

Image Recognition with IBM Cloud Visual Recognition

Phase-4

INTRODUCTION:

Continuing the journey to enhance the image recognition system, explore how to integrate IBM Cloud Visual Recognition with AI-generated captions. This phase will delve into the implementation of the image classification process through the powerful IBM Cloud Visual Recognition API. Furthermore, it will introduce the concept of natural language generation to automatically create descriptive captions for the images that the system recognizes. This integration of advanced technologies promises to take the image recognition system to a new level, enabling it to not only identify images but also communicate their content in human-readable language. Let's dive into the details of this exciting development!

PROCEDURE:

Login to the IBM account with IBMid. Then enter your password.

The home page of the IBM account will be:

The screenshot shows the IBM Cloud dashboard. At the top, there's a navigation bar with tabs for Catalog, Manage, and a user profile. Below the navigation is a search bar and a 'Create resource' button. The main area is titled 'Dashboard'. It features a 'For you' section with several cards:

- Build**: Explore IBM Cloud with this selection of easy starter tutorials and services.
- Get Started with Watson Studio**: Get started with using AI and Cloud Object Storage in 15 minutes.
- Use Watson Studio**: Watson Studio provides a suite of tools and a collaborative environment for data scientists, developers and domain experts.
- Build with Watson**: Chatbots, Insights, recognizers, and more. Explore the AI platform for business.
- Use Watson Assistant**: Watson Assistant lets you build conversational interfaces into any application, device, or channel.
- Get started with machine learning + Watson Studio**: Build, run and manage AI models. Prepare data and build models anywhere using open source code or visual modeling. Predict and optimize your outcomes.

Below the 'For you' section are four cards: News, Recent support cases, Planned maintenance, and IBM Cloud status. A sidebar on the left contains icons for various services like VMs, Databases, and Container registries. A message at the bottom says 'Waiting for cloud.ibm.com...'.

Now, Click on the help icon and then click on Docs in the drop-down menu

This screenshot is similar to the one above, showing the IBM Cloud dashboard. However, the help icon in the top right corner has been clicked, opening a dropdown menu. The 'Docs' option is highlighted in blue, indicating it is selected. The rest of the menu includes Support center, Guided tours, and Send feedback.

Then click the search icon. Search for Visual Recognition in the search bar.

The screenshot shows a web browser window with three tabs: 'ibm cloud account - Yahoo Search', 'IBM Cloud', and 'IBM Cloud Docs'. The 'IBM Cloud Docs' tab is active, showing a search results page for 'visual recognition'. The search bar at the top contains the query 'visual recognition'. Below the search bar, the page title is 'Find answers Cloud docs'. A search input field says 'Search IBM Cloud docs' and a link 'Define with search operators' is available. The search results list includes: 'visual recognition', 'visual recognition demo', 'visual recognition project', 'visual recognition api', 'visual recognition case study', 'visual recognition model', and 'visual recognition software'. To the right of the search results, there is a sidebar with sections for 'Solutions', 'Product guides', 'Tutorials', 'API & SDK reference', and 'FAQs'. Each section has a brief description and a corresponding icon. A 'Feedback' button is located in the bottom right corner of the sidebar.

Then open the DocumentAnalytics.VisualRecognitionClassifier actions.

The screenshot shows a web browser window with three tabs: 'ibm cloud account - Yahoo Search', 'IBM Cloud', and 'IBM Search'. The 'IBM Search' tab is active, showing a search results page for 'visual recognition'. The search bar at the top contains the query 'visual recognition'. Below the search bar, the URL 'ibm.com/search?lang=en&cc=in&q=visual%20recognition' is displayed. The search results list includes: 'Visual Recognition with IBM Watson' (with a link to 'community.ibm.com/community/user/watsonstudio/blogs/lommy-gerald1/2021/04/01/visual-recognition-with-ibm-watson'), 'DocumentAnalytics.VisualRecognitionClassifier actions' (with a link to 'ibm.com/docs/en/datacap/9.1.5/documentanalytics/visualrecognitionclassifieractions'), and another 'DocumentAnalytics.VisualRecognitionClassifier actions' entry (with a link to 'ibm.com/docs/en/datacap/9.1.4/documentanalytics/visualrecognitionclassifieractions'). A 'Site feedback' button is located in the bottom right corner of the sidebar.

DocumentAnalytics.VisualRecognitionClassifier actions page will be like this.

This screenshot shows the 'DocumentAnalytics.VisualRecognitionClassifier actions' page in the IBM Datacap 9.1.5 documentation. The left sidebar lists various actions under 'DocumentAnalytics.VisualRecognitionClassifier actions', including VisualRecogClassify, VisualRecogSetCredentials, VisualRecogSetMinConfidence, VisualRecogSetURL, VisualRecogTrain, DocumentAnalytics.NaturalLanguageClassifier actions, Email actions, Equalize actions, and Ewsmail actions. The main content area describes the Visual Recognition Classifier as an IBM Watson service for classifying image-based documents. It details how to use the default classifier or create a custom one by providing training data (two compressed .zip files). Once trained, it can classify other documents with confidence scores. Below this, a section titled 'Properties and methods:' lists the four actions mentioned in the sidebar. The URL in the browser bar is https://ibm.com/docs/en/datacap/9.1.5?topic=actions-documentanalyticsvisualrecognitionclassifier&mhsrc=ibmsearch_a&mhq=visual%20recognition.

This screenshot shows the 'DocumentAnalytics.NaturalLanguageClassifier actions' page in the IBM Datacap 9.1.5 documentation. The left sidebar lists actions under 'DocumentAnalytics.NaturalLanguageClassifier actions', including visualrecogSetCredentials, VisualRecogSetMinConfidence, VisualRecogSetURL, VisualRecogTrain, and DocumentAnalytics.NaturalLanguageClassifier actions. The main content area starts with a 'Properties and methods:' section listing the four actions from the sidebar. Below this is a 'Parent topic:' section pointing to 'Global actions'. At the bottom of the page, there is a note about inclusive language and a copyright notice. The URL in the browser bar is <https://www.ibm.com/docs/en/datacap/9.1.5?topic=actions-documentanalyticsnaturallanguageclassifier>.

Then open DocumentAnalytics.NaturalLanguageClassifier actions,

the page will be

This screenshot shows a web browser window displaying the IBM Datacap 9.1.5 Documentation. The URL is ibm.com/docs/en/datacap/9.1.5?topic=actions-documentanalyticsnaturallanguageclassifier. The page title is "DocumentAnalytics.NaturalLanguageClassifier actions". The content area includes a brief description of the Natural Language Classifier service, instructions for creating and training a classifier, and information about performing OCR before classification. A sidebar on the left lists various actions, with "DocumentAnalytics.NaturalLanguageClassifier actions" currently selected. Under this section, the "NLClassify" action is expanded, showing its description and properties.

Natural Language Classifier is an IBM Watson service, which can be trained to classify documents or sections of documents. That based on the text that is contained in the document or the section of a document. The default instance name: DocumentAnalytics.NaturalLanguageClassifierActions.

To do this, you must first create and train a classifier using some representative set of training data containing text from a set of sample documents. Once the classifier has been trained, you can give it other similar documents and classifier attempts to classify them according to its training. The Classifier returns a confidence score that is associated with the classification.

Because Natural Language Classifier works on the text of the document, you must first perform OCR on the document before calling the Classify action to do the classification.

Properties and methods:

- [NLClassify](#)

This screenshot shows the same documentation page as the previous one, but the "NLClassify" action is now fully expanded. It provides a detailed description of what the action does, its properties, and its methods. The expanded content includes:

- **NLClassify**
Identifies a page by using the IBM Natural Language technology.
- **NLClassifyText**
Classifies the specified text by using the IBM Natural Language technology.
- **NLCSetCredentials**
Sets the credentials to be used to do the classification.
- **NLCSetLanguage**
Sets the language of the page to be classified.
- **NLCSetMinConfidence**
Sets the minimum confidence score for classification matching.
- **NLCTrain**
Creates or replaces an NLC Classifier.

NLClassify

It Identifies a page by using the IBM Natural Language technology.

The screenshot shows the IBM Datacap 9.1.5 Documentation page for the NLClassify action. The left sidebar shows the navigation menu with 'NLClassify' selected. The main content area displays the action's purpose, syntax, parameters, and returns. The syntax section shows the code: `bool NLClassify (string ClassifierName)`. The parameters section specifies that `ClassifierName` is the name of the classifier. The returns section states that the action returns `True` if successful and `False` otherwise.

NLClassify

Last Updated: 2021-03-04

Identifies a page by using the IBM Natural Language technology.

Syntax

```
bool NLClassify (string ClassifierName)
```

Parameters

string **ClassifierName** - Name of the Classifier to be used.

Returns

True, action is successful. Otherwise, **False**.

The screenshot shows the IBM Datacap 9.1.5 Documentation page for the NLClassify action, focusing on the 'Level' and 'Details' sections. The 'Level' section indicates that the action is at the 'Page level'. The 'Details' section provides a detailed description of the action's functionality, mentioning that it identifies a page using IBM Natural Language technology, analyzes full text, and attempts to find matches within defined classes. It also notes that if a match is found, the 'Page type' is populated with the ID of the category matched, and if no match is found, the page type is set to 'Other'. The section also describes how classification results are stored in variables like 'MatchingCategoryX' and 'MatchingCategoryConfX', and how classification can be run without updating the page type. It further explains that because matching relies on full page text, a full page recognition action must be called before using the Classify action. The final part of the details section lists the order of text retrieval from layout.xml, .txt file, and .cco file.

True, action is successful. Otherwise, **False**.

Level

Page level.

Details

This action identifies a page by using the IBM Natural Language technology. This technology analyzes the full text of pages and attempts to find match within the classes that have been defined for the selected classifier. If a match is found, the Page type is populated with the ID of the category that was matched.

If a match is not found, the page type is set to "Other".

When classification is complete, a list of matches and their confidence values are stored in the "MatchingCategoryX" and "MatchingCategoryConfX" variables. The number of matches is stored in the variable "MatchingCategoriesCount".

To run classification without updating the page type, set the variable "UpdateDCOType" to "0" before calling this action. In this case classification will not update the page type, but the variables that are mentioned above will still be populated.

Because the matching relies on a page's full text, a full page recognition action must be called before using the Classify action.

This action gets the text for the page in the following order:

- from the layout.xml that is file generated by the Recognize action
- from the .txt file that is generated by the RecognizeToFile action
- from the .cco file that is generated by the RecognizePage action

The screenshot shows a web browser window with the URL ibm.com/docs/en/datacap/9.1.5?topic=actions-nlclassify. The page is titled "NLClassify - IBM Documentation". The left sidebar shows navigation categories: "IBM Datacap" (selected), "Change version" (set to 9.1.5), "Show full table of contents", "Filter on titles", and a list of actions: "NLClassifyText", "NLSetCredentials", "NLSetLanguage", "NLSetMinConfidence", "NLTrain", "Email actions", "Equalize actions", and "Ewsmail actions". The main content area describes the "NLClassifyText" action, stating it classifies specified text using IBM Natural Language technology. It includes notes about matching categories, examples of how to use it in layout.xml or .txt files, and a code example:

```
    NLSetLanguage("en")
    NLSetCredentials("@APPVAR(values/gen/NLCLibName)", "BAPPVAR(values/adv/NLCPassword)")
    NLSetMinConfidence(0, 9)
    Recognize()
    NLClassify("@APPVAR(values/gen/NLClassifyName)")
```

NLCClassifyText

Classifies the specified text by using the IBM Natural Language technology.

The screenshot shows a web browser window with the URL ibm.com/docs/en/datacap/9.1.5?topic=actions-nlcclasstext. The page is titled "NLCClassifyText - IBM Documentation". The left sidebar shows navigation categories: "IBM Datacap" (selected), "Change version" (set to 9.1.5), "Show full table of contents", "Filter on titles", and a list of actions: "NLClassifyText", "NLSetCredentials", "NLSetLanguage", "NLSetMinConfidence", "NLTrain", "Email actions", "Equalize actions", and "Ewsmail actions". The main content area starts with a breadcrumb trail: "All products / IBM Datacap / 9.1.5 /". It includes a "Was this topic helpful?" button. The page title is "NLCClassifyText" and was last updated on 2021-03-04. It describes the action as classifying specified text using IBM Natural Language technology. It includes sections for "Syntax", "Parameters", and "Returns". The "Syntax" section shows the C# code:

```
bool NLCClassifyText (string ClassifierName, string TextToClassify)
```

The "Parameters" section lists "ClassifierName" (Name of the Classifier to be used) and "TextToClassify" (Text to be classified by using the Natural Language Classifier. This parameter supports SmartParameters). The "Returns" section states "True if the action is successful. Otherwise, False".

The screenshot shows the IBM Datacap 9.1.5 documentation for the `NLClassifyText` action. The left sidebar lists various actions under the `NLClassify` category, with `NLClassifyText` selected. The main content area describes the `TextToClassify` parameter as text to be classified by the Natural Language Classifier, supporting SmartParameters. It also specifies the `Returns` as `True` if successful, otherwise `False`. The `Level` is described as page level. The `Details` section notes that this action classifies specified text using IBM Natural Language technology, attempting to find matches within defined classes. It also mentions that when classification is complete, it stores matches and confidence values in `MatchingCategoryX` and `MatchingCategoryConfX`, and the number of matches in `MatchingCategoriesCount`. Both `ClassifierName` and `TextToClassify` can be smart parameters. An example code snippet is provided:

```
NLCSetCredentials("@APPVAR(values/gen/NLCUserName)","@APPVAR(values/adv/nlcpassword")  
NLCSetMinConfidence(0.9)  
NLClassifyText("APPVAR(values/gen/NLCClassifierName)","Sample Text")
```

NLCSetCredentials

Sets the credentials to be used to do the classification.

The screenshot shows the IBM Datacap 9.1.5 documentation for the `NLCSetCredentials` action. The left sidebar lists various actions under the `NLClassify` category, with `NLCSetCredentials` selected. The main content area starts with a note that this action sets the credentials to be used for classification. The `Syntax` is shown as:

```
bool NLCSetCredentials (string UserName , string Password)
```

The `Parameters` are `UserName` and `Password`. The `Returns` is described as `True` if the action succeeds, otherwise `False`.

The screenshot shows a browser window with the URL ibm.com/docs/en/datascap/9.1.5?topic=actions-nlcsetcredentials. The page is titled "IBM Datacap" and "Documentation". The sidebar on the left lists various actions: NLClassify, NLClassifyText, **NLCSetCredentials**, NLCSetsLanguage, NLCSetsMinConfidence, NLCTrain, Email actions, Equalize actions, and Ewsmail actions. The main content area starts with the "Returns" section, which states "True, if the action succeeds. Otherwise, False." Below this is the "Level" section, indicating "All level." The "Details" section contains a note about setting credentials for classification, mentioning smart parameters and security recommendations. The "Example" section shows the following code snippet:

```
NLCSetsLanguage("en")
NLCSetsCredentials("@APPVAR(values/gen/NLCUserName)","@APPVAR(values/adv/NLCPassword)")
NLCSetsMinConfidence(0.9)
Recognize()
NLCClassify("@APPVAR(values/gen/NLCClassifierName)")
```

NLCSetLanguage

Sets the language of the page to be classified.

The screenshot shows a browser window with the URL ibm.com/docs/en/datascap/9.1.5?topic=actions-nlcsetlanguage. The page is titled "IBM Datacap" and "Documentation". The sidebar on the left lists: NLCClassify, NLCClassifyText, NLCSetCredentials, **NLCSetLanguage**, NLCSetsMinConfidence, NLCTrain, Email actions, Equalize actions, and Ewsmail actions. The main content area includes the "Syntax" section with the code `bool NLCSetLanguage (string LanguageCode)`, the "Parameters" section (describing `LanguageCode` as a two-letter code), and the "Returns" section (stating "True, if the action succeeds. Otherwise, False."). The top right corner has a "Was this topic helpful?" button with thumbs up and down icons.

The screenshot shows a browser window with the URL ibm.com/docs/en/datacap/9.1.5?topic=actions-nlcsetlanguage. The page is titled "IBM Datacap" and "Documentation". The search bar says "Search in IBM Datacap 9.1.5". The main content area is titled "Returns" and states "True, if the action succeeds. Otherwise, False." Below this is a section titled "Level" with the text "All level.". A section titled "Details" follows, describing the action as setting the language of the page to be classified. It lists supported Language codes: en (English), ar (Arabic), fr (French), de (German), it (Italian), ja (Japanese), ko (Korean), pt (Portuguese (Brazilian)), and es (Spanish). A note at the bottom of the "Details" section states: "This action must be called before the Classify action."

This screenshot shows the same documentation page as above, but with additional content. In the "Details" section, there is a note: "This action must be called before the Classify action." Below this is a "Example" section containing a code snippet:

```
NLCSetLanguage("en")
NLCSetCredentials("@APPVAR(values/gen/NLCUserName)", "@APPVAR(values/adv/NLCPassword)")
NLCSetMinConfidence(0.9)
Recognize()
NLCClassify("@APPVAR(values/gen/NLCClassifierName)")
```

At the bottom of the page, under "Parent topic:", there is a link to "DocumentAnalytics.NaturalLanguageClassifier actions".

NLCSetMinConfidence

Sets the minimum confidence score for classification matching.

The screenshot shows a web browser window with the URL ibm.com/docs/en/datacap/9.1.5?topic=actions-nlcsetminconfidence. The page is titled "NLCSetMinConfidence". It includes a sidebar with navigation links for various actions like NLClassify, NLClassifyText, etc., and a search bar at the top right.

Syntax:

```
bool NLCSetMinConfidence (string MinScore)
```

Parameters:

string **MinScore** - Minimum score for classification matching. Valid values are fractional values between zero and one (for example: 0.0 and 1.0)

Returns:

True, if the parameter value is between the valid range of zero to one (0.0 and 1.0) Otherwise, **False**.

The screenshot shows a web browser window with the URL ibm.com/docs/en/datacap/9.1.5?topic=actions-nlcsetminconfidence. The page is titled "NLCTrain". It includes a sidebar with navigation links for various actions like NLClassify, NLClassifyText, etc., and a search bar at the top right.

Level:

All level.

Details:

When Classify searches for a classification match, a score between zero (no match) and one (a positive match) is calculated. This action sets the minimum score that a match must be considered a match. Any matches with a score less than the value specified is rejected. With this action, you can control the tolerance for documents matching an existing example.

When setting up the parameter in your application, use the decimal character from the system locale that is defined for the application in the Taskmaster Application Manager. For example, when the decimal character is a period, use a value from 0.0 to 1.0. When the decimal character is a comma, use a value in the range 0,0 - 1,0.

The **MinScore** can be a Smart Parameter. This action must be called before the Classify action.

Example:

```
NLCTrain()  
NLCCredentials("@APPVAR(values/gen/NLCUserName)", "@APPVAR(values/adv/NLCPassword)")  
NLCCsetMinConfidence(0.9)  
Recognize()  
NLCClassify("@APPVAR(values/gen/NLCClassifierName)")
```

NLCTrain

Creates or replaces an NLC Classifier.

The screenshot shows the IBM Documentation interface for IBM Datacap 9.1.5. The left sidebar lists various actions: NLCClassify, NLCClassifyText, NLCSetCredentials, NLCSetLanguage, NLCSetMinConfidence, and NLCTrain. NLCTrain is currently selected. The main content area is titled "NLCTrain" and describes it as "Creates or replaces an NLC Classifier." It includes a "Syntax" section with the code `bool NLCTrain (string CSVDirectory, string Name, string deleteExisting, string appendToCSV)`, a "Parameters" section with detailed descriptions for CSVDirectory, Name, and deleteExisting, and a "Returns" section stating "True, action is successful. Otherwise, False."

This screenshot shows the same documentation page for NLCTrain, but with more expanded details. The "Parameters" section now includes the description for the appendToCSV parameter. The "Returns" section remains the same. A new "Level" section is present, stating "Batch level." A new "Details" section provides a detailed explanation of how the action works, mentioning that pages of the batch are used as training data and that the batch needs to be divided into documents where the 'Type' associated with the document is a class created in NLC.

The screenshot shows a web browser window with the URL ibm.com/docs/en/datacap/9.1.5?topic=actions-nlctrain. The page is titled 'NLCTrain - IBM Documentation'. On the left, there's a navigation sidebar for 'IBM Datacap' version 9.1.5, with sections for 'Change version' (set to 9.1.5), 'Show full table of contents' (unchecked), and 'Filter on titles'. Below these are several actions: NLClassify, NLClassifyText, NLCSetCredentials, NLCSetLanguage, NLCSetMinConfidence, and **NLCTrain**, which is currently selected. Under 'NLCTrain', there are sub-sections for 'Email actions', 'Equalize actions', and 'Ewsmail actions'. The main content area is titled 'Details' and describes the 'NLCTrain' action as creating or replacing an NLC Classifier at the batch level. It provides an example code snippet:

```
NLCSetLanguage("en")
NLCSetCredentials("@APPVAR(values/gen/NLCUserName)","@APPVAR(values/adv/NLCPassword)")
NLCTrain("@APPATH(runtime)+..+'\CSV","@APPVAR(values/gen/NLCClassifierName)",0,1)
```

Below the code, it says 'Parent topic:' followed by a link to 'DocumentAnalytics.NaturalLanguageClassifier actions'.

CONCLUSION:

In conclusion, the integration of IBM Cloud Visual Recognition and AI-generated captions represents a significant step in advancing the capabilities of the image recognition system. Implementation of the image classification process with the IBM Cloud Visual Recognition API, have harnessed the power of cutting-edge computer vision technology. Additionally, incorporating natural language generation to create captions for recognized images has made the system more informative and user-friendly. This seamless blend of visual and textual information enhances the overall user experience and opens doors to various applications in fields like content management, accessibility, and more.