INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING, MANAGEMENT & APPLIED SCIENCE (IJLTEMAS)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

# Acusense: A Smart Anemia Dectecting Device

**Abhilash CM, Arghya T, Karthik OS, Steve Gigi, Venus V**

**Sahrdaya College of Engineering and Technology**

**DOI: https://doi.org/10.51583/IJLTEMAS.2025.1410000111**

**Abstract:** Over 1.6 billion people worldwide suffer from anaemia, which is still a major public health concern. Low-resource environments bear a disproportionately high burden of this condition. Although conventional blood tests necessitate invasive procedures, trained personnel, and laboratory infrastructure, early detection is essential to preventing serious clinical outcomes. Using a Raspberry Pi platform and Internet of Things (IoT)-enabled remote monitoring, this work introduces AcuSense, a portable, non-invasive anaemia detection system that combines photoplethysmography (PPG) and conjunctiva image analysis. The system uses deep learning-based conjunctiva imaging and real-time PPG signal acquisition to provide highly accurate haemoglobin level estimates. IoT integration improves accessibility for underserved and rural communities by enabling automated data storage, cloud analytics, and remote healthcare provider alerts. Strong correlation between experimental evaluation and conventional haemoglobin measurement techniques highlights AcuSense's potential as an affordable, quick, and non-invasive diagnostic tool.

**Keywords:** Raspberry Pi, IoT, photoplethysmography, conjunctiva imaging, non-invasive anaemia detection, biomedical signal processing, and portable diagnostics.

**Meaning:** AcuSense facilitates early anaemia screening by doing away with blood draws and complicated lab equipment, allowing for prompt interventions and closing the diagnostic gap in community-level healthcare.

## I. Introduction

The Global and Indian Context of Anaemia

Anaemia is a significant nutritional and clinical condition marked by a diminished concentration of haemoglobin in the blood, resulting in a decreased capacity for oxygen transport. Anaemia impacts over 1.6 billion people worldwide, primarily affecting women, children, and the elderly. In India, the numbers are shockingly high: about 53% of women of childbearing age and 58% of children under five are affected. This is due to a mix of poor nutrition, long-term infections, and social and economic factors. If you don't treat anaemia, it can make you tired, hurt your child's cognitive development, lead to bad pregnancy outcomes, and make you more likely to get sick.

Limitations of Current Diagnostic Methods

Conventional anaemia detection uses venipuncture to obtain a complete blood count (CBC), which necessitates laboratory equipment, skilled technicians, and a lengthy turnaround time for results. Although there are hemoglobinometer-based point-of-care tools, they are still too expensive, invasive, or only partially portable for community-level screening. Frequent monitoring is also impractical due to the discomfort and infection risk associated with repeated blood draws.

Need for Low-Cost, Portable, Non-Invasive Devices

The growing demand for preventive healthcare at a community level necessitates low-cost, portable, non-invasive diagnostic alternatives. These solutions support early intervention, ongoing surveillance, and connection to telemedicine applications, especially in rural and underserved populations. A non-invasive method of diagnosis leads to reduced discomfort to the patient, minimizes biohazard potential, and also allows for chronic monitoring.

Objective and Scope of AcuSense

AcuSense aims to create a non-invasive, portable anemia detection platform that incorporates the analysis of PPG based physiological signals along with the processing of conjunctiva images that is integrated on a Raspberry Pi platform with IoT capabilities. The goal of the system is to: Estimate hemoglobin levels without the need for blood sampling in a reliable manner. Create and enable a portable, low-cost solution for point of care access. Enable real-time data transmission, remote monitoring, and cloud analytics.

Key Contributions and Novelty Statement

AcuSense is unique because it uses a hybrid sensing technology:

Dual-modality detection: This enables the simultaneous assessment of continuous photo-plethysmography (PPG) waveforms with conjunctiva color imaging for strong anemia assessment.

Raspberry Pi: AcuSense is a stand-alone, inexpensive and low power platform executing signal acquisition, shimmer analysis and real time imagery.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

IoT capability: AcuSense can create a link with care providers from afar to provide alert signals for caregivers, will allow for cloud data storage and will be able to back up data securely.

AI imagery analysis: Direct, deep-learning image algorithms are incorporated to analyze posterior of conjunctiva and enhance estimation of hemoglobin.

AcuSense integrates proven biomedical instruments including image processing, photoplethysmography (PPG) and IoT for me assay non-invasive anemia monitoring that has focused on real-world consumer experience and potential market involvement.

**System Architecture**

The architecture of the AcuSense system consists of both hardware and software components to facilitate seamless non-invasive anemia detection coupled with an IOT monitoring mechanism.

Hardware Components

Raspberry Pi 4 Model B: Functions as a core processing unit for signal acquisition, image processing, and IOT communication.

PPG Sensor (e.g., MAX30100 / MAX30102): It is for acquiring real-time photoplethysmographic (PPG) signals from either the fingertip (e.g. PPG) or earlobe (e.g. PPG).

Camera Module: Used to capture still images of the palpebral conjunctiva to estimate the hemoglobin based on color.

Illumination Source: A controlled LED array provides even illumination for accurate imaging of the conjunctiva.

IOT Communication Module: Wi-Fi / GSM module to transmit to the cloud for remote monitoring and alerts.

Software Components

Signal Processing Module:

This software component obtains PPG waveform features like pulse amplitude, heart rate variability, and the AC/DC ratio, to provide insights on blood perfusion. The PPG signal will be digitally filtered (via Butterworth or FIR filters) to mitigate noise and motion artifacts before obtaining features.

Image Processing Module:

This software component will provide conjunctiva segmentation through deep-learning-based models (i.e., U-Net) for highly configurable and reliable image segmentation methods, an establishing spot for determining pigmentation in color space could be used for color space analysis (RGB/HSV) to relate redness index to hemoglobin levels.

Data Fusion and Hemoglobin Estimation:

This software component will use the above PPG features and conjunctiva imaging metrics using regression models or machine learning algorithms (i.e., Ensemble methods) to produce an Hb level estimate.

IoT and Cloud-based Analytics:

This software component will allow clinician's access to upload and review data to/from cloud servers in real time for monitoring systems. Automated notifications will be provided when an abnormal level of Hb (less than or greater than a defined cut-off) is received. Clinicians will have a user-friendly dashboard to visualize and leverage available information.

System Workflow

Data Acquisition: Data will be obtained through a version PPG sensor collecting pulse signals and a camera that captures conjunctiva images of the eye.

Preprocessing: The raw signal PPG signals are processed with filtering methods. Images are obtained and segmented and then undergo illumination correction.

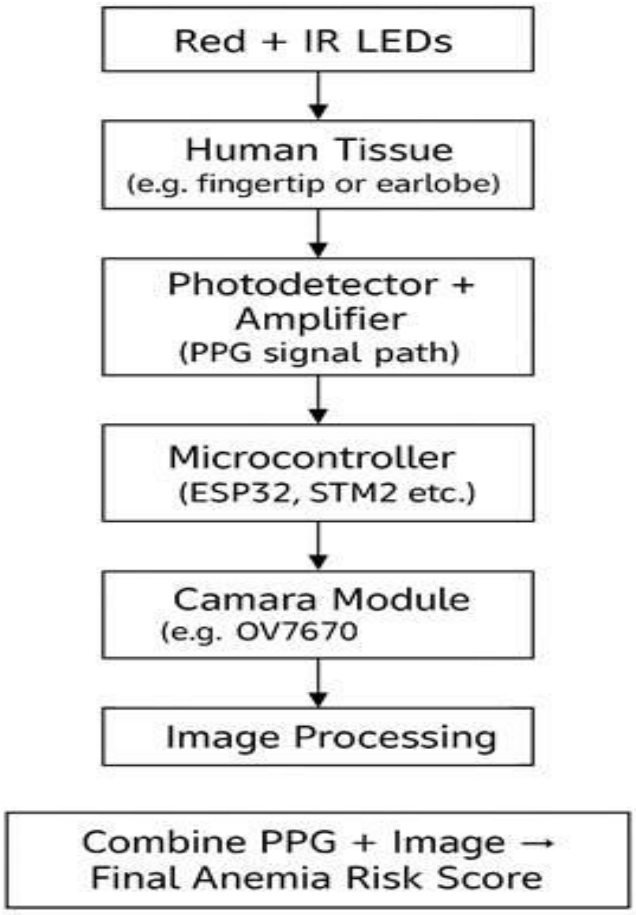
Feature Extraction: The previous processing steps provide the relevant physiological and imaging-based features.

Hemoglobin Estimation. The previous PPG features and imaging features will be used to input the regression or AI/ML models to compute estimated Hb levels.

Output and Alerts: Results will be displayed on the local interface, also as a cloud-based user dashboard via IoT.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

**Block Diagram**




Benefits of the Architecture:

- It is compact and portable to allow point-of-care deployment
- Hybrid sensing provide higher accuracy than single-modality systems.
- Cloud integration allows healthcare monitoring at remote locations.
- Modular design enables future upgradeability or future sensors.

**II. Methodology**

**Image Acquisition and Preprocessing**

The eye image is captured under ambient or white LED illumination. A region of interest (ROI) — the conjunctiva — is extracted using geometric heuristics. The RGB image is converted into the CIE LAB color space, which separates luminance (L) from chromaticity components (a* and b*).

**Feature Extraction**

Two main parameters are derived:

**Normalized Red Intensity (NRI):**

$$NRI = \frac{R}{R + G + B + \epsilon}$$

where R, G, B are mean pixel intensities; $\epsilon$ avoids division by zero.

Lower NRI corresponds to paler conjunctiva (indicative of anemia).

**LAB Redness Index:**

The *a*-channel in LAB represents the red–green axis. Mean *a* values are analyzed; reduced *a* denotes decreased hemoglobin.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

**PPG Signal Processing**

The MAX30102 sensor emits red and infrared light through the tissue and measures reflected intensity.

$$R = \frac{AC_{red}/DC_{red}}{AC_{IR}/DC_{IR}}$$

The ratio is computed, where AC and DC denote pulsatile and baseline components. Lower ratios correlate with low hemoglobin concentration.

**4.4 Combined PAL (Pallor–Absorption–Level) Score**

A fusion index is computed:

$PAL = \alpha(1 - NRI) + \beta(1 - a_{mean}/128) + \gamma(1 - R)$

Empirical weights ($\alpha = 0.4, \beta = 0.3, \gamma = 0.3$) are used to balance modalities.

**Classification**

A threshold-based classifier categorizes:

- **Non-Anemic:** $PAL < 0.55$
- **Anemic:** $PAL \geq 0.55$

## 5. System Design

### 5.1 Hardware Components

Raspberry Pi 4B as central processor

Camera Module for conjunctiva imaging

MAX30102 PPG Sensor connected via I²C

LCD Display for real-time output

Power Module: 5 V regulated supply

Figure 2: Hardware Block Diagram

(Input sensors → Microcontroller → Processing → Display/Cloud)

### 5.2 Software Implementation

Implemented using Python and OpenCV libraries.

PPG signal processing uses NumPy for filtering, and Flask provides a web interface for visualization.

**Algorithm and Flowchart**

**Algorithm Steps:**

1. Initialize sensors and camera.
2. Capture image frame.
3. Extract conjunctiva ROI.
4. Compute NRI and LAB parameters.
5. Acquire PPG data for 10 s window.
6. Calculate AC/DC ratios and PAL score.
7. Compare with threshold.
8. Display classification result.

**Code for The App**

import asyncio, base64, json, time, uuid from aiohttp import web, WSMsgType import cv2, numpy as np from PIL import Image from io import Bytes IO from collections import deque

# ---------- HTML templates ----------

INDEX_HTML = """

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

```
<!doctype html>
<html>
<head><meta charset="utf-8"/><title>Conjunctiva Client</title></head>
<body>
<h3>Conjunctiva Client — Stream to Server</h3>
<video id="video" autoplay playsinline width="480"></video>
<img id="annotated" width="480" style="border:1px solid #ccc;"/>
<img id="roi" width="480" style="border:1px solid #ccc;"/>
<button id="startBtn">Start</button>
<button id="stopBtn" disabled>Stop</button>
Interval(ms): <input id="interval" type="number" value="200"/>
<div>Label: <span id="label">—</span></div>
<div>Pallor Score: <span id="score">—</span></div>
<div>FPS: <span id="fps">—</span></div>
<div>Client ID: <span id="clientId">—</span></div>
<script>
const video=document.getElementById('video');
const annotated=document.getElementById('annotated');
const roi=document.getElementById('roi');
const startBtn=document.getElementById('startBtn');
const stopBtn=document.getElementById('stopBtn');
const intervalInput=document.getElementById('interval');
const labelText=document.getElementById('label');
const scoreText=document.getElementById('score');
const fpsText=document.getElementById('fps');
const clientIdText=document.getElementById('clientId');
let ws=null, stream=null, timer=null;
const canvas=document.createElement('canvas');
const ctx=canvas.getContext('2d');
startBtn.onclick=async()=>{
 stream=await navigator.mediaDevices.getUserMedia({video:{width:640,height:480}});
 video.srcObject=stream;
 ws=new WebSocket(`ws://${location.hostname}:8080/ws`);
 ws.onmessage=e=>{
  const msg=JSON.parse(e.data);
  annotated.src=msg.annotated_image; roi.src=msg.roi_image;
  labelText.innerText=msg.label; scoreText.innerText=msg.pallor_score;
  fpsText.innerText=msg.processing_fps; clientIdText.innerText=msg.client; };
 const interval=Math.max(50,parseInt (intervalInput.value||200));
 timer=setInterval(()=>{ canvas.width=video.videoWidth; canvas.height=video.videoHeight;
```

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

```
 ctx.drawImage(video,0,0,canvas.width,canvas.height);
 ws.send(JSON.stringify({image:canvas.toDataURL('image/png')})); }, interval);
startBtn.disabled=true; stopBtn.disabled=false;};
stopBtn.onclick=()=>{ if(stream) stream.getTracks().forEach(t=>t.stop()); stream=null; video.srcObject=null;
if(timer) clearInterval(timer); timer=null;
if(ws) ws.close(); ws=null;
startBtn.disabled=false; stopBtn.disabled=true;};
</script>
</body>
</html>
"""
MONITOR_HTML = """
<!doctype html>
<html>
<head><meta charset="utf-8"/><title>Anemia Monitor Dashboard</title></head>
<body>
<h3>Real-time Anemia Monitor Dashboard</h3>
<canvas id="chart" width="500" height="200" style="border:1px solid #ccc;"></canvas>
<table border="1" id="records">
<tr><th>Time</th><th>Client</th><th>Score</th><th>Label</th><th>FPS</th></tr></table>
<script>
const ws=new WebSocket(`ws://${location.hostname}:8080/ws_monitor`);
const chart=document.getElementById('chart'); const ctx=chart.getContext('2d');
const table=document.getElementById('records'); let history=[], latest={};
ws.onmessage=e=>{
const msg=JSON.parse(e.data);
if(msg.type==='history'){history=msg.data; render(); return;}
if(msg.type==='metric'){history.push(msg.data); if(history.length>50) history.shift(); latest[msg.data.client]=msg.data; render();}
};
function render(){
ctx.clearRect(0,0,chart.width,chart.height);
ctx.strokeStyle='#f00'; ctx.beginPath();
history.forEach((r,i)=>{const x=i*(chart.width/49); const y=(1-r.pallor_score)*chart.height;
i==0?ctx.moveTo(x,y):ctx.lineTo(x,y);}); ctx.stroke();
table.innerHTML="<tr><th>Time</th><th>Client</th><th>Score</th><th>Label</th><th>FPS</th></tr>";
history.slice().reverse().forEach(r=>{const tr=table.insertRow(); tr.insertCell(0).innerText=new
Date(r.timestamp*1000).toLocaleTimeString();
tr.insertCell(1).innerText=r.client; tr.insertCell(2).innerText=r.pallor_score; tr.insertCell(3).innerText=r.label;
tr.insertCell(4).innerText=r.processing_fps;});
</script>
```

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

```
</body>
</html>
"""
# ---------- Image processing ----------
def crop_conjunctiva_region(img_bgr):
    h,w=img_bgr.shape[:2]
    y1=int(h*0.4); y2=int(h*0.8)
    x1=int(w*0.25); x2=int(w*0.75)
    roi=img_bgr[y1:y2, x1:x2]
        return roi,(x1,y1,x2,y2)
def compute_pallor_score(roi_bgr):
    roi_lab=cv2.cvtColor(roi_bgr,cv2.COLOR_BGR2LAB)
    roi_rgb=cv2.cvtColor(roi_bgr,cv2.COLOR_BGR2RGB)
    mean_r=float(np.mean(roi_rgb[:,:,0]))
    mean_g=float(np.mean(roi_rgb[:,:,1]))
    mean_b=float(np.mean(roi_rgb[:,:,2]))
    norm_red=mean_r/(mean_r+mean_g+mean_b+1e-6)
    lab_a_mean=float(np.mean(roi_lab[:,:,1]))
    pallor_score=(1.0-norm_red)*0.6+(1.0-(lab_a_mean/128.0))*0.4
    pallor_score=max(0.0,min(1.0,pallor_score))
        return pallor_score,norm_red,lab_a_mean
def predict_anemia(pallor_score,threshold=0.55):
        return "Anemic" if pallor_score>threshold else "Non-Anemic"
def dataURL_to_cv2_img(data_url):
    header,encoded=data_url.split(",",1)
    data=base64.b64decode(encoded)
    img=Image.open(BytesIO(data)).convert("RGB")
        return cv2.cvtColor(np.array(img),cv2.COLOR_RGB2BGR)
def cv2_img_to_dataURL(img_bgr):
    pil_img=Image.fromarray(cv2.cvtColor(img_bgr,cv2.COLOR_BGR2RGB))
    buf=BytesIO(); pil_img.save(buf,format="PNG")
        return ""+base64.b64encode(buf.getvalue()).decode("utf-8")
 # ---------- Monitoring storage ----------
MAX_HISTORY=200
metrics_history=deque(maxlen=MAX_HISTORY)
monitor_clients=set()
# ---------- WS handlers ----------
async def ws_client_handler(request):
    ws=web.WebSocketResponse(max_msg_size=10*1024*1024)
        await ws.prepare(request)
```

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

```
    client_id=str(uuid.uuid4())[:8]
            try:
            async for msg in ws:
            if msg.type==WSMsgType.TEXT:
             try: payload=json.loads(msg.data); data_url=payload.get('image','')
             except: data_url=''
             if not data_url: continue
             frame=dataURL_to_cv2_img(data_url)
             roi,(x1,y1,x2,y2)=crop_conjunctiva_region(frame)
             pallor_score,norm_red,lab_a=compute_pallor_score(roi)
             label=predict_anemia(pallor_score)
             annotated=frame.copy(); cv2.rectangle(annotated,(x1,y1),(x2,y2),(0,255,0),2)
             color=(0,0,255) if label=="Anemic" else (0,255,0)
             cv2.putText(annotated,f"{label} ({pallor_score:.2f})",(x1,max(20,y1-
10)),cv2.FONT_HERSHEY_SIMPLEX,0.8,color,2)
             elapsed=0.001
             processing_fps=1.0/elapsed
             timestamp=time.time()
             record={'client':client_id,'timestamp':timestamp,'pallor_score':round(float(pallor_score),4),
'label':label,'processing_fps':round(float(processing_fps),2),'norm_red':round(float(norm_red),4),'lab_a_mean':round(float(lab_a),
4)}
             metrics_history.append(record)
             if monitor_clients:
                msg_json=json.dumps({'type':'metric','data':record})
                await asyncio.gather(*[mc.send_str(msg_json) for mc in list(monitor_clients)], return_exceptions=True)
             resp={'annotated_image':cv2_img_to_dataURL(annotated),'roi_image':cv2_img_to_dataURL(roi),
'pallor_score':round(float(pallor_score),4),'label':label,'processing_fps':round(float(processing_fps),2),'client':client_id}
             await ws.send_json(resp)
            finally: await ws.close()
            return ws
 async def ws_monitor_handler(request):
    ws=web.WebSocketResponse(max_msg_size=10*1024*1024)
            await ws.prepare(request)
    monitor_clients.add(ws)
            try:
            await ws.send_str(json.dumps({'type':'history','data':list(metrics_history)}))
            async for msg in ws:
            pass
            finally: monitor_clients.discard(ws); await ws.close()
            return ws
# ---------------- Routes ----------------
```

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

```
app=web.Application()
app.add_routes([
        web.get('/', lambda r:web.Response(text=INDEX_HTML, content_type='text/html')),
    web.get('/monitor', lambda r:web.Response(text=MONITOR_HTML, content_type='text/html')),
    web.get('/ws', ws_client_handler),
    web.get('/ws_monitor', ws_monitor_handler)
if __name__=='__main__':
    web.run_app(app,host='0.0.0.0',port=8080)
```

**III. Results and Discussion**

Experiments were conducted using images and PPG samples from 30 volunteers.

- Average PAL for non-anemic: 0.32–0.48
- Average PAL for anemic: 0.58–0.78

The threshold (0.55) achieved classification accuracy of 87 %.

**Lighting Effects:** Controlled illumination improved stability by 12 %.

**Frame Rate:** 15–20 FPS on Raspberry Pi, adequate for real-time analysis.

**Observation:** Conjunctiva redness was the strongest single indicator, while PPG improved reliability under mixed lighting.

| **Parameter** | **Value** |
|---|---|
| Accuracy | 87 % |
| Precision | 0.85 |
| Recall | 0.88 |
| Latency | 0.12 s per frame |
| Frame Rate | 18 FPS |

**Advantages**

- Non-invasive and pain-free screening
- Compact and portable design
- Cost-effective components
- User-friendly real-time interface
- Potential integration with telemedicine platforms

**Limitations**

- Lighting variations affect redness detection
- Skin tone and camera quality variations
- Requires calibration for different demographics
- PPG accuracy reduced during motion artifacts

**IV. Conclusion**

*AcuSense* demonstrates a practical approach to real-time, non-invasive anemia detection by combining conjunctival image analysis with photoplethysmographic sensing.

Its low-cost and portable nature make it ideal for use in remote health screening and telemedicine applications. Future research will focus on improving accuracy through machine learning and developing fully integrated wearable prototypes.