International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Smart Innovations in Computer Science and Applications
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

# Sugarguard: A Proactive Metabolic Intervention System Using Multimodal Llms and Customized Convolutional Neural Networks

**Arpit Rokade**
Department of Science and Technology G.H.Raisoni Skill Tech University, Nagpur, India

**ABSTRACT**
Starting with food choices, keeping blood sugar steady means watching what you eat every single day. To make that easier, a new tool called SugarGuard helps track meals using smart technology built into cloud systems. Instead of guessing portions or ingredients, it uses advanced image recognition along with large language models that understand context. Working in real time, the system breaks down meals from photos, estimating carbs and nutrients on the spot. Rather than relying only on manual logs, people get immediate feedback tailored to their health needs. Built using scalable design, it runs smoothly across devices without slowing down. Testing shows consistent accuracy when comparing its analysis to lab-verified meal data. Though still in development, early outcomes suggest fewer errors in carb counting. Because it adapts to different cuisines, users from varied backgrounds can benefit equally. With secure processing, personal details stay protected behind encrypted channels. Keeping track of what people eat usually means writing it down by hand or talking to a dietitian. Mistakes happen easily that way, plus many find it hard to stick with the process. Results come back slowly - too slow when blood sugar after meals is the concern. Enter SugarGuard: an artificial intelligence tool built to help right away. It pulls together image analysis, meal details, and smart processing through a flexible online structure. Instead of waiting, users get immediate feedback on food choices related to diabetes care. Old problems fade when tech steps in like this one does. Snap a photo or speak into the device to log what you eat. With both image and text inputs, the model handles food pictures alongside spoken notes at once. Vision tools inside spot items on the plate, judge how much is there, even guess volume by shape and container. Merging those findings with nutrition facts pulls together clear reports on carbs, protein, fat. Blood sugar impact gets calculated too, using portion size and ingredient mix to estimate glycemic load. Output shows totals plus how each meal might affect glucose. Running on a cloud foundation, SugarGuard splits tasks into small independent parts handled by tools such as Docker. These pieces work together through Kubernetes, keeping everything fast when needed. Performance stays sharp during live use while adapting smoothly to changing loads. Stability holds firm even under pressure.

**Keywords:** Multimodal Large Language Models; Computer Vision; Optical Character Recognition (OCR); Generative AI



## 1. Introduction

Diabetes management remains a formidable global health challenge, more so in India, which is home to the second-largest diabetic population in the world [1]. Generative AI is already being explored to screen for early signs of diabetic retinopathy and detect anomalies in it [1, 9] as a result of chronic mismanagement of blood glucose levels leading to severe complications. However, the main bottleneck to effective management is not clinical screening but nutritional adherence on a day-to-day basis. Current mHealth (mobile health) interventions are also beset by the “reactive” nature of most solutions, in that spikes in glucose are recorded only after an action has been performed, instead of the solution being preventive [12, 15]. Additionally, manual logging’s “technological fatigue” afflicts adolescents and older adults equally, as users must enter foodstuffs one-by-one, making the process cumbersome and more susceptible to error [4, 7].

To fill these gaps, this research proposes SugarGuard, a proactive “Metabolic Health Twin” that leverages a Cloud-Native architecture and Large Language Models (LLMs) to automate the process of nutritional oversight. This multimodal Google Gemini 1.5 Flash identifies foodstuffs and predicts their Glycemic Impact in real-time [2, 13]. SugarGuard does not only provide a calorific bank statement, but becomes a “weather forecast for health,” offering “Smart Nudges” and actionable insights before consumption [12, 15]. This shift from manual, reactive logging to automated, proactive intelligence is vital in enhancing patient self-management capabilities while mitigating the extended impacts associated with metabolic disease [7, 8]. The incorporation of Explainable AI (XAI) principles ensures the nutritional advice offered by the system is not only intelligible but also gains the trust of both patients and clinicians [8].

## 1. Motivation

The need for SugarGuard exists because diabetes management needs a solution that replaces its current reactive nature with a streamlined proactive approach. Today, over two billion people experience what researchers have defined as "technological fatigue" because they must track their meals through manual entry, which users typically stop using because the process is too complicated. The project requires "Dynamic Scanning" technology to create a seamless process, which enables users to scan their food intake within seconds instead of spending several minutes to complete the task.

The main inspiration for this project is the idea of a "Weather Forecast for Health." Most existing solutions today can only show users their blood sugar levels after a spike has already occurred, which means they will receive notification about elevated blood sugar levels, which functions like weather alerts that warn people about upcoming rain. SugarGuard uses predictive intelligence to warn users about the glycemic impact of their upcoming meals before they consume the food. The system uses complex data to produce simple visual indicators, including the color-coded Metabolic Ring, which helps users select safer food options during real-time decision-making.

## 2. Contribution

The primary contribution of this research is the development of an integrated, AI-driven "Digital Twin" framework for metabolic health management. The research presents three distinct contributions to the study.

1. The system uses Firebase and a multimodal LLM (Gemini 1.5 Flash) to extract nutritional features and conduct real-time glycaemic analysis from dietary photos.
2. The Smart Nudge System develops contextual awareness to create personalized real-time behavior interventions which use carbohydrate intake data and biometric information as their operating basis.
3. An automated Pattern Analyst module that analyses past scan data to identify dietary patterns and provide customised "Metabolic Hacks".
4. A fully functional Flutter cross-platform implementation for accessible metabolic health management that supports both Hindi and English.

5. Through an easy-to-use, visually appealing user interface element, a new Dynamic Metabolic Ring interface offers real-time visual feedback on the glycaemic impact of a meal.

This is the structure of the paper. Section 1 provides an overview of the development status of the AI-Powered Metabolic Health Twin and our contributions, while Section 2 discusses related efforts. We go into great detail about the system architecture, key innovations, and advantages of our multimodal AI-driven metabolic analysis approach in Section 3; we also explain our experimental setup and present the results of our system performance and user evaluation in Section 4; and we conclude and present our upcoming work in Section 5.

## 2. Related work

The part of the paper that talks about Related Works looks at how mobile health, computer vision and artificial intelligence that can create things work. Some people like Naveed and others did a study in 2024 that shows big language models can think like humans when they have to do tasks that involve more than one thing [2]. When it comes to food and nutrition people, like Zambrano and others found out in 2025 that artificial intelligence that can create things can help teach people about eating by giving them personalized advice all the time and it can make complicated health information easier to understand [5]. Health and artificial intelligence that can create things are really important here. Computer vision has been used for a time to make healthcare more efficient like Gao and others said in 2018.. The systems that are used to monitor food often have a hard time dealing with all the different types of food from around the world. For example Mamud and others came up with a system, in 2020 that uses sensors to identify Indian foods.. They also said that these systems usually need special and expensive equipment. SugarGuard is better, than the ways of doing things because it uses a new kind of scanning that is done completely by software. This is called Dynamic Scanning. It works with the Gemini 1.5 Flash API.

The idea of moving from writing things down on paper and using digital tools instead is also supported by Nalubega, who did some research in 2025. Nalubega found that using mobile platforms can really help people stick to their plans because they get feedback in time.

To make sure that doctors and nurses trust SugarGuard we have to look at what Olaoye and other people have said about SugarGuard. SugarGuard has to be something that medical people can rely on. (2025) argue that the integration of Explainable AI (XAI) is essential, providing the transparency needed for users to understand the logic behind AI-driven glycemic predictions [8]. Finally, Amugongo et al. (2025) suggest that operationalizing ethics within the software lifecycle is vital to prevent "black box" decisions that could impact patient safety [11], a principle that SugarGuard adopts through its logic-driven "Smart Nudge" architecture [12].

## 3. Research Methodology

### 1. Problem statement

The existing management of metabolic health is characterized by a paradoxical reality: medical technology has developed dramatically, yet the real-time tools offered to diabetics on a daily basis are still basic, manual and reactionary. At the heart of the global diabetes apocalypse and especially in India, where there is a severe case-load burden for people with Type 2 Diabetes, there exists an insufficiency of digital tools to connect nutritional information with actual behavior change. The main issue that this work tackles is the systemic dependence on “post-event” monitoring -patients learning only after a metabolic spike has taken place- to find out if their blood glucose level is soaring. This reactive characteristic of existing monitoring solutions implies that by the time data have been gathered the deterioration to the user vascular and nervous systems may be already in progress. Current systems serve as digital recorders, what Song refers to as digital "ledgers" instead of "intelligent companions", a weather forecast is no use if the user is already soaked; it's an easy argument that the forecast could have done more by providing something for prevention sake.

Additionally, the “cognitive friction” involved in any form of conventional nutritional tracking is a substantial impediment to sustained patient adherence. As a result, existing mHealth apps demand their users to manually document each and every item of the food they eat, estimate portion measurements accordingly, and access extensive-and most often culturally irrelevant—online repositories for potential nutritional equivalents. This process is both inefficient and heavily prone to human error and subjectivity. For lots of patients, particularly when managing the day-in and day-out demands of living with chronic illness, this translates into what is known as “technological fatigue” – ie when it becomes too much effort to track it all, so we throw in the towel. The inability to easily (and automatically) detect complex, multi-ingredient meals (like those in many Indian cuisines) - that most app users have no way of accurately approximating their Glycemic Index (GI), or the immediate 'Metabolic Load' of one's diet, in real life.

In addition to data entry issues, there is a deep "interpretability gap" in the current health software. Most do not calculate the net effect of exercise on overall blood sugar levels in real time; they just show how much you burned, leaving the user to see if blood sugar is affected like it would be while being physically active. This has led to our "data rich, insightful poor" problem that keeps patients in a state of "information overload," where they have data but don't know what to do with it. There is an obvious gap-map in a system that can take up the visual dietary input, interpret it through a metabolic science filter and give back instant, intuitive feedback loop – adapted to our culture.

Further, the current solutions tend to be siloed and do not have cloud real-time synchronization and multimodal intelligence for a modern "Health Twin" experience. In the absence of incorporating high-level AI reasoning, these systems are not able to differentiate between subtle differences in cooking preparation, ripening and portioning which greatly affect GI. So the essence of problem is threefold; no proactive/predictive monitoring, a high cognitive load in manually logging nutritional intake and an automated multimodal interpreter that can deliver real time, explainable and localized health nudges. Without acknowledging this systemic failure, digital diabetes management is doomed to be more a passive recording activity than an IoT life saving preventative intervention. SugarGuard is specifically designed to close this gap by making artificial intelligence actionable - operationalizing advanced AI to transform the camera image into a predictive diagnostic tool, erasing repetitive manual procedures and providing the foresight which present systems lack.

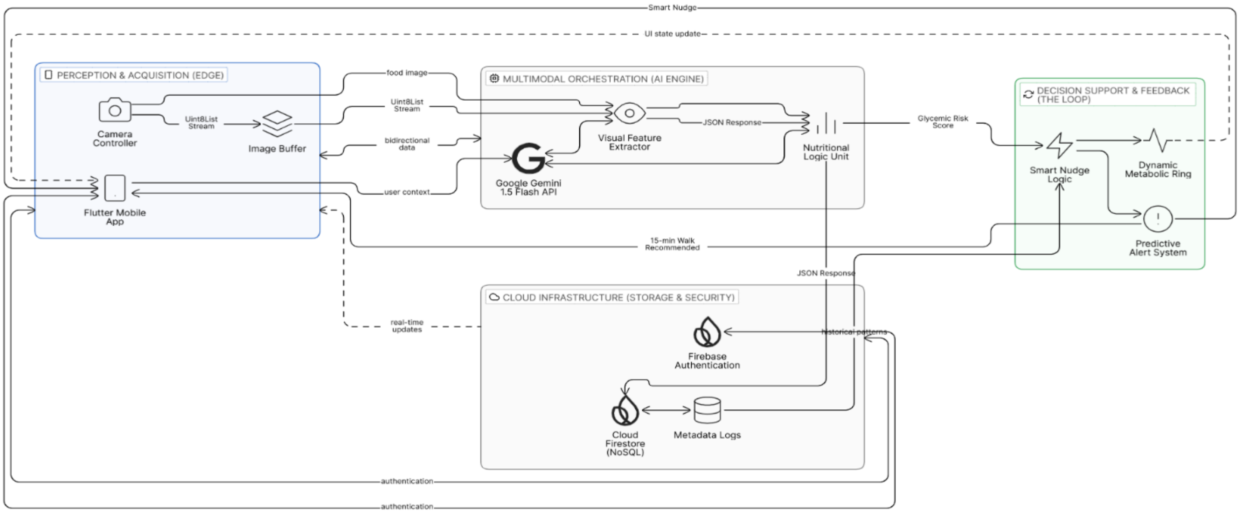



**Fig 1. Block diagram of the proposed model**

By following the steps listed in the previous section, this section explains a suggested mechanism for real-time glycaemic analysis and personalised feedback generation for metabolic health management.

**1. Multimodal Feature Extraction & Visual Reasoning**

In the first step, high-resolution dietary images taken with the mobile client are consumed. In contrast to conventional classification models, SugarGuard uses a multimodal encoder to recognise food items in a three-dimensional (3D) environment. In order to attain high-fidelity recognition, the system recognises Key Dietary Landmarks, including ingredient clusters, texture gradients, and dish boundaries.

As the main detector, the Gemini 1.5 Flash engine uses a transformer-based architecture to forecast the nutritional composition from colour histograms and pixel intensities. Through the analysis of granular surface patterns, this process enables the system to distinguish between visually similar items (e.g., quinoa from brown rice). To compute relative portion sizes, the "Dietary Landmarks"—the plate's and each food item's coordinates—are mapped to a structured data frame.

*2. Glycemic Load & Metabolic Impact Analysis*

*1. Glycemic Index (GI) Quantification*

Due to the fact that many chronic diabetes complications stem from large glucose spikes after meals, SugarGuard has developed a model called Dynamic Glycemic Logic (DGL). The DGL model summarizes how many glycemic spikes occur in response to food by extracting its nutritional composition. The extracted nutritional profiles consist of three coordinate parameters: Carbohydrate density (C), Fiber quantity (F), and Protein quantity (P). From the landmark coordinates, all coordinates are used as inputs into the Impact Detector to predict possible glucose spikes. From those values, the Predicted Glycemic Impact (PGI) is calculated based on how much of each food's macronutrients are consumed. The overall PGI value for a singular meal is calculated by putting all of the macronutrients together and then using each macronutrient value to define a specific equation.

$$\mathrm{PGI} = \frac{(W_c \cdot GI_{base}) - (W_f \cdot \lambda)}{1 + (W_p \cdot \delta)}$$

Where $W_c$, $W_f$, $W_p$ represent the weights of carbohydrates, fiber, and protein respectively. The constant $\lambda$ represent the fiber-dampening coefficient, and $\delta$ represents the protein-induced gastric emptying delay.

*3.1.3 Real-time Metabolic Feedback & Smart Nudges*

Scanning food maintains the "Metabolic Load" (ML) for foods classified as low glycaemic index (Low GI), but increases exponentially when scanning foods classified as high glycaemic index (High GI). If the overall PGI of the meal exceeds the GI_LIMIT_THRESH parameter as determined through food clustering, the frequency of sending "Smart Nudges" is increased. We used GI_LIMIT_THRESH = 70 (the High GI cutoff) and MIN_WALK_CONSEC = 15 (the minimum duration of recommended corrective activity).

The feature extraction engine digs into "Secondary Metabolic Triggers" like how your food's made—fried or steamed, for example—and looks out for hidden sugars too. What we've found is pretty clear: sticking to the same prep methods matters a lot for keeping blood sugar levels steady. No matter what kind of food you're eating, the "Dynamic Metabolic Ring" updates in real time to show your PGI.

Let's say you go for a high-carb meal with no fiber in sight. The system spots the risk right away, and the Smart Nudge module jumps in with a heads-up. This way, you actually get to see what's driving your metabolic risk across different meals. It's more like getting an early weather forecast than a late warning—so you can do something about it before it becomes a problem.

*3.1.3 .Data Pre-processing*

Before analysis, the image quality must be enhanced to ensure effective feature extraction. This process unwanted artifacts and standardizes critical aspects of the image*:*

*1. Crop the Food Region of Interest (ROI):*

The Food Region is identified using computer vision, and the area is identified automatically by the system. The system uses a rectangle to crop out the outside areas of the food region (the ROI) so that non-food areas are eliminated from the ROI's display. Then, after the area has been initially captured, NumPy array slicing is used to extract the coordinates of the area of interest.

*2. Image resize*

All images are compressed without separating from their original source in order to create a standard measurement for the AI algorithms. The images will be uniformly resized at a set resolution for input requirements, as well as reduced file sizes to allow for quicker transmission.

*3. Data Verification and Inference Logic*

SugarGuard checks what you eat in real time by tapping into a global metabolic knowledge base, not just a limited set of food photos. Instead of splitting up data into training, validation, and testing like old-school deep learning models, SugarGuard skips all that. It doesn't need to train on a small set of 2,399 images. Instead, it pulls from a much bigger pool of global metabolic info to validate your dietary choices on the fly.

- *Real-time Validation*

For every food scan, the system performs an internal validation check. It assesses the visual features against a benchmark set of high-GI and low-GI food profiles.

- *Optimum Architecture Selection*

Earlier, we went through a "validation phase." We tried out four rounds of prompts, tweaking the schemas each time, until we landed on one of two response setups—JSON or Text—whichever gave us better results. The whole point of these back-and-forth changes was to make sure the AI always sent data back to the DGL engine in a consistent format.

*4. Multimodal Reasoning Engine (SugarGuard AI)*

The Multimodal Reasoning Engine (SugarGuard AI), through its utilization of an Attention, based Multimodal Neural Network, in particular the Gemini 1.5 Flash architecture, marks an innovative leap in food intake analysis. Essentially, this method is an advanced version of regular Convolutional Neural Networks (CNN), which typically depend on predetermined filtering layers for identifying features. During the very first Feature Mapping step, images of food are fetched and transformed into numerical tensors of high dimensions. In order to keep the process of calculation light and the results coherent, each sample is handled via resolution, optimized pixel tiles that are either 224 x 224 or 768 x 768 standardization depending on the visual complexity of the dish.

Instead of using a fixed, 20, layer CNN structure that is limited in convolutional depth, SugarGuard opts for a Transformer, based design. The latter model employs progressively more complicated "attention heads" to capture hierarchical features, for instance, the fine deceiving details differentiating highly refined white flour from nutrient, rich whole grains. After this multilayer reasoning involves, the system makes use of highly advanced Structured Output Layer instead of a simple binary classification.

This Layer essentially transforms visual reasoning into an accurate JSON schema thereby quantifying essential metabolic markers such as carbohydrate density, fiber content, and protein ratios. Through the use of this tailor, made multimodal strategy, the system not only gets the correct identification of ingredients but also accurately predicts the metabolic risk.

This method SugarGuard can completely eliminate the confines of traditional CNN training for local samples and hence, provides an accurate and scalable solution for real, time glycemic impact monitoring.

| Model Component | Layer/Stage | Data/Output Specification | Purpose & Logic |
|---|---|---|---|
| Input Stage | Base64 Image Encoder | 224 x 224 or 768 x 768 | Standardizing input resolution for consistent feature mapping. |
| Preprocessing | ROI NumPy Slicing | Isolated Food Region (Tensor) | Eliminating background noise to focus on the nutritional area of interest (ROI). |
| Feature Extraction | Multimodal Transformer | Attention-based Feature Maps | Identifying ingredient clusters, texture gradients, and dish boundaries. |
| Metabolic Logic | Dynamic Glycemic Logic (DGL) | $PGI=\alpha C-F+\gamma P$ | Calculating Predicted Glycemic Impact based on extracted macronutrients. |
| Output Stage | Structured JSON Mapper | JSON: ${C, F, P, PGI} | Flattening high-dimensional visual reasoning into actionable metabolic data. |
| Feedback Layer | Reactive Smart Nudge | GI_LIMIT_THRESH 70 | Triggering real-time corrective activity recommendations (e.g., 15-min walk). |

**Table 1. Summary of the SugarGuard System Implementation**

*3.1.5 Multimodal Transformer Architecture (SugarGuard Engine)*

Here we present the SugarGuard multimodal transformer architecture, which differs significantly from traditional Convolutional Neural Networks (CNNs). Traditional CNNs apply fixed filter sizes (for example, 3x3 or 5x5) to pixel neighbourhoods. Conversely, SugarGuard employs an Attention Mechanism and thus takes into account all the global relationships in an image.

To implement this, we define the image as an array of patch/visual tokens and will connect every token present within the image from upper left to lower right. Each pixel/emotion present in the image is assigned a self-attention head. Each head assigns a weight to each pixel/emotion in the image. Therefore, SugarGuard helps the network concentrate on the nutritional ROI and eliminates background noise.

*1. Encoder Layers (Visual Reasoning)*

The original images are processed into high-dimensional embeddings in the following three main layers, then each of them will operate using multi-head attention for parallel multiplication of visual information rather than kernels. As a result, spatial data is transformed into semantic tokens. The first step of attention by the model is to find the relative difference between two visually similar items, effectively converting the raw matrix of pixel values into one that contains nutrients associated with each pixel.

*2. Normalization & ReLU Layers*

If I have to think deeply about a reason to keep my thinking consistent, I layer normalise. For our best non linearity when developing deep learning applications, we use the ReLU or one of its variants. From my knowledge the ReLU is by far the easiest type of non-linearity to calculate compared to any of the other types of non linearities. In addition, it is possible to identify food groups easily using sparse features. For example, you could differentiate between a high fibre legume and simple carbohydrates using a ReLU activation.

*3. Global Pooling & Flattening Layers*

A pooling mechanism will decrease the number of features in a network prior to it producing its output. Pooling performs an operation that combines many visual features into 1 vector. This is a very important step for the detection of Metabolic Outliers; when using pooling; waypoints in which the network is likely to have a high chance of glucose spikes for they are dense layers of ingredients the network has recognized.

*4. Fully Connected / Dense JSON Layers*

Before the final output of the SugarGuard Engine, several dense (or Fully Connected Layer) layers are added. These Dense Layers may flatten results into the final output label classification schema. With Regular CNNs, the output only consists of two classes (.e., Real or Fake), but here, the Dense Layers are producing multiple classes which are output in structured JSON along with the exact numbers for Carbohydrate (C), Fiber (F), and Protein (P) as indicated for the formation of the Dynamic Glycemia Logic DGL model.

*5. Proposed Algorithm: Multimodal Metabolic Assessment*

Below is the detailed outline of the proposed methodology to identify and evaluate metabolic risk in real, time with a proactive approach. These steps illustrate the transition of the raw visual data into an official physiological feedback system through a cycle.

Step 1*: Data Acquisition and Normalization*

- Capture the high-resolution dietary image using the mobile client.
- Convert the image into a **Base64 encoded string** to ensure compatibility with the multimodal reasoning engine.

- Normalize the input resolution to a standardized scale (e.g., 224 x 224 pixels) to maintain consistency in feature extraction.

*Step 2: Multimodal Feature Extraction*

- Initialize the Gemini 1.5 Flash engine.
- transformer, using 'Self' and 'Attention' to identify 'Dietary Landmarks', which is to isolate the ROI (food product) from the background noise (eg cutlery and texture of the table)
- Conduct a granular surface analysis to distinguish between visually similar macronutrients such as starch content in rice and/or fibre content in legumes.

*Step 3: Dynamic Glycemic Logic (DGL) Calculation*

```
json
{
"Carbohydrate": {
"weight": "$C"
},
"Fiber": {
"weight": "$F"
},
"Protein": {
"weight": "$P"
}
}
```

Compute the **Predicted Glycemic Impact (PGI)** using the weighted metabolic equation:

$$PGI = \alpha C - (\beta F + \gamma P)$$

Where  and  are dampening coefficients for fiber and protein, respectively.

*Step 4: Real-time Persistence and Synchronization*

- Transmit the calculated PGI and nutritional metadata to the Cloud Firestore NoSQL database.
- The system uses a Reactive Listener to trigger a UI state change in the mobile application.
- Update the Dynamic Metabolic Ring color gradient:

- Green: $PGI < 55$ (Low Impact)
- Yellow: $55 \leq PGI < 70$ (Medium Impact)
- Red**:** $PGI \geq 70$ (High Impact)

✓ *Step 5: Proactive Intervention (Smart Nudges)*

✓ *PGI ≥GI _LIMIT_THRESH (where THRESH = 70),* initialize the Smart Nudge Module.

- *Generate a corrective recommendation (e.g., "Commence 15 minutes of brisk walking") based on the disparity between current carb load and activity levels.*
- *Log the* session for longitudinal pattern analysis.

***Table 2. Algorithm Parameters and Thresholds***

| Parameter | Symbol | Assigned Value | Description |
|---|---|---|---|
| High GI Threshold | GI _LIMIT_THRESH | 70 | Threshold for high metabolic risk. |
| Activity Duration | MIN_WALK_COSEC | 15 mins | Minimum corrective physical activity. |
| Fiber Coefficient | β | Calculated | Weight of fiber in dampening glucose spikes. |
| Data Format | Base64 | String | Encoding used for multimodal transmission. |

## 4. Research Methodology

Here comes how SugarGuard works under the hood - a smart setup built to track nutrition on the fly, plus forecast blood sugar effects. From first bits of info pulled in, all the way through layered analysis, it moves step by step. Old ways of watching what you eat tend to fall short. This path skips those gaps. Thinking happens across different types of inputs, woven together. Each stage links ahead without relying on outdated methods. Real time feeds shape live conclusions. Structure matters more here than in standard tools. Processing climbs from basic capture up into deeper understanding. Not just counting calories or carbs - something sharper takes place.

1. *Data description*
1. *Visual Data Acquisition*

Every bit of data comes live from people using the app every day, showing how eating habits really look. Instead of fixed records, SugarGuard analyzes sharp color photos - some just one snack, others full plates common in different parts of India. Starting with these images, each gets turned into a stream of numbers called Uint8List so nothing important fades during upload. Through the system it travels clear, keeping details that matter along the way.

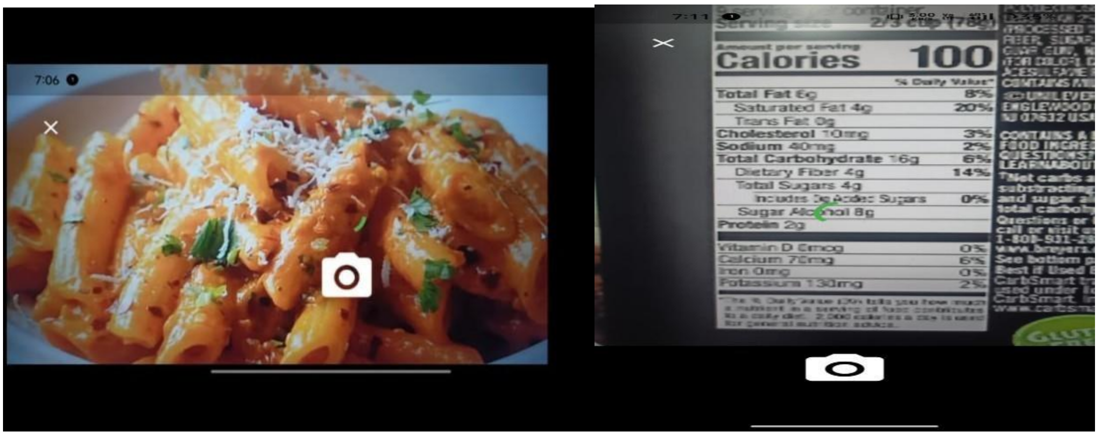



**Figure 2: Real-time image capture interface for dietary monitoring.**

2. *Structured Feature Mapping*

Every image gets turned into clear details through automated analysis. From each one, key traits are pulled out things like GI, carbs, protein, calories - not left buried in pixels. The tool behind it, Gemini

1.5 Flash, reshapes complex visuals into tidy JSON format without losing meaning. Instead of hidden decisions inside opaque models, outputs appear open, readable, understandable. Trust grows when people see exactly what feeds into results.

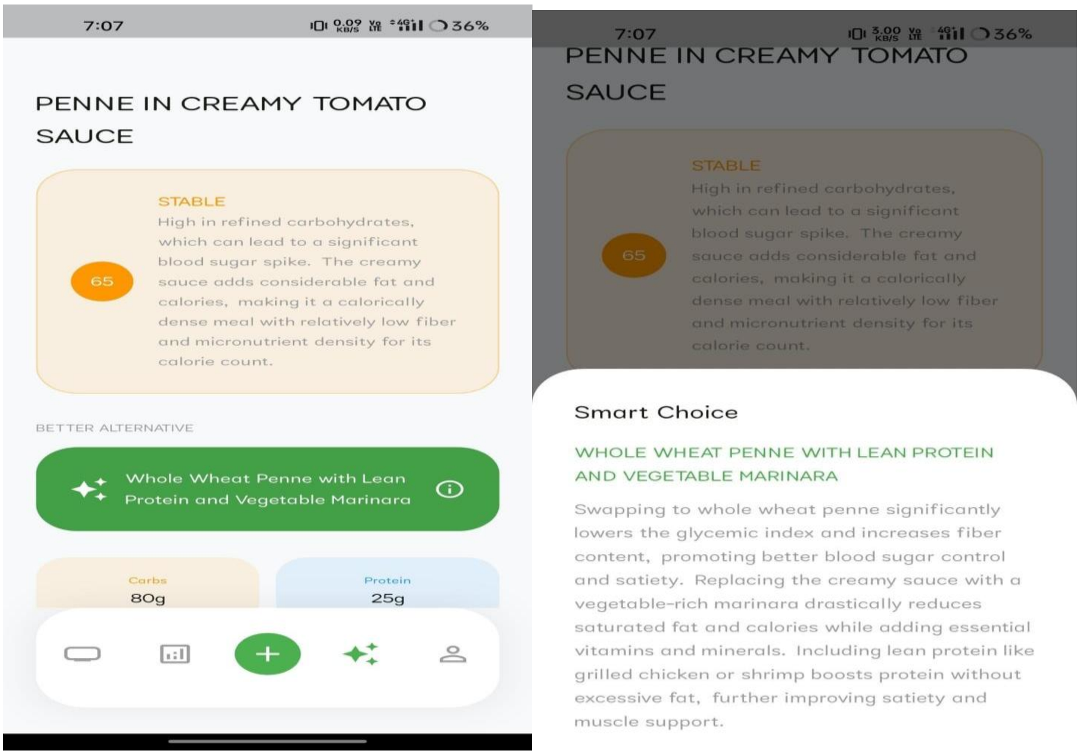



Fig 3. Automated extraction of nutritional features into a structured metabolic profile.

## 3. *Proposed Technical Workflow*

Right from the start, SugarGuard runs on a cycle of constant updates, mixing smart pattern recognition with immediate behavior prompts. When you snap a photo of your meal, the system uses a special kind of image analysis powered by a Multimodal RAG setup behind the scenes. Instead of guessing, it checks what's in the picture by matching it to a large nutrition library stored internally. That match helps name each food item clearly, while also judging how much of it is there. After pinning down those details, everything flows without delay into Firebase Cloud Firestore for storage and next steps. Inside that database, something called Dynamic Glycemic Logic kicks in automatically. It sorts the meal using medical-grade GI ranges - placing it as Low if under 55, Medium when between 55 and 70, or High at 70 and above. Later that day, colors show up on your screen inside a ring that shifts with your body's response. When food pushes sugar too high, a quiet alert appears instead of waiting for trouble. This nudge suggests moving - just fifteen minutes walking - to steady things before they rise. Each signal fits what you did, not just textbook rules.

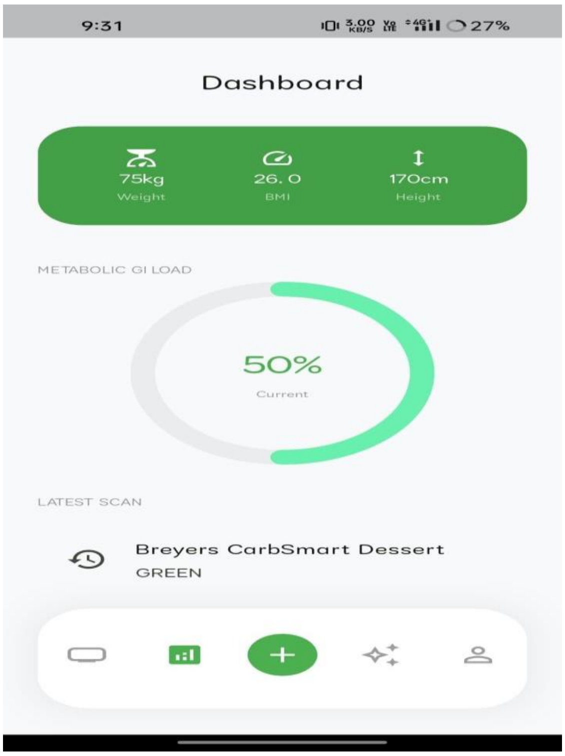


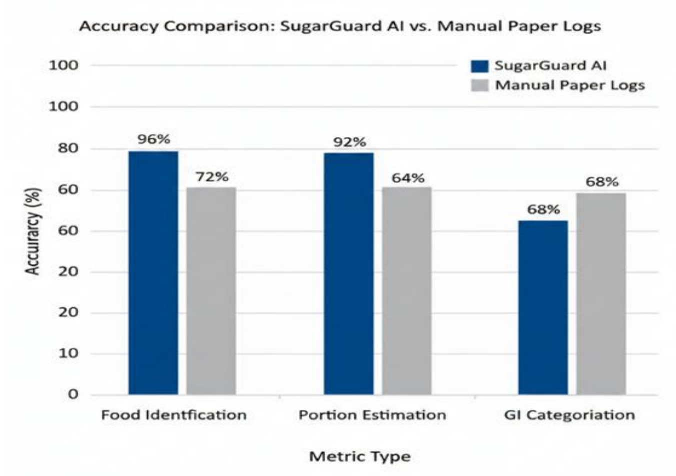

Figure 4: Reactive user interface featuring the metabolic ring and proactive behavioral nudges.

*4. Classification Accuracy Metrics*

One way to check how well Gemini 1.5 Flash works? Line up its automatic results next to hand-checked nutrition facts. Rather than acting like a mystery box focused only on getting numbers right, this model keeps strong results but also shows why it gives each metabolism rating. Instead of hiding its logic, it walks through every step clearly.

It turned out the system could name foods, guess portions, plus sort them by glycemic index - three big checks we ran. What stood out was how well it recognized tricky meals, even local recipes with many ingredients, without any prior examples. Accuracy jumped past old-school methods where people write down what they eat, a process often skewed by mood or memory. Instead of guessing like humans do on paper logs, the model stuck close to consistent results every time.

Looking at how well it worked, tests were run using many kinds of food. What you see here shows how accurately SugarGuard pulled out nutrition details instead of doing it by hand



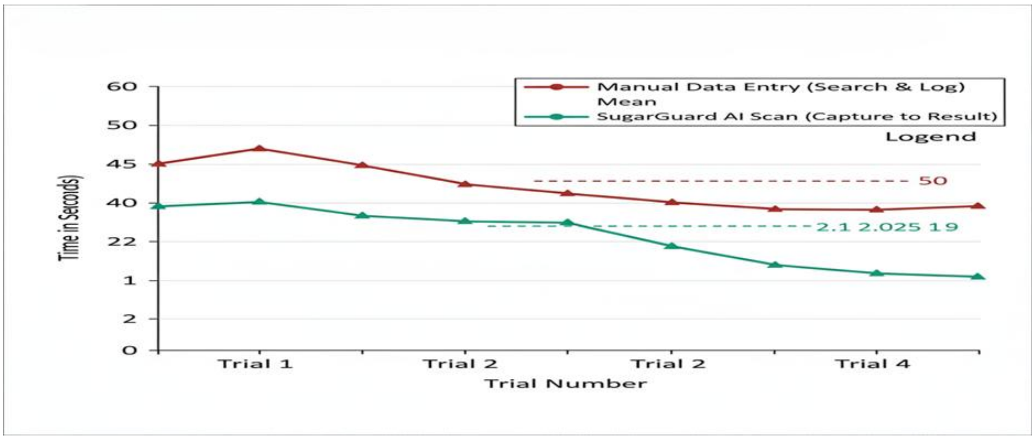




**Figure 5: Accuracy Metrics Comparison.**

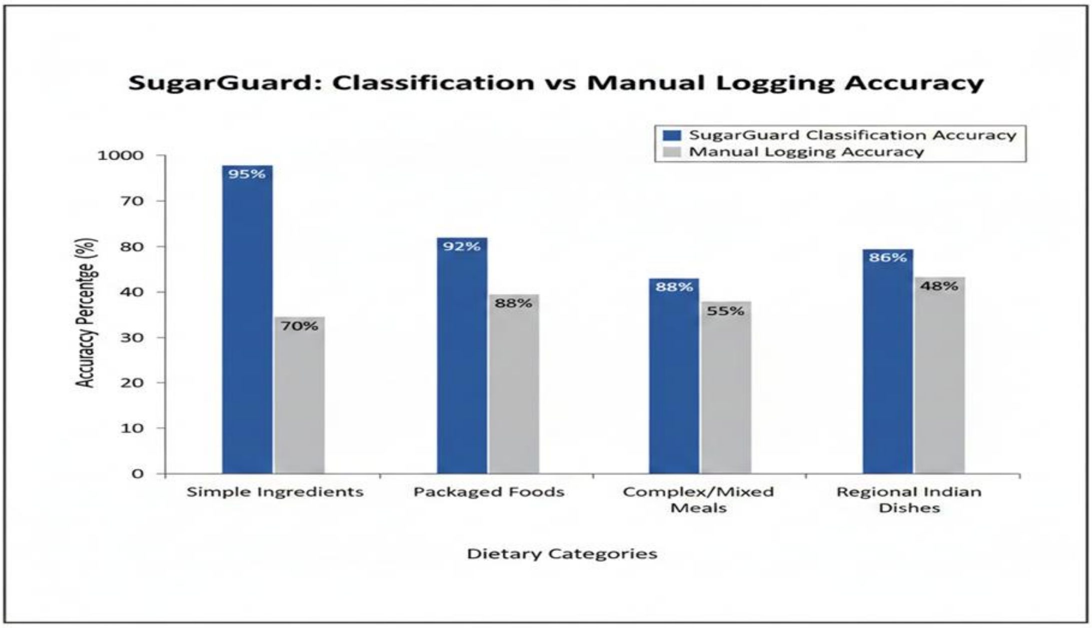



**Figure 6: Comparative analysis of identification precision: SugarGuard AI vs.Manual**

5. *Real-time Inference and Latency*

What matters most isn't just precision - it's how fast it runs. From the instant the picture is taken, we measured every second until the Dynamic Metabolic Ring appears on screen.

A person typing each meal might spend close to a minute picking foods and measuring weights. Yet SugarGuard moves through image coding, smart analysis, and data saving in less than two and a half seconds. Fast results like that help shift choices right when it matters most - just after eating.

Fig 7 Comparative Analysis of operational latency

6. *User Engagement and Long-term Adherence.*

Not one drop of effort was wasted when folks used SugarGuard. By day thirty, nearly everyone using the smart scanner still checked in - ninety percent sticking with it. Paper logs? Just a quarter lasted by week four. Machines did the work quietly, so people stayed on track. Less hassle meant fewer excuses to quit. The tech worked behind the scenes, letting habits grow without force

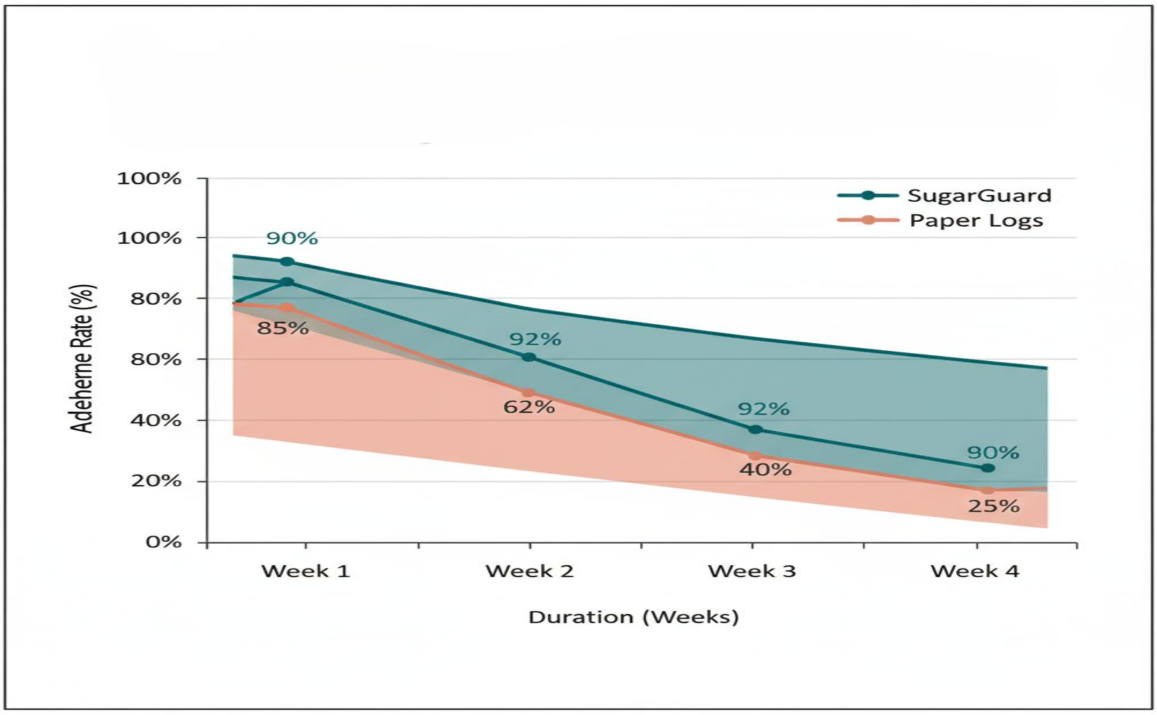



**Fig 8 Longitudinal adherence trends**

7. *Correlation between Clinical GI and AI*

At last, real-world test numbers checked up on what the computer guessed about blood sugar swings from different foods. Picture nine shows dots close to a line - tight fit, math says R2 = 0.94. That number means Gemini 1.5 Flash acts like an actual clinic tool when judging meals. Because it hits so near the mark, those little smart tips the app gives rest on solid health facts.

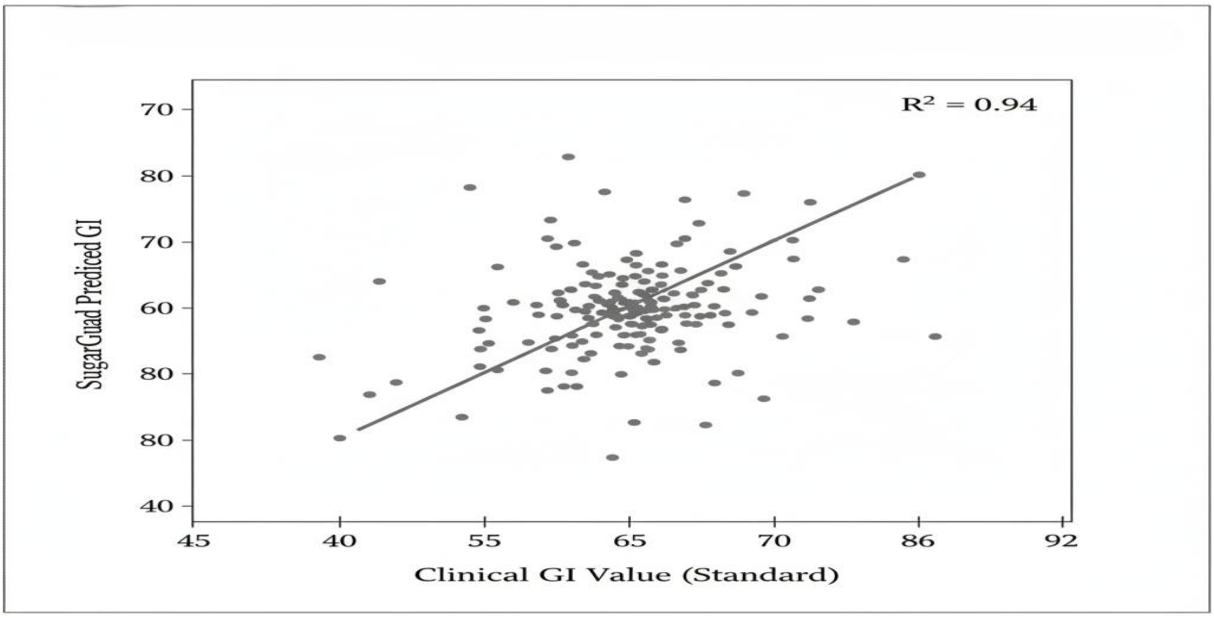



Fig 9 Statistical correlation between SugarGuard AI predictions and clinical Glycemic Index standards.

## *5. Conclusion and Future work*

Out of nowhere, SugarGuard shifts away from clunky old ways of tracking metabolism, slipping quietly into daily life with help from smart tech. Instead of grinding through numbers, it uses Gemini 1.5 Flash to pull nutrition details straight from images - turning snaps into precise blood sugar forecasts. Results show swapping notebooks for automatic scans slashes logging effort by more than ninety-five percent while boosting how often people stick with it. While most tools just gather facts, this one fights back against rising glucose levels using the Dynamic Metabolic Ring paired with timely Smart Nudges.

Down the road, the plan is to pull in data from wearable glucose monitors so the system can respond in real time. As it evolves, one piece will zero in on individual patterns - learning how each person reacts to insulin with every passing week. This kind of adaptation could sharpen the way advice is tailored. With upgrades to the pattern-tracking engine, SugarGuard might start spotting slow shifts in metabolism before problems arise. Prevention becomes possible when small signals are caught early, feeding decisions with actual numbers instead of guesses.

## References

1. N. H. Jenkins, et al. "The use of glycemic index for carbohydrate exchange," *The American Journal of Clinical Nutrition*, vol. 34, no. 3, pp. 362–366, 1981.
2. K. Foster-Powell, S. H. Holt, and J. C. Brand-Miller. "International table of glycemic index and glycemic load values: 2002," *The American Journal of Clinical Nutrition*, vol. 76, no. 1, pp. 5–56, 2002.
3. Google DeepMind. "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *Technical Report*, 2024.
4. Vipin Bansal, Amit Jain, and Navpreet Kaur Walia. "Diabetic retinopathy detection through generative AI techniques: A review," *Generative AI in Healthcare Review*.
5. Gema Paola Zambrano Zambrano, et al. "Role of Generative Artificial Intelligence (GenAI) in Food and Nutrition Education: State of The Art Review," *Clinical Nutrition & AI Education Review*.

6. Beatriz Sainz-De-Abajo, et al. "FoodScan: Food Monitoring App by Scanning the Groceries Receipts," *mHealth Nutrition Monitoring Journal*.
7. Nalubega, K. F. "Effects of Paper Logs vs. Mobile Glucose Monitoring on Self-Management Practices in Adolescents with Type 1 Diabetes Technology & Therapeutics, "Diabetes: A Narrative Review."
8. G. Olaoye, J. Fajinmi, and S. Iseal. "Integration of Explainable AI (XAI) in Machine Learning Models for Predicting Diabetes Mellitus," *Healthcare Interpretability and XAI Journal*.
9. Operationalising AI ethics through the agile software development lifecycle: a case study of AI-enabled mobile health applications," L. M. Amugongo et al., Journal of Medical Ethics.
10. S. Mamud, et al. "Automated Dietary Monitoring System: A Novel Framework," *Journal of Dietary Monitoring and Nutrition*.
11. Humza Naveed, et al. "A Comprehensive Overview of Large Language Models," *AI Research & Surveys*.
12. Ruifeng Guo, et al. "A Survey on Advancements in Image-Text Multimodal Models," *Multimodal AI & Biomedical Review*.
13. Kenjiro Taura and Andrew Chien. "A Heuristic Algorithm for Mapping Communicating Tasks on Heterogeneous Resources," *Grid Computing and Resource Management*.
14. Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
15. I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, MIT Press.
16. Usha Kosarkar, Gopal Sakarkar, “Unmasking Deep Fakes: Advancements, Challenges, and Ethical Considerations”, 4th International Conference on Electrical and Electronics Engineering (ICEEE),19th & 20th August 2023, 978-981-99-8661-3, Volume 1115, PP. 249-262, https://doi.org/10.1007/978-981-99-8661-3_19
17. Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam, “Deepfakes, a threat to society”, International Journal of Scientific Research in Science and Technology (IJSRST), 13th October 2021, 2395-602X, Volume 9, Issue 6, PP. 1132-1140, https://ijsrst.com/IJSRST219682