```
import os

from dotenv import load_dotenv
from termcolor import colored

import swarms.prompts.security_team as stsp
from swarm_models import GPT4VisionAPI
from swarms.structs import Agent

# Load environment variables and initialize the Vision API
load_dotenv()
api_key = os.getenv("OPENAI_API_KEY")

llm = GPT4VisionAPI(openai_api_key=api_key)

# Image for analysis
# img = "IMG_1617.jpeg"
img = "ubase1.jpeg"
img2 = "ubase2.jpeg"

# Initialize agents with respective prompts for security tasks
crowd_analysis_agent = Agent(
    agent_name="Crowd Analysis Agent",
    llm=llm,
    sop=stsp.CROWD_ANALYSIS_AGENT_PROMPT,
    max_loops=1,
```

```
    multi_modal=True,
)

weapon_detection_agent = Agent(
    agent_name="Weapon Detection Agent",
    llm=llm,
    sop=stsp.WEAPON_DETECTION_AGENT_PROMPT,
    max_loops=1,
    multi_modal=True,
)

surveillance_monitoring_agent = Agent(
    agent_name="Surveillance Monitoring Agent",
    llm=llm,
    sop=stsp.SURVEILLANCE_MONITORING_AGENT_PROMPT,
    max_loops=1,
    multi_modal=True,
)

emergency_response_coordinator = Agent(
    agent_name="Emergency Response Coordinator",  # "Emergency Response Coordinator
    llm=llm,
    sop=stsp.EMERGENCY_RESPONSE_COORDINATOR_PROMPT,
    max_loops=1,
    multi_modal=True,
)
```

```
colored("Security Team Analysis", "green")
colored("Inspect the scene for any potential threats", "green")
colored("Weapon Detection Analysis", "green")
weapon_detection_analysis = weapon_detection_agent.run(
    "Inspect the scene for any potential threats", img
)


colored("Surveillance Monitoring Analysis", "cyan")
surveillance_monitoring_analysis = surveillance_monitoring_agent.run(
    "Monitor the overall scene for unusual activities", img
)

colored("Emergency Response Analysis", "red")
emergency_response_analysis = emergency_response_coordinator.run(
    "Develop a response plan based on the scene analysis", img
)
```