new.cpp

```cpp
#include <iostream>
#include <curl/curl.h>

// Replace these values with your IBM Cloud Visual Recognition credentials
const std::string apiKey = "YOUR_API_KEY";
const std::string serviceUrl = "YOUR_SERVICE_URL";

// Function to perform the HTTP POST request
size_t WriteCallback(void* contents, size_t size, size_t nmemb, std::string* output) {
    size_t totalSize = size * nmemb;
    output->append((char*)contents, totalSize);
    return totalSize;
}

// Function to classify an image using IBM Visual Recognition
std::string classifyImage(const std::string& imagePath) {
    CURL* curl;
    CURLcode res;

    curl_global_init(CURL_GLOBAL_DEFAULT);
    curl = curl_easy_init();

    if (curl) {
        // Set the API endpoint
        std::string url = serviceUrl + "/v3/classify?api_key=" + apiKey;

        // Set the image file for classification
        struct curl_httppost* post = NULL;
        struct curl_httppost* last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "images_file", CURLFORM_FILE, imagePath.c_str(), CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_URL, url.c_str());
        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, WriteCallback);

        std::string response;
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);

        // Perform the request
        res = curl_easy_perform(curl);

        // Clean up
        curl_easy_cleanup(curl);
        curl_formfree(post);

        if (res != CURLE_OK) {
            std::cerr << "curl_easy_perform() failed: " << curl_easy_strerror(res) << std::endl;
        } else {
            return response;
        }
    }

    return "";
}

```

```cpp
int main() {
    // Replace with the path to the image you want to classify
    std::string imagePath = "path/to/your/image.jpg";

    std::string response = classifyImage(imagePath);

    // Handle the response as needed (e.g., parse JSON)
    std::cout << "Response: " << response << std::endl;

    return 0;
}

```