**KGISL INSTITUTE OF TECHNOLOGY**

(AFFILIATAED TO ANNA UNIVERSITY)

Saravanampatti, Coimbatore – 641035



**Cloud computing**

*Submitted by*

*Abarna M* (711721106001)

Eswari k (711721106034)

Maghaa Miruthaa B (711721106060)

Hariharan R (711721106036)

Jaya Ganesh R (711721106048)

1**. Set up IBM Cloud Visual Recognition:**

- Sign up for an IBM Cloud account if you haven't already.

- Create a Visual Recognition service instance from the IBM Cloud catalog.

- Obtain the API keys and credentials for your Visual Recognition service.

**2.Collect and Preprocess Images:**

- Gather a dataset of images you want to classify and describe.

- Preprocess the images, ensuring they are in a suitable format for the Visual Recognition API.

**3. Integrate IBM Cloud Visual Recognition:**

- You can use various programming languages (e.g., Python) to interact with the Visual Recognition API. Here's a Python example using the `ibm-watson` SDK:

```python

from ibm\_watson import VisualRecognitionV3

from ibm\_watson.visual\_recognition\_v3 import FileWithMetadata

api\_key = 'YOUR\_API\_KEY'

service\_url = 'YOUR\_SERVICE\_URL'

version = '2018-03-19'

visual\_recognition = VisualRecognitionV3(

version=version,

iam\_apikey=api\_key,

url=service\_url

)

with open('image.jpg', 'rb') as image\_file:

classes = visual\_recognition.classify(

image\_file,

threshold='0.6',

owners=["me"]).get\_result()

```

- Adjust the parameters and threshold based on your specific use case.

**4. Generate Image Captions:**

- After classifying the image using Visual Recognition, you can use a Natural Language Processing (NLP) library or model to generate captions for the recognized objects. One popular choice is using the OpenAI GPT-3 model.

- You will need to use the GPT-3 API to generate captions. You can do this in Python using the `openai` library:

```python

import openai

openai.api\_key = 'YOUR\_OPENAI\_API\_KEY'

response = openai.Completion.create(

engine="davinci",

prompt="Generate a caption for this image: 'a cat sitting on a table'",

max\_tokens=30

)

caption = response.choices[0].text

```

- Replace 'a cat sitting on a table' with the result from the Visual Recognition step.

**5. Display the Captions:**

- Once you have the generated captions, you can display them alongside the recognized images, for example, on a website or in a mobile app.

**6. Repeat for Multiple Images:**

- Loop through your dataset, applying the Visual Recognition and caption generation process to each image.

Remember to handle errors, rate limits, and any additional processing or filtering of captions as per your project's requirements. Additionally, you can fine-tune the GPT-3 model to better suit your image captioning needs. This is a simplified example, and the actual implementation can be more complex, depending on your specific use case and the technologies you are using.

Certainly! Let's dive deeper into building the image recognition system by integrating IBM Cloud Visual Recognition and using AI-generated captions. Here's a more detailed step-by-step guide:

**7. Set Up IBM Cloud Visual Recognition**

- Sign in to your IBM Cloud account or create one if you don't have an account.

- Create a Visual Recognition service instance from the IBM Cloud catalog.

- Obtain the API key and URL for your Visual Recognition service.

**8. Collect and Prepare Your Image Data:**

- Assemble a dataset of images that you want to classify and describe. Ensure your dataset is diverse and represents the types of images your system will encounter.

- Preprocess your images, which may include resizing, cropping, and converting to a format supported by the Visual Recognition API (commonly JPEG or PNG).

**9. Integrate IBM Cloud Visual Recognition:**

- You can use the IBM Watson Python SDK (ibm-watson) to interact with the Visual Recognition API. Install the SDK using pip:

```bash

pip install ibm-watson

```

- Sample Python code for image classification using Visual Recognition:

```python

from ibm\_watson import VisualRecognitionV3

from ibm\_watson.visual\_recognition\_v3 import FileWithMetadata

api\_key = 'YOUR\_API\_KEY'

service\_url = 'YOUR\_SERVICE\_URL'

version = '2018-03-19'

visual\_recognition = VisualRecognitionV3(

version=version,

iam\_apikey=api\_key,

url=service\_url

)

with open('image.jpg', 'rb') as image\_file:

classes = visual\_recognition.classify(

image\_file,

threshold='0.6',

owners=["me"]).get\_result()

```

**10. Generate Image Captions Using AI:**

- After classifying an image using Visual Recognition, you can use an AI model like GPT-3 for generating captions. This step may involve using the OpenAI GPT-3 API, as shown in the previous response.

- Provide the recognized objects and their probabilities from the Visual Recognition results as input to GPT-3 to generate descriptive captions.

**11. Display the Captions:**

- Once you have the generated captions, you can associate them with the respective images and display them on a website, app, or in any medium relevant to your project.

**12. Scaling and Efficiency:**

- Depending on your project's scale, you might need to implement efficient processes for bulk image classification and caption generation.

- Consider using multi-threading or parallel processing to speed up the classification and captioning tasks, especially for large datasets.

**13. Error Handling and Fine-tuning:**

- Implement error handling to address any issues that may arise during the image recognition and caption generation process.

- Fine-tune the GPT-3 model for better caption quality, as you can provide feedback to the model to generate more accurate and contextually relevant captions.

**14. User Interaction:**

- If your image recognition system is user-facing, consider building a user interface for users to interact with your system, upload images, and view the generated captions.

**15. Testing and Evaluation:**

- Test your system with a variety of images to ensure that it accurately recognizes objects and generates meaningful captions.

**16. Deployment:**

- Deploy your image recognition system on your preferred platform, whether it's a web server, cloud service, or a mobile application.

Please note that building an image recognition system and implementing AI-generated captions is a complex and resource-intensive task. Consider the needs and constraints of your specific project, and allocate resources accordingly. Regularly update and fine-tune your system to improve its performance and accuracy over time.