

DS Lab

Exp - 3

Name: Vaibhav Boudh

Division: D20B

Roll no: 60

Aim: To build a Cognitive image based application to understand context for a Customer service application/ Insurance/ Healthcare Application/ Smarter Cities/ Government etc.

Theory:

Cognitive image-based context recognition is an emerging field within computer vision and artificial intelligence that focuses on interpreting images in a manner akin to human cognitive processes. This discipline integrates various techniques to extract meaningful information from visual data, using that information to understand context or make informed decisions.

1. **Cognitive Image Processing:** This aspect of the field involves analyzing and interpreting images by simulating human cognitive abilities. Unlike basic image recognition, cognitive image processing aims to deduce context, emotions, and intentions from visual inputs. It draws from cognitive science, psychology, and neuroscience to model human perception and understanding of images.
2. **Context Recognition:** Context recognition refers to the capacity to identify the setting or situation depicted in an image. This includes:
 - **Scene Understanding:** Recognizing the general environment, such as a park, office, or street.
 - **Object Recognition:** Identifying and classifying objects within the scene, like cars, people, or buildings.
 - **Activity Recognition:** Determining what actions or interactions are taking place, such as someone running or two people conversing.
3. **Feature Extraction:** Feature extraction plays a vital role in cognitive image-based context recognition. Features are distinct patterns or characteristics within an image that help in understanding its content. These features can be:
 - **Low-Level Features:** Basic elements like color, texture, and edges.
 - **High-Level Features:** More complex attributes like object shapes, spatial relationships, and semantic meaning.
4. **Applications:** This technology has a wide range of applications, including:
 - **Autonomous Vehicles:** Understanding road environments, detecting pedestrians, and assessing traffic situations.
 - **Surveillance:** Analyzing and interpreting activities in security footage.
 - **Augmented Reality (AR):** Adding contextual information to real-world environments.

- **Healthcare:** Assisting in diagnosis and treatment planning by analyzing medical images.
5. **Challenges:** The field faces several challenges, such as:
- **Image Variability:** Variations in lighting, angle, and occlusion can impact recognition accuracy.
 - **Complex Scenes:** Analyzing scenes with multiple interacting objects and activities can be challenging.
 - **Contextual Ambiguity:** Images can be ambiguous, making it difficult to infer the correct context without advanced reasoning and additional data.
6. **Future Directions:** The field is continuously evolving, with ongoing research focusing on:
- **Enhancing Model Robustness:** Creating models capable of handling a wide variety of image conditions.
 - **Real-Time Processing:** Improving the speed and efficiency of context recognition for real-time applications.
 - **Human-AI Collaboration:** Developing systems that can work alongside humans, offering context-aware assistance and insights.

Code:

```
import numpy as np
from transformers import BlipProcessor, BlipForConditionalGeneration
from PIL import Image

def describe_image(image_path):
    # Load the pre-trained model and processor from Hugging Face
    processor =
BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-base")
    model =
BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning
-base")

    # Open and process the image
    image = Image.open(image_path)

    # Prepare the image for the model
    inputs = processor(images=image, return_tensors="pt")

    # Generate the caption
    out = model.generate(**inputs)
    description = processor.decode(out[0], skip_special_tokens=True)

    return description

def print_binary_matrix(image_path, threshold=128):
```

```

# Load the image
image = Image.open(image_path).convert('L') # Convert to grayscale

# Convert the grayscale image to a numpy array
image_array = np.array(image)

# Apply a threshold to create a binary image
binary_array = (image_array > threshold).astype(int)

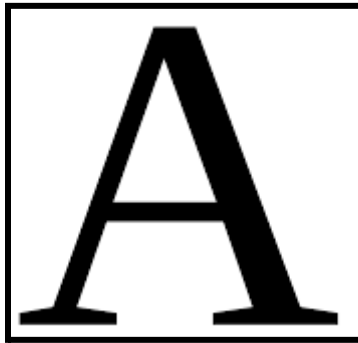
# Print the binary matrix
for row in binary_array:
    print(' '.join(str(cell) for cell in row))

# Example usage
print_binary_matrix('images.jpg')
image_description = describe_image("images.jpg")
print()
print("Image description: "+image_description)

```

Output:

Example 1 : The letter “A”



Binary matrix

Image description: the letter a is a black and white font

Output

Example 2: Car with Neon lights



Binary Matrix

Image description: a futuristic car with neon lights

Output

Conclusion: Therefore, we have studied and built a cognitive image based application to understand context for a Customer service application.