

CN PROJECT REPORT(CSC611)

Team-14

**(Image Description Generation Using Client-Server
Connection with Different Networks)**

Submitted by

Prithwish Dey (CSE/22065/919)

Amit Mandhana (CSE/22014/868)

Anindya Bhaumik (CSE/22018/872)

Aditya Paul (CSE/22004/858)

Chirag Shukla (CSE/22038/892)

Adrish Roy (CSE/22007/861)

Akash Chauhan (CSE/22009/863)

Bachelor of Technology in Computer Science and
Engineering.



Indian Institute of Information Technology Kalyani, West Bengal,
India, 741235, Spring 2025.

Implementation and Analysis of Image Description Generation Using Client-Server Connection Over Email Protocols

Demo Video Link: [📺 Image Description Generation Using Client - Server Connection ...](#)

Our Github Link:

❖ Introduction

As remote communications and automated systems continue to advance rapidly, this project is looking to see if it is possible to generate **image descriptions through a client-server system** over standard email protocols. The main idea emphasized is transferring image data from one computer (the **server**) to another computer (the **client**) through the **SMTP** and **POP** protocols. The server receives the image sent by the client, processes the image through a **classification model**, and sends back a response containing the generated description.

❖ Aim of the Project

In order to create a **distributed system** that utilizes client computers to send image data to a client computer that is located in **another network**. The client will classify the image using a trained model and send the classification result back to the client making the system realistic in **simulating a real-time**, fully autonomous image understanding pipeline.

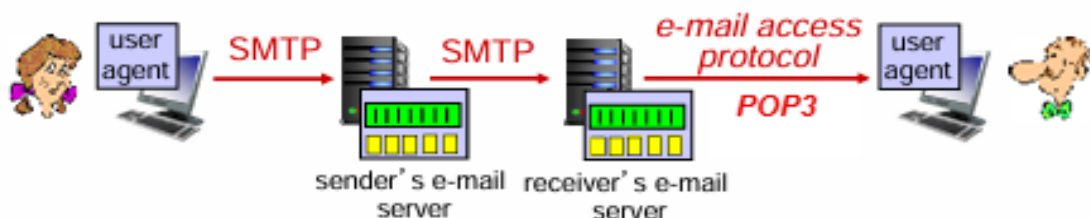
● Reasons for Choosing This Topic

1. **Real-world Application:** Email protocols are **universally accessible**, making this system applicable in any environment with basic internet connectivity.
2. **Flexibility and Independence:** As the system operates over common internet protocols, it is very **flexible** and not dependent on any particular hardware.

3. **Educational Value:** It integrates **machine learning (image classification)**, networking protocols (**SMTP/POP**), and designing of system (**client-server model**).
4. **Scalability:** The system can be extended with **integration of cloud service or multi-client communication**.

❖ SMTP & POP Protocols

- **SMTP (Simple Mail Transfer Protocol):** The client uses the Simple Mail Transfer Protocol (SMTP) protocol for sending an image file as an email attachment. **TCP/IP** is used by SMTP for **reliable delivery of messages**.
- **POP (Post Office Protocol):** The server system uses POP (Post Office Protocol) to fetch messages from the **mail server**. To extract the image data, Emails can be parsed.

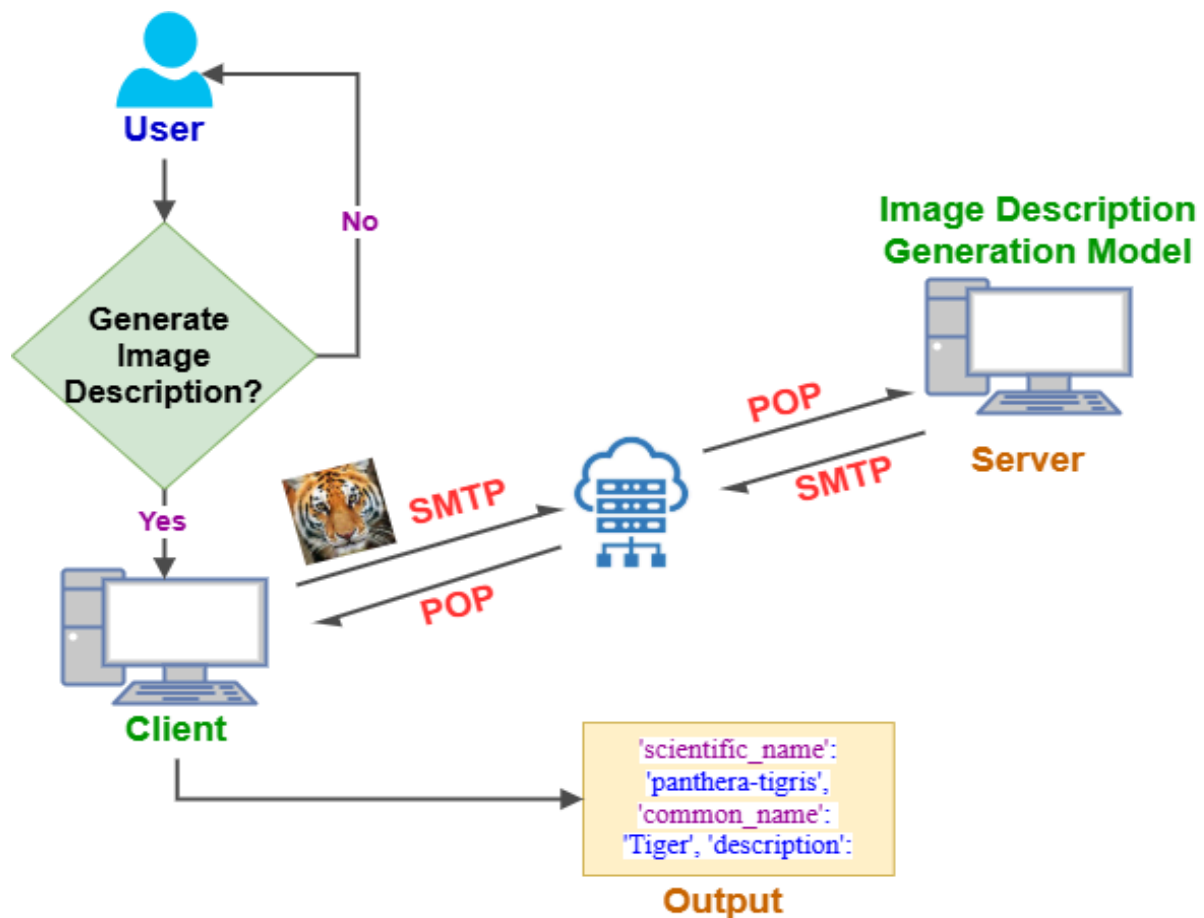


These protocols are used here as a **simple and universal transport mechanism** for sending files between **two distant systems**, without the need for special servers, rather than to **communicate via human-readable email**.

❖ System Architecture

- **Client(PC 1):** Captures or selects an image by webcam, sends it to the client using the SMTP protocol.
- **Server (PC 2):** Receives the image using POP, classifies it with a machine learning model, and sends the classification result back to the server.

Once the programs are operating, there is **no need for manual intervention** because the entire exchange is **self-sufficient**. The **block diagram** illustrates it.



❖ Result

The server processes and classifies the images after the system successfully sends them via email from the client. For example, the server reponses with the following when a **Tiger** image is sent:

```

{"scientific_name": 'panthera-tigris', 'common_name': 'Tiger',
'description': 'The largest of the big cats, recognized by its orange
coat with black stripes.', 'habitat': 'Forests, grasslands, and
wetlands across Asia', 'endangered': 'endangered', 'dangerous':
'True', 'poisonous': 'False', 'venomous': 'False', 'probability':
97.7931797504425}
  
```

This shows the **accuracy and dependability** of the model in **real-time real classification**.

❖ **Conclusion**

This project highlights a simple but powerful way of performing remote photo classification using standard network protocols. It demonstrates how email can be utilized as transportation for basic inter-device communication functions, establishing a foundation for use in remote regions or low-resource settings. The server-client pipeline was effective to cinematize the flow of an image, while leaving an excess amount of image classification possible for accuracy and responsiveness.

❖ **Future Work**

- **Include encryption** and layers of security in the data transfer.
- **Implement more complex models** like object detection or scene understanding.
- **Permit multi-client support** for collaborative classification systems.
- **Develop a user interface (UI)** to track communication logs and outputs in real-time.

-----*The End*-----