In [24]:
```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Shows how to send multiple images with the Converse API with an accompanying text prompt to Anthropic Claude 3 Sonn
"""

import logging
import boto3
import os
import time

from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logging.basicConfig(level=logging.INFO)

def generate_conversation(bedrock_client,
                          model_id,
                          input_text,
                          input_image,
                          num_images=20):
    """
    Sends a message to a model with multiple copies of the same image.
    """

    logger.info("Generating message with model %s and %d copies of %s", model_id, num_images, input_image)

    # Get image extension and read in image as bytes
    image_ext = input_image.split(".")[-1]
    with open(input_image, "rb") as f:
        image_bytes = f.read()

    # Create content array starting with text
    content = [
        {
            "text": input_text
        }
    ]
```

```
    # Add the same image multiple times
    for i in range(num_images):
        content.append({
            "image": {
                "format": image_ext,
                "source": {
                    "bytes": image_bytes
                }
            }
        })

    message = {
        "role": "user",
        "content": content
    }

    messages = [message]

    # Send the message.
    response = bedrock_client.converse(
        modelId=model_id,
        messages=messages
    )

    return response

def test_image_file(bedrock_client, model_id, image_path, num_images):
    """Test a single image file with specified number of copies"""

    # Get file size for reporting
    file_size = os.path.getsize(image_path) / (1024 * 1024)  # MB

    print(f"\n{'='*60}")
    print(f"Testing: {image_path}")
    print(f"File size: {file_size:.2f} MB")
    print(f"Number of copies: {num_images}")
    print(f"{'='*60}")

    input_text = f"What's in these {num_images} images? (They're all the same {os.path.basename(image_path)} for te

    try:
```

```
        response = generate_conversation(
            bedrock_client, model_id, input_text, image_path, num_images)

        output_message = response['output']['message']

        print(f"✅ SUCCESS!")
        print(f"Role: {output_message['role']}")

        for content in output_message['content']:
            print(f"Text: {content['text'][:200]}...")  # Truncate long responses

        token_usage = response['usage']
        print(f"Input tokens:  {token_usage['inputTokens']}")
        print(f"Output tokens:  {token_usage['outputTokens']}")
        print(f"Total tokens:  {token_usage['totalTokens']}")
        print(f"Tokens per image: ~{(token_usage['inputTokens'] - 20) / num_images:.0f}")  # Rough calc
        print(f"Stop reason: {response['stopReason']}")

        return True, token_usage

    except ClientError as err:
        message = err.response['Error']['Message']
        logger.error("A client error occurred: %s", message)
        print(f"❌ FAILED: {message}")
        return False, None

# Initialize client
model_id = "anthropic.claude-3-sonnet-20240229-v1:0"
bedrock_client = boto3.client(service_name="bedrock-runtime")

print("Setup complete! Ready to test images.")
```

```
Setup complete! Ready to test images.
```

In [25]:

```
print("Testing under1mb.jpeg with 20 copies...")
success1, tokens1 = test_image_file(bedrock_client, model_id, "under1mb.jpeg", 20)
```

```
INFO:__main__:Generating message with model anthropic.claude-3-sonnet-20240229-v1:0 and 20 copies of under1mb.jpeg
```

```
Testing under1mb.jpeg with 20 copies...

============================================================
Testing: under1mb.jpeg
File size: 0.85 MB
Number of copies: 20
============================================================
✅ SUCCESS!
Role: assistant
Text: The image shows a scenic natural pool or cenote surrounded by rocky cliffs. The water in the pool is a strik
ing turquoise-green color. There are many people swimming and wading in the pool or gathered...
Input tokens:  30811
Output tokens:  99
Total tokens:  30910
Tokens per image: ~1540
Stop reason: end_turn
```

In [26]:
```
print("Testing over1mb.jpeg with 20 copies...")
success2, tokens2 = test_image_file(bedrock_client, model_id, "over1mb.jpeg", 20)
```

```
INFO:__main__:Generating message with model anthropic.claude-3-sonnet-20240229-v1:0 and 20 copies of over1mb.jpeg
Testing over1mb.jpeg with 20 copies...

============================================================
Testing: over1mb.jpeg
File size: 1.31 MB
Number of copies: 20
============================================================
ERROR:__main__:A client error occurred: Input is too long for requested model.
❌ FAILED: Input is too long for requested model.
```

In [32]:
```
print("Testing 3mb.jpeg with 7 copies...")
success3, tokens3 = test_image_file(bedrock_client, model_id, "3mb.jpeg", 7)
```

```
INFO:__main__:Generating message with model anthropic.claude-3-sonnet-20240229-v1:0 and 7 copies of 3mb.jpeg
```

```
Testing 3mb.jpeg with 7 copies...

============================================================
Testing: 3mb.jpeg
File size: 2.48 MB
Number of copies: 7
============================================================
✅ SUCCESS!
Role: assistant
Text: The images depict a scenic natural swimming pool located inside a large rocky cave or cavern. The pool has v
ibrant blue-green water, with rocky ledges and formations surrounding it.

There are many pe...
Input tokens:  10804
Output tokens:  168
Total tokens:  10972
Tokens per image: ~1541
Stop reason: end_turn
```

In [30]:
```
print("Testing 3mb.jpeg with 8 copies...")
success4, tokens4 = test_image_file(bedrock_client, model_id, "3mb.jpeg", 8)
```

```
INFO:__main__:Generating message with model anthropic.claude-3-sonnet-20240229-v1:0 and 8 copies of 3mb.jpeg
Testing 3mb.jpeg with 8 copies...

============================================================
Testing: 3mb.jpeg
File size: 2.48 MB
Number of copies: 8
============================================================
ERROR:__main__:A client error occurred: Input is too long for requested model.
❌ FAILED: Input is too long for requested model.
```

In [33]:
```
print(f"\n{'='*60}")
print("FINAL SUMMARY:")
print(f"{'='*60}")

results = [
    ("under1mb.jpeg (20 copies)", success1, tokens1),
    ("over1mb.jpeg (20 copies)", success2, tokens2),
```

```
    ("3mb.jpeg (7 copies)", success3, tokens3),
    ("3mb.jpeg (8 copies)", success4, tokens4)
]

for test_name, success, token_usage in results:
    status_icon = "✅" if success else "❌"
    print(f"{status_icon} {test_name}: {'Success' if success else 'Failed'}")

    if success and token_usage:
        print(f"   Input tokens: {token_usage['inputTokens']}")
        print(f"   Total tokens: {token_usage['totalTokens']}")

print(f"\nAll tests completed with model {model_id}.")
```

```
============================================================
FINAL SUMMARY:
============================================================
✅ under1mb.jpeg (20 copies): Success
   Input tokens: 30811
   Total tokens: 30910
❌ over1mb.jpeg (20 copies): Failed
✅ 3mb.jpeg (7 copies): Success
   Input tokens: 10804
   Total tokens: 10972
❌ 3mb.jpeg (8 copies): Failed

All tests completed with model anthropic.claude-3-sonnet-20240229-v1:0.
```