Purpose of Adding AI/ML Features

The SDK already provides raw data streams (images, audio, sensor events). AI/ML enhances these by turning raw data into intelligent insights:

 Why Add AI/ML?

Image/Video Analysis: Detect objects, faces, or hazards in captured media.

Audio Processing: Speech-to-text, emotion detection, or real-time translation.

Predictive Insights: Use sensor data (battery, memory, usage logs) for anomaly detection or predictive maintenance.

Context Awareness: Automatically trigger SDK actions (e.g., start recording when AI detects a person speaking).

Personalization: Tailor responses or actions based on user behavior patterns.

 Example Use Cases

Glasses capture a photo AI model classifies objects SDK overlays labels or triggers alerts.

PCM audio stream ML speech-to-text real-time subtitles in AR display.

Battery + usage logs ML predicts “low battery soon” SDK auto-triggers warning.

 In short: The SDK gives you control + data streams, while AI/ML adds intelligence + automation. Together, they transform the glasses from a device controller into a smart assistant.

Would you like me to design a layered architecture diagram (SDK layer AI/ML inference UI/alerts) so you can visualize exactly where to plug in your models?

 System Architecture for AI/ML Integration

1. Device Layer (Smart Glasses + SDK)

SDK APIs handle:

Scanning & connecting (BleScannerHelper, BleOperateManager)

Media capture (photos, video, audio)

Event monitoring (battery, memory, volume, OTA updates)

Wi-Fi transfer for large media files

AI triggers (image recognition, thumbnails)

2. Data Handling Layer

Raw Data Streams:

Images via getPictureThumbnails, Wi-Fi sync

Audio (PCM) via voiceFromGlasses

Sensor logs battery, memory, events

Preprocessing:

Resize, normalize, clean data

Convert PCM spectrograms for audio ML

Metadata tagging (timestamp, device info)

3. AI/ML Inference Layer

Image Models: Object detection, OCR, hazard recognition

Audio Models: Speech-to-text, emotion detection, translation

Predictive Models: Battery/memory anomaly detection, usage forecasting

Decision Logic:

If AI detects “person” → trigger auto-record

If AI predicts “low battery soon” → push alert

If AI recognizes “fire/smoke” → send disaster warning

4. Application Layer (User Interface)
Mobile App / Dashboard:

Displays AI insights (labels, transcripts, alerts)

Provides safety tips or contextual actions

Allows manual override of AI decisions

Notifications:

Real-time alerts (low battery, hazard detected)

Visual overlays (object labels in AR view)

Audio feedback (voice prompts)

5. Cloud/Backend (Optional)
Heavy ML models (e.g., ensemble disaster prediction) run in the cloud.

Glasses send data → backend inference → results returned.

Ensures lightweight on-device processing with fallback to cloud.

→ Purpose of AI/ML Here
Enhance intelligence: Turn raw sensor/media data into actionable insights.

Automate actions: Trigger SDK functions based on AI predictions.

Improve safety: Disaster prediction, anomaly detection, hazard recognition.

Personalize UX: Adapt glasses behavior to user context and preferences.

 In short: The SDK is your data + control layer, AI/ML is the intelligence layer, and the app/UI is the delivery layer. Together, they make the glasses a smart assistant rather than just a recording device.
Device Layer (top): Smart glasses + SDK APIs for scanning, connecting, capturing media, and monitoring events.

Data Handling Layer: Preprocesses raw streams (images, audio, sensor logs) into usable formats.

AI/ML Inference Layer: Runs models for image recognition, speech-to-text, predictive analytics, and decision logic.

Application Layer (UI): Displays AI insights, overlays, alerts, and notifications in the mobile app/dashboard.

Cloud/Backend (optional): Handles heavy ML tasks remotely, sending results back to the device for lightweight usage.