IMPLEMENTATION OF INTELLIGENT SOFTWARE USING IBM WATSON AND BLUEMIX

Juraj COLLINÁSZY, Marek BUNDZEL, Iveta ZOLOTOVÁ
Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, Tel.: +421 55 602 2564,
E-mail: j.collinaszy@gmail.com, marek.bundzel@tuke.sk, iveta.zolotova@tuke.sk

ABSTRACT

We present the implementation of intelligent software for image classification on the IBM Bluemix platform using the IBM Watson cognitive services. Our system is voice or text controlled cloud-based service accessible via implemented RESTful API. The core idea behind this system is the ability to learn new classifiers without the need of acquiring training data from the user. To this end, we developed multiple methods aiming to maximize classification accuracy of the system. The system uses web search services to download data used for training the classifiers. We have experimentally evaluated the service and proved that our software is capable of learning and creating new classifiers based upon the user entered keyword.

Keywords: *automatic learning, image classification, IBM Bluemix, IBM Watson*

1. INTRODUCTION

DeepQA Project of IBM addressed the problems of answering questions asked the computer in natural language. *Watson* supercomputer [1] named after Thomas J. Watson, the founder of IBM, is the result of the project. *Watson* is often referred to as the first cognitive system and earned its fame by winning the *Jeopardy!* game against two of the former champions. As of 2017 the name *Watson* designates a scale of cognitive services based in the *IBM Bluemix* Open Cloud Architecture implementation rather than a supercomputer [2]. Watson is available as a set of open APIs and SaaS products. Developers can build applications by combining multiple APIs. Among the offered services are natural language processing and image classification. We have built a software based on these capabilities with the motivation to evaluate the potential of the cognitive services. Many facts and information published in this paper are based on the documentation published by the IBM company. The services we mention here refer to the services provided by the IBM company as of June, 2016. Some of the services have been discontinued since.

The system we have implemented serves for image classification. The majority of the services for object recognition and image classification are based on deep convolutional neural networks. Today, many architectures are used but the pioneering work was done among others by Fukushima on the *Neocognitron* [3] and by LeCun on the *LeNet* networks [4]. More on the topic of deep learning can be found in [5]. However, the developers unskilled in neural networks are usually not able to train or fine tune such networks for their own purposes. The *Visual Recognition* service provided through the *Watson Developer Cloud* platform enables to train image classifiers. The user must provide positive and negative examples of the objects of interest. Compiling the training set is usually tedious and time consuming. Therefore we have decided to implement a system that will learn to recognize new objects based just on the verbally entered class name. We have implemented methods for automated classifier production and created a system for demonstration of the *IBM Bluemix* and *IBM Watson* capabilities.

2. IBM BLUEMIX AND IBM WATSON

IBM Bluemix [6] is a cloud offering based on IBM's *Cloud Foundry* project which is an open source platform as a service. It enables the developers to create, deploy and manage applications on the cloud via the *DevOps* service. It was open to the public in 2014. *Bluemix* supports several programming languages and enables three different ways to run a code. The code can be executed on *Cloud Foundry* infrastructure directly, in a container representing an environment providing everything the application needs and on a virtual server isolating the solution on a public cloud. This platform also supports two types of applications: mobile application that run outside of *Bluemix* and web applications loaded inside the platform. The cognitive services are available via the *Watson Developer Cloud* [7]. These services utilize the methods of machine learning and artificial intelligence. Not all of the services provided are cognitive but the majority of the services is oriented on processing of unstructured data and natural language. The services are divided into four subcategories and several of the services were available as beta versions during writing of this paper. IBM is very active in the development of the services and therefore the *Watson Developer Cloud* environment is dynamic. Upgrades and releases of new versions are frequent. We have used the services described in the following subsections. More details on the services are available at the IBM websites listed in the references.

2.1. Visual Recognition

Visual Recognition uses machine learning and semantic classifiers to recognize visual entities such as environments, objects and events depending on the image properties such as color, texture and shape. This service is able to recognize a set of pre-trained classes based on the *IBM Multimedia Analysis and Retrieval System* [8]. The service enabled users to train new classifiers in December 2015.

2.2. Concept Expansion and Concept Insights

Concept Expansion is a service applying a fast algorithm for pattern comparison to expand the entity terminology. The purpose is to find terms that have similar concept or meaning as the given term. The *Concept Insights* service performs conceptual analysis and indexing of documents selected by the user. The service builds a conceptual model based on the given documents and uses the model to search for conceptually similar documents. The relations between the documents are modeled in a graph that is also offered to the user. The system downloads data from the free online encyclopedia *Wikipedia.*

2.3. Speech to Text and Text to Speech

These services aid in building natural language user interfaces. *Speech to Text* provides a transcript of natural language into text. Artificial intelligence is used to combine grammatical and language structures with the processing of voice signal for a more accurate identification of words. On the other hand, *Text to Speech* service synthesizes speech from a text file while adjusting rhythm and intonation. The words are synthesized in real time in several languages.

2.4. Tradeoff Analytics

Tradeoff Analytics is a service oriented on multi-criteria decision making utilizing Pareto optimization. The service's input is the problem formulated in JSON data interchange format [9] and the output is the problem analysis also in JSON format. The output is difficult to read and therefore IBM offers libraries for further processing of the output.

3. BUILDING THE SYSTEM FOR AUTOMATED IMAGE CLASSIFICATION

The application utilizes the possibility given by the *Watson Visual Recognition* service to build new classifiers and to extend the functionality of the in-built classifiers. By the term "automated image classification" we understand a system that is able to train new classifiers automatically based on the context or based on the user request without the need to explicitly define the training set. The user enters the request verbally by stating the name of the new object to be recognized in the images. The system recognizes the concept, extends it, and downloads an appropriate set of images containing positive and negative examples of the object to be recognized. Besides the system for automated training of classifiers we have implemented also an interface for voice communication with the system. The complete system capable of voice communication with the user exists in the form of a cloud service. This minimizes the need for local programming and ensures platform independence.

3.1. Automated Classifier Training

The classifiers trained by the *Visual Recognition* service are discriminative. Therefore the training set must be comprised of positive and negative examples of the object to be recognized. We have used *Bing Search API* [10] published by Microsoft to download the images from the web. Together we have used 15000 positive and 37500 negative examples of 15 objects. The positive examples are searched for using the keyword verbally entered by the user. The problem arises when searching for the negative examples. We have used the *Concept Expansion* service to find words that are conceptually similar to the object to be recognized. When the keywords for positive and negative examples have been established the images are searched for and downloaded via the *Bing Search API*. The training set is primarily compiled so that it contains equal number of positive and negative examples. Although the *Visual Recognition* service does not provide any means to alter the training parameters or the architecture of the classifier, we compile a balanced training set because it is a good practice when training artificial neural networks. The images are then used to train a new classifier as it can be seen in Fig. 1. We call this process the "simple training".

Fig. 1 Simple training

3.2. Double Training

The disadvantage of the simple training is that many of the positive examples downloaded are in fact negative examples misinterpreted by *Bing*. The results are sorted by relevance and the likelihood of a correct match is high by the first results decreasing gradually. We have developed a "double training" method aiming to minimize noise in the training set. Our algorithm first downloads 1000 search results. Then an auxiliary classifier is trained using the first 100 results as positive and the last 100 results as the negative examples. The training set compiled for the simple training is then validated by the auxiliary classifier and only the images that are labeled the same by *Bing* and by the auxiliary classifier are used to build the final classifier. The process is shown in Fig. 2.

Fig. 2 Double training

With this type of training the number of the training images and the number of concepts used are not fixed because of different number of images passing the auxiliary classifier for each concept. The classifier training time increases significantly because of the need to classify a large amount of images during the training. The

advantage of this training method is the reduced noise in the training set and the disadvantage is the long training time and some amount of uncertainty introduced by varying sizes of training data.

3.3. Training with Adaptive Concepts

Both of the earlier mentioned methods as well as majority of the tested commercially available solutions tend to produce larger amounts of false positive results. To address this issue we have developed a method we call the "training with adaptive concepts". More than a method to compile a training set for a single classifier it is a method to train and retrain a group of classifiers. The aim of this method is to achieve high accuracy rates by creating a system for discrimination between smaller amounts of objects rather than a general object classification system. Here we do not determine the negative examples using the *Concept Expansion* service but we use all the concepts the system already knows as negative examples. At first we train a new classifier using the simple training method but we use the *Concept Insights* service to determine the negative examples. The *Concept Insights* service generates concepts equal or very similar to those generated by *Concept Expansion* service and it was chosen for this method because it was more stable. Every new classifier after the first one uses the positive examples of other already trained classifiers as the negative examples for itself. The goal is to train the group of classifiers to be mutually exclusive. After a new classifier is trained all the existing classifiers are retrained using the positive examples of the new classifier as their own negative examples. This is also a significant disadvantage making this approach suitable for training of only a limited number of classifiers. Process of the training with adaptive concepts is shown in Fig. 3.

Fig. 3 Training with adaptive concepts

Unless the number of needed classifiers is known beforehand, training of each of the new classifiers requires retraining of all of the old classifiers increasing the training time significantly. In order to avoid imbalance between the positive and negative examples in the training sets of the individual classifiers random sampling of the negative examples is used. This method is still functional after the *Concept Expansion* service was discontinued in June 2016.

3.4. The Structure of the Application

The system for automated image recognition can be structured based on the purpose (the presentation and the experimental part) or based on the location where the applications are running (cloud or local). The structure of the system is illustrated in the Fig. 4. The cloud applications are on the right designated "Bluemix" the local applications are on the left, designated "Local".

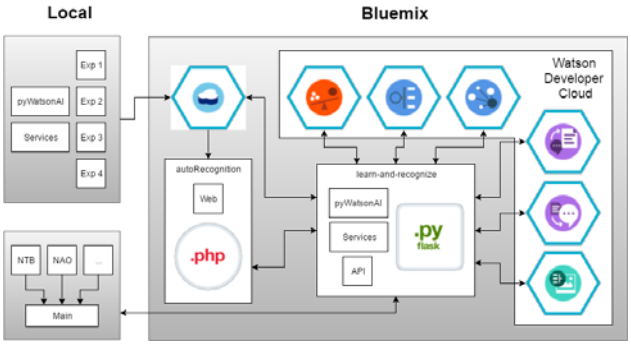

Fig. 4 Structure of the system

The implemented cloud API (learn-and-recognize) running on *Bluemix* is the core of the system. It communicated with the *Bluemix* services (Watson Developer Cloud) and with the local applications. The core API uses Flask web development framework for Python [11] to provide the API services. The core API is also the main component of the presentation part that is responsible for the conversation with the user. The communication module implements the main loop and uses libraries to communicate with various peripherals (Nao humanoid robot of SoftBank Robotics and MS Windows in our case). Based on the user input new classifiers are trained by the core API using the above described approaches. The user input is processed by the *Speech to Text* service, interpreted and executed. The response is formulated and presented either visually or in natural language using the *Text to Speech* service. The cloud API reduces the need for local applications to the minimum. The locally running apps record and play sound files and acquire the images from the local camera.

The purpose of the experimental part is to execute automated experiments. The results of the experiments (Exp. 1. – Exp. 4.) are stored in the *Cloudant noSQL* database on the *Bluemix* cloud. The experiments are started locally and use a separate instance of the *Visual Recognition* service. On the contrary to the presentation part that runs online and builds new classifiers upon the user's request the experimental part downloads and prepares all data before building the classifiers. We have used the experiments to quantify the usability of the system under various setups.

Besides the applications described above we have also implemented a website for reading the experimental results and management of the classifiers via a web interface. The web page written mainly in *Javascript* and *PHP* ran on the *Bluemix* platform under the name *autoRecognition* but it is not operational anymore. We have also implemented the *pyWatsonAI* and *Services* libraries that manage communication with the *Watson Developer Cloud* and implement supporting functions. The supporting functions for example download and read information about an object from *Wikipedia.* The source codes are available for non-commercial use upon request to the authors.

3.5. Conversation with the System

The conversation with the system is scripted. This means that although variables are used in the conversation the system recognizes only certain requests and responds with certain answers.

Table 1 Conversation with the system

Request	Answer	Purpose
What do you know?	I know *X*, ..., *Y* and *Z*.	Lists known objects of classification.
Learn *X*!	Now I know *X*.	Trains a new classifier.
Forget *X*!	I do not know *X* anymore.	Erases an existing classifier.
What is *X*?	*Wiki*	Reads information from *Wikipedia* about the object X.
What is this?	This is *X*. / I do not know.	Classifies an object shown to the local camera.
Tell me about this!	*Wiki* / I do not know this.	Classifies an object shown to the local camera. Then it either reads its description from *Wikipedia* or responds that it did not recognize the object.
Is this *X*?	Yes. / No.	Classification of the object shown to the local camera. New classifier is trained automatically.
Study *X*!	Now I know everything about *X*.	Trains a new classifiers for object X and other objects conceptually related to object X.
Thank you.	No problem.	Terminates the communication.

The system recognizes 9 requests listed in Tab. 1. The *X*, *Y* and *Z* letters represent variables. The *Wiki* string represents description of the concept retrieved from *Wikipedia* generated by the *Bluemix* cloud *Concept Insights* service. An example of a conversation with the system via a Nao humanoid robot is available at https://youtu.be/j56aLvmcyhY (Feb. 2017).

4. EXPERIMENTAL EVALUATION

The aim of these experiments was to find the optimal number of training samples and concepts for negative samples as well as examining various methods of training and comparing their classification accuracies to commercially available image content analysis services. For the experiments we have repeatedly trained the system to recognize 15 objects. 150 images not used in the training were used for testing. The same images were used to test the accuracy of commercially available image content analysis services. We have selected the objects to represent simpler classification tasks such as apples that do not vary in shape much and more complex such as lamps that come in many shapes and colors. The classifiers that are constructed by the *Watson Visual Recognition* service must be trained on positive and negative examples (images). The images are automatically selected by a web search that is not 100% accurate therefore the training set is inherently noisy. Usually the more training samples are used the more general the resulting classifier is, assuming there is little noise in the training set. We have empirically established that the trained classifiers reach the peak performance when trained on between 200 and 300 examples. Please note that the individual dichotomic discriminative classifiers are combined into the multiclass classification system the performance of which is also evaluated.

Table 2 The negative concepts generated by *Concept Expansion*, example

Input	Negative concepts
Apples	Apples, Pears, Peaches, Strawberries, Apricots, Blueberries
Banana	Banana, Pineapple, Coconut, Strawberry, Mango, Chocolate Walnut
Door	Door, Dead bolt, Swung open, Flung open, Hallway, Propped open
Flower	Flower, Blossom, Petals, Lisianthus, Hydrangea, Sweetpeas
Lamp	Lamp, Bulb, Halogen, Dimmable, Light fixture, Compact fluorescent
Mug	Mug, Tumbler, Coffee cup, Shot glasses, Ceramic coffee, Tea cup
Smartphone	Smartphone, Handset, Andriod powered, Samsungs, Phones, Android based

The negative examples of every object were found using the keywords determined by the *Concept Expansion* service. We call these keywords the “negative concepts”. Tab. 2 shows examples of the generated negative concepts. The next problem was to determine how many negative concepts are to be used in the training. For this,

we have trained the classifiers using the simple training method described above on training sets that have been algorithmically assembled and contained examples of 1-5 negative concepts. We have established that the optimal number of negative concepts for the classifier to be trained on is 2. The results of the experiment are summarized in Tab. 3.

Table 3 Relation between the number of negative concepts used and the classifiers' performances

No. of negative concepts	Average accuracy	False positives	False negatives
1	60 %	29 %	11 %
2	64 %	26,7 %	9,3 %
3	60,3 %	28,7 %	11 %
4	62,3 %	24,3 %	13,3 %
5	62,3 %	23,3 %	14,3 %

We have performed experiments to test how the noise in the training sets influences the resulting classifiers' quality. We have assembled the training sets manually, hand picking the positive and negative examples of the objects. These sets did not contain contradicting examples. The average accuracy of the classifiers trained on the manually assembled training sets was 62.7% with 25% of false positives and 12.3% of false negatives. This is not an improvement over the use of the algorithmically assembled training sets.

We have also tested the double training and training with adaptive concepts methods. The double training method addresses the problem that the search results for positive examples often do not contain the object of interest. The auxiliary classifier is constructed and variable number of images is used for the training depending on the auxiliary classifiers decision. The average accuracy of the classifiers trained with double training was 59.7% with 19.7% of false positives and 20.7% of false negatives. We deem this approach as unsuitable. The reason for the failure is the problematic performance of the auxiliary classifier. When we have compared the results of simple training, double training and the training with manually assembled training sets we have concluded that adding the noise in the training set causes slow deterioration of the classification accuracy but that other factors are more important for constructing a highly accurate classifier.

The training with adaptive concepts addresses the problem of a high number of false positive classifications of the trained classifiers. The system using this type of training achieved the average accuracy of 86.7% with 8.7% of false positives and 4.7% of false negatives. This results are superior to the other methods we have implemented and superior to tested commercially available services, as shown in Tab. 4. The problem is that the system is limited to classify only a limited set of objects. This method is able to retrain smaller sets of classifiers fast and while it is not great at general recognition tasks, it performs greatly at tasks that require discrimination between a few objects at a time.

Table 4 Comparison of accuracies including commercial solutions

Service/method	Accuracy	Note
Training with adaptive concepts	86.7%	Small amount of objects.
Clarifai	84.3%	[12], many false positives.
Google Vision API	71.3%	[13].
Simple training	64%	200 samples, 2 negative concepts
Alchemy Vision	60.7%	Part of *Watson Developer Cloud*, no false positives.
Double training	59.7%	
Watson Visual Recognition	52.3%	In beta testing as of June 2016.

5. CONCLUSIONS

We have created a voice controlled learning system for automated object recognition. The system is based on IBM *Bluemix* platform using the cognitive services of IBM *Watson.* We have used several cognitive services to compile a working cloud-based application and we have tested the options for improvement of the overall classification performance of the system. We have concluded that with the state-of-the-art cognitive services available it is possible to build intelligent software by developers with only a limited knowledge of the methods of artificial intelligence (mainly computer vision in our case). We have also concluded that using the knowledge on the assumed methods the providers of the intelligent services can greatly improve the performance of the resulting system as we have shown with the training with adaptive concepts method. The system that we have developed is a suitable solution for problems where a limited set of up to app. 50 objects is to be recognized. We have added the explanatory functionality to the system by using and reading the information that *Wikipedia* stores about the objects of interest. The implemented system is available on a wide range of end devices. We have demonstrated its function on a Nao humanoid robot and on MS Windows equipped computers.

ACKNOWLEDGMENTS

This publication is the result of the Project implementation: Grant KEGA - 001TUKE-4/2015 (50%) and by University Science Park TECHNICOM for

Innovation Applications Supported by Knowledge Technology, the second phase of project USP TECHNICOM, supported by the Research & Development Operational Programme funded by the ERDF (50%).

REFERENCES

[1] FERRUCI, D. et al.: Building Watson: An Overview of the DeepQA Project, *AI Magazine,* Vol. 31, No. 3, 2010, pp. 59-79.

[2] Watson, https://www.ibm.com/watson, retrieved Feb. 2017.

[3] FUKUSHIMA, K.: Neocognitron. A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biological Cybernetics, 36 (4): 93–202, 1980.

[4] LECUN, Y. et al.: Handwritten digit recognition with a back-propagation network, in Touretzky, David (Eds), Advances in Neural Information Processing Systems (NIPS 1989), 2, Morgan Kaufman, Denver, CO, 1990.

[5] LECUN, Y. – BENGIO, Y. – HINTON, G.: Deep learning, Nature 521, 436–444 (28 May 2015) doi:10.1038/nature14539.

[6] Bluemix, https://www.ibm.com/cloud-computing/bluemix/, retrieved Feb. 2017.

[7] Watson Developer Cloud, https://www.ibm.com/watson/developercloud/, retrieved Feb. 2017.

[8] NATSEV, A. et al.: IBM multimedia analysis and retrieval system. In CIVR '08 Proceedings of the 2008 international conference on Content-based image and video retrieval, Niagara Falls, Canada, July 07 - 09, 2008, pp. 553-554.

[9] The JSON Data Interchange Format (PDF), http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf, ECMA International, October 2013, retrieved Feb. 2017.

[10] Bing Search API, http://datamarket.azure.com/dataset/bing/search, retrieved Dec. 2016.

[11] Flask (A Python Microframework), http://flask.pocoo.org/, retrieved Feb. 2017.

[12] Clarifai, An Image Recognition API, https://www.clarifai.com/, retrieved Feb. 2017.

[13] Vision API – Image Content Analysis, Google Cloud Platform, https://cloud.google.com/vision/ retrieved Feb. 2017.

Received February 13, 2017, accepted March 31, 2017

BIOGRAPHIES

Juraj Collinászy has attained his bachelor's degree in Informatics at the Department of the Computers and Informatics of the Faculty of Electrical Engineering and Informatics at Technical University of Košice in 2013. His interest in artificial intelligence led him to transfer to the department of Cybernetics and Artificial Intelligence at the same university, where he earned master's degree in Intelligent Systems in 2016. His research interests are focused specifically on Machine Learning and Computer Vision.

Marek Bundzel is with the Technical University of Košice, Department of Cybernetics and Artificial Intelligence, Slovakia. He is active in teaching and research. His expertise includes mainly methods of computational intelligence like Artificial Neural Networks, Support Vector Machines and Evolutionary algorithms and applications of the above methods in pattern recognition, forecast and robotics. Marek Bundzel has spent two years at Waseda University, Tokyo where he was developing a model based on the memory-prediction framework, a theory of brain function, for the purposes of object recognition in mobile robot.

Iveta Zolotová graduated at the Department of Technical Cybernetics of the Faculty of Electrical Engineering, Technical University of Košice, Slovakia in 1983. She defended her C.Sc. in the field of hierarchical representation of digital image in 1987. Since 2010 she has been working as a Professor at the Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Slovakia. Her scientific research is focused on networked control and information systems, supervisory control, data acquisition, human machine interface and web labs. She also investigates issues related to digital image processing.