"""

Swarm of multi modal autonomous agents for manufacturing!

Health Security agent: Agent that monitors the health of working conditions: input image of factory output: health safety index 0.0 - 1.0 being the highest

Quality Control agent: Agent that monitors the quality of the product: input image of product output: quality index 0.0 - 1.0 being the highest

Productivity agent: Agent that monitors the productivity of the factory: input image of factory output: productivity index 0.0 - 1.0 being the highest

Safety agent: Agent that monitors the safety of the factory: input image of factory output: safety index 0.0 - 1.0 being the highest

Security agent: Agent that monitors the security of the factory: input image of factory output: security index 0.0 - 1.0 being the highest

Sustainability agent: Agent that monitors the sustainability of the factory: input image of factory output: sustainability index 0.0 - 1.0 being the highest

Efficiency agent: Agent that monitors the efficiency of the factory: input image of factory output: efficiency index 0.0 - 1.0 being the highest

Agent:

health security agent -> quality control agent -> productivity agent -> safety agent -> security agent -> sustainability agent -> efficiency agent

"""

import os

```
from dotenv import load_dotenv
from termcolor import colored

from swarm_models import GPT4VisionAPI
from swarms.structs import Agent

load_dotenv()
api_key = os.getenv("OPENAI_API_KEY")

# GPT4VisionAPI
llm = GPT4VisionAPI(openai_api_key=api_key, max_tokens=2000)

assembly_line = (
    "examples/demos/swarm_of_mma_manufacturing/assembly_line.jpg"
)
red_robots = (
    "examples/demos/swarm_of_mma_manufacturing/red_robots.jpg"
)
robots = "examples/demos/swarm_of_mma_manufacturing/robots.jpg"
tesla_assembly_line = (
    "examples/demos/swarm_of_mma_manufacturing/tesla_assembly.jpg"
)


# Define detailed prompts for each agent
tasks = {
```

```
"health_safety": (
    "Analyze the factory's working environment for health safety."
    " Focus on cleanliness, ventilation, spacing between"
    " workstations, and personal protective equipment"
    " availability."
),
"productivity": (
    "Review the factory's workflow efficiency, machine"
    " utilization, and employee engagement. Identify operational"
    " delays or bottlenecks."
),
"safety": (
    "Analyze the factory's safety measures, including fire exits,"
    " safety signage, and emergency response equipment."
),
"security": (
    "Evaluate the factory's security systems, entry/exit"
    " controls, and potential vulnerabilities."
),
"sustainability": (
    "Inspect the factory's sustainability practices, including"
    " waste management, energy usage, and eco-friendly processes."
),
"efficiency": (
    "Assess the manufacturing process's efficiency, considering"
    " the layout, logistics, and automation level."
```

```
    ),
}



# Define prompts for each agent
health_safety_prompt = tasks["health_safety"]
productivity_prompt = tasks["productivity"]
safety_prompt = tasks["safety"]
security_prompt = tasks["security"]
sustainability_prompt = tasks["sustainability"]
efficiency_prompt = tasks["efficiency"]



# Health security agent
health_security_agent = Agent(
    llm=llm,
    sop_list=health_safety_prompt,
    max_loops=1,
    multi_modal=True,
)

# Quality control agent
productivity_check_agent = Agent(
    llm=llm,
    sop=productivity_prompt,
    max_loops=1,
```

```
    multi_modal=True,
    autosave=True,
)

# Security agent
security_check_agent = Agent(
    llm=llm,
    sop=security_prompt,
    max_loops=1,
    multi_modal=True,
    autosave=True,
)

# Efficiency agent
efficiency_check_agent = Agent(
    llm=llm,
    sop=efficiency_prompt,
    max_loops=1,
    multi_modal=True,
    autosave=True,
)

print(colored("Running the agents...", "green"))


print(colored("Running health check agent initializing...", "cyan"))
```

```
# Add the first task to the health_security_agent
health_check = health_security_agent.run(
    "Analyze the safety of this factory", robots
)



print(
    colored(
        "--------------- Productivity agents initializing...", "green"
    )
)
# Add the third task to the productivity_check_agent
productivity_check = productivity_check_agent.run(
    health_check, assembly_line
)


print(
    colored(
        "--------------- Security agents initializing...", "green"
    )
)
# Add the fourth task to the security_check_agent
security_check = security_check_agent.run(
    productivity_check, red_robots
)
```

```
print(
    colored(
        "---------- Efficiency agents initializing...", "cyan"
    )
)
# Add the fifth task to the efficiency_check_agent
efficiency_check = efficiency_check_agent.run(
    security_check, tesla_assembly_line
)
```