SCIENCE CHINA
Information Sciences

• Supplementary File •

# RS-Agent: Automating Remote Sensing Tasks through Intelligent Agent

Wenjia Xu[1†*], Zijian Yu[1†], Boyang Mu[1], Jiuniu Wang[2], Zhiwei Wei[3*] & Mugen Peng[1]

[1]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*City University of HongKong, Hong Kong 999077, China*
[3]*School of Geographic Sciences, Hunan Normal University, Changsha 410081, China*

## A Detailed Prompt Designs

This section provides the complete prompt templates used in RS-Agent's Central Controller for task inference and keyword extraction. These prompts are carefully designed to incorporate remote sensing domain knowledge and guide the LLM to make accurate task planning decisions.

### A.1 Task Inference Prompt

The Task Inference prompt guides the Central Controller to analyze user queries and infer the appropriate task types from the supported task list. The detailed prompt is illustrated as follows.

```
Role Definition
You are RS-Agent, a remote sensing expert responsible for inferring the required task types from a user query. Your goal is to analyze the user
's remote-sensing analysis intent and select the minimal and sufficient set of task types from the supported task list.

Remote Sensing Reasoning Cues
When interpreting the query, consider the following remote-sensing factors:
1.Sensor/modality (optical, SAR, thermal, etc., or unknown);
2.Spatial granularity (object-level/scene-level/region-level);
3.Temporal setting (single-date, pre-post, or multi-temporal);
4.Analytical operation (detection, segmentation/mapping, classification, change analysis, severity/impact assessment, counting/statistics, or
description/QA).

Knowledge-Assisted Inference (Optional)
If the task type cannot be confidently inferred from the query alone, you may first perform a knowledge search to retrieve relevant remote-
sensing background information, and then re-evaluate the task type based on the acquired knowledge.
Prefer high-level analytical tasks over low-level processing steps, and avoid redundant or implied tasks.

Consistency Check
Before producing the final output, ensure that:
1.All task types are selected from supported_tasks;
2.The task set is minimal, non-redundant, and sufficient to satisfy the user's analytical intent.

Inputs:
- supported_tasks: {supported_tasks}
- user_query: {Q}

Output (JSON only):
{{"task_type":["task1","task2","task3"]}}
```

### A.2 Keyword Extraction Prompt

The Keyword Extraction prompt is used in the DualRAG mechanism to extract important keywords from user queries with weighted importance scores. This enables more effective knowledge retrieval by prioritizing remote sensing domain-specific terminology. The format of the keyword generation prompt is shown below:

* Corresponding author (email: xuwenjia@bupt.edu.cn, trentonwei@whu.edu.cn)

```
Role Definition
You are a remote sensing keyword extractor for RAG. From the query, extract concise, searchable keywords and assign importance (1-10).
Prioritize remote-sensing professional terminology and abbreviations; omit trivial words.

Extraction Principles
When extracting keywords, follow these principles:
- Prioritize remote-sensing professional terminology, standard abbreviations, and well-defined technical concepts;
- Omit trivial, conversational, or low-information words;

Remote Sensing Prioritization Order
Prefer keywords in the following descending order of importance: Task or analysis target; Analytical or processing concept; Sensor, modality,
or platform; Indices, variables, or domain-specific attributes; Temporal setting ; Spatial scale or resolution; Area of interest or location.

Importance Scoring Rules
Assign an importance score to each keyword:
9-10: core constraints essential for correct retrieval;
6-8: important qualifiers that refine the query;
3-5: broad contextual information;
Omit keywords with negligible retrieval value.

Inputs:
- user_query: {Q}
Output (JSON only):
{"keywords":[{"term":"...","importance":1-10}]}
```

## B Multi-Tool End-to-End Task Success Rate

To complement the task planning accuracy reported in Table 6 of the main text, we further evaluate multi-tool end-to-end task success rate, measured by final-answer correctness on the RSVQA-LR benchmark [1]. As shown in Table S1, the Baseline, $+T_{\text{inf}}$, and $+S_{\text{retr}}$ variants achieve comparable end-to-end accuracy (84.24%, 84.34%, and 84.57%, respectively), with notably low Rural/Urban accuracy (20%–29%), indicating that partial orchestration components alone are insufficient for complex multi-tool workflows. In contrast, the full RS-Agent ($+T_{\text{inf}}+S_{\text{retr}}$) achieves 90.88% overall, with Rural/Urban accuracy jumping to 97%, demonstrating that the synergistic combination of Task Inference and Solution Retrieval is essential for planning effective multi-step tool pipelines.

**Table S1** Multi-tool end-to-end task success rate on RSVQA-LR.

| Method | Rural/Urban | Presence | Compare | Avg. |
|---|---|---|---|---|
| Baseline | 20.00% | 87.23% | 83.62% | 84.24% |
| $+T_{\text{inf}}$ | 23.00% | 87.58% | 83.48% | 84.34% |
| $+S_{\text{retr}}$ | 29.00% | 87.58% | 83.73% | 84.57% |
| $+T_{\text{inf}}+S_{\text{retr}}$ | **97.00%** | **91.07%** | **90.58%** | **90.88%** |

## C Complete Task Planning Accuracy

Table 2 in the main text reports task planning accuracy on 10 representative tasks. Table S2 below provides the complete results for all 18 evaluation tasks. The ranking across LLMs remains consistent with the main-text subset.

## D Tool List

Table S3 provides a complete overview of the 27 tools integrated into the RS-Agent framework, organized by modality.

## E Multi-Tool Collaboration Example

This section demonstrates RS-Agent's capability to handle complex queries that require coordinated invocation of multiple tools. Figure S1 illustrates a complete workflow where the agent first infers the task type through Prompt1, retrieves task-specific guidance from the Solution Searcher, and then plans the tool execution sequence through Prompt2 to produce the final answer.

**Table S2** Complete Task Planning Accuracy of RS-Agent with Different LLMs (18 evaluation tasks).

| Task | ChatGPT | | | LLaMa 3.1 | | Qwen2.5 | | | DeepSeek |
|---|---|---|---|---|---|---|---|---|---|
| | 3.5-turbo-1106 (87.71t/s) | 3.5-turbo (65.03t/s) | 4o-mini (58.87t/s) | 8B (100.78t/s) | 70B (17.71t/s) | 14B (69.61t/s) | 32B (36.77t/s) | 72B (16.24t/s) | r1:70B (18.25t/s) |
| Cloud Removal | 95.00% | 95.00% | 100% | 100% | 100% | 100% | 95.00% | 100% | 100% |
| Image Dehazing | 30.00% | 95.00% | 100% | 100% | 100% | 100% | 100% | 100% | 75.00% |
| Super Resolution | 100% | 100% | 100% | 0% | 100% | 100% | 100% | 100% | 95.00% |
| Denoising | 90.00% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 90.00% |
| Image Captioning | 55.00% | 45.00% | 90.00% | 15.00% | 60.00% | 70.00% | 80.00% | 80.00% | 10.00% |
| Object Detection | 75.00% | 60.00% | 95.00% | 30.00% | 90.00% | 90.00% | 85.00% | 100% | 85.00% |
| Optical Plane Classif. | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 95.00% |
| Scene Classification | 20.00% | 90.00% | 100% | 80.00% | 90.00% | 90.00% | 100% | 100% | 50.00% |
| SAR Detection | 30.00% | 100% | 100% | 75.00% | 95.00% | 100% | 100% | 100% | 100% |
| SAR Plane Classif. | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 90.00% |
| Knowledge Search | 100% | 100% | 100% | 100% | 80.00% | 100% | 100% | 100% | 10.00% |
| Building Damage Det. | 100% | 100% | 100% | 100% | 100% | 95.00% | 100% | 100% | 100% |
| Building Extraction | 10.00% | 70.00% | 100% | 55.00% | 100% | 100% | 100% | 100% | 100% |
| Road Extraction | 15.00% | 55.00% | 100% | 65.00% | 100% | 100% | 100% | 100% | 100% |
| Horizontal Detection | 20.00% | 55.00% | 100% | 95.00% | 100% | 100% | 100% | 100% | 100% |
| Rotated Detection | 15.00% | 35.00% | 100% | 85.00% | 90.00% | 100% | 100% | 100% | 100% |
| Semantic Segmentation | 60.00% | 100% | 100% | 80.00% | 100% | 100% | 100% | 100% | 80.00% |
| Land Use Classif. | 15.00% | 100% | 100% | 75.00% | 100% | 100% | 100% | 95.00% | 95.00% |
| Average Accuracy | 57.22% | 82.50% | 99.17% | 75.28% | 94.72% | 96.94% | 97.78% | 98.61% | 81.94% |

[1] "B" denotes the number of parameters in billions.
[2] Numbers in parentheses (t/s) indicate model inference speed in tokens per second on NVIDIA RTX4090 GPUs.

**Table S3** Overview of the RS-Agent Toolkit. Tools are organized by sensing modality and analysis level.

| **Modality** | **Level** | **Tool Name** | **Tool Description** |
|---|---|---|---|
| Optical | Low-Level | Super Resolution (×2) | Enhance spatial resolution of optical imagery (2× upscaling) |
| | | Super Resolution (×4) | Enhance spatial resolution of optical imagery (4× upscaling) |
| | | Image Denoising | Remove noise from optical images |
| | | Cloud Removal | Remove cloud cover from optical imagery |
| | | Dehazing | Remove haze effects from optical imagery |
| | High-Level | Visual Question Answering | Answer natural-language questions about optical images |
| | | Optical Object Detection | Detect objects in optical imagery |
| | | Object Cropping | Crop and extract regions of interest from images |
| | | Aircraft Type Recognition | Classify aircraft types in optical images |
| | | Scene Classification | Classify scene categories of optical images |
| | | Aircraft Counting | Count the number of aircraft in optical images |
| | | Building Damage Assessment | Assess building damage levels from optical images |
| | | Building Extraction | Extract building footprints from optical imagery |
| | | Road Extraction | Extract road networks from optical imagery |
| | | Land-Cover Classification | Classify land cover types in optical images |
| | | Knowledge-Based Reasoning | Retrieve and reason over remote sensing knowledge |
| | | Horizontal Object Detection | Detect objects with horizontal bounding boxes |
| | | Rotated Object Detection | Detect objects with oriented bounding boxes |
| SAR | Low-Level | Despeckling | Remove speckle noise from SAR images |
| | High-Level | SAR Object Detection | Detect targets in SAR imagery |
| | | Aircraft Type Recognition | Classify aircraft types in SAR images |
| | | Rotated Detection | Detect objects with oriented bounding boxes in SAR imagery |
| | | Land-Cover Classification | Classify land cover types in SAR imagery |
| | | Semantic Segmentation | Perform pixel-level semantic segmentation on SAR images |
| | | Sea-Land Segmentation | Segment sea and land regions in SAR images |
| | | Change Detection | Detect changes between multi-temporal SAR images |
| | | Target Recognition | Recognize and classify targets in SAR images |

**User**

How many airplanes are parked in this image, and what are their categories?

**Prompt 1**

**Role Definition:**
You are RS-Agent, a remote sensing expert responsible for inferring the required task types from a user query. Your goal is to analyze the user's remote-sensing analysis intent and select the minimal and sufficient set of task types from the supported task list.
**Remote Sensing Reasoning Cues:**
When interpreting the query, consider the following remote-sensing factors:
1.Sensor/modality (optical, SAR, thermal, etc., or unknown);
2.Spatial granularity (object-level/scene-level/region-level);
3.Temporal setting (single-date, pre-post, or multi-temporal);
4.Analytical operation (detection, segmentation/mapping, classification, change analysis, severity/impact assessment, counting/statistics, or description/QA).
**Knowledge-Assisted Inference:**
If the task type cannot be confidently inferred from the query alone, you may first perform a knowledge search to retrieve relevant remote-sensing background information, and then re-evaluate the task type based on the acquired knowledge.
Prefer high-level analytical tasks over low-level processing steps, and avoid redundant or implied tasks.
**Consistency Check**
Before producing the final output, ensure that:
1.All task types are selected from supported_tasks;
2.The task set is minimal, non-redundant, and sufficient to satisfy the user's analytical intent.
**Inputs:**
- supported_tasks: {supported_tasks}
- user_query: {Q}
**Output (JSON only):**
{{"task_type":["task1","task2","task3"]}}
\end{lstlisting}

**RS-Agent**

{{"task_type":["Object Detection", "Plane Classification"]}}

**Solution Searcher**

**Input:**["Object Detection", "Plane Classification"]
**Output:** ["To solve this kind of task, you should directly use tool `classify_image' to determine image type, then use `optical_detection' or `sar_detection' to get answer.", "To solve this kind of task, you should directly use tool 'optical_plane_tyoe' to get answer"]

**Prompt 2**

You are an intelligent assistant named RS-Agent. Your task is to efficiently call tools and provide precise answers based on the user's needs. You can maintain context across multiple turns of conversation and directly answer the user's questions using the results from the tools. Please respond in the language the user is using, and ensure your answers are concise, natural, and appropriate, reflecting your professionalism and friendliness as an intelligent assistant.
**User Input:** {Query}
**Solution Guidance:** {Solution Guidance}
**Conversation History:** {conversation_history}

**RS-Agent**

The user wants to perform detection on an image located at `../test.jpg`. Before proceeding with the detection, it's important to determine the image type.
**Action:** classify_image
**Action Input:** ../test.jpg
To solve your problem, I need to call the tool classify_image, which is currently running...
I have got the output of the tool and am generating the answer...
**Observation:** Optical image
Thought: The image has been classified as an optical image. The next step is to perform detection on this optical image.
**Action:** optical_detection
**Action Input:** **../**test.jpg
To solve your problem, I need to call the tool optical_detection, which is currently running...
image 1/1 ../result.jpg: 1024x1024 162.5ms
**Observation:** {'message': '3 planes', 'extract_image_path': '../result.jpg', 'cropped_image_path': ['../plane0.jpg', '../plane1.jpg', '../plane2.jpg']}
Thought:The optical detection tool has identified that there are 3 planes in the image located at `../result.jpg`.
**Action:** optical_plane_type
**Action Input:** ['../plane0.jpg', '../plane1.jpg', '../plane2.jpg']
**Observation:** {'message': ['Boeing 737', 'Boeing 737', 'Boeing 737']
**Thought:** I have identified the number of planes and their types. I can now give the final answer.

**Final Answer:** The image contains 3 planes, and they are all Boeing 737.

**Figure S1** Illustration of the RS-Agent workflow. Prompt1 infers the task type, the Solution Searcher retrieves task-specific guidance, and Prompt2 plans tool execution to produce the final answer.

# F User Interface

This section presents the user interface of RS-Agent, which provides an intuitive platform for users to interact with the system through natural language. As shown in Figure S2, users can upload remote sensing images and perform various tasks such as object detection and building damage assessment by simply describing their analysis needs in plain language, without requiring expertise in remote sensing or programming.

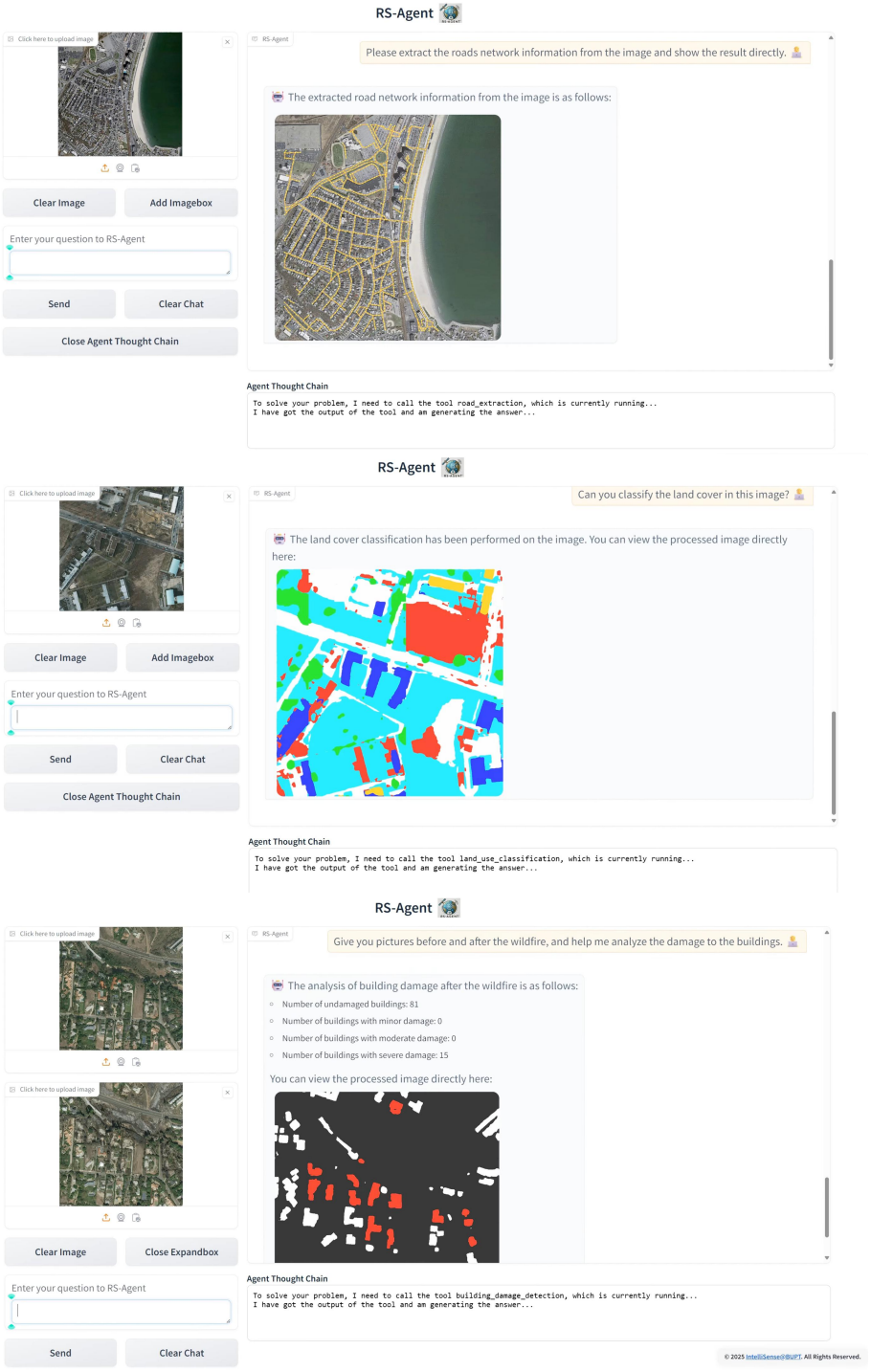


**Figure S2** The user interface of the RS-Agent. Users can upload remote sensing images and interpret remote sensing data through natural language. The example shows object detection and building damage detection task.

**References**

1 Lobry S, Marcos D, Murray J, et al. Rsvqa: Visual question answering for remote sensing data. IEEE Transactions on Geoscience and Remote Sensing, 2020, 58: 8555–8566