DYNA
http://dyna.medellin.unal.edu.co/

UNIVERSIDAD NACIONAL DE COLOMBIA

Smart audit in real time: automated visual analysis from mobile devices

Francisco Gerardo Huamanchumo-Trujillo [a], Alejandro Roman Campos-Gamarra [a], Rodrigo Alonso Guevara-Saldaña [a] & Alberto Carlos Mendoza-de los Santos [b]

[a] *Escuela de Ingeniería de Sistemas, Universidad Nacional de Trujillo, Trujillo, Perú. t1023300821@unitru.edu.pe, t1513300121@unitru.edu.pe, t1053300621@unitru.edu.pe*
[b] *Departamento de Ingeniería de Sistemas, Universidad Nacional de Trujillo, Trujillo, Perú, amendozad@unitru.edu.pe*

Received: June 8th 2025. Received in revised form: November 4th, 2025. Accepted: November 21st, 2025

Abstract
This study presents Smart Audit in Real Time, a mobile system designed to automate the visual inspection of technological infrastructures using OpenAI GPT-4o Vision. The application captures images, identifies auditable elements, and generates structured reports aligned with international IT auditing standards. A dataset of 50 images from different operational environments was analysed, producing 207 findings grouped into four categories: physical security, ergonomics, environmental conditions, and regulatory compliance. The system achieved an overall accuracy of 83% and an average F1-score of 0.65, with a mean response time of 15.8 seconds per evaluation. A chi-square test applied to the confusion matrix confirmed a statistically significant association between predicted and actual categories, supporting the reliability of the classification. The results show that automated visual auditing from mobile devices can efficiently detect operational risks and produce consistent evaluative outputs without expert intervention. This research contributes to the digital transformation of IT auditing by improving accessibility, precision, and real-time decision-making.

Keywords: digital inspection; visual recognition; contextual verification; operational management; automated review; intelligent processing.

Auditoría inteligente en tiempo real: análisis visual automatizado desde dispositivos móviles

Resumen
Este estudio presenta Smart Audit in Real Time, un sistema móvil diseñado para automatizar la inspección visual de infraestructuras tecnológicas mediante OpenAI GPT-4o Vision. La aplicación captura imágenes, identifica elementos auditables y genera reportes estructurados alineados con normas internacionales de auditoría de TI. Se analizó un conjunto de 50 imágenes procedentes de diversos entornos operativos, obteniéndose 207 hallazgos clasificados en cuatro categorías: seguridad física, ergonomía, condiciones ambientales y cumplimiento normativo. El sistema alcanzó una exactitud global del 83% y un F1-score promedio de 0.65, con un tiempo medio de respuesta de 15.8 segundos por evaluación. Una prueba de chi-cuadrado aplicada a la matriz de confusión confirmó una asociación estadísticamente significativa entre las categorías reales y las predichas, respaldando la confiabilidad del modelo. Los resultados evidencian que la auditoría visual automatizada desde dispositivos móviles puede detectar eficientemente riesgos operativos y generar evaluaciones consistentes sin intervención experta. Esta investigación contribuye a la transformación digital de la auditoría de TI, mejorando la accesibilidad, la precisión y la toma de decisiones en tiempo real.

Palabras clave: inspección digital; reconocimiento visual; verificación contextual; gestión operativa; revisión automatizada; procesamiento inteligente.

1 Introduction

In recent years, IT audits have changed substantially as organizations operate in more distributed settings that demand strict compliance and dependable operational security. In this landscape, mobile tools for assessing technological environments offer a practical alternative to conventional approaches, which tend to rely on static procedures and often miss physical or contextual risks that do not appear in documentation. Ongoing digital transformation together with

How to cite: Huamanchumo-Trujillo, F.G., Campos-Gamarra, A.R., Guevara-Saldaña, R.A. and Mendoza-de los Santos, A.C., Smart audit in real time: automated visual analysis from mobile devices DYNA, (93)240, pp. 27-35, January - March, 2026.

mobile work practices and a growing dependence on interconnected infrastructures has increased the need for quicker, more flexible, and automated audit processes. As a result, the way risks are detected, recorded, and addressed is shifting, especially in facilities where physical conditions directly affect the reliability of information systems.

Audits can no longer focus solely on software checks or network monitoring; they also need visual evidence from the physical environment of cable organization, ventilation, access control, and equipment layout among others. Building on this idea, this work introduces a smart audit approach for real-time assessments using a mobile device. The device camera captures environmental images, and artificial-intelligence models identify auditable elements. The system flags issues such as tangled wiring, missing signage, blocked areas, or poor airflow and produces an automatically structured report that states the problem, the relevant regulation, a suggested corrective action, and the associated risk. Because the process is automated end-to-end, personnel with limited audit experience can still obtain consistent, professional results.

The spread of accessible, portable technologies has changed how auditors interact with systems. Mobile platforms make it possible to document environmental aspects of order, physical security, and ventilation that are frequently overlooked in purely logical reviews [1]. In public-sector settings, AI helps surface vulnerabilities that merit both ethical and technical attention [2]. Beyond recognizing deficiencies, intelligent tools standardize how findings are captured and traced. Automatically generated reports derived from image analysis align observations with international standards, improving documentation structure, traceability, and transparency throughout the audit process [3]. This brings tangible advantages where fast, accurate reporting is essential.

Complementary advances reinforce this direction. The combination of blockchain and cloud accounting with AI strengthens the integrity and traceability of audit evidence [4]. Reviews of AI methods in cybersecurity show that automated analysis can reveal complex or subtle vulnerabilities that are hard to detect with manual procedures [5]. In management and performance audits, intelligent models have shortened analysis cycles and improved effectiveness [6,7]. On the computer-vision side, techniques such as quantized scene detection and mobile augmented reality enable real-time identification of physical risks directly from smartphones [8-13]. Moreover, lightweight AI models have proven practical for mobile deployment, supporting device-based audit tasks without excessive resource demands [14]. Taken together, these developments point to a more holistic view of IT auditing in which visual inspection complements logical analysis to improve accuracy, speed, and the quality of recorded evidence. The intention is not to replace human auditors, but to extend their capabilities through context-aware AI and mobile computing.

Despite these benefits, mobile intelligence audits face limitations. Dependence on network connectivity can restrict operation in constrained environments, and image capture raises privacy and ethical concerns when sensitive content or personnel appear in the frame. Ensuring interpretability, transparency, and responsible use of AI tools is therefore essential for sustainable adoption in professional practice.

2 Design

The proposed software design focuses on developing a functional mobile application built with the Dart programming language and the Flutter framework, ensuring high compatibility with Android devices. The system architecture is divided into three modules: the user interface, the image-processing engine, and the PDF-report generator. This modular structure allows each component to be updated or extended independently, enabling future integrations and improvements.

The main workflow begins when the user captures or selects an image from the device's camera or gallery. The image is then encoded and sent to the analytics service, which processes it in the background and returns a structured audit report containing the detected findings, the applicable regulatory framework, the suggested corrective actions, and the corresponding risk level.

A dedicated REST API manages communication between the mobile client and the analysis service, while an integrated control converts the generated report into a ready-to-archive or shareable PDF file.

From a technical standpoint, specific performance optimizations were implemented for mobile operation. The adoption of deep-learning models for real-time detection on devices with limited computational resources validates the feasibility of this system [15]. In addition, the mobile computer-vision framework is regarded as a technological revolution that enables contextual recognition and intelligent image capture for audit purposes [16]. Finally, super-resolution techniques were applied to enhance image clarity without compromising performance, improving the quality of the visual reports produced directly on the mobile device [17].

3 Materials and Methods

The following technologies were used:

- Mobile device: A mobile device with an Android operating system (version 14) was used to capture the images of the environment to be audited. The device has a high-resolution camera, which allows clear visual evidence to be obtained for automated analysis.
- Make/Model: Galaxy A55 5G
- Operating system: Android version 14
- Dart: It is a modern programming language optimized for the development of user interfaces created by Google. It is an object-oriented language with static typing that allows you to develop efficient and high-performance applications. Dart is the primary language used in Flutter, making it easy to create reactive, cross-platform mobile interfaces from a single codebase.
- Flutter: It is an open-source framework developed by Google for the creation of mobile, web, and desktop applications. It allows you to develop native apps for Android and iOS using a single codebase written in Dart. Flutter provides a rich set of customizable widgets,

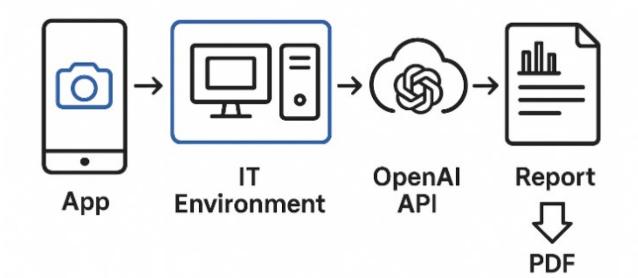

Figure 1. Process of generating a computer audit report.
Source: Authors.

responsive architecture, and built-in tools to facilitate modern interface design and agile development.

- OpenAI GPT-4 Vision: It is an artificial intelligence service capable of interpreting both text and images. This version of the GPT-4 model has multimodal capabilities, allowing it to analyze visual content and generate natural language responses from images. In this project, it is used to perform automatic audits from photographs captured from mobile devices. The image is sent to the model along with a prompt text, and the model returns a detailed analysis in the form of text.
- Internet connection: The system requires an active and stable internet connection to communicate with OpenAI's API. This connection is essential to send images to the AI server, receive the corresponding textual analysis, and subsequently process this information within the mobile app.
- Output format and PDF generation: Once the AI-generated analysis is received, this text is transformed into a PDF document using libraries available in Flutter. This PDF file serves as evidence of the audit performed and can be stored locally, shared digitally, or printed. The generation of PDFs allows the results to be structured and preserved in a widely accepted and portable format.

The system was developed in four phases, each aimed at solving a key component of the system. Each of these phases was developed and integrated in a modular way to ensure the correct functioning of the complete system. Fig. 1 illustrates the process of generating a computer audit report. Phases are briefly described below:

- Phase 1: Interface development.
- Phase 2: Image selection functionality.
- Phase 3: OpenAI API integration
- Phase 4: Generating the PDF file

3.1 Interface development:

Since the main purpose of the software is to capture an image of the environment in real-time to perform an automated computer audit, the development of a mobile application was necessary because of the need to use the device's camera immediately and accessible to the user.

For its implementation, Flutter, a cross-platform development framework created by Google, was chosen. Flutter allows you to build native apps for both Android and iOS from a single codebase, speeding up development and making it easier to maintain your project.

The graphical interface was designed with a minimalist and functional approach, allowing the user to perform the main actions clearly and directly. Next, the code used in the rendering of the interface is presented in Fig. 2. This visual foundation sets the starting point for the next phases of the system, focused on user interaction and intelligent processing of the captured image.

3.2 *Image select functionality*

This phase focuses on incorporating the functionality that allows the user to provide the image that will serve as an input for the computer audit. Two options are provided, as shown in Fig. 3: Capture a new photo or select an existing image from the device's gallery.

```
Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text("Auditoria por Imagen")),
      body: SafeArea(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            children: [
              if (_imagen != null)
                Image.file(_imagen!, height: 200, fit: BoxFit.cover),
              const SizedBox(height: 12),
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceAround,
                children: [
                  ElevatedButton.icon(
                    onPressed: () => _seleccionarImagen(ImageSource.gallery),
                    icon: const Icon(Icons.image),
                    label: const Text("Galeria"),
                  ),
                  ElevatedButton.icon(
                    onPressed: () => _seleccionarImagen(ImageSource.camera),
                    icon: const Icon(Icons.camera),
                    label: const Text("Cámara"),
                  ),
                ],
              ),
              const SizedBox(height: 12),
              ElevatedButton.icon(
                onPressed: _imagen == null || _procesando ? null : _enviarImagen,
                icon: const Icon(Icons.send),
                label: const Text("Hacer auditoria"),
              ),
              const SizedBox(height: 16),
              Expanded(
                child: _procesando
                    ? const Center(child: CircularProgressIndicator())
                    : TextField(
                        controller: _respuestaController,
                        maxLines: null,
                        readOnly: true,
                        expands: true,
                        decoration: const InputDecoration(
                          border: OutlineInputBorder(),
                          hintText: "Aqui aparecerá el resultado de la auditoria...",
                        ),
                      ),
              ),
              const SizedBox(height: 12),
              Padding(
                padding: const EdgeInsets.only(top: 8.0),
                child: ElevatedButton.icon(
                  onPressed: _respuesta.isEmpty ? null : _descargarPDF,
                  icon: const Icon(Icons.download),
                  label: const Text("Descargar PDF"),
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
```

Figure 2. The code snippet corresponding to the construction of the interface
Source: Authors.

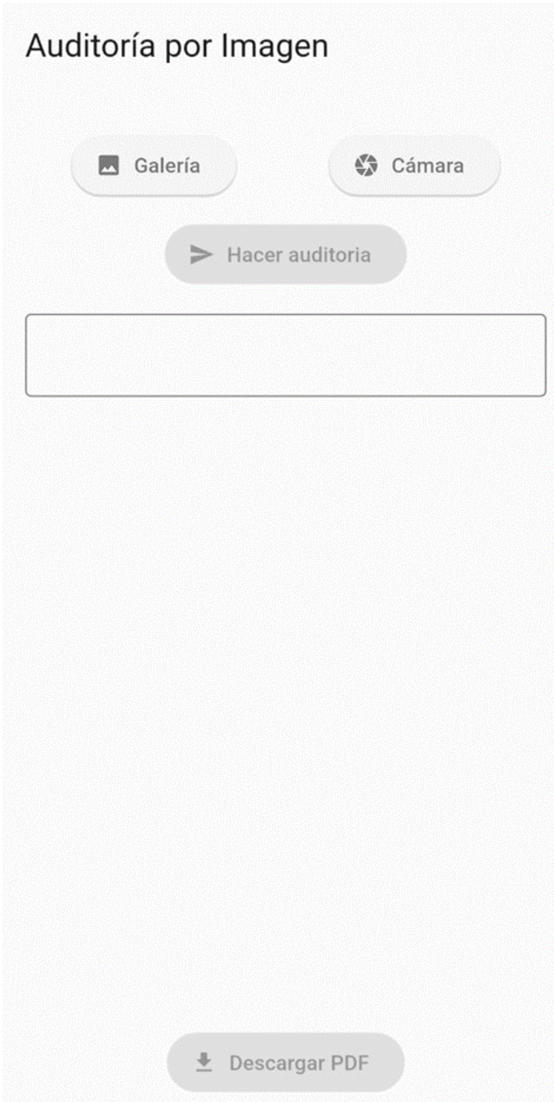

Figure 3. Mobile device interface.
Source: Authors.

3.1.1 Main functionality: Select image

The function responsible for handling image selection is presented in Fig. 4, which includes the corresponding implementation code. This feature allows you to get an image from the specified source (camera or gallery) using the image_picker package. Once selected, the image is saved locally, and the status of the application is updated.

```
Future<void> _seleccionarImagen(ImageSource origen) async {
    final XFile? seleccionada = await _picker.pickImage(source: origen);
    if (seleccionada != null) {
      setState(() {
        _imagen = File(seleccionada.path);
        _respuesta = '';
        _respuestaController.clear();
      });
    }
  }
```

Figure 4. The function in charge of managing the selection of the image
Source: Authors.

```
inAxisAlignment: MainAxisAlignment.spaceAround,
children: [
  ElevatedButton.icon(
    onPressed: () => _seleccionarImagen(ImageSource.gallery),
    icon: const Icon(Icons.image),
    label: const Text("Galería"),
  ),
  ElevatedButton.icon(
    onPressed: () => _seleccionarImagen(ImageSource.camera),
    icon: const Icon(Icons.camera),
    label: const Text("Cámara"),
  ),
],
```

Figure 5. Button code for image selection.
Source: Authors.

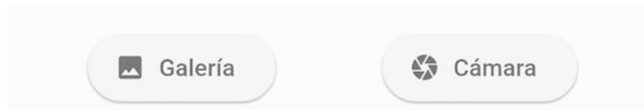

Figure 6. User interface for image selection or capture.
Source: Authors.

```
final bytes = await _imagen!.readAsBytes();
final base64 = base64Encode(bytes);
final dataUrl = 'data:image/jpeg;base64,$base64';
```

Figure 7. Base64 encoded image in data URL format
Source: Authors.

3.2.1 Selection Interface: Camera or Gallery

The user interface offers two clearly differentiated buttons, one to access the gallery, and one to use the device's camera. The implementation code is shown in Fig. 5, and the interface is shown in Fig. 6, where the user can select an image for processing.

3.3 OpenAI API integration

Once the image has been obtained from the camera or gallery, the next step is to send it to the OpenAI API, where the captured environment will be visually analyzed with GPT-4o and the results will be exported to an audit report.

3.3.1 Converting the image to base format64

Before sending, the image must be base64 encoded in data URL format, which is the format required by the API to process images, the code used is in Fig. 7.

Message preparation and payload, the message that will be sent to the API is constructed. This includes:

- A prompt with detailed instructions on how to audit the image.
- The image in base format64.
- The artificial intelligence model to be used (GPT-4o).

Next, in Fig. 8 the message that will be sent to the API is constructed. This includes:

```
final payload = {
  "model": "gpt-4o",
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": "..."
        },
        {
          "type": "image_url",
          "image_url": {"url": dataUrl}
        }
      ]
    }
  ],
  "max_tokens": 4096
};
```

Figure 8. Code to call the artificial intelligence model to use (gpt-4o).
Source: Authors.

Inside "text" will go the prompt which will be: You are a professional computer auditor. Analyze the submitted image in detail, showing a technology-related facility (e.g., computers, servers, networks, cabling, infrastructure, etc.). Identifies all visible problems, deficiencies or bad practices that may affect the safety, operation, availability or regulatory compliance of technological systems.\n\nConsider aspects such as:\n- Physical security\n- Organization of the environment\n- Cable and energy management\n- Environmental conditions and ventilation\n- Ergonomics and accessibility\n- Signage, cleaning and order\n- Fire protection, theft or accidents\n- Asset management or tagging\n- Physical access controls\n- Operational continuity\n- Any other relevant aspect that can be observed\n\nFor each finding detected:\n1. Clearly describe the observed problem.\n2. It mentions the applicable regulatory framework (ISO/IEC, NIST, COBIT, ITIL, EIA/TIA, etc.). \n3. Propose an action plan or specific corrective measures.\n4. Indicate the risks associated if the problem is not corrected.\n\nAnswer in Spanish and structure the report by separating each finding as if it were part of a professional audit. If no visible problems are detected, it indicates that the installation complies with observable good practices.

To process OpenAI API response, the content is decoded and the analysis generated by the model is extracted. Then, the interface is updated to show the results to the user. The code used for this section is shown in Fig.9.

```
final respuesta = await http.post(
  Uri.parse("https://api.openai.com/v1/chat/completions"),
  headers: {
    "Content-Type": "application/json; charset=UTF-8",
    "Authorization": "Bearer api_key"
  },
  body: jsonEncode(payload),
);

final data = json.decode(utf8.decode(respuesta.bodyBytes));
final contenido = data["choices"][0]["message"]["content"];
```

Figure 9. Code to extract the analysis generated by the model
Source: Authors.

This section transforms the picture into a detailed report that identifies various findings related to security, environment organization, technology infrastructure, and regulatory compliance. Each finding includes a description of the observed problem, the applicable regulatory framework, a proposed action plan, and the associated risks in the event of failure to implement the suggested corrective measures. Considering that the "Authorization" will be the api_key generated on the OpenAI website.

3.4 Generation of the pdf file:

Once the structured analysis of the image has been received by the API, the application allows you to generate a file in PDF format containing all the findings; the coding used is shown in Fig. 10. This functionality is essential for documenting, sharing, or backing up analysis results in a readable and portable format.

```
Future<void> _descargarPDF() async {
    try {
      if (_respuesta.isEmpty) {
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(content: Text("No hay contenido para exportar.")),
        );
        return;
      }

      final pdf = pw.Document();

      pdf.addPage(
        pw.MultiPage(
          pageFormat: PdfPageFormat.a4,
          build: (pw.Context context) => [
            pw.Text(
              _respuesta,
              style: pw.TextStyle(fontSize: 12),
            ),
          ],
        ),
      );

      final directory = await getApplicationDocumentsDirectory();
      final path = '${directory.path}/auditoria_respuesta.pdf';
      final file = File(path);
      await file.writeAsBytes(await pdf.save());

      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("PDF guardado en: $path")),
      );

      await OpenFile.open(path);
    } catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("Error al generar PDF: $e")),
      );
    }
}
```

Figure 10. Code to extract the PDF from everything analyzed by the function
Source: Authors.

The PDF document is built from the previously generated textual content and organizes each finding in a clear structure, facilitating its reading and subsequent analysis. Each section of the report is composed of the following elements:

- Observed problem: Describes the inconvenience detected from the visual analysis of the image, such as security failures, disorder, bad wiring practices, among others.
- Applicable regulatory framework: Indicates which norms, standards or good practices (such as ISO/IEC, NIST, COBIT, ITIL, etc.) are related to the problem identified.
- Action plan: Details of the concrete measures that must be applied to correct the problem observed.
- Associated risks: It sets out the possible consequences that could arise if the suggested action plan is not carried out.

An actual extract of the content that can be generated after analyzing an image is shown in Fig. 11 below. For testing purposes, an image of the development environment used by one of the authors was used.

Figure 11. Image of the environment where the application is developed
Source: Authors.

The audit findings were:

Technology Installation Audit Report
1. Organization of the Environment
- *Observed Problem*: The artboard is disorganized with various elements, including a fan on top of documents or notebooks.
- *Applicable Regulatory Framework*: ITIL, IT Service Management, focus on an orderly work environment.
- *Action Plan*: Organize the workspace, removing unnecessary elements and ensuring that devices such as fans are not on documents that may be important.
- *Associated Risks*: Disorganization can lead to loss of important documents, distractions, and reduced efficiency.

2. Cable and Power Management
- *Observed Problem*: The cables are placed without visible organization, which could lead to clutter or accidents.
- *Applicable Regulatory Framework*: ISO/IEC 14763-2 on the management and design of cabling installations.
- *Action Plan*: Implement cable management systems such as trunking or clips to keep wiring tidy and prevent tripping.
- *Associated Risks*: Possible physical accidents, service interruptions or damage to equipment.

3. Environmental Conditions and Ventilation
- *Observed Problem*: The use of fans indicates that there could be ventilation or temperature control problems.
- *Applicable Regulatory Framework*: ASHRAE, standards for environmental conditions in data centers.
- *Action Plan*: Assess the need for more efficient or suitable HVAC systems for the workspace.
- *Associated Risks*: Overheating of equipment, reduction of hardware life, less comfort for the user.

4. Ergonomics and Accessibility
- *Observed Issue*: The keyboard and mouse layout may not be ergonomic, which may lead to long-term discomfort or injury.
- *Applicable Regulatory Framework*: ISO 9241 ergonomics guidelines.
- *Action Plan*: Evaluate and adjust the position of the input devices according to the best ergonomic practices.
- *Associated Risks*: Injuries may repetitive strain, reduced productivity.

5. Fire Protection
- *Observed Problem*: No smoke detectors or fire extinguishers visible in the image.
- *Applicable Regulatory Framework*: NFPA 75, Fire Protection of Information Processing Equipment.
- *Action Plan*: Ensure the installation of appropriate smoke detectors and fire extinguishers in the vicinity of the technological installation.
- *Associated Risks*: Risk of fire damage to equipment and data, jeopardizing safety and operational continuity.

Conclusions
The installation shows several opportunities for

improvement in terms of organization, safety and ergonomics. We recommend implementing the aforementioned corrective measures to ensure a safe, efficient, and compliant environment.

4 Results

To evaluate the effectiveness of the AI-based IT auditing system, a representative dataset of 50 images was created, reflecting diverse technological environments. Stratified sampling was used to ensure coverage of common audit settings: 17 corporate offices, 10 remote workspaces, 11 software development areas, and 12 computer rooms or educational labs.

Based on the number of auditable findings, images were classified as simple (1–3 elements, 15 images), moderately complex (4–6 elements, 22 images), and complex (over 6 elements, 13 images). This categorization allowed analyzing how visual complexity affected system performance. An analysis of the dataset identified 207 auditable findings distributed across four predefined categories: Physical Security, Ergonomics, Environmental Issues, and Regulatory Compliance (Table 1). The findings averaged 4.14 per image, reflecting the complexity of real technological environments where multiple issues often coexist. The simplest image contained a single finding, corresponding to a well-organized remote workspace, while the most complex image showed nine findings, associated with an educational laboratory exhibiting multiple deficiencies.

Table 1.
Auditable findings categories

Category	Quantity
security	53
ergonomics	57
environmental	48
compliance	49
Total	**207**

Source: Authors.

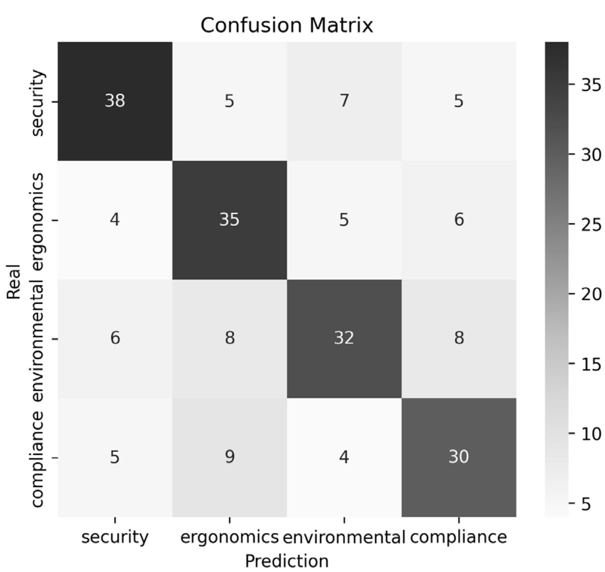

Figure 12. Confusion matrix elaborated using data collected from analysis.
Source: Authors.

Table 2.
Results of Chi-square test applied to the confusion matrix.

Statistic	Value	df	p-value	Critical Value	Decision
χ^2	180.87	9	3.32×10^{-34}	16.91	Reject H_0

Source: Authors.

With the dataset constructed, the IT auditing system was evaluated to assess its accuracy in identifying and classifying the 207 findings. The quantitative results were summarized in a confusion matrix (Figure 12), which highlights both correct detections and characteristic error patterns.

A chi-square (χ^2) test of independence was applied to the confusion matrix to verify whether the distribution of classifications differed significantly among the four categories. The null hypothesis (H_0) assumed no association between predicted and actual findings. As shown in Table 2, the test confirmed a statistically significant relationship between both variables, indicating that the audit system classification performance is consistent and reliable across all categories.

Performance metrics for each category were also calculated and are shown in Table 3. The application exhibited its best performance in identifying issues related to physical security and environmental problems, achieving 72% and 61% accuracy, respectively. These values are consistent with the visual nature of these issues, which tend to present clear, common, and distinguishable patterns suitable for automated analysis. Conversely, findings related to ergonomics and regulatory compliance achieved lower accuracy rates, both at 61%, which may be attributed to the wide variety of existing regulatory frameworks and their respective versions.

Additionally, tests were performed to obtain the application's response time. Using the same set of analyzed images, response time was collected across all stages involved in generating the audit report, as shown in Table 4.

The evaluation of the intelligent IT auditing system revealed consistent performance across multiple categories of auditable findings, with an overall F1 score of 0.65 and an average processing time of 15.8 seconds per image. These

Table 3.
Performance metrics

Category	Precision	Sensitivity	F1 Score	Accuracy
security	0.72	0.69	0.70	0.85
ergonomics	0.61	0.70	0.65	0.82
environmental	0.67	0.59	0.63	0.82
compliance	0.61	0.63	0.62	0.82
Average	0.65	0.65	0.65	0.83

Source: Authors.

Table 4.
Response time per stage

Stage	Average response time (sec)
Image capture	2.3
Image Processing	10.7
Preparation of the PDF report	2.8
Total	**15.8**

Source: Authors

results indicate a satisfactory balance between detection accuracy and operational efficiency, supporting the viability of image-based auditing as a complementary approach to traditional IT audit practices.

The findings align with the broader trend of technology-assisted auditing described in [18], where it was demonstrated that technology-based audit techniques significantly improve efficiency and task completion rates. Similarly, the developed system achieves substantial time reductions by automating visual inspection tasks that would otherwise require manual evaluation. This is consistent with the strategies proposed in [2], that emphasize the use of artificial intelligence, machine learning, and automation to enhance IT auditing through anomaly detection and process optimization. While both studies focus on data-centric or procedural automation, the present system extends these concepts into computer vision, allowing the automated identification of auditable findings directly from images of technological environments. This adaptation exemplifies how AI can evolve from purely data-driven analytics to visually contextual intelligence, expanding its applicability in IT auditing.

In the context of artificial intelligence applications to auditing, [19] conceptualized AI as a transformative enabler for real-time monitoring, anomaly detection, and data-driven decision-making, while [3] empirically demonstrated that AI enhances the accuracy and speed of financial audits by automating fraud detection and anomaly analysis. Building on both perspectives, the proposed system operationalizes these principles within IT auditing, applying AI to the visual domain rather than numerical or textual data. This integration of computer vision enables the recognition of physical, ergonomic, and environmental non-compliance indicators, broadening the scope of AI-based auditing beyond financial or transactional contexts. The system thus provides tangible evidence that AI can effectively support not only abstract data analysis but also the spatial and contextual assessment of technological infrastructures.

Furthermore, [20] argued that Agile methodologies enhance IT audit effectiveness by promoting adaptability, iterative improvement, and stakeholder collaboration. Although the proposed system does not explicitly follow an Agile framework, its modular architecture and rapid-feedback cycle align with Agile principles, enabling continuous refinement of detection models and real-time response to new risk patterns. Integrating this system into Agile audit workflows could therefore strengthen its adaptability and extend its practical use in dynamic organizational environments.

Finally, as noted by [21] and reinforced by [7], the rapid emergence of AI technologies has outpaced the evolution of auditing standards, creating a regulatory gap concerning transparency, interpretability, and ethical responsibility. The present study reflects this dual reality: while AI-driven systems, such as the one developed here, achieve substantial efficiency and precision improvements, they also depend on data representativeness and require human oversight to ensure trustworthiness. These findings underscore the urgent need for updated standards and ethical frameworks that explicitly address the integration of AI in auditing, ensuring the reliability, fairness, and accountability of such systems.

5 Conclusions and recommendations

The developed intelligent auditing application successfully automates the evaluation of technological environments through image recognition. Using a dataset of 50 images containing 207 auditable findings, the system achieved an average F1 score of 0.65 and an overall accuracy of 83%, with a mean processing time of 15.8 seconds per image. A chi-square (χ^2) statistical test applied to the confusion matrix ($\chi^2 = 180.88$, df = 9, $p < 0.001$) confirmed a significant association between the predicted and actual findings, indicating that the audit system performance is not due to chance. These results validate the reliability and effectiveness of the proposed solution as a consistent and efficient tool to support IT audit processes. Moreover, as discussed previously, the system's performance aligns with trends reported in the literature, where artificial intelligence and automation significantly enhance audit efficiency and precision. The integration of visual-intelligence techniques extends these advances to IT auditing, demonstrating that image-based analysis can complement traditional approaches and contribute to the digital transformation of auditing practices.

Although the system relies on an internet connection to send images and generate the audit via GPT, this does not represent a significant limitation, as it is designed for IT environments where connectivity is generally available, and running an AI model directly on a mobile device would be too resource intensive. Using GPT involves a low associated cost, and in the future, local models could be considered to reduce dependence on cloud computing services. Regarding privacy, data is limited to the generated report, which is stored only on the user's device, so its security depends on who has access to the phone. Finally, image interpretation is exclusively aimed at identifying issues in the facility and generating a useful report for corrective actions; it is not intended for malicious purposes, highlighting the need for ethical and responsible use of the tool.

References

[1] Jessica-del-Milagro, P.T., Artificial intelligence in government auditing: emerging ethical challenges. Ciencia Latina Revista Científica Multidisciplinar, 8(4), pp. 5984–5998, 2024. DOI: https://doi.org/10.37811/cl_rcm.v8i4.12812

[2] Reyes, A., AI strategies applied to computer auditing. Technology Rain Journal, 2(2), pp. e18, 2023. DOI: https://doi.org/10.55204/trj.v2i2.e18

[3] Valladares-Albarracín, J.J., and Ordóñez-Parra, Y.L., The application of artificial intelligence in accounting auditing. Multidisciplinary Journal of Research Perspectives, [online]. 4(special), pp. 73–85, 2024. Available at: https://rperspectivasinvestigativas.org/index.php/multidiscipinaria/article/view/172

[4] Tapia-Salvatierra, A.J., Impact of artificial intelligence, blockchain and cloud accounting on the transformation of accounting and auditing practices in Mexico: opportunities, challenges and integration strategies. Ciencia Latina Revista Científica Multidisciplinar, 8(4), pp. 12491–12510, 2024. DOI: https://doi.org/10.37811/cl_rcm.v8i4.13450

[5] Torres, A.B., Rendón, F.G., and Gutiérrez, J.F., Revisión de las técnicas de inteligencia artificial aplicadas en seguridad informática. Revista Ontare, 7, pp. 98–115, 2020. DOI: https://doi.org/10.21158/23823399.v7.n0.2019.2612

[6] Galán-Toral, A.X., and Torres-Palacios, M.M., Optimization of management audit processes through the application of artificial intelligence. EMAC case. Cienciamatria, 10(2), pp. 189–209, 2024. DOI: https://doi.org/10.35381/cm.v10i2.1325
[7] Tosca-Magaña, S.A., Vázquez-Vidal, V., and Martinez-Ortiz, M., The digital revolution in accounting: the impact of artificial intelligence on auditing. FACE: Journal of the Faculty of Economic and Business Sciences, 24(2), pp. 71–78. 2024. DOI: https://doi.org/10.24054/face.v24i2.3119
[8] Ignatov, A., Malivenko, G., Timofte, R., Chen, S., Xia, X., Liu, Z., Zhang, Y., Zhu, F., Li, J., Xiao, X., Tian, Y., Wu, X., Kyrkou, C., Chen, Y., Zhang, Z., Peng, Y., Lin, Y., Dutta, S., Das, S.D., and Shah, N.A., Fast and accurate quantized camera scene detection on smartphones, Mobile AI 2021 Challenge: report. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2558–2568. 2021. DOI: https://doi.org/10.1109/cvprw53098.2021.00289
[9] Bustelo, J.L., The intersection between artificial intelligence (AI), complex thinking, and bias audit methodology. Ibero-American Journal of Complexity and Economic Sciences, 2(4), pp. 5–16, 2024. DOI: https://doi.org/10.48168/ricce.v2n4p5
[10] Lucero, L., The role of computer auditing in the era of personal data protection in Ecuador. Technology Rain Journal, 2(2), pp. e17, 2023. DOI: https://doi.org/10.55204/trj.v2i2.e17
[11] Alcocer-Yunda, V.H., Ausay-Carrillo, C.F., Ávila-Pita, S.P., and Sani-Paguay, A.A., Artificial intelligence and digital transformation in auditing to combat fraud and corruption. Revista Científica Multidisciplinaria InvestiGo, 6(15), pp. 316–328. 2025. DOI: https://doi.org/10.56519/39hf4095
[12] Dastagir, R.V., Jami, J.T., Chanda, S., Hafiz, F., Rahman, M., Dey, K., Rahman, M.M., Qureshi, M., and Chowdhury M.M., AI-driven smartphone solution for digitizing rapid diagnostic test kits and enhancing accessibility for the visually impaired. ArXiv (Cornell University). Art. 7, 2024. DOI: https://doi.org/10.48550/arxiv.2411.18007
[13] Li, X., Tian, Y., Zhang, F., Quan, S., and Xu, Y., Object detection in the context of mobile augmented reality. 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 156–163. 2020. DOI: https://doi.org/10.1109/ismar50242.2020.00037
[14] Li, Y., Dang, X., Tian, H., Sun, T., Wang, Z., Ma, L., Klein, J., and Bissyandé, T.F., An empirical study of AI techniques in mobile applications. Journal of Systems and Software, 219, Art. 112233, 2025. DOI: https://doi.org/10.1016/j.jss.2024.112233
[15] Feng, R., Chen, S., Xie, X., Meng, G., Lin, S.W., and Liu, Y., A performance-sensitive malware detection system using deep learning on mobile devices. IEEE Transactions on Information Forensics and Security, 16, pp. 1563–1578, 2021. DOI: https://doi.org/10.1109/tifs.2020.3025436
[16] Suo, J., Zhang, W., Gong, J., Yuan, X., Brady, D.J., and Dai, Q., Computational imaging and artificial intelligence: the next revolution of mobile vision. Proceedings of the IEEE, 111(12), pp. 1607–1639, 2023. DOI: https://doi.org/10.1109/jproc.2023.3338272
[17] Ignatov, A., Perevozchikov, G., Timofte, R., Zhang, Z., Gao, T., Yang, Y., Zhu, S., Wang, S., Yoon, K., Gankhuyag, G., Moon, H.C., Jeong, T., Kim, Y., Lee, S., Baek, J., Jeong, J., Park, E., Lee, J., Lee, H., and Kim, S., Quantized image super-resolution on mobile NPUs, Mobile AI 2025 Challenge: Report. 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1899–1912. 2025. DOI: https://doi.org/10.1109/cvprw67362.2025.00179
[18] Eulerich, M., Masli, A., Pickerd, J., and Wood, D.A., The impact of audit technology on audit task outcomes: evidence for technology-based audit techniques. Contemporary Accounting Research, 40(2), pp. 981–1012. 2023. DOI: https://doi.org/10.1111/1911-3846.12847
[19] Leocádio, D., Malheiro, L., and Reis, J., Artificial intelligence in auditing: a conceptual framework for auditing practices. Administrative Sciences, 14(10), Art. 238, 2024. DOI: https://doi.org/10.3390/admsci14100238
[20] Ilori, O., Nwosu, E., and Naiho, A., Enhancing IT audit effectiveness with agile methodologies: a conceptual exploration. Engineering Science and Technology Journal, 5(6), pp. 1969–1994, 2024. DOI: https://doi.org/10.51594/estj.v5i6.1217
[21] Gauthier, J., and Brender, N., How do the current auditing standards fit the emergent use of blockchain? Managerial Auditing Journal, 36(3), pp. 365–385. 2021. DOI: https://doi.org/10.1108/MAJ-12-2019-2513

F. G. Huamanchumo Trujillo
Systems Engineering Student School of Systems Engineering National University of Trujillo.
ORCID: 0009-0007-1937-4036

A. R. Campos Gamarra
Systems Engineering Student School of Systems Engineering National University of Trujillo
ORCID: 0009-0009-8492-1133

R. A. Guevara Saldaña
Systems Engineering Student School of Systems Engineering National University of Trujillo
ORCID: 0009-0002-6323-3517

A. C. Mendoza de los Santos
PhD in Systems Engineering and Informatics Department of Systems Engineering National University of Trujillo
ORCID: 0000-0002-0469-915X