

```
import os
```

```
from dotenv import load_dotenv
```

```
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 = "bank_robbery.jpg"
```

```
# Initialize agents with respective prompts for security tasks
```

```
crowd_analysis_agent = Agent(
```

```
    llm=llm,
```

```
    sop=stsp.CROWD_ANALYSIS_AGENT_PROMPT,
```

```
    max_loops=1,
```

```
    multi_modal=True,
```

```
)
```

```
weapon_detection_agent = Agent(
```

```

    llm=llm,

    sop=stsp.WEAPON_DETECTION_AGENT_PROMPT,

    max_loops=1,

    multi_modal=True,

)

surveillance_monitoring_agent = Agent(

    llm=llm,

    sop=stsp.SURVEILLANCE_MONITORING_AGENT_PROMPT,

    max_loops=1,

    multi_modal=True,

)

emergency_response_coordinator = Agent(

    llm=llm,

    sop=stsp.EMERGENCY_RESPONSE_COORDINATOR_PROMPT,

    max_loops=1,

    multi_modal=True,

)

# Run agents with respective tasks on the same image

crowd_analysis = crowd_analysis_agent.run(

    "Analyze the crowd dynamics in the scene", img

)

weapon_detection_analysis = weapon_detection_agent.run(

```

```
"Inspect the scene for any potential threats", img
)

surveillance_monitoring_analysis = surveillance_monitoring_agent.run(
    "Monitor the overall scene for unusual activities", img
)

emergency_response_analysis = emergency_response_coordinator.run(
    "Develop a response plan based on the scene analysis", img
)

# Process and output results for each task

# Example output (uncomment to use):
print(f"Crowd Analysis: {crowd_analysis}")
print(f"Weapon Detection Analysis: {weapon_detection_analysis}")
print(
    "Surveillance Monitoring Analysis:"
    f" {surveillance_monitoring_analysis}"
)
print(f"Emergency Response Analysis: {emergency_response_analysis}")
```