

Importing Image Descripton module with Rorchaque VQA

In [2]:

```
from imgdescbackend import process_image
```

```
# Example usage
```

```
image_path = r"cards/Rorschach_blot_08.jpg"
```

```
result = process_image(image_path)
```

```
print(result)
```

```
D:\Projects\Absolute image description\Main\venv\lib\site-packages\transformers\generation\utils.py:1273: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum length of the generation.
```

```
warnings.warn(
```

```
{'descriptions_blip': 'a painting of a flower on a white background', 'descriptions_ved': 'a painting of an animal with a flower in it', 'descriptions_noamrot': 'the image features a red crab and a red and pink flower in the foreground, with a white and blue sky in the background the caption suggests that the image is related to a painting', 'What do you see in the image?': 'banana', 'Where in the image does your attention focus the most?': 'face', 'What features or elements in the image influenced your perception?': 'face', 'Are there any common or recognizable elements in the image?': 'yes', 'How would you describe the overall style or characteristics of the image?': 'both'}
```

In [3]:

```
print(result["descriptions_blip"])
```

```
print(result["descriptions_ved"])
```

```
print(result["descriptions_noamrot"])
```

a painting of a flower on a white background

a painting of an animal with a flower in it

the image features a red crab and a red and pink flower in the foreground, with a white and blue sky in the background the caption suggests that the image is related to a painting

In [4]:

```
def dict_to_string(input_dict):
    result_str = ""
    for key, value in input_dict.items():
        result_str += f"{key}: {value}\n"
    return result_str
```

```
formatted_str = dict_to_string(result)
```

```
print(formatted_str)
```

descriptions_blip: a painting of a flower on a white background

descriptions_ved: a painting of an animal with a flower in it

descriptions_noamrot: the image features a red crab and a red and pink flower in the foreground, with a white and blue sky in the background the caption suggests that the image is related to a painting

What do you see in the image?: banana

Where in the image does your attention focus the most?: face

What features or elements in the image influenced your perception?: face

Are there any common or recognizable elements in the image?: yes

How would you describe the overall style or characteristics of the image?: both

Importing llama2 module

In [5]:

```
from llama2backend import generatetext
```

```
from llama2backend import generatetext
```

```
prompt = f""" Generate an absolute merged description of the following image descriptions
:
    {formatted_str}
    Now, provide a single, comprehensive description that merges all the informat
ion from the individual descriptions above.
    """

# Example usage
result = generatetext(prompt)
print(result)
```

Llama.generate: prefix-match hit

Absolutely! Here is a merged description of the image based on the given instructions: In the image, we see a painting of a flower on a white background, with an animal (animal) featuring a red crab and a red and pink flower in the foreground, against a white and blue sky in the background. The caption suggests that the image is related to a painting. Our attention is drawn to the face of the crab, which seems to be the most prominent feature in the image. The image's style or characteristics are both recognizable and unique, with a mix of vibrant colors and soft brushstrokes that give it a distinctive look. Additionally, there is a common element of nature present in the image, specifically the flower and the crab, which adds to its overall aesthetic appeal.

Rorchaque based Image description generator

In [1]:

```
%%time

from imgdescbackend import process_image

# Example usage
image_path = r"cards/Rorschach_blot_08.jpg"
resultdict = process_image(image_path)

print(resultdict)
print(resultdict["descriptions_blip"])
print(resultdict["descriptions_ved"])
print(resultdict["descriptions_noamrot"])

def dict_to_string(input_dict):
    result_str = ""
    for key, value in input_dict.items():
        result_str += f"{key}: {value}\n"
    return result_str

formatted_str = dict_to_string(resultdict)
from llama2backend import generatetext

prompt = f""" Generate an absolute merged description of the following image descriptions
:
    {formatted_str}
    Now, provide a single, comprehensive description that merges all the informat
ion from the individual descriptions above.
    """

# Example usage
result = generatetext(prompt)
print(result)
```

D:\Projects\Absolute image description\Main\venv\lib\site-packages\transformers\models\vit\feature_extraction_vit.py:28: FutureWarning: The class ViTFeatureExtractor is deprecated and will be removed in version 5 of Transformers. Please use ViTImageProcessor instead.
warnings.warn(
D:\Projects\Absolute image description\Main\venv\lib\site-packages\transformers\generation\utils.py:1273: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum length of the generation.
warnings.warn(

We strongly recommend passing in an ``attention_mask`` since your `input_ids` may be padded. See <https://huggingface.co/docs/transformers/troubleshooting#incorrect-output-when-padding-tokens-arent-masked>.

You may ignore this warning if your ``pad_token_id`` (50256) is identical to the ``bos_token_id`` (50256), ``eos_token_id`` (50256), or the ``sep_token_id`` (None), and your input is not padded.

```
{'descriptions_blip': 'a painting of a flower on a white background', 'descriptions_ved': 'a painting of an animal with a flower in it', 'descriptions_noamrot': 'the image features a red crab and a red and pink flower in the foreground, with a white and blue sky in the background the caption suggests that the image is related to a painting', 'What do you see in the image?': 'banana', 'Where in the image does your attention focus the most?': 'face', 'What features or elements in the image influenced your perception?': 'face', 'Are there any common or recognizable elements in the image?': 'yes', 'How would you describe the overall style or characteristics of the image?': 'both'}
a painting of a flower on a white background
a painting of an animal with a flower in it
the image features a red crab and a red and pink flower in the foreground, with a white and blue sky in the background the caption suggests that the image is related to a painting
```

```
AVX = 1 | AVX2 = 1 | AVX512 = 0 | AVX512_VBMI = 0 | AVX512_VNNI = 0 | FMA = 1 | NEON = 0
| ARM_FMA = 0 | F16C = 1 | FP16_VA = 0 | WASM_SIMD = 0 | BLAS = 1 | SSE3 = 1 | SSSE3 = 0
| VSX = 0 |
```

Certainly! Based on the given descriptions, I can provide an absolute merged description of the image:

In the image, we see a red crab and a red and pink flower in the foreground, with a white and blue sky in the background. The caption suggests that the image is related to a painting. Our attention focuses on the face of the crab, which appears to be looking directly at us. The image features several recognizable elements, including the red and pink flower, the white and blue sky, and the crab's face. The overall style or characteristics of the image can be described as a painting with a realistic depiction of a crab and a flower in a natural setting.

CPU times: total: 1min 35s

Wall time: 2min 12s

Score evaluation

In [7]:

```
from rouge_score import rouge_scorer
from nltk.translate.bleu_score import sentence_bleu
from nltk.translate.meteor_score import meteor_score
from nltk.tokenize import word_tokenize
import pandas as pd

def calculate_scores(reference_text, generated_text):
    # Tokenize texts
    reference_tokens = word_tokenize(reference_text)
    generated_tokens = word_tokenize(generated_text)

    # ROUGE Score
    scorer = rouge_scorer.RougeScorer(['rouge1', 'rougeL'], use_stemmer=True)
    rouge_score = scorer.score(reference_text, generated_text)

    # BLEU Score
    bleu_score = sentence_bleu([reference_tokens], generated_tokens)

    # METEOR Score
    meteor_score_value = meteor_score([reference_tokens], generated_tokens)

    return rouge_score, bleu_score, meteor_score_value

def evaluate_with_multiple_references(generated_text, reference_texts):
    results = []

    for ref in reference_texts:
        rouge_score, bleu_score, meteor_score_value = calculate_scores(ref, generated_text)
```

```

        results.append({
            'Reference Text': ref,
            'ROUGE-1 Score': rouge_score['rouge1'].fmeasure,
            'ROUGE-L Score': rouge_score['rougeL'].fmeasure,
            'BLEU Score': bleu_score,
            'METEOR Score': meteor_score_value
        })

    return pd.DataFrame(results)

# Example usage
generated_text = result
reference_texts = [
    resultdict["descriptions_blip"],
    resultdict["descriptions_ved"],
    resultdict["descriptions_noamrot"],
    resultdict["descriptions_blip"] + resultdict["descriptions_ved"] + resultdict["d
escriptions_noamrot"]
]

df = evaluate_with_multiple_references(generated_text, reference_texts)

df.head()

```

Out[7]:

| | Reference Text | ROUGE-1 Score | ROUGE-L Score | BLEU Score | METEOR Score |
|---|---|---------------|---------------|------------|--------------|
| 0 | a painting of a flower on a white background | 0.122449 | 0.122449 | 0.048628 | 0.248490 |
| 1 | a painting of an animal with a flower in it | 0.135135 | 0.135135 | 0.043408 | 0.304918 |
| 2 | the image features a red crab and a red and pi... | 0.404624 | 0.369942 | 0.138603 | 0.488662 |
| 3 | a painting of a flower on a white backgrounda ... | 0.515789 | 0.452632 | 0.207926 | 0.557566 |

In [8]:

```

def replace_reference_text(df):
    # Define the new reference texts
    new_references = [
        "Salesforce/blip-image-captioning-base",
        "jaimin/image_caption",
        "noamrot/FuseCap_Image_Captioning",
        "Combined Descriptions"
    ]

    # Replace the first 3 rows of the reference text column
    df.loc[:3, 'Reference Text'] = new_references
    df.rename(columns={'Reference Text': 'Models'}, inplace=True)
    return df

df = replace_reference_text(df)

df.head()

```

Out[8]:

| | Models | ROUGE-1 Score | ROUGE-L Score | BLEU Score | METEOR Score |
|---|---------------------------------------|---------------|---------------|------------|--------------|
| 0 | Salesforce/blip-image-captioning-base | 0.122449 | 0.122449 | 0.048628 | 0.248490 |
| 1 | jaimin/image_caption | 0.135135 | 0.135135 | 0.043408 | 0.304918 |
| 2 | noamrot/FuseCap_Image_Captioning | 0.404624 | 0.369942 | 0.138603 | 0.488662 |
| 3 | Combined Descriptions | 0.515789 | 0.452632 | 0.207926 | 0.557566 |

In [9]:

```
df.to_excel('model_scores_df.xlsx', index=False)
```

Salesforce/blip-image-captioning-base: ROUGE-1 and ROUGE-L Scores: 0.0 BLEU Score: 0.0 METEOR Score: 0.0

Analysis: These scores indicate no overlap between the generated text and the reference texts. This could mean that the model's outputs are entirely different from the references, potentially indicating poor performance or a significant difference in the type of content being generated compared to the f e2 .nce. jaimin/image_caption: ROUGE-1 Score: 0.136986 ROUGE-L Score: 0.109589 BLEU Score: 0.046209 METEOR Score: 0.301215 Analysis: These scores suggest a low to moderate level of similarity with the reference texts. The model shows some ability to capture relevant content (as indicated by the METEOR score), but the overall alignment with the ree e text s 3 . is modest. noamrot/FuseCap_Image_Captioning: ROUGE-1 Score: 0.349398 ROUGE-L Score: 0.265060 BLEU Score: 0.092045 METEOR Score: 0.539405 Analysis: These are the highest scores among the three models, indicating a relatively better performance in terms of matching with the reference texts. The scores suggest a good balance of content accuracy and fluency.The scores you've provided for the "Combined Descriptions" model in the context of ROUGE, BLEU, and METEOR metrics suggest a moderate level of performance. To understand these scores better, let's analyze them individually: ROUGE-1 Score (0.427586): This score measures the overlap of unigrams (single words) between the generated text and the reference texts. A score of approximately 0.43 suggests a moderate level of word overlap, indicating that about 43% of the words in the generated text are also found in the reference text. ROUGE-L Score (0.262069): The ROUGE-L score focuses on the longest common subsequence and is generally considered a measure of the fluency and structure of the text. A score of around 0.26 is on the lower side, suggesting that the sequence of words in the generated text moderately aligns with that in the reference text. BLEU Score (0.1107189): The BLEU score measures the precision of n-grams in the generated text against the reference texts. A score over 0.11 in BLEU is generally considered low, particularly in contexts like machine translation. However, it's important to note that BLEU scores can be less informative for tasks outside of translation or when the reference and generated texts are highly creative or varied. METEOR Score (0.434109): The METEOR score is an improvement over BLEU, as it also considers synonyms and paraphrasing. A score of approximately 0.43 is moderate, suggesting some level of semantic and syntactic alignment with the reference texts. Overall Assessment: These scores indicate a moderate level of performance, with some room for improvement, especially in terms of fluency and structural alignment (as indicated by the ROUGE-L score). The context in which these scores are being evaluated is crucial. For instance, if this is a creative writing task or a task where exact word overlap with reference texts is not critical, these scores might be quite acceptable. It's also important to compare these scores against a baseline or control model to better understand their significance. For example, if these scores are significantly higher than those of previous models or iterations, they represent an improvement. Lastly, these scores are quantitative measures and should ideally be supplemented with qualitative assessments to get a fuller picture of the model's performance. In summary, while these scores are not exceptionally high, they do indicate a reasonable level of performance, particularly in terms of word overlap and semantic alignment. Depending on the specific requirements and context of your task, they could be seen as a solid foundation for further refinement.

In [10]:

```
print(resultdict)

{'descriptions_blip': 'a painting of a flower on a white background', 'descriptions_ved': 'a painting of an animal with a flower in it', 'descriptions_noamrot': 'the image feature s a red crab and a red and pink flower in the foreground, with a white and blue sky in th e background the caption suggests that the image is related to a painting', 'What do you see in the image?': 'banana', 'Where in the image does your attention focus the most?': 'face', 'What features or elements in the image influenced your perception?': 'face', 'Are there any common or recognizable elements in the image?': 'yes', 'How would you describe the overall style or characteristics of the image?': 'both'}
```

Qualitative Analysis of the image description

In [11]:

```
import pandas as pd
from llama2backend import generatetext

def create_gpt_based_assessment(generated_text, aspect):

    funcprompt = f"Please provide an assessment of the following image description text i n terms of its {aspect}:\n\n'{generated_text}'"

    response = generatetext(funcprompt)

    return response

def create_qualitative_assessment_df(generated_text, reference_texts):
```

```
aspects = ['Coherence', 'Relevance', 'Creativity', 'Factual Accuracy',
           'Grammatical Correctness', 'Style', 'Engagement']

qualitative_assessment_df = pd.DataFrame(aspects, columns=['Qualitative Aspect'])
qualitative_assessment_df['Generated Text'] = generated_text

qualitative_assessment_df['Score/Comments'] = qualitative_assessment_df['Qualitative
Aspect'].apply(
    lambda aspect: create_gpt_based_assessment(generated_text, aspect))

return qualitative_assessment_df

# Example usage
generated_text = result
reference_texts = [
    resultdict['descriptions_blip'],
    resultdict['descriptions_ved'],
    resultdict['descriptions_noamrot']
]

df2 = create_qualitative_assessment_df(generated_text, reference_texts)
df2
```

Llama.generate: prefix-match hit
Llama.generate: prefix-match hit
Llama.generate: prefix-match hit
Llama.generate: prefix-match hit
Llama.generate: prefix-match hit
Llama.generate: prefix-match hit
Llama.generate: prefix-match hit

Out[11]:

| | Qualitative Aspect | Generated Text | Score/Comments |
|---|-------------------------|---|---|
| 0 | Coherence | Absolutely! Here is a merged description of ... | Based on the provided image description text... |
| 1 | Relevance | Absolutely! Here is a merged description of ... | Thank you for providing the image descriptio... |
| 2 | Creativity | Absolutely! Here is a merged description of ... | Thank you for providing the image descriptio... |
| 3 | Factual Accuracy | Absolutely! Here is a merged description of ... | I'm just an AI, I don't have personal opinio... |
| 4 | Grammatical Correctness | Absolutely! Here is a merged description of ... | Based on the provided image description text... |
| 5 | Style | Absolutely! Here is a merged description of ... | Thank you for providing the image descriptio... |
| 6 | Engagement | Absolutely! Here is a merged description of ... | Based on the provided image description text... |

In [12]:

```
df2.to_excel('qualitative_assessment_df.xlsx', index=False)
```

prompt for project report

In [2]:

```
generated_text = """ Absolutely! Here is a merged description of the image based on the g
iven instructions:
In the image, we see a painting of a flower on a white background, accompanied by an anim
al with a flower in it. The red crab and red and pink flower are positioned in the foregr
ound, while a white and blue sky can be seen in the background. The caption suggests that
the image is related to a painting. Our attention is drawn to the face of the crab, which
appears to be the most prominent feature in the image. The image features several recogni
zable elements, including the flower, the crab, and the sky. The overall style or charact
eristics of the image can be described as a mix of realistic and abstract, with a focus o
n the use of bold colors and simple shapes."""
```

In [3]:

```
from llama2backend import generatetext
```

```
import pandas as pd

scores_df = pd.read_excel('model_scores_df.xlsx')
qualitative_assessment_df = pd.read_excel('qualitative_assessment_df.xlsx')

prompt = f""" Create me a 1000 word Proposed model evaluation report if this is the {generated_text} by the model.
                This is the evaluation Score evaluation {str(scores_df.iloc[len(scores_df)-1])}.
                This is the qualitative assessment {qualitative_assessment_df["Score/Comments"]}
            """

# Example usage
result = generatetext(prompt)
print(result)
```

Proposed Model Evaluation Report:

Introduction:

The following report presents an evaluation of a proposed model based on the given image description. The model is designed to generate a description of the image, and the evaluation is conducted using various automated metrics and qualitative assessment.

Automated Metrics:

ROUGE-1 Score: 0.515789
 ROUGE-L Score: 0.452632
 BLEU Score: 0.207926
 METEOR Score: 0.557566

Qualitative Assessment:

Based on the provided image description text, the model has generated a clear and concise description of the image. The description highlights the prominent features of the image, including the red crab and red and pink flower in the foreground, while the white and blue sky is visible in the background. The model's use of bold colors and simple shapes creates a visually appealing description that effectively conveys the overall style or characteristics of the image.

Strengths:

- * The model has accurately described the main elements of the image, including the crab and flower.
- * The use of bold colors and simple shapes creates a visually appealing description.

Weaknesses:

- * The model could have provided more detail on the background sky.
- * The description could be more specific in terms of the colors used in the image.

Recommendations for Future Improvement:

- * Incorporate more detailed information about the background sky to provide a more comprehensive description of the image.
- * Use more specific and vivid language to describe the colors used in the image, such as "bright blue" or "pastel pink."

Conclusion:

Based on the evaluation results, the proposed model has demonstrated adequate performance in generating a description of the given image. However, there is room for improvement, particularly in terms of providing more detail about the background sky and using more specific language to describe the colors used in the image. With these recommendations in mind, the model can be further refined to produce even more accurate and informative descriptions in the future.

In [5]:

```
print(prompt)
```

Create me a 1000 word Proposed model evaluation report if this is the Absolutely! Here is a merged description of the image based on the given instructions:

In the image, we see a painting of a flower on a white background, accompanied by an animal with a flower in it. The red crab and red and pink flower are positioned in the foreground, while a white and blue sky can be seen in the background. The caption suggests that the image is related to a painting. Our attention is drawn to the face of the crab, which appears to be the most prominent feature in the image. The image features several recognizable elements, including the flower, the crab, and the sky. The overall style or characteristics of the image can be described as a mix of realistic and abstract, with a focus on the use of bold colors and simple shapes. by the model.

This is the evaluation Score evaluation Models Combined Descriptions

```
ROUGE-1 Score      0.515789
ROUGE-L Score      0.452632
BLEU Score         0.207926
METEOR Score       0.557566
```

```
Name: 3, dtype: object.
```

```
          This is the qualitative assessment 0      Based on the provided image descr
ption text...
```

```
1      Thank you for providing the image descriptio...
2      Thank you for providing the image descriptio...
3      I'm just an AI, I don't have personal opinio...
4      Based on the provided image description text...
5      Thank you for providing the image descriptio...
6      Based on the provided image description text...
```

```
Name: Score/Comments, dtype: object
```

```
In [ ]:
```