Innovation assignment:
Microsoft’s CaptionBot

Artificial intelligence

Primer cuatrimestre curso 2019-2020

Daniel Torres Cirina
Jose Ignacio Bustamante Vargas
Albert Mercade Plasencia

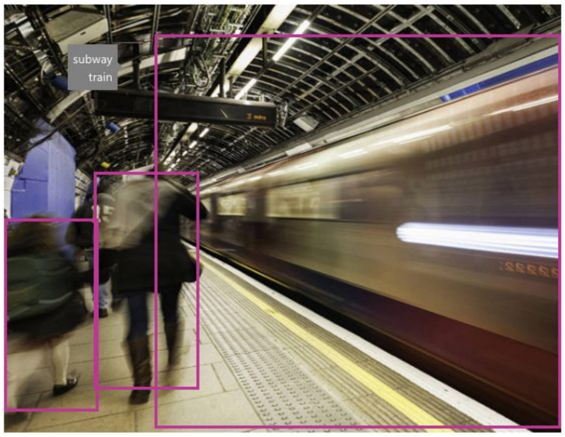
Index

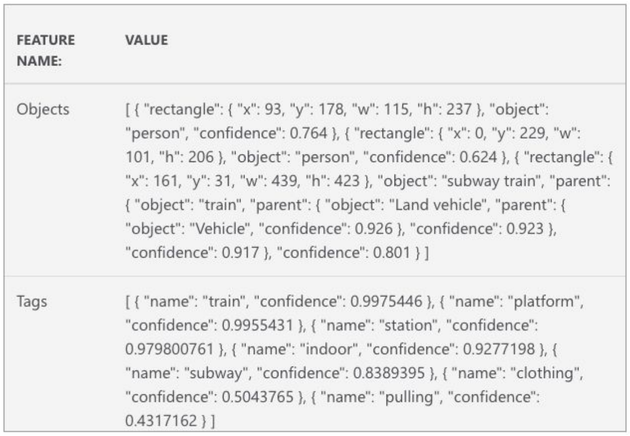
1. Introduction

For this project in innovation we have chosen a Microsoft product called CaptionBot. It uses artificial intelligence for any given photo to analyze it and extract complete information of the images to classify and process visual data.

Specifically, for parts of the paper we focus on one of CaptionBot's aspects. Which is the API developed to analyze a picture, identify the objects or landmarks and add a tag.

This tag is used to describe said object or landmark based on a level of confidence.





FEATURE NAME:	VALUE
Objects	[{ "rectangle": { "x": 93, "y": 178, "w": 115, "h": 237 }, "object": "person", "confidence": 0.764 }, { "rectangle": { "x": 0, "y": 229, "w": 101, "h": 206 }, "object": "person", "confidence": 0.624 }, { "rectangle": { "x": 161, "y": 31, "w": 439, "h": 423 }, "object": "subway train", "parent": { "object": "train", "parent": { "object": "Land vehicle", "parent": { "object": "Vehicle", "confidence": 0.926 }, "confidence": 0.923 }, "confidence": 0.917 }, "confidence": 0.801 }]
Tags	[{ "name": "train", "confidence": 0.9975446 }, { "name": "platform", "confidence": 0.9955431 }, { "name": "station", "confidence": 0.979800761 }, { "name": "indoor", "confidence": 0.9277198 }, { "name": "subway", "confidence": 0.8389395 }, { "name": "clothing", "confidence": 0.5043765 }, { "name": "pulling", "confidence": 0.4317162 }]

- **Figure 1**. CaptionBot. To the left, drawing of the bounding boxes for 2 persons and a subway train. To the right, objects returned by the API composed of the tags and levels of confidence.

2. Product description

CaptionBot.ai is powered by deep learning technology that identifies and captions photos. When a photo is uploaded, it is sent to Microsoft for image analysis to return a caption. It can model objects in photographs so that a computer can understand them.

The system can recognise a broad range of visual concepts and also performs entity extraction.

It incorporates three separate services to process the images. The Computer Vision API identifies the components of the photo, it mixes that with data from the Bing Image Search API, and runs any faces it spots through their Emotion API. This analyses human facial expressions to detect anger, contempt, disgust, fear, happiness, sadness or surprise. It can also recognize animals and describe landscapes, although it does respond with "I am not really confident" to quite a few images.

The machine has been trained to understand how a human understands the image.

3. Description of the AI techniques that have been used

CaptionBot is made of the Computer Vision API, the Emotion API and the Bing Image API.[1]

Azure's Computer Vision service provides access to advanced algorithms that process images and return information, depending on the visual features of interest. For example, Computer Vision can determine if an image contains adult content, or it can find all of the human faces in an image. It’s possible to analyze images to detect and provide insights about their visual features and characteristics.

With the action “Tag visual features” from the Computer Vision API it’s possible to identify and tag visual features in an image, from a set of thousands of recognizable objects, living things, scenery, and actions. When the tags are ambiguous or not common knowledge, the API response provides hints to clarify the context of the tag. Tagging isn't limited to the main subject, such as a person in the foreground, but also includes the setting (indoor or outdoor), furniture, tools, plants, animals, accessories, gadgets, and so on.[2]

And with “Detect objects” the API returns the bounding box coordinates for each tag applied. For example, if an image contains a dog, cat and person, the Detect operation will list those objects together with their coordinates in the image. This functionality can be used to process further relationships between the objects in an image. It will also let us know when there are multiple instances of the same tag in an image.

Through the use of these API actions, it’s possible to tag and caption (almost) any image.

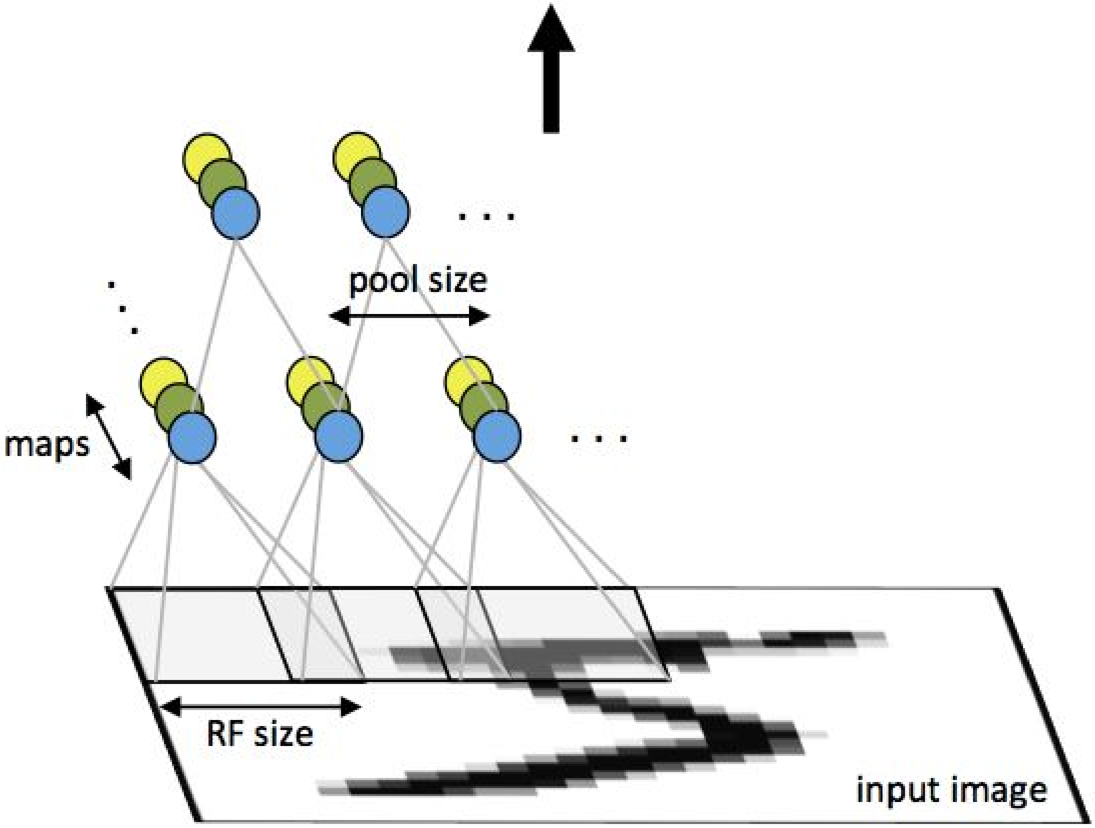
4. Description of how the techniques have been used

First, it predicts the words that are likely to appear in a legend, using a convolutional neural network (also known as CNN) to recognize what is in the image. The convolutional neural network is trained with many examples of images and legends, and automatically learns features such as fragments of a single color, shapes and other features. Next, it uses a language model to take that set of words and create possible coherent legends. Finally, it implements a verifier that measures the general semantic similarity between the legend and the image, to choose the best possible one.[2]

The architecture of a CNN is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal). This is achieved with local connections and tied weights followed by some form of pooling which results in translation invariant features. Another benefit of CNNs is that they are easier to train and have many fewer parameters than fully connected networks with the same number of hidden units.

Architecture of a convolutional neural network[3]

A CNN consists of a number of convolutional and subsampling layers optionally followed by fully connected layers. The input to a convolutional layer is a m x m x r image where m is the height and width of the image and r is the number of channels, e.g. an RGB image has r = 3. The convolutional layer will have k filters (or kernels) of size n x n x q where n is smaller than the dimension of the image and q can either be the same as the number of channels r or smaller and may vary for each kernel. The size of the filters gives rise to the locally connected structure which are each convolved with the image to produce k feature maps of size $m−n+1$. Each map is then subsampled typically with mean or max pooling over p x p contiguous regions where p ranges between 2 for small images (e.g. MNIST, a large database of handwritten digits that is commonly used for training various image processing systems) and is usually not more than 5 for larger inputs. Either before or after the subsampling layer an additive bias and sigmoidal (a mathematical function having a characteristic "S"-shaped curve or sigmoid curve) nonlinearity (means the output is not simply a constant scaling of the input variables) is applied to each feature map.



- **Figure 2**. First layer of a convolutional neural network with pooling. Units of the same color have tied weights and units of different colors represent different filter maps.

After the convolutional layers there may be any number of fully connected layers.

In order to use gradient based optimization, a back propagation algorithm is needed to compute the gradient with respect to the parameters of the model.

Back Propagation[3]

Let $\delta^{(l+1)}$ be the error term for the (l+1)-st layer in the network with a cost function *J(W,b;x,y)* where (*W,b*) are the parameters and (*x,y*) are the training data and label pairs. If the l-th layer is densely connected to the (l+1)-st layer, then the error for the l-th layer is computed as:

$$\delta^{(l)} = ((W^{(l)})^T\delta^{(l+1)}) \cdot f'(z^{(l)})$$

- **Figure 3**. Error for the l-th layer in the network.

and the gradients are:

$$\nabla_{W^{(l)}} J(W,b;x,y) = \delta^{(l+1)}(a^{(l)})^T,$$
$$\nabla_{b^{(l)}} J(W,b;x,y) = \delta^{(l+1)}$$

- **Figure 4**. The gradients.

If the l-th layer is a convolutional and subsampling layer then the error is propagated through as:

$$\delta^{(l)}_k = \mathrm{upsample}((W^{(l)}_k)^T\delta^{(l+1)}_k) \cdot f'(z^{(l)}_k)$$

- **Figure 5**. The error propagated through if the l-th layer is a convolutional and subsampling layer.

Where *k* indexes the filter number and $f'(z^{(l)}_k)$ is the derivative of the activation function. The *upsample* operation has to propagate the error through the pooling layer by calculating the error *w.r.t* to each unit incoming to the pooling layer. For example, if we have mean pooling then *upsample* simply uniformly distributes the error for a single pooling unit among the units which feed into it in the previous layer. In max pooling the unit which was chosen as the max receives all the error since very small changes in input would perturb the result only through that unit.

Finally, to calculate the gradient *w.r.t* to the filter maps, we rely on the border handling convolution operation again and flip the error matrix $\delta^{(l)}_k$ the same way we flip the filters in the convolutional layer.

$$\nabla_{W^{(l)}_k} J(W,b;x,y) = \sum_{(i=1,m)} [\ (a^{(l)}_i) * \mathrm{rot90}(\delta^{(l+1)}_k,2)\ ],$$
$$\nabla_{b^{(l)}_k} J(W,b;x,y) = \sum_{(a,b)} [\ (\delta^{(l+1)}_k)_{a,b}\ ].$$

- **Figure 6**. Calculation for the gradient *w.r.t* to the filter maps.

Where $a^{(l)}$ is the input to the *l*-th layer, and $a^{(1)}$ is the input image. The operation $(a^{(l)}_i)*\delta^{(l+1)}_k$ is the “valid” convolution between *i*-th input in the *l*-th layer and the error *w.r.t.* the *k*-th filter.

5. Why it is an innovative product and the nature of innovation

As explained in the paper presented by Microsoft researchers Kenneth Tran and Xiaodong He[4], on which CaptionBot is partly based, the systems that existed until now for caption generation hadn't been successful at all, as a matter of fact they only worked kind of well under controlled environments where the images fed to system were very similar to the training examples used. These systems haven't been tested in an open environment and it isn't clear how they would perform. Moreover, these systems usually only gave a generic description of the picture without acknowledging the presence in the image of key entities such as celebrities and landmarks, which, as Tran and He explain in the paper, are very present and important in our common sense and the way we think and look at these images.

The approach taken by Microsoft using the technologies and techniques explained above have resulted in a system that generally works in an open domain environment. On top of that, it's also able to identify well known entities to all of us such as celebrities and landmarks, which previous systems weren't able to do. Moreover, to further enhance the caption generation system they created a confidence model that is able to estimate a confidence score for the captions based on the visual and text features thus providing the system with a caption even in the most difficult situations and pictures for the system.

In order to test the quality of the captioning done by this new system, they carried out human evaluation experiments where the humans gave their impressions on the results generated. The results showed that this system outperforms previous similar systems in both in- and out-of-domain datasets. And when tested with the most challenging dataset, which sourced random images from Instagram, the results showed that this system improves the human satisfaction rate by 94.9%.

Thus, we can conclude that this system is innovative in the sense that it is using previously existing technologies such as CNNs and newly created ones such as the confidence model mentioned above to generate captions in an open environment that are producing never before seen results.

6. The product impact on the company

Microsoft offer IA solutions in order to provide “Cognitive Services”.[5]
Cognitive Services bring AI within reach of every developer—without requiring machine-learning expertise. All it takes is an API call to embed the ability to see, hear, speak, search, understand, and accelerate decision-making into your apps.

In general terms goods results from the product allows to Microsoft offers and sell this kind of solutions to customers that are interested in it and provide more investors who finance the company.

Some examples of similar products that could improve using this technique is Uber boosts platform security with facial recognition.

7. The impact of the product on the user

The product itself is not useful for the standard user. It works to make “siblings” products that give some real value to the user. The most used by now is for leisure. Some examples are “Celebs like me” or “My moustache”. But if the technique improves its accuracy the sibling that try to predict people’s age from an image could be used for medical purposes.

Another issue to take into consideration is about privacy and security. All the data used for training the models should be properly secured and the terms should be clear and transparents to the final user. Another concern is about the usage (for example using edited images to bully or blackmailing). Consequently could be interesting study a way to identify real images to fake ones (making that every algorithm generates a watermark for example).

8. Bibliography

[1]

https://www.captionbot.ai/Home/Magic

[2]

https://blogs.microsoft.com/ai/picture-this-microsoft-research-project-can-interpret-caption-photos/

http://deeplearning.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/

[3]

https://docs.microsoft.com/en-gb/azure/cognitive-services/computer-vision/concept-tagging-images

[4]

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/06/ImageCaptionInWild-1.pdf

[5]

https://azure.microsoft.com/en-us/services/cognitive-services/