```
import os
from unittest.mock import MagicMock, Mock, patch

import pytest
import torch
from PIL import Image
from transformers import NougatProcessor, VisionEncoderDecoderModel

from swarm_models.nougat import Nougat


@pytest.fixture
def setup_nougat():
    return Nougat()


def test_nougat_default_initialization(setup_nougat):
    assert setup_nougat.model_name_or_path == "facebook/nougat-base"
    assert setup_nougat.min_length == 1
    assert setup_nougat.max_new_tokens == 30


def test_nougat_custom_initialization():
    nougat = Nougat(
        model_name_or_path="custom_path",
        min_length=10,
```

```
        max_new_tokens=50,
    )
    assert nougat.model_name_or_path == "custom_path"
    assert nougat.min_length == 10
    assert nougat.max_new_tokens == 50


def test_processor_initialization(setup_nougat):
    assert isinstance(setup_nougat.processor, NougatProcessor)


def test_model_initialization(setup_nougat):
    assert isinstance(setup_nougat.model, VisionEncoderDecoderModel)


@pytest.mark.parametrize(
    "cuda_available, expected_device",
    [(True, "cuda"), (False, "cpu")],
)
def test_device_initialization(
    cuda_available, expected_device, monkeypatch
):
    monkeypatch.setattr(
        torch,
        "cuda",
        Mock(is_available=Mock(return_value=cuda_available)),
```

```
    )
    nougat = Nougat()
    assert nougat.device == expected_device


def test_get_image_valid_path(setup_nougat):
    with patch("PIL.Image.open") as mock_open:
        mock_open.return_value = Mock(spec=Image.Image)
        assert setup_nougat.get_image("valid_path") is not None


def test_get_image_invalid_path(setup_nougat):
    with pytest.raises(FileNotFoundError):
        setup_nougat.get_image("invalid_path")


@pytest.mark.parametrize(
    "min_len, max_tokens",
    [
        (1, 30),
        (5, 40),
        (10, 50),
    ],
)
def test_model_call_with_diff_params(
    setup_nougat, min_len, max_tokens
```

```
):
    setup_nougat.min_length = min_len
    setup_nougat.max_new_tokens = max_tokens

    with patch("PIL.Image.open") as mock_open:
        mock_open.return_value = Mock(spec=Image.Image)
        # Here, mocking other required methods or adding more complex logic would be necessary.
        result = setup_nougat("valid_path")
        assert isinstance(result, str)


def test_model_call_invalid_image_path(setup_nougat):
    with pytest.raises(FileNotFoundError):
        setup_nougat("invalid_path")


def test_model_call_mocked_output(setup_nougat):
    with patch("PIL.Image.open") as mock_open:
        mock_open.return_value = Mock(spec=Image.Image)
        mock_model = MagicMock()
        mock_model.generate.return_value = "mocked_output"
        setup_nougat.model = mock_model

        result = setup_nougat("valid_path")
        assert result == "mocked_output"
```

```
@pytest.fixture
def mock_processor_and_model():
    """Mock the NougatProcessor and VisionEncoderDecoderModel to simulate their behavior."""
    with patch(
        "transformers.NougatProcessor.from_pretrained",
        return_value=Mock(),
    ), patch(
        "transformers.VisionEncoderDecoderModel.from_pretrained",
        return_value=Mock(),
    ):
        yield



@pytest.mark.usefixtures("mock_processor_and_model")
def test_nougat_with_sample_image_1(setup_nougat):
    result = setup_nougat(
        os.path.join(
            "sample_images",

"https://plus.unsplash.com/premium_photo-1687149699194-0207c04bc6e8?auto=format&fit=crop&
q=80&w=1378&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D
%3D",
        )
    )
    assert isinstance(result, str)
```

```
@pytest.mark.usefixtures("mock_processor_and_model")
def test_nougat_with_sample_image_2(setup_nougat):
    result = setup_nougat(os.path.join("sample_images", "test2.png"))
    assert isinstance(result, str)


@pytest.mark.usefixtures("mock_processor_and_model")
def test_nougat_min_length_param(setup_nougat):
    setup_nougat.min_length = 10
    result = setup_nougat(
        os.path.join(
            "sample_images",

"https://plus.unsplash.com/premium_photo-1687149699194-0207c04bc6e8?auto=format&fit=crop&
q=80&w=1378&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D
%3D",
        )
    )
    assert isinstance(result, str)


@pytest.mark.usefixtures("mock_processor_and_model")
def test_nougat_max_new_tokens_param(setup_nougat):
    setup_nougat.max_new_tokens = 50
```

```
    result = setup_nougat(
        os.path.join(
            "sample_images",

"https://plus.unsplash.com/premium_photo-1687149699194-0207c04bc6e8?auto=format&fit=crop&
q=80&w=1378&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D
%3D",
        )
    )
    assert isinstance(result, str)


@pytest.mark.usefixtures("mock_processor_and_model")
def test_nougat_different_model_path(setup_nougat):
    setup_nougat.model_name_or_path = "different/path"
    result = setup_nougat(
        os.path.join(
            "sample_images",

"https://plus.unsplash.com/premium_photo-1687149699194-0207c04bc6e8?auto=format&fit=crop&
q=80&w=1378&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D
%3D",
        )
    )
    assert isinstance(result, str)
```

```
@pytest.mark.usefixtures("mock_processor_and_model")
def test_nougat_bad_image_path(setup_nougat):
    with pytest.raises(
        Exception
    ):  # Adjust the exception type accordingly.
        setup_nougat("bad_image_path.png")



@pytest.mark.usefixtures("mock_processor_and_model")
def test_nougat_image_large_size(setup_nougat):
    result = setup_nougat(
        os.path.join(
            "sample_images",

"https://images.unsplash.com/photo-1697641039266-bfa00367f7cb?auto=format&fit=crop&q=60&w
=400&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHx0b3BpYy1mZWVkfDJ8SnBnNktpZGwtSGt8fGVufDB
8fHx8fA%3D%3D",
        )
    )
    assert isinstance(result, str)



@pytest.mark.usefixtures("mock_processor_and_model")
def test_nougat_image_small_size(setup_nougat):
    result = setup_nougat(
```

```
        os.path.join(
            "sample_images",

"https://images.unsplash.com/photo-1697638626987-aa865b769276?auto=format&fit=crop&q=60&w=400&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHx0b3BpYy1mZWVkfDd8SnBnNktpZGwtSGt8fGVufDB8fHx8fA%3D%3D",
        )
    )
    assert isinstance(result, str)


@pytest.mark.usefixtures("mock_processor_and_model")
def test_nougat_image_varied_content(setup_nougat):
    result = setup_nougat(
        os.path.join(
            "sample_images",

"https://images.unsplash.com/photo-1697469994783-b12bbd9c4cff?auto=format&fit=crop&q=60&w=400&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHx0b3BpYy1mZWVkfDE0fEpwZzZLaWRsLUhrfHxlbnwwfHx8fHw%3D",
        )
    )
    assert isinstance(result, str)


@pytest.mark.usefixtures("mock_processor_and_model")
```

```
def test_nougat_image_with_metadata(setup_nougat):
    result = setup_nougat(
        os.path.join(
            "sample_images",

"https://images.unsplash.com/photo-1697273300766-5bbaa53ec2f0?auto=format&fit=crop&q=60&w
=400&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHx0b3BpY1mZWVkfDE5fEpwZzZLaWRsLUhrfHxlbnw
wfHx8fHw%3D",
        )
    )
    assert isinstance(result, str)
```