```python
import os

from dotenv import load_dotenv

from swarm_models import GPT4VisionAPI, OpenAIChat
from swarms.prompts.xray_swarm_prompt import (
    TREATMENT_PLAN_PROMPT,
    XRAY_ANALYSIS_PROMPT,
)
from swarms.structs.agent import Agent

# Load environment variables
load_dotenv()
openai_api_key = os.getenv("OPENAI_API_KEY")

# Function to analyze an X-ray image
multimodal_llm = GPT4VisionAPI(
    openai_api_key=openai_api_key,
)

# Initialize Language Model (LLM)
llm = OpenAIChat(
    openai_api_key=openai_api_key,
    max_tokens=3000,
)
```

```
# Function to analyze an X-ray image

analyze_xray_agent = Agent(

    llm=multimodal_llm,

    autosave=True,

    sop=XRAY_ANALYSIS_PROMPT,

    multi_modal=True,

)



# Treatment Plan Agent

treatment_agent = Agent(

    llm=multimodal_llm,

    autosave=True,

    sop=TREATMENT_PLAN_PROMPT,

    max_loops=4,

)



# Function to generate a treatment plan

def generate_treatment_plan(diagnosis):

    treatment_plan_prompt = TREATMENT_PLAN_PROMPT.format(diagnosis)

    # Using the llm object with the 'prompt' argument

    return treatment_agent.run(treatment_plan_prompt)
```

```
# X-ray Agent - Analyze an X-ray image
xray_image_path = "examples/demos/xray/xray2.jpg"


# Diagnosis
diagnosis = analyze_xray_agent.run(
    task="Analyze the following XRAY", img=xray_image_path
)

# Generate Treatment Plan
treatment_plan_output = generate_treatment_plan(diagnosis)

# Print and save the outputs
print("X-ray Analysis:", diagnosis)
print("Treatment Plan:", treatment_plan_output)

with open("medical_analysis_output.txt", "w") as file:
    file.write("X-ray Analysis:\n" + diagnosis + "\n\n")
    file.write("Treatment Plan:\n" + treatment_plan_output + "\n")

print("Outputs have been saved to medical_analysis_output.txt")
```