Speed	Detection	Reliability	Overall
YOLOv8	Slow	High	Good	Best
SSD	Fast	None	Poor	Worst
EfficientDet	Medium	Medium	High	Average

C. Training Performance

The training performance of the model is shown in Fig. 6.

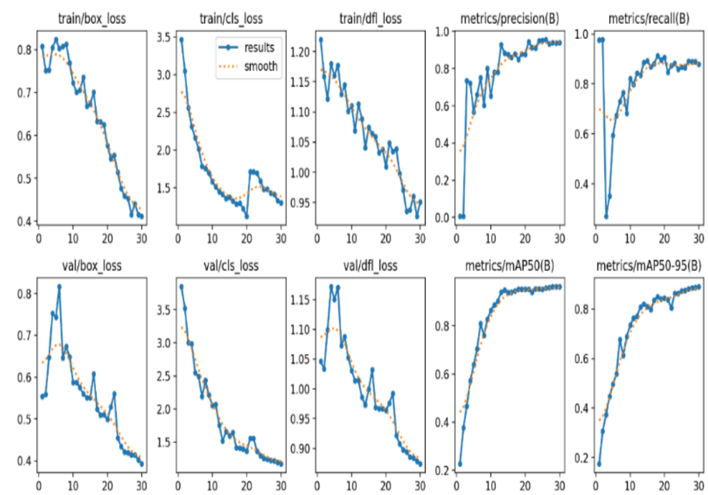


Fig. 6 Training Performance

From the training graphs, it can be seen that the loss decreases over time and the performance improves gradually. The model becomes more stable after a few epochs, and the metrics also show consistent improvement.

Overall, the system is working fine for this setup. The detection is reasonably accurate, and the full flow from detection to showing the information is working properly. There are some issues in certain conditions, but for this basic implementation, the results are fine.

V. CONCLUSION

In this project, a simple system was developed to improve the in-store shopping experience using object detection and a basic augmented display. The system is able to detect selected retail products using a YOLOv8 model and show relevant details directly on the screen. and also a small feature is added where users can ask questions about the detected product and get a response.

The system was tested with a small dataset, and the results were quite good for the selected products. The detection worked well in normal conditions, and the full flow from capturing the image to showing the information was smooth. A basic cart feature was also added to show how the system can be extended later.

There are still a few limitations. For example, in low lighting or when the image is not clear, the detection sometimes doesn't work properly. Also, right now the system supports only a limited number of products.

Overall, this work shows that combining object detection with simple user interaction can make shopping a bit easier and more interactive. It is not a full commercial system, but it still gives a good idea of how these technologies can be used together in real-world use.

VI. FUTURE ENHANCEMENT

There are a few areas where this system can be improved. One of the main things is increasing the number of products in the dataset, so that it can handle more items. Right now, it supports only a limited set.

Another area to improve is detection under different conditions. In some condition like low lighting or when the image is not very clear, the performance can reduce a bit. This can be improved by using more training data or by improving the model.

The system can also be extended by adding a proper payment feature instead of the current demo version. This would make it more useful in a real-world setup.

Along with this, the user interface can be improved to make it more easier to use. and Features like better product suggestions or recommendations can also be added.

Overall, the system can be developed further into a more complete solution with some additional improvements.

REFERENCES

[1] Javornik D., Augmented reality in retail and its impact on customer experience, 2016 https://www.sciencedirect.com/science/article/pii/S0969698916300473

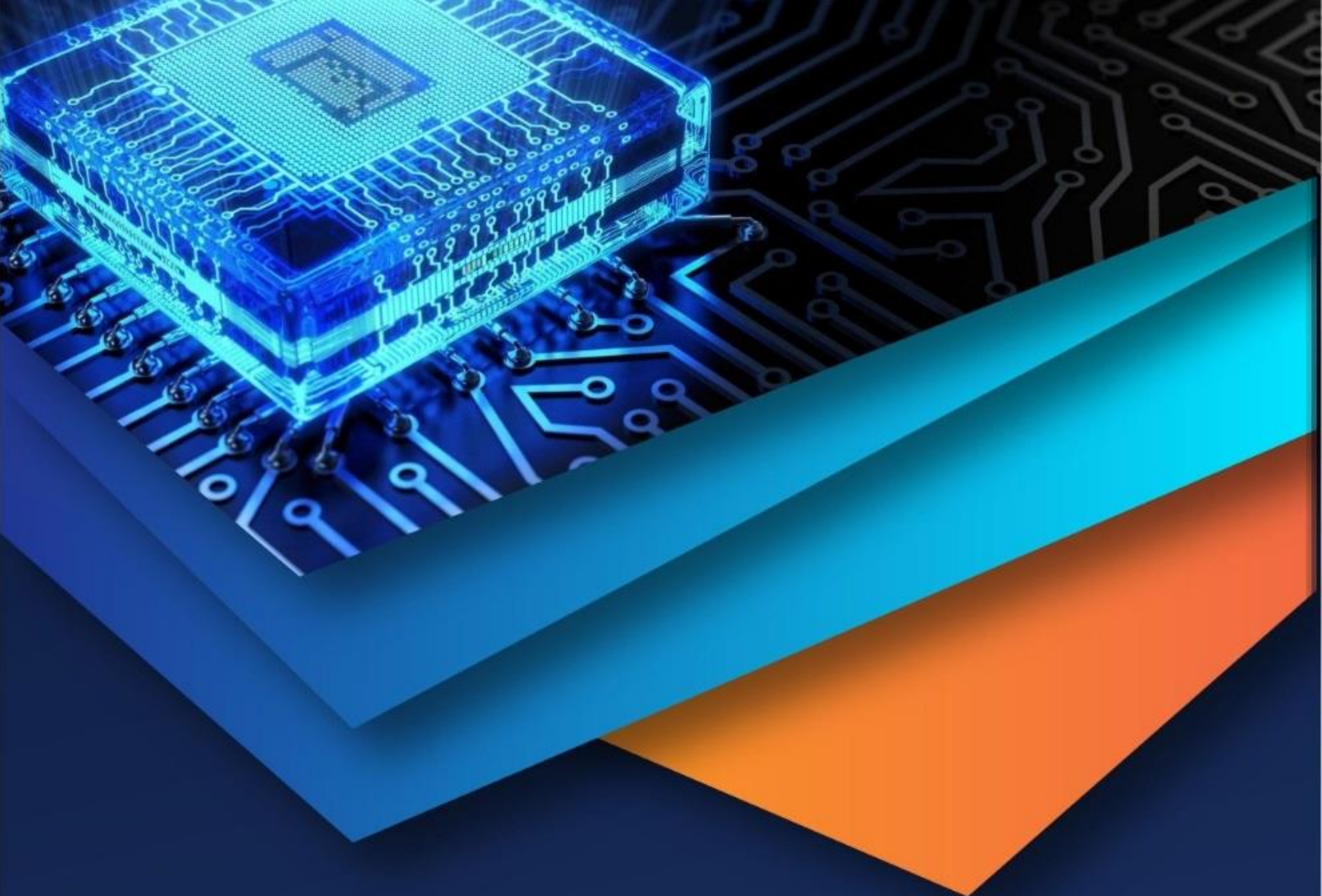
[2] Poushneh A., Vasquez-Parraga C., Impact of augmented reality on shopping experience, 2017 https://www.sciencedirect.com/ science/ article/pii/S0969698916302782

[3] Hilken J. et al., Use of augmented reality in retail environments, 2017 https://www.sciencedirect.com/science/article/pii/S0022435916300786

[4] Pantano E. et al., Smart technologies in retail, 2017 https://link.springer.com/book/10.1007/978-3-319-44263-0

[5] Redmon J. et al., YOLO: real-time object detection system, 2016 https://arxiv.org/abs/1506.02640

[6] Ollama, running LLM models locally https://ollama.com



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY