AIM :To write simple code for Image captioning

Algorithm for Basic Image Captioning Display

1. **Define a Function to Load and Preprocess Images:**
   - Use OpenCV (`cv2`) to load images from file paths.
   - Convert the color space from BGR to RGB.
   - Resize images to a consistent size suitable for display and processing (e.g., 224x224 pixels).
   - Normalize pixel values to a range of [0, 1].
2. **Prepare Example Images and Captions:**
   - Define a list of image paths (`image_paths`) pointing to the images you want to display.
   - Define a list of captions (`captions`) corresponding to each image.
3. **Load and Preprocess Images:**
   - Iterate through `image_paths`, applying the `load_image` function to load and preprocess each image.
   - Store the preprocessed images in a list (`images`).
4. **Display Images with Captions Using Matplotlib:**
   - Create a figure (`fig`) and axes (`axes`) using `plt.subplots`.
   - Iterate through the preprocessed images and captions, using `enumerate` to track the index (`i`) and corresponding image/caption pair.
   - For each image/caption pair, use `axes` to display the image with `imshow` and set the caption as the title using `set_title`.
   - Adjust axis visibility using `axis('off')` to remove axis ticks and labels.
5. **Adjust Layout and Show the Plot:**
   - Use `plt.tight_layout()` to adjust the spacing between subplots to prevent overlapping captions.
   - Finally, use `plt.show()` to display the plot with all images and captions.

CODE

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Function to load and preprocess an image
def load_image(image_path):
    img = cv2.imread(image_path)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  # Convert BGR to RGB
    img = cv2.resize(img, (224, 224))  # Resize to 224x224 (input size for ResNet50)
    img = img.astype(np.float32) / 255.0  # Normalize pixel values
    return img

# Example images and paths (replace with your own)
image_paths = [
    'image1.jpg',
    'image2.jpg',
    'image3.jpg',

]

# Load and preprocess images
images = [load_image(image_path) for image_path in image_paths]

# Example captions (replace with your own)
captions = [
    'horse and a girl are in camp fire',
    'children playing in swimming pool',
    'man is watching the sunset'
]

# Ensure the number of captions matches the number of images
assert len(captions) == len(images)

# Display images with captions
fig, axes = plt.subplots(2, 5, figsize=(15, 7))  # Adjust figsize as needed
```

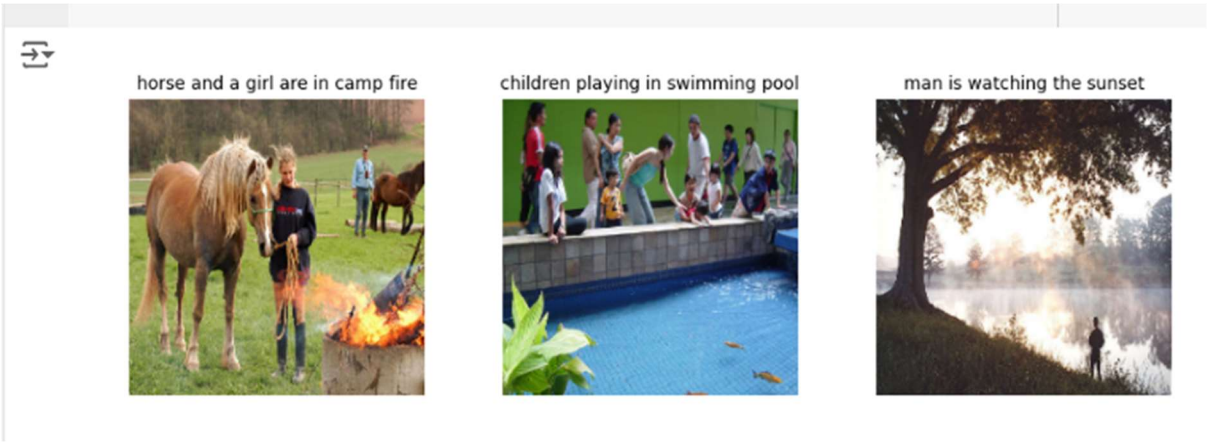
```
# Ensure the number of captions matches the number of images
assert len(captions) == len(images)

# Display images with captions
fig, axes = plt.subplots(2, 5, figsize=(15, 7))  # Adjust figsize as needed

for i, (image, caption) in enumerate(zip(images, captions), 1):
    ax = axes[(i - 1) // 5, (i - 1) % 5]
    ax.imshow(image)
    ax.set_title(caption, fontsize=10)  # Adjust fontsize as needed
    ax.axis('off')

plt.tight_layout()  # Adjust layout to prevent overlapping
plt.show()
```

OUTPUT


horse and a girl are in camp fire
children playing in swimming pool
man is watching the sunset