ISM 647

North Carolina Herpetology Image Identification Application Development Assignment
(Snakes on a Plane)

Assignment 3

By

Mark McSwain and Jeanne Reppert

April 29, 2020

Executive Summary

In 2006 Samuel L. Jackson was faced with a plane-load of venomous snakes. To quote Neville Flynn: “I have had it with these *[expletive deleted]* snakes, on this *[expletive deleted]* plane!”. Thankfully, in North Carolina we only have six kinds of venomous snakes to contend with and really only one of those has potential to pop up in our daily lives. Nevertheless when hiking trails throughout North Carolina, the adventurer is apt to come into contact with any of them. An application designed to use image recognition/classification to help determine whether or not a snake is venomous or non-venomous is the ultimate goal of this kind of work.

After investigating the various capabilities of a number of cognitive applications products provided by Microsoft Azure and IBM Watson, we decided to compare Watson Studio’s Visual Recognition Service Object Classifier with Azure Custom Vision Classifier. We used training and test sets of images for both of the applications in an attempt to determine which one is easier to use and more reliable for the task it has been given.

After the task of compiling and sorting a set of about 800 snake images into venomous and non-venomous examples, the images were uploaded into both programs and trained. Subsequently a test set of a little more than 100 images was used to assess the accuracy of both models. This process was repeated with a second set of train images and then tested again in order to verify the results.

Overall we were impressed with the results that both applications provided. In most quantitative measures, both models performed accurately an average of over 90% of the time. In terms of accuracy Azure’s Custom Vision Classifier performed slightly better than Watson’s Object Classifier. However, Watson’s application was much more user friendly particularly when it came to uploaded images and processing test results. For that reason, Watson was the preferred application. While both of these applications were free and thus limited in capacity, both programs demonstrated the potential usefulness of object identifier programs for a variety of tasks within a variety of contexts.

Snakes in North Carolina:
The reason the application was developed.

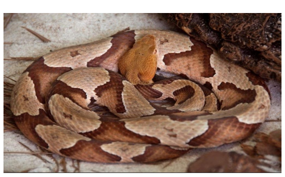
(Describing the problem for which cognitive application was developed.)

Why snakes? An image classification project should be fun and should help people to organize pictures of their kids or puppies. A project to identify venomous snakes may seem repugnant, however, a tool that could allow people to break down a barrier to nature is a worthwhile effort. To many people, snakes are mythical beasts that lay in wait, hoping for the chance to strike against a wayward hiker or wanderer. The reality is that this is just not the case. Snakes are reclusive creatures and in North Carolina there is only one that is common to most areas, the copperhead.[1] They can be found in urban and rural areas and for that reason, they are the one snake that everyone would do well to be able to identify.

In addition to simply being able to identify snakes that are happened upon during a hike or a gardening task, knowing the difference between types of snake is important for preserving a safe natural environment. Non-venomous snakes are often mistaken for venomous snakes but should not be eliminated when found since they serve as predators of more dangerous snakes like copperheads and thus would keep those populations in check. Likewise, a harmless brown snake may cause one to startle, however, they serve an important purpose of keeping the rodent population in check. Knowing which snakes are dangerous and which snakes to encourage in our environment has lasting positive effects on our surroundings.

The venomous snakes[2] in North Carolina are:

Copperhead



The most common snake and responsible for most snakebites in North Carolina. This snake can be found everywhere in North Carolina except the Outer Banks. It is recognized by a copper and brown coloring with hourglass-shaped markings on top of its head. Most adults are between 2 and 4 feet long and juvenile snakes have a greenish color at the tip of their tail. This is a pit viper with a diamond shaped head.

[1] Alvin Braswell, "NC SNAKE FAQ," North Carolina Museum of Natural Sciences, https://naturalsciences.org/learn/faqs/snake-faq#1, Date unknown.

[2] "Venomous Snakes in North Carolina", Times-News, https://www.thetimesnews.com/photogallery/NC/20190628/NEWS/628009980/PH/1, June 28, 2019

Eastern Cottonmouth

These snakes can typically be found near the water and are mostly found in coastal regions of North Carolina. They are heavy bodied snakes with olive, brown or blackish coloring with lighter crossbands. Their name comes from the color of their open mouth. This is a pit viper with a diamond shaped head.

Timber Rattlesnake

This large snake is the most common rattlesnake in North Carolina and can be found throughout the state. They are pinkish to blackish in color with dark, light-centered blotches and crossbands. Adults have black tails and they range from 3 to 6 feet long. This is a pit viper with a diamond shaped head.

Pigmy Rattlesnake

Mostly found in the coastal region and southern Piedmont, this snake has coloring ranges from gray to brown to reddish, with dark brown splotches. They have smaller rattles and adults are between 15 and 25 inches long. This is a pit viper with a diamond shaped head.

Eastern Diamondback Rattlesnake

This snake is rarely found in North Carolina and resides in the southern most coastal areas. They have large rattles on their tails and are identified by their diamond-shaped pattern on brown or grayish backgrounds. They are between 6 and 6 feet long, but may be larger. This is a pit viper with a diamond shaped head.

Eastern Coral Snake

The deadliest snake in North Carolina lacks the anatomy to typically bite unless handled, the coral snake is colorful and known by its yellow-red-yellow-black ringed pattern. “Red on yellow, kill a fellow: red on black, venom lack”. This snake is not a pit viper and is related to the cobra. It can grow up to 3 feet long and is often mistaken for the scarlet king snake.

North Carolina was chosen because of its locus to the members of this team as well as the relative ease with which its venomous snakes can be grouped and identified. Five of the six venomous snakes in North Carolina are Pit Vipers which can be characterized by distinctive features in its head in addition to the markings on its body that are typically used to differentiate types of snakes. The sixth snake, the Eastern Coral Snake has very unique markings and color patterns that do distinguish it from its doppelganger, the Scarlet King Snake.

Selecting an Environment for Image Recognition

(Selection and justification of the selection of one of the two environments.)

For this project two tools were investigated: IBM Watson Studio and Microsoft Azure.

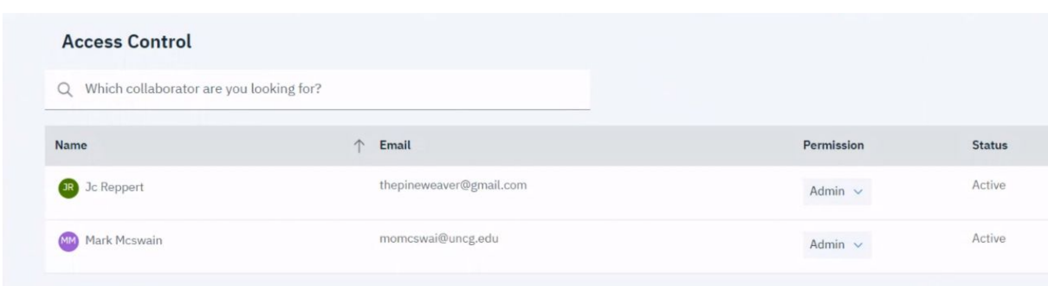
- IBM Watson products have a variety of options for cognitive computing tasks including visual recognition, decision optimization, deep learning, natural language processing, natural language classification, language translator, tone analyzer, personality insights, and text to speech.
- Microsoft Azure offers similar tools such as anomaly detector, immersive reader, text analytics, translator text, language understanding, QnA maker, speech recognition, speech to text, computer vision, face, ink recognizer, and form recognizer. While some of the language oriented tools were explored, it was concluded that the Custom Vision image classifier tools would be explored.

The Environments Watson versus Azure

(Learning the environment and demonstrating it by developing a basic cognitive application
with no to minimal programming requirement.)

Both environments operate in a similar manner. They utilize cloud accounts and manage the training, testing and deployment of applications in a similar fashion. What we have discovered is that, overall, the Watson interface is much easier to use and deploy, while the Azure application yielded more accurate results given the size of the training set of data.

Watson allows the user to easily set up multiple users on the same project and facilitates group development.



The interface can manage uploads of compressed folders of files that can then be grouped according to their class. One drawback to the version of Watson that we used is that there is a

250MB limitation on the size of folders that can be uploaded. However, given that limitation, the ability to train and test with compressed folders makes Watson a very efficient engine with which to classify images. The output from testing is also easy to manage as it outputs a group of tiles with the image name within the tile. Properly named images make it much easier to determine if Watson accurately classified a snake by comparing the image name with the classification. Deployment of the eventual code into an application is made easier with a diverse option of code available for different uses.

Azure, while easy to set up, does not readily facilitate multiple users in the portal that we were using. The setup, however, is straight forward, but it takes quite a bit longer to upload the uncompressed training set of images. Also, once the model is trained, the test images must be loaded and evaluated one by one with evaluation of those tests being one at a time as well. This meant that instead of uploading groups of photos at a time, in Azure 120 test photos had to be uploaded and analyzed by hand, making it a longer and less efficient process. It is possible to ascertain the name of the image, but it is not as apparent as Watson, which made the task of determining false positives and false negatives a manual process.

The Overall Experience with Azure and Watson and Basic Capabilities of each Service

(Accurately describing the development process and commenting on the overall experience.)
(Demonstration of familiarity with the basic capabilities of the selected environment.)

Image Selection:

At first, Watson Studio was employed to classify images for the selected subject, however, we decided to compare our results with Microsoft Azure after exploring both tools.

On the face of things the processes for training, validating, and testing the objects for image classification were very similar. The largest part of the overall process began with identifying appropriate data, processing the data into a usable form and then organizing it for use in IBM Watson's Visual Recognition tool and Microsoft Azure's Custom Vision. As we familiarized ourselves with the program we learned that the training part of the process in studio was limited to 250mb. This limitation meant that the number of images used to train the model would need to be selected carefully. Much effort was made to obtain a dataset of snake images, however the few that were available were not suitable for this limited task.

For example, the best dataset of snake jpeg's was one compiled by an organization called AIcrowd.[3] This organization created a contest where the challenge was snake identification by species as a public health service particularly for the purpose of quick identification so that

[3] LifeCLEF, "Snake Species Identification Challenge", www.aircrowd.com, https://www.aicrowd.com/challenges/snake-species-identification-challenge

antivenom could be administered efficiently and appropriately. The datasets used, however, were compiled of snakes found throughout the world and ranged from 20G to 40G in size. Because the metadata for the jpegs were included in a separate file, significant level of data manipulation and programming would be necessary to process and reduce these jpegs into a usable resource for the tools we were using.

In order to have a collection of photographs large enough for training and testing but small enough to meet the memory requirements of Watson, we collected around 800 photographs and divided them into training and testing sets. Around 120 photographs were used for testing while the remaining photographs were used for training the models. Of these photographs about 10 percent of them were non-snake images. These were introduced to the model at the suggestion of both applications to be used as a control. This created a third class that we only used to help create the model but did not include in our overall results. The rest of the images were evenly divided between images of venomous and non-venomous snakes. We intentionally chose non-venomous snakes that could easily be confused with the six venomous snakes that are native to North Carolina, making the identification process a bit more difficult for our cognitive applications. We chose this approach since one of the difficulties in identifying snakes is that venomous snakes tend to have non-venomous look-alikes.

Training and testing the models:

Watson:

The steps for training the IBM Watson Studio Visual Recognition included registering for an account with cloud access for storage. After the studio was loaded, various tools could be selected for use. Some of these tools required payment for api calls, however, the image classifier tool was free to use with some limitations to the amount of memory used. Once the Visual Recognition tool was loaded into IBM Watson Studio and permissions were shared for both of us to use, the tool was ready to load images for training.

The first task for loading the training images was to identify categories. Two classes were created: venomous and nonvenomous. The tool permitted the user to browse local computer files and upload them to the system. Most conveniently, zip files were able to be used. These files were already divided into venomous training sets and nonvenomous training sets. Each jpeg was named so that the species of the snake could be quickly recognized by the user for analysis. Additionally, since IBM Watson strongly suggests a third category be utilized in which non-snake items would be introduced in order to help the model to learn. For this category a zip file was uploaded with 32 non-snake images. These images included photos of items that might be confused with snakes like ropes, belts, hoses, braided hair and toy snakes. In total, 431 images were uploaded for the initial training phase. This included 219 images of a variety of non-venomous snakes found in North Carolina and 180 images of the six categories of venomous snakes found in North Carolina. While for machine learning, this number of photos would be a relatively small amount of images for an effective model, we were somewhat bound by the constraints of the tool which limited the amount of memory that could be utilized by the task. These photos, along with the non-snake images, brought us very close to the upper bounds of the memory that was permitted in the training process of the program.

The next step once the photos were uploaded and tagged, was to train the model. This required about 30 minutes of processing on the website. Once the training was completed a test set of 60 non-venomous snake images, 60 venomous snake images, and 25 images of items that were not snakes were uploaded into the model. In this stage, the tool required that images were selected separately rather than in a zip file, in order to upload them. Multiple images could however be selected at one time so this step went relatively quickly. The test images were then processed in less than a couple of minutes.

Once the first round of results were gathered and analyzed, a second round of images was added to the system and the model was retrained using the same process as outlined above. After the training process was completed for a second time, the test set was again uploaded and the results were again gathered and analyzed. In the case of Watson, Each of the 120 test images was labeled as to the class that it was assigned to originally and what the predicted class was along with a percentage of certainty. However, the tool did not aggregate the results or provided any metrics for overall accuracy. Thus, we chose to aggregate the results ourselves and calculated the accuracy measures based on a confusion matrix. We constructed ourselves.

Azure:

Model training in Azure was similar to the Watson application with a couple of exceptions. Groups of photos were added to Azure and then tagged based on classes. Because Azure did not accept zip files, this process was a bit lengthier to perform. Like Watson, Azure also strongly suggested a third class of non-snake items be added to the model. Once the images were added and tagged then the model was trained.

Testing the model followed. Unlike Watson, Azure required that one test photo was added at a time. Once the output was given and the photo was classified, this photo was added to a test set automatically but was not labeled clearly enough to be useful for results later. For that reason, the results for each individual photo had to be tallied manually as each individual photo was uploaded. This ended up being a time consuming process for the 120 test photos.

Once again, an additional set of photos was added to the model for additional training. Once those photos were added and the model was trained for a second time, the test photos were uploaded one at a time and the results were again recorded manually. Like Watson, with the exception of a few measures that were applied to the training set, Azure did not provide any overall metrics of how the model performed. As a result, once again, the hand tallied results were utilized to compile a confusion matrix and calculate some accuracy measures so that the two models could be compared objectively. The results from both tests are reviewed in the next section.

How Each Application Performed

(Providing actual evidence of the application and how it is used.)

Overall, both Watson and Azure performed their tasks well. We were surprised, given the limited number of images that were allowed, that the models identified snakes as venomous or non-venomous as accurately as they did. Azure did perform a little better than Watson overall but the differences in the measures were not significant. The number of false positives were higher than false negatives for both models. Interestingly, the model did not improve much (it actually got a little worse in some ways) with the second training set. We also kept track of the number of times that each model recognized the non-snake images accurately. Azure had no trouble identifying non-snake images accurately but Watson was not as successful. However, after a second set of images was added in the second round of training the Watson model performed much better with non-snake images.

We intentionally chose non-snake images that could appear like snakes to a computer (mimicking texture and shape.) This was done mostly out of curiosity to see if the identifier could be misled with these types of images. It is possible that if the non-snake images were not quite as similar to snakes that the model overall may have had a higher accuracy rate. It would be interesting to redo the model training with this in mind to see if the applications performed better or worse than these results.

Confusion Matrices: Outcomes For Training Set

IBM Watson	
TP 56	FN 2
FP 11	TN 50

Test set: 119 non-venomous and venomous snake images (Watson rejected one image) and 25 non-snake images
Accuracy for non-snake images: 11 identified as non-snake, 14 identified as snake

Microsoft Azure	
TP 57	FN 1
FP 7	TN 52

Test set: 117 non-venomous and venomous snake images, (Azure rejected three images due to size)
Accuracy for non-snake images: Azure identified all non-snake images as non-snake.

First Test after Training - Quantitative Measures:

	Accuracy: (TP+TN)/ (TP+TN+FP+FN)	**Specificity:** TN/(TN+FP)	**Precision:** TP/(TP+FP)	**Sensitivity:** TP/(TP+FN)	**F - measure:** (2*Sens.*Prec.)/ Sens. + Prec.
IBM Watson	.89	.82	.84	.97	.90
Microsoft Azure	.93	.88	.89	.98	.93

In every category, the Microsoft Azure model performed better than the IBM Watson model. In these models sensitivity is perhaps the most critical category for identifying venomous versus non-venomous snakes (one would not want to have a snake identified as non-venomous if it was really venomous.) For these cases both models' sensitivity scores were relatively close to one another and fairly high. Nonetheless, I wouldn't handle a snake in this case unless the sensitivity rating of the model was 100% correct!

Results below are after adding the 200 validation set images (venomous and non-venomous) along with another 25 non-snake images. In total that amounted to about 600 non-venomous and venomous images and about 55 non-snake images. With adding all the images together (test images as well) this amounted to about 800 images used. The second round did not perform quite as well as the first but still was fairly close. Azure still outperformed Watson. We maxed out the storage capacity with this amount of images for Watson. (We actually crashed Watson and had to rebuild the model with the second training set.) Azure had limits but we were able to not max them out with the images we used (with the exception of a few images that were rejected due to size - there was a 4mb limit per image.)

Confusion Matrices: Outcomes For Training Set with Validation Images Added

IBM Watson	
TP 53	FN 5
FP 7	TN 55

Test set: 120 Venomous images and Non-Venomous snake images and 25 non-snake images
Accuracy for non-snake images: 5 identified as snake, 20 identified as non-snake (an improvement)

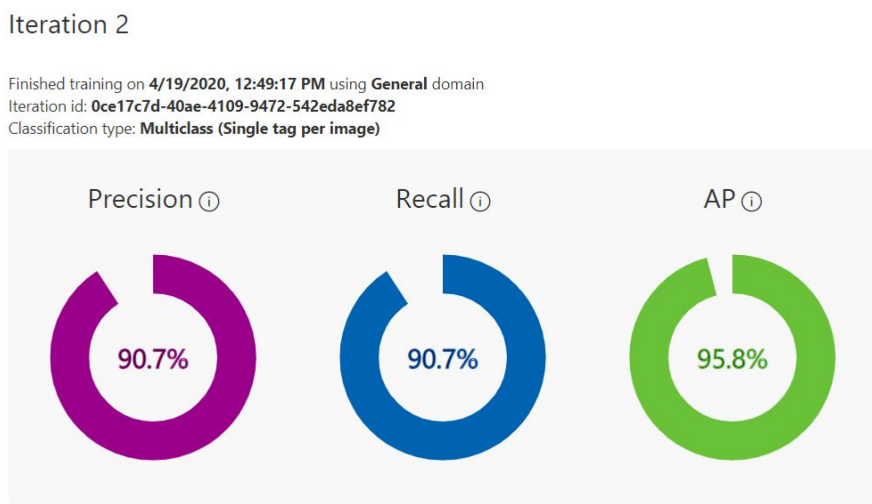
Microsoft Azure	
TP 56	FN 2
FP 7	TN 54

Test set: 119 Venomous Images and Non-Venomous Images, (Azure rejected one image)
Accuracy for non-snake images: Azure identified all non-snake images as non-snake.

Second Test after Adding Validation Set to the Training Set - Quantitative Measures:

	Accuracy: (TP+TN)/ (TP+TN+FP+FN)	**Specificity:** TN/(TN+FP)	**Precision:** TP/(TP+FP)	**Sensitivity:** TP/(TP+FN)	**F - measure:** (2*Sens.*Prec.)/ Sens. + Prec.
IBM Watson	.9	.89	.88	.91	.89
Microsoft Azure	.92	.89	.89	.97	.93

For Azure adding additional images in the second training appeared to decrease the model's overall ratings of the training set as well. Below is a screenshot of Azure's ratings of the training set. These metrics were generated before the test images were added.

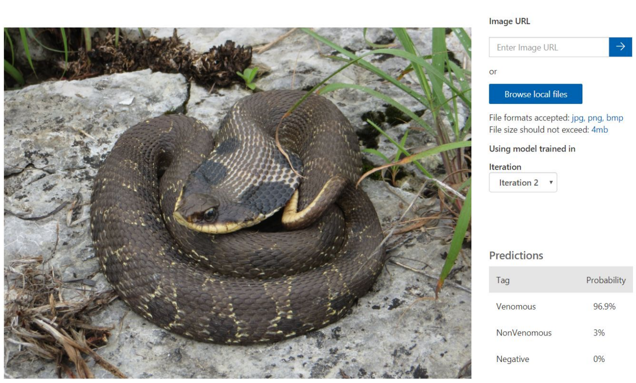


How The Applications Solved The Problem of Identifying Snakes

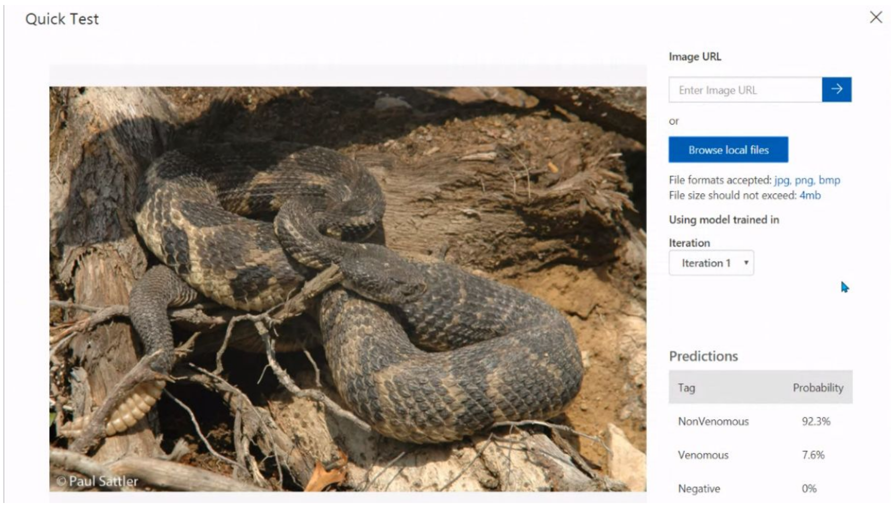
(Justifying the developed cognitive application solves the problems stated above.)

The test set revealed several difficulties with identification within the model. However, in the majority of cases, venomous and non-venomous cases were identified correctly and items that were not snakes were tagged appropriately as well. While both applications performed very well, the Azure solution gave greater precision given the same training set of original image data. Azure had fewer false positives, where non-venomous snakes were wrongly identified as venomous and fewer false negatives, where venomous snakes were wrongly identified as non-venomous.

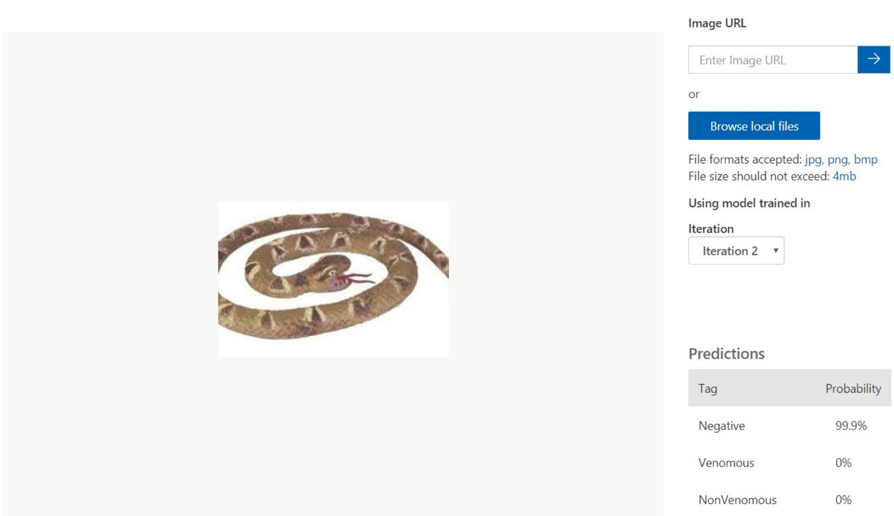
This is an example of a false positive as identified by Azure. This is a hognose snake, also known as a spreading adder or puff adder. The snake has two primary means of defense which include playing dead and spreading the skin around its neck similar to a cobra. In this picture the snake has “spread” its neck which may have caused Azure to mistake it’s head for the triangular head of a pit viper (venomous trait) or it could have mistaken its markings for one of the rattlesnakes (venomous) in the database. Azure identified the snake as venomous with 96.9% probability.



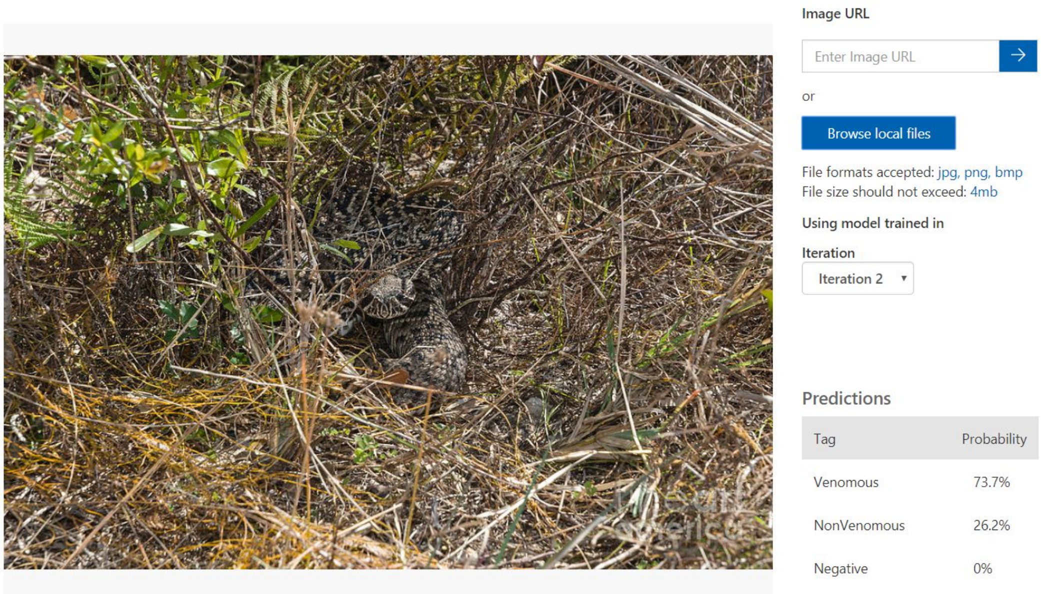
The photo below was the only false positive that Azure returned in the first test set. The model was convinced with a 92.3% probability that this was a non-venomous snake. Interestingly, this snake would be relatively easy to identify by eye due to the clear rattle at the end of this snake's tale even though the head is somewhat obscured. This misclassification is a good example as to why one would want a false negative classification rate extremely close to zero if one was to choose to interact with a snake based on these results.



Azure did a good job of identifying objects that were not snakes. One example is a plastic/rubber snake that was identified with a 99.9% probability as negative (not a snake).

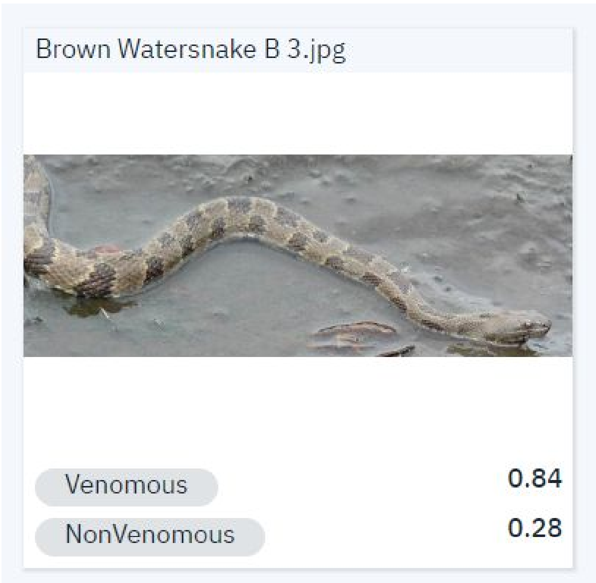


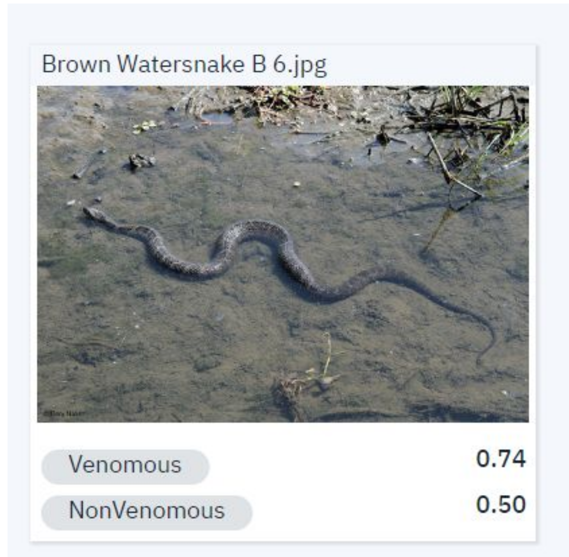
Also of note is Azure’s ability to identify a snake who is very heavily camouflaged.



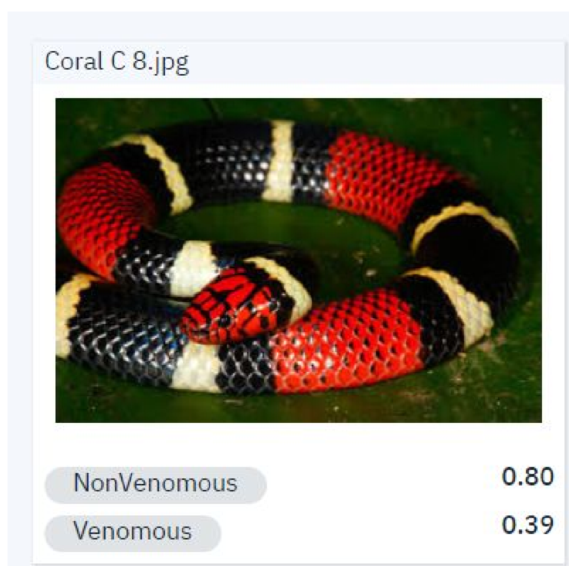
Watson also had difficulty determining that the Eastern Hognose snake was venomous with a 91% probability. Again, the flared head may have confused the classification with a pit viper or the bright markings may have confused this snake with a rattlesnake. In any event, it was a false positive.

The Brown Watersnake was another example of a false positive for Watson. In the two following images, it is unclear whether the water, the markings or the angle of the head confused classification.

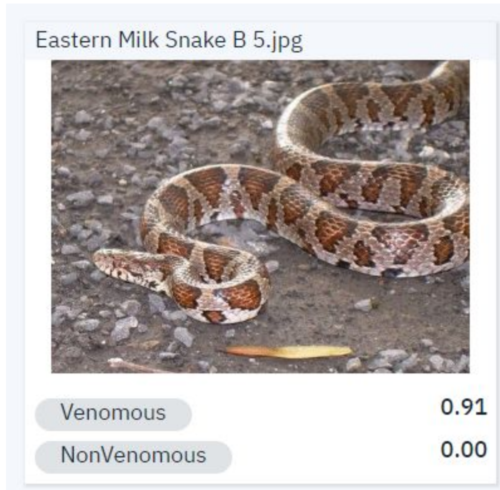




On two occasions Watson returned false negatives identifying two different images of Coral snake as non-venomous. Again, this type of false negative could prove to be highly dangerous. Is interesting however that in most cases the models could tell the difference between the venomous coral snake and the non-venomous snake even though they look almost identical with the exception that the order of the rings of color are different.



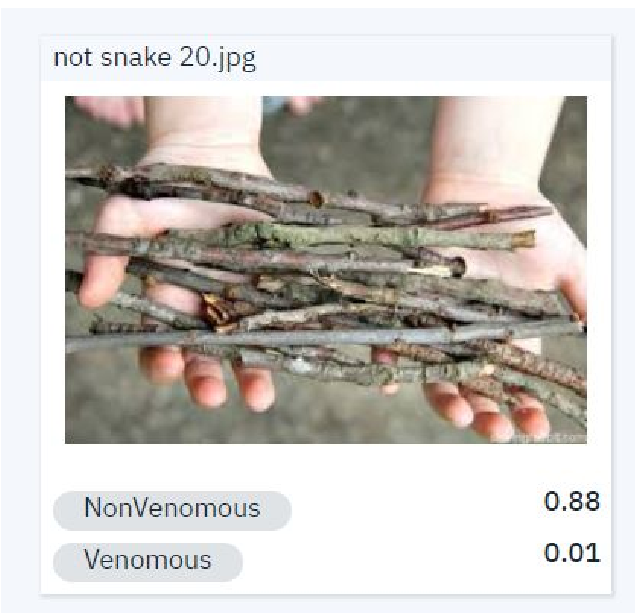
Watson also identified Corn Snakes and milk snakes venomous. Corn Snakes and Copperheads are often mistaken, so this was in some respects to be expected.[4]



Watson returned three false negatives on the Eastern Cottonmouth Snake which is a bit more concerning.

[4] Less Wright, “Building an AI snake identifier with FastAI - first test: Copperhead v Corn Snake”, Data Driven Investor, https://medium.com/datadriveninvestor/building-an-ai-snake-identifier-with-fastai-first-test-copperhead-vs-corn-snake-15b10ec1c807, 01/20/2019

Where Azure did a good job of classifying non-snakes as such, Watson had difficulty classifying this bundle of sticks. Luckily these are non-venomous sticks.



How the Applications Could be Improved

(Explanation of how the application could have been improved if additional resources were available.)

As this is an identification project, the number and variety of images available in the training set will determine the ability of the application to recognize and make decisions. With a minimal set of images an application might be expected to determine if the image submitted in the final phase is “snake” or “not a snake”. More images would allow a determination between “venomous” and “non-venomous”. And, the ultimate, which would perhaps require tens of thousands of images, might be expected to identify a specific snake from a submitted image.

With a paid account, a significant number of images could be added to the training sets for the models. These sets would have to be compiled by the user since few snake databases are available and none that are restricted to the southeast area of the United Snakes. Nonetheless, this would not be a very difficult task.

With some programming and compilation and a significant amount of additional storage and computing power, the 20-40 gig datasets of snake images from around the world could be processed in the same way. In one case, a user of this set recorded created an app that could be used for identification of snakes. An app that utilized location settings could hone in on possible species based on the location of the user and the likelihood that a snake would be encountered in that region.

One problem that would be difficult to overcome with the use of these apps would be how the results are accumulated. In our case, those results were aggregated and then accuracy measures were computed without the use of an application. An application that could output an aggregation of the analysis of the test set along with the appropriate quantitative measures would add a significant improvement to this application.

Finally, it is noted that there were export features that exceeded the scope of this assignment. Potentially a user would be able to utilize a variety of code that could be exported to a website for further interactivity and usage. Additionally, it appeared that the models in Watson could be easily adjusted with a R interface that was present on the tool. Additionally, these models could potentially be used to create a useful mobile application.

Demo to Potential Users

(Showing the demo of the cognitive application to a few potential users and documenting their feedback.)

Given the time and place where this project was completed, sharing the demo of the Watson and Azure engines was restricted to sharing with our families. This project was completed within each of the applications and not deployed through a website or Visual Studio to a mobile app for public consumption. However, one family member, who is a snake enthusiast, found the application to be a unique and potentially valuable tool. Another member, who is an avid hiker affirmed the interest of being able to have such a tool to identify photos of snakes that are sometimes stumbled upon during outdoor adventures. Another family member simply found that looking at the variety of snakes was disconcerting enough and did not find the experience quite as enjoyable. As this was a guided demonstration, all of the participants were impressed that both classification models achieved remarkable results with such a relatively small training set of images. Several were amused by the non-snake items that Watson initially inadvertently identified as snakes (particularly the non-venomous snakes.) A guide was used for initial identification as well as reference for the demonstration.[5]

Notes for Image Recognition Developers

(Developing a "How-to" document for aspiring Cognitive Technology application developers who do not know anything about either Watson or Azure.)

Both applications require cloud accounts which provide server and storage for the user. Before either applications can be loaded the user must set up an IBM Watson account as well as a Microsoft Azure account. From there the processes, while similar are outlined herein:

IBM Watson Studio's Visual Recognition Service Object Classifier

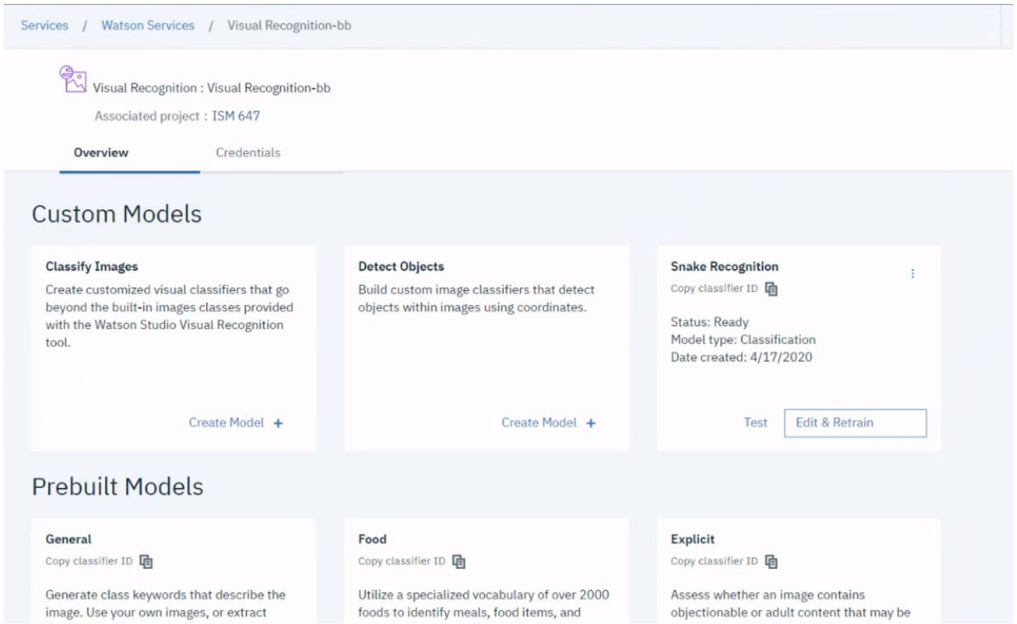
A. **Defining the Problem:** The problem involves gathering and grouping images in such a way that Watson can digest a training set, a test set and ultimately a validation set of data. As no substantive databases of snakes could be found to load en mass, it was determined that the easiest way to manage the images was to rename them according to their class and type. For example, a snake image file might be labeled "Copperhead A 1.jpg" which would indicate that this is a venomous copperhead class. The "A" denotes that it is used in a training set and the "1" means that it is the first image in the file.

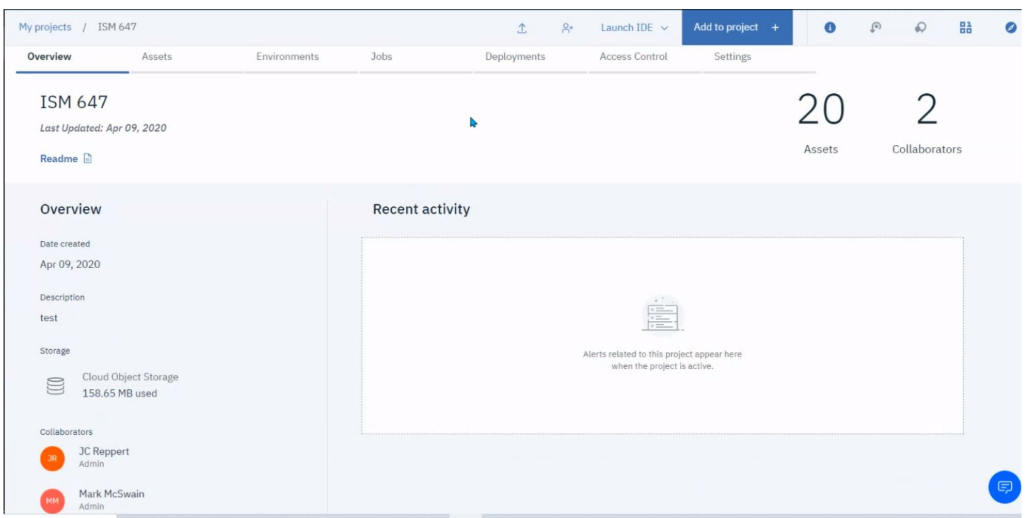
B. **Researching and Acquiring Images:** For the "Snakes" project it was difficult to find databases of images related to snakes indigenous to North Carolina, however such databases could be handled easily with Watson's ability to manage large zip files of images.

C. **Setting up the Project:** Start a new project with an Associated Visual Recognition Service and create a new Visual Recognition Model. Edit the name to reflect its purpose.

---

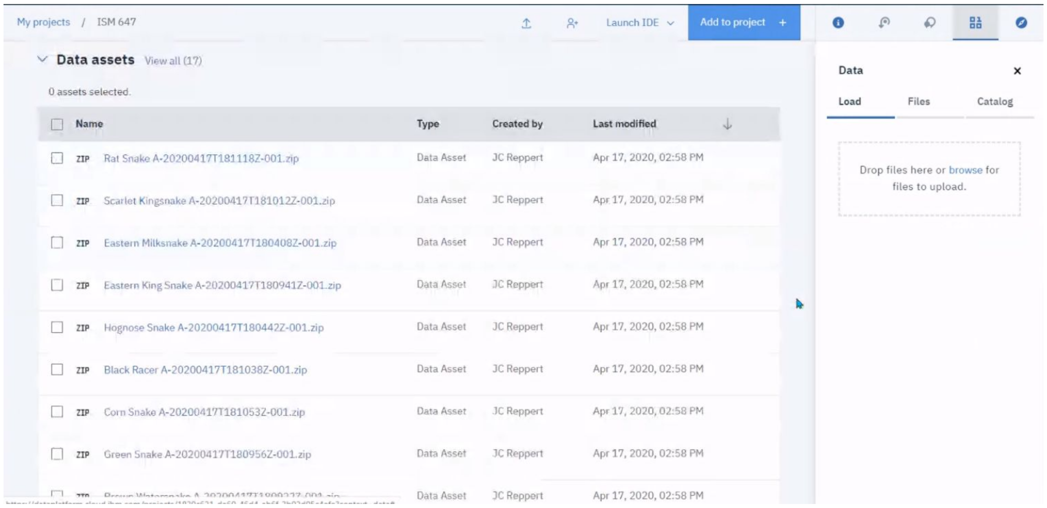
[5] "Amphibians and Reptiles of North Carolina", http://herpsofnc.org/snakes/, copyright 2020

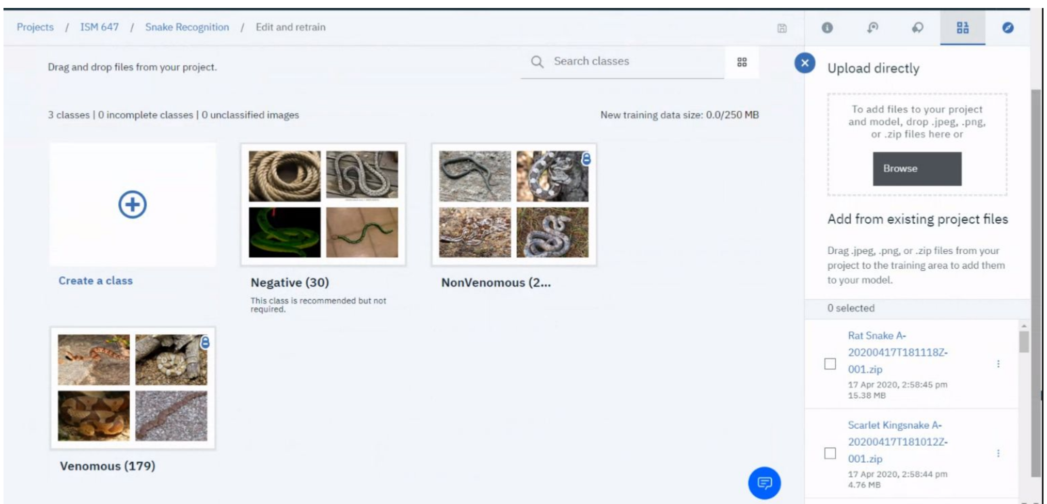


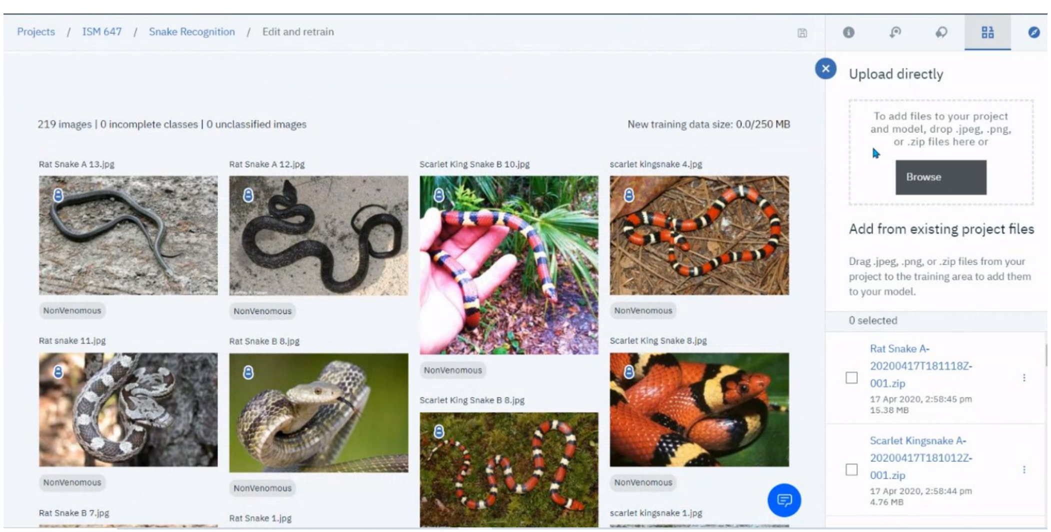


D. **Loading the Training Set:** Watson allows the user to drag and drop zip files of images that are grouped into classes. Multiple groups can be created and files are easily loaded using drag and drop into the space indicated. Once the files are loaded, select each zip file and add them to the model.

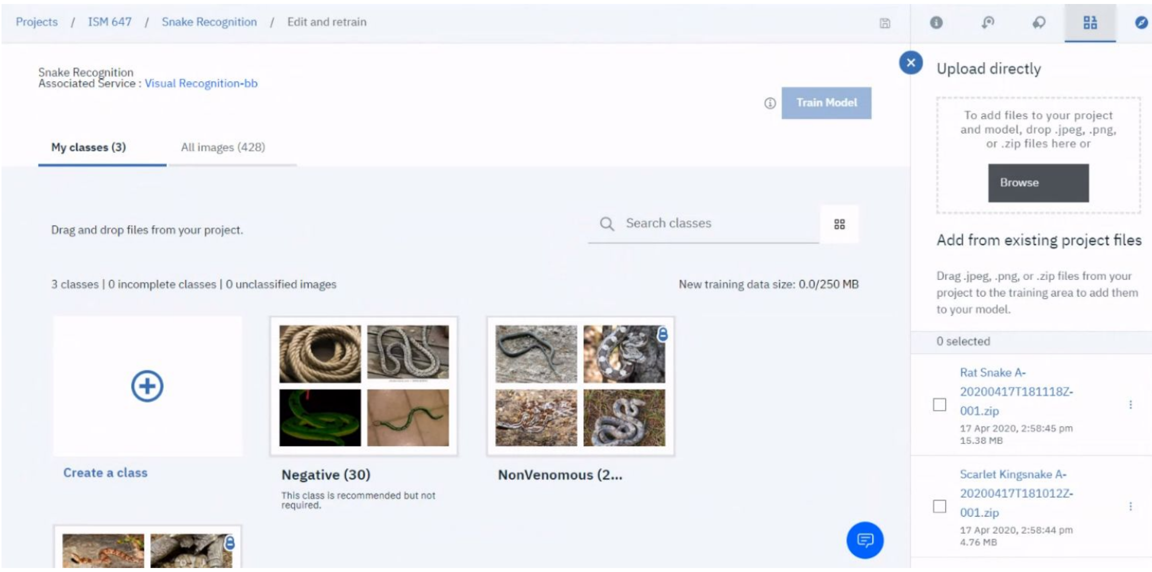
Examples of initial loading of images and creating tagged classes:



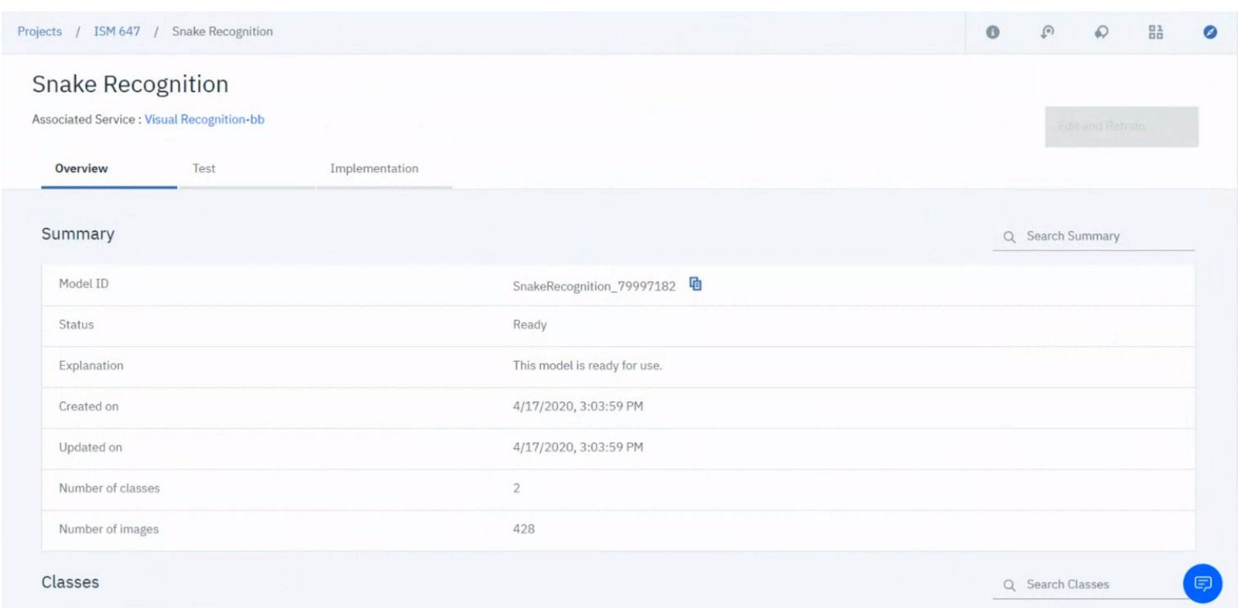




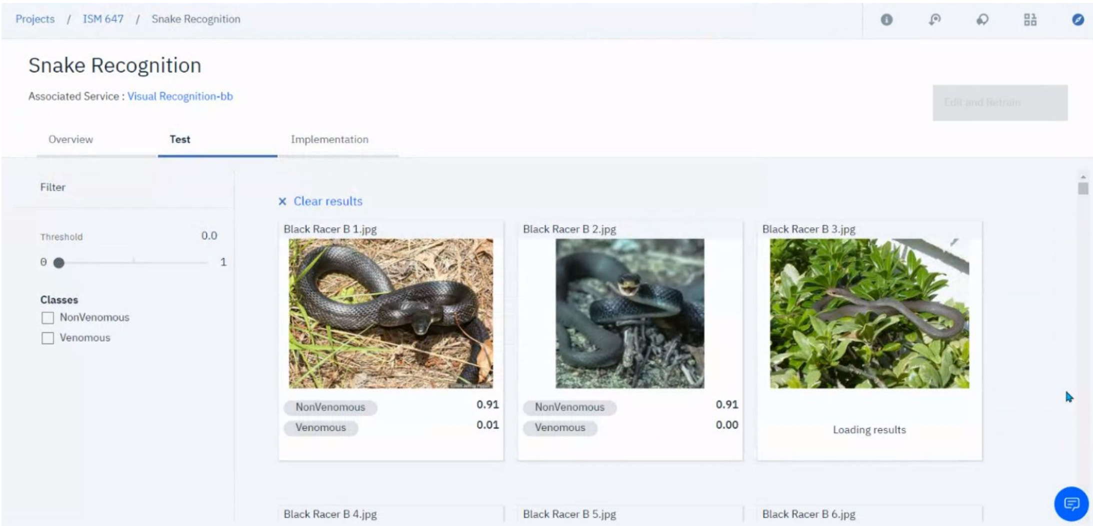
E. **Training the Model:** Verify that all of the images were added. Select a class to view all of the images within that class. When you are satisfied with the classes, click on the “Train the Model” icon in the upper right of the application. Once finished, view the overview tab to inspect the list of classes created.



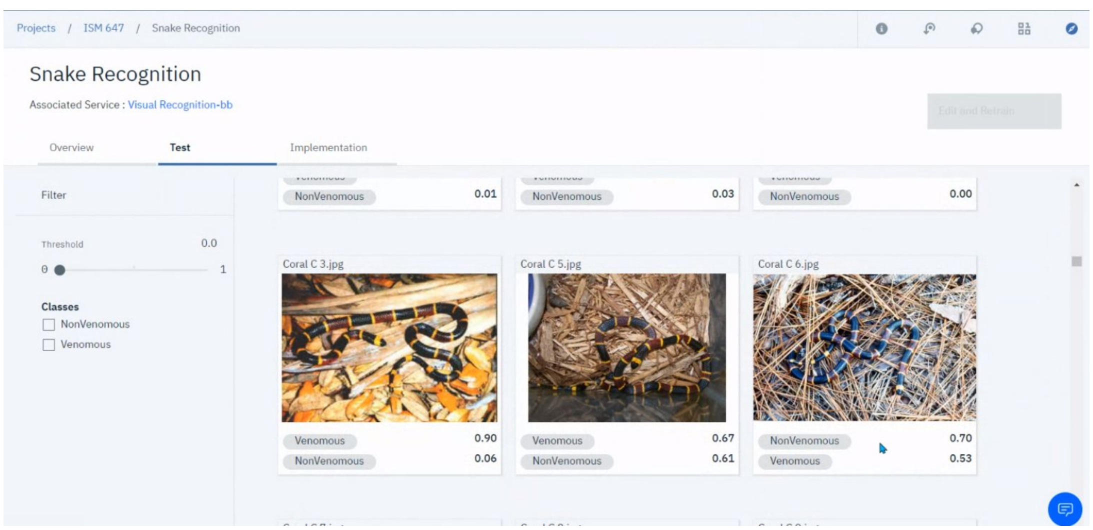
F. **Loading the Test Set:** On the Test tab, you can add images to analyze. Just like the training set, you are able to drag and drop zip files of test images into the application. Results show the matching classes and a confidence score for each image. You can clear results and drag in new images for further testing. Filters can be used for each class.



Correctly identifying non-venomous snakes:

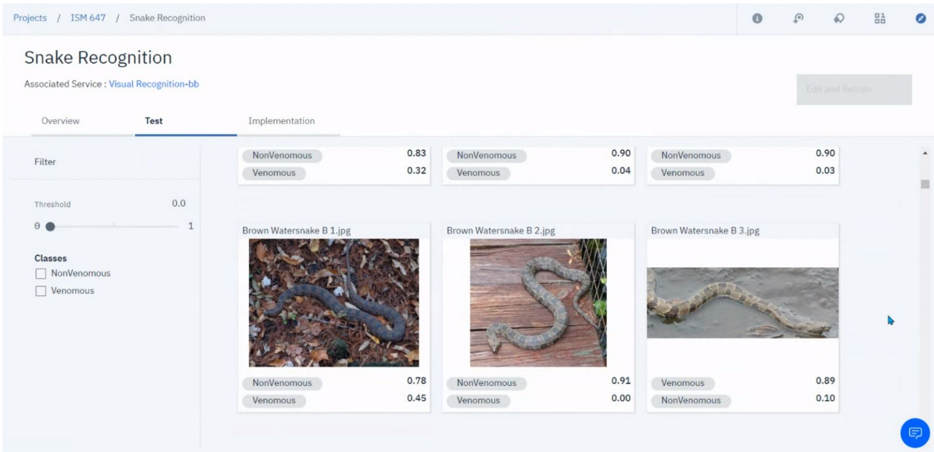


Example of Watson's performance with Coral snakes:

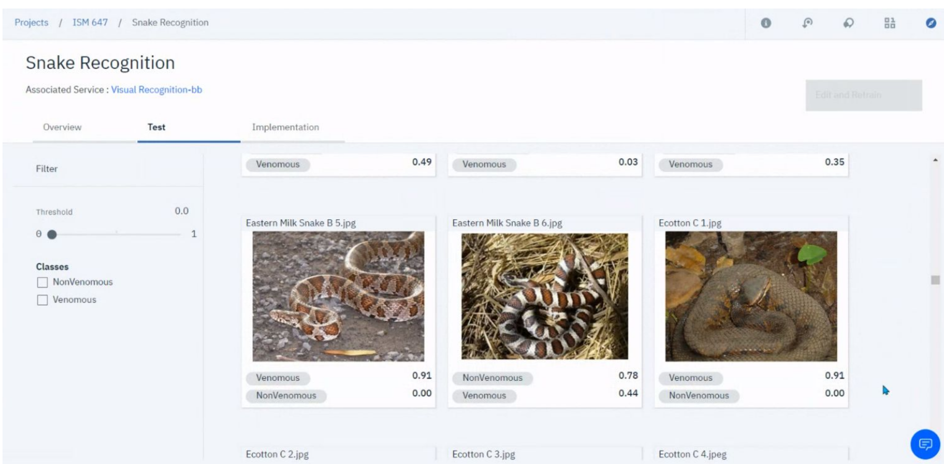


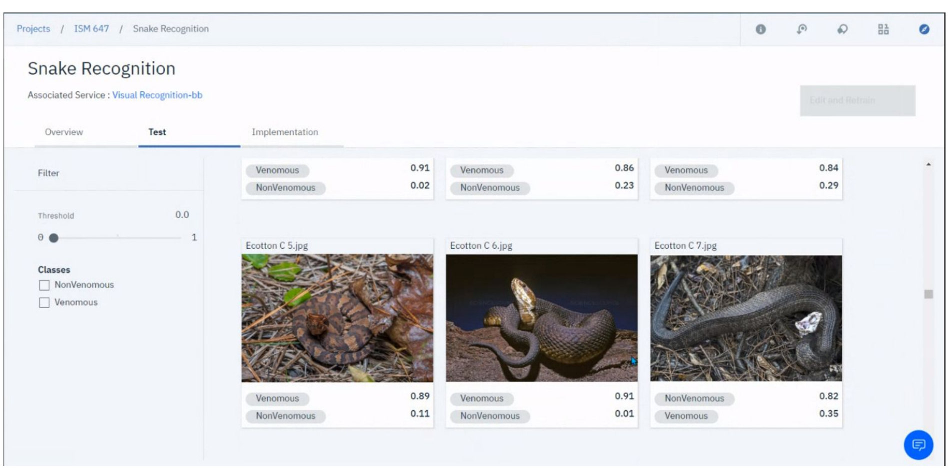
G. **Loading the Validation Set if applicable:** While a "validation set" isn't strictly required, we used an additional set of data loaded as a second test in order to further experiment with the model. Both applications appeared to have the option to retrain the model as many times as the user wished as long as memory requirements were not exceeded.

An example of a misclassification of a non-venomous snake:

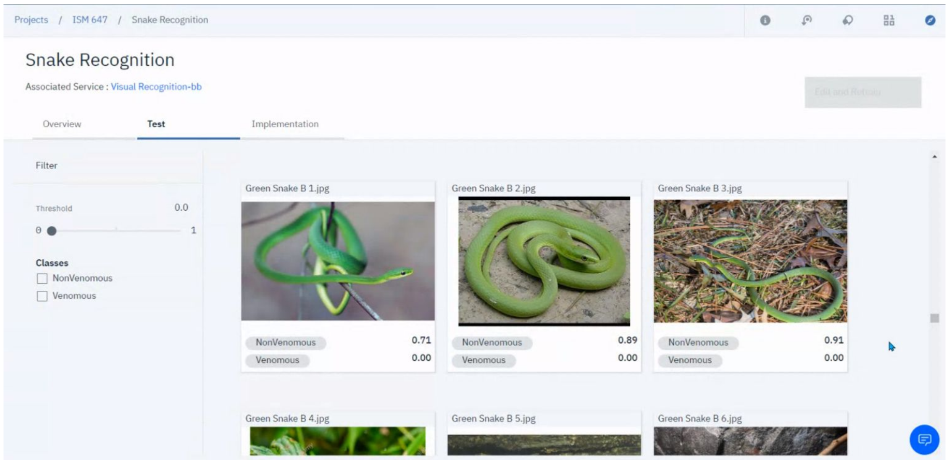


A couple of other misclassifications of non-venomous snakes:

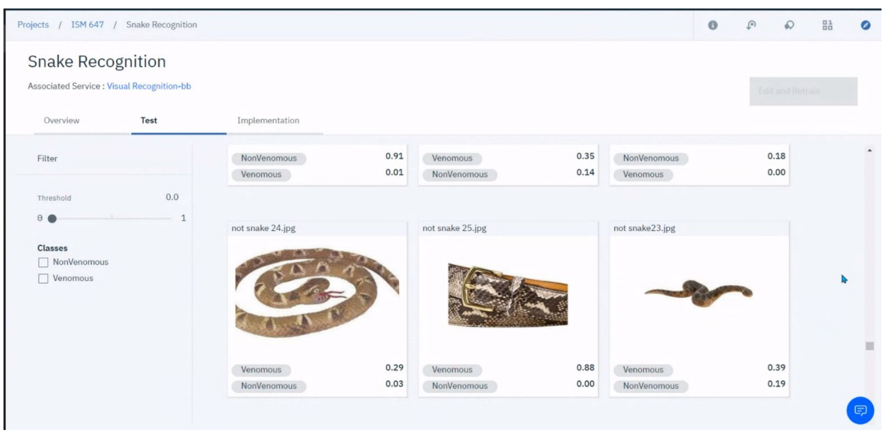


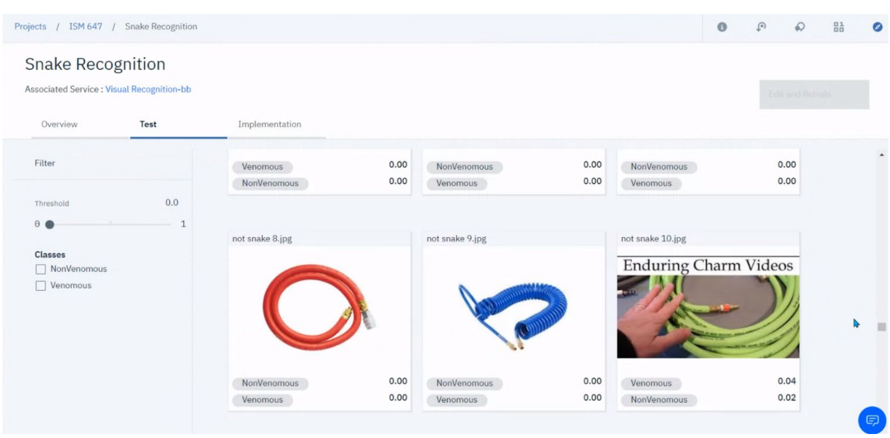


Several correct classifications:



Watson’s responses for non-snake items were different from Azure’s. Azure simply classified as not snake where Watson would still try to classify the item as venomous/non-venomous even though this was an additional category.

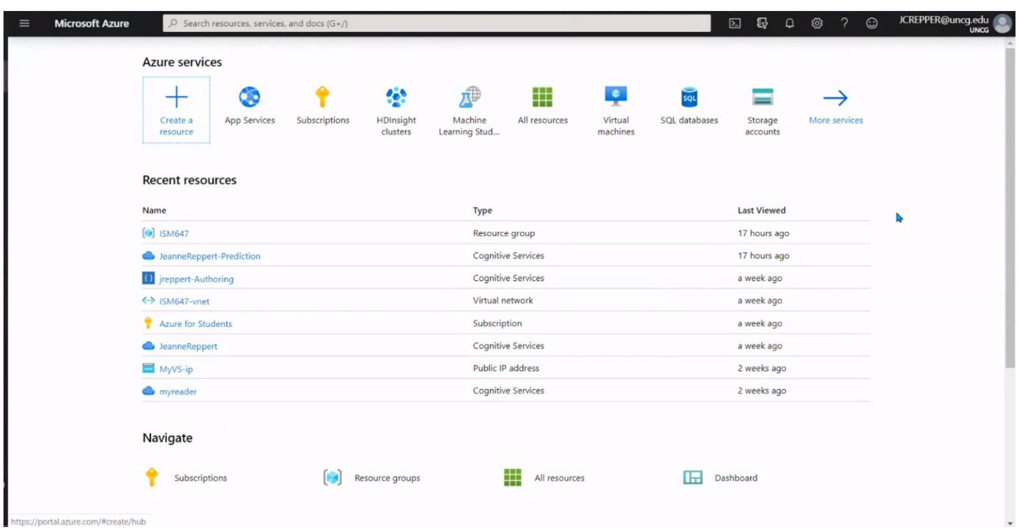


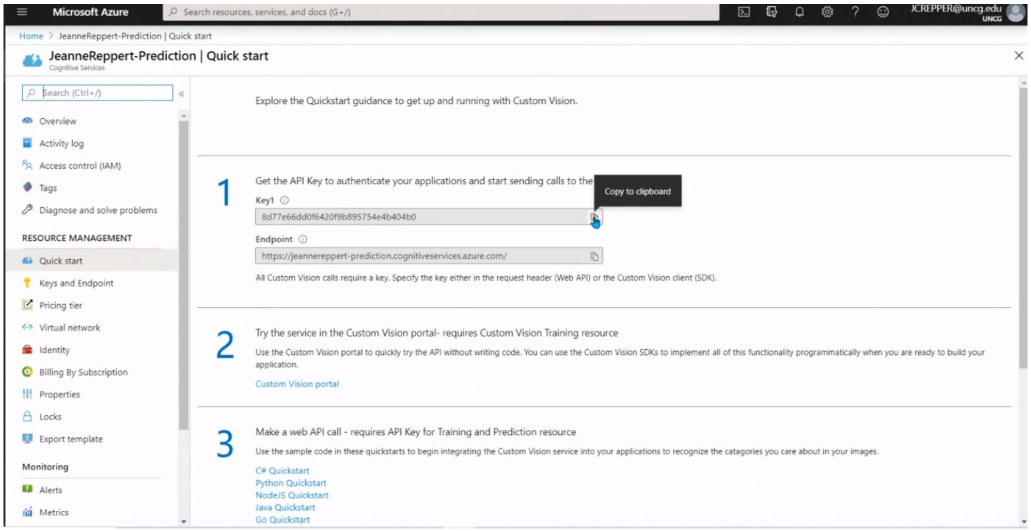


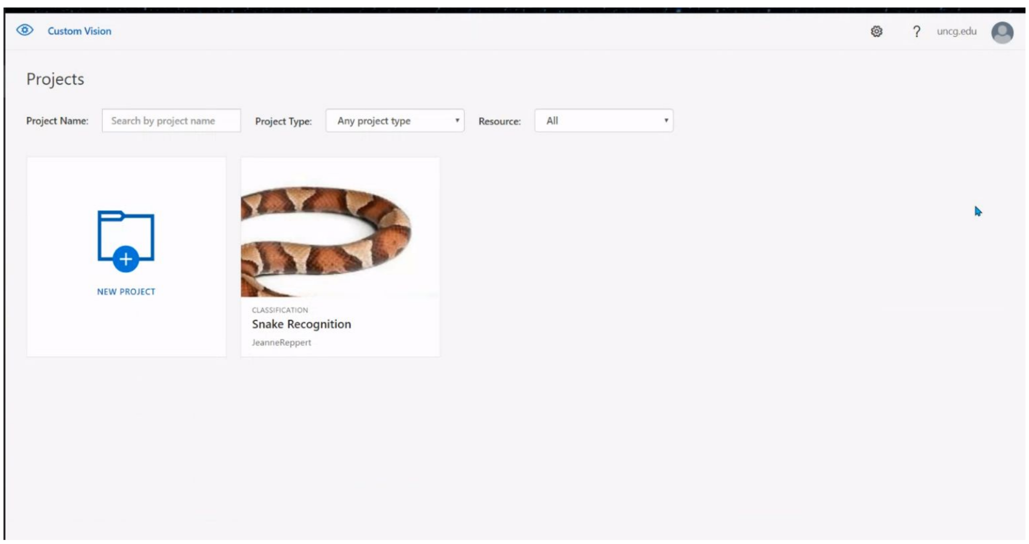
H. **Deploying the Application:** The project Screen contains an implementation tab from which snippets of code can be copied and pasted into the app’s final location for use.

Microsoft Azure’s Custom Vision

A. **Defining the Problem:** The method used to collect data for the Watson project is best suited to Azure’s single file loading method. Given the student user account access, limited loading options require individual image files to be loaded.
B. **Researching and Acquiring Images:** We used the same data set as was collected for the Watson project so no additional steps were required: For the “Snakes” project it was difficult to find databases of images related to snakes indigenous to North Carolina, however such databases could be handled easily with Watson’s ability to manage large zip files of images.
C. **Setting up the Project:** Start a new project from the Custom Vision site. Once there select the “New Project " option to create the new project. Completing the new item form is straightforward with the exception of three fields.
    a. **Project Type:** Here is where you have the option to tell the model whether this is a classification model (recognizes and compares one image to another) or an object detection model where one or more elements of an image are detected based on a reference set of images.
    b. **Classification Type:** Here Multilabel (Multiple tags per image) or Multiclass (Single tag per image) can be selected. Given our relatively small set of images, we chose Multiclass as we only asked Azure to identify whether the snake is venomous or non-venomous.
    c. **Domains:** This allows you to select a larger category to describe the kind of image files contained in the model.

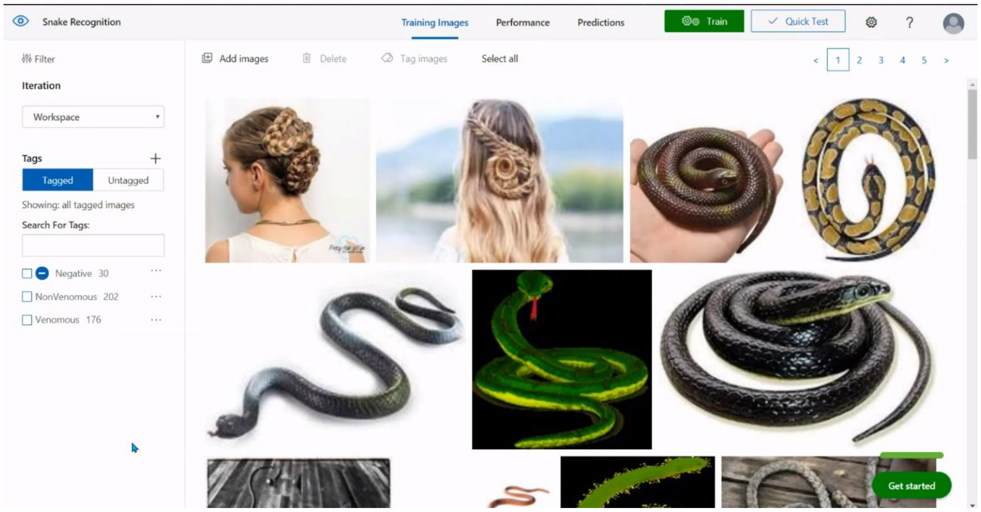




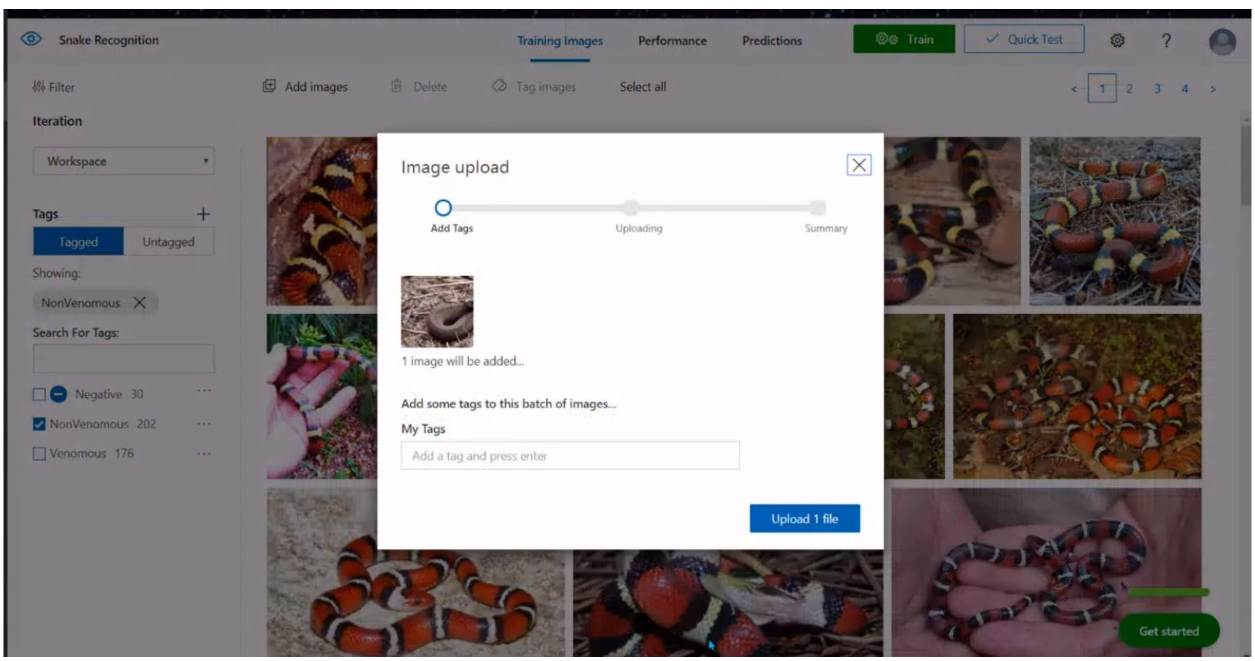


D. **Loading the Training Set:** Click the "Add Images" button to begin the process of loading training images. Unlike Watson, images must be loaded in matches of single images and for that reason, this process takes considerably longer to populate the training set. Images should be grouped according to type so that they may be tagged as one group. Multiple tags may be selected.

Note that in Azure, the names of photos are not included and the images are not tagged unless the user hovers over the image. This made processing results very difficult.

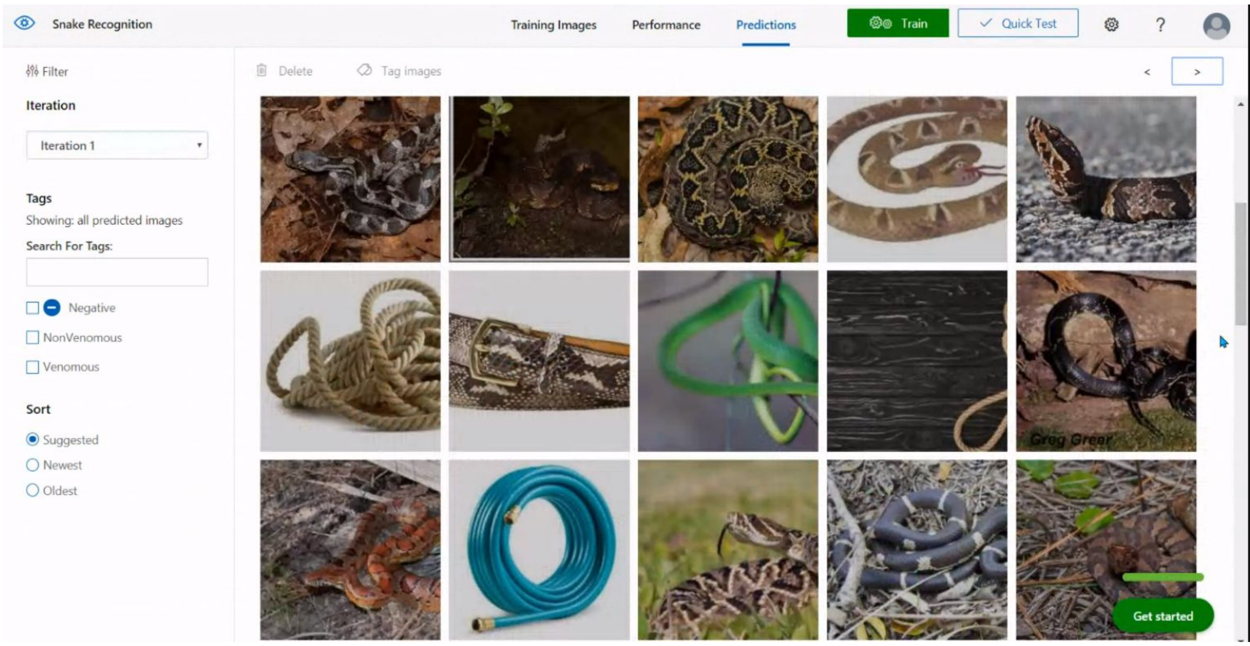


Images were tagged as they were added to the set.

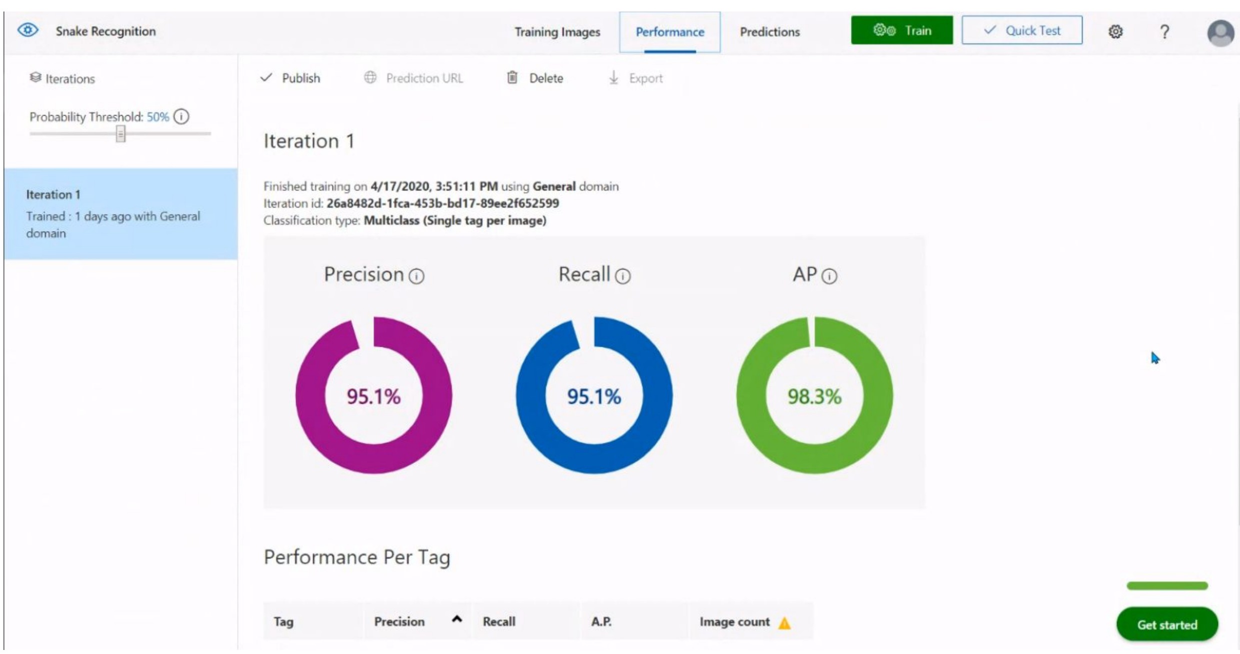


E. **Train the Model:** Once the images have been loaded and tagged, training the model is merely a matter of selecting the “Train” option next to the “Quick Test” option. Once the model has been trained, Azure returns performance metrics on how well Azure was able to classify the images in the training set. High scores are desirable indicators for both measures.

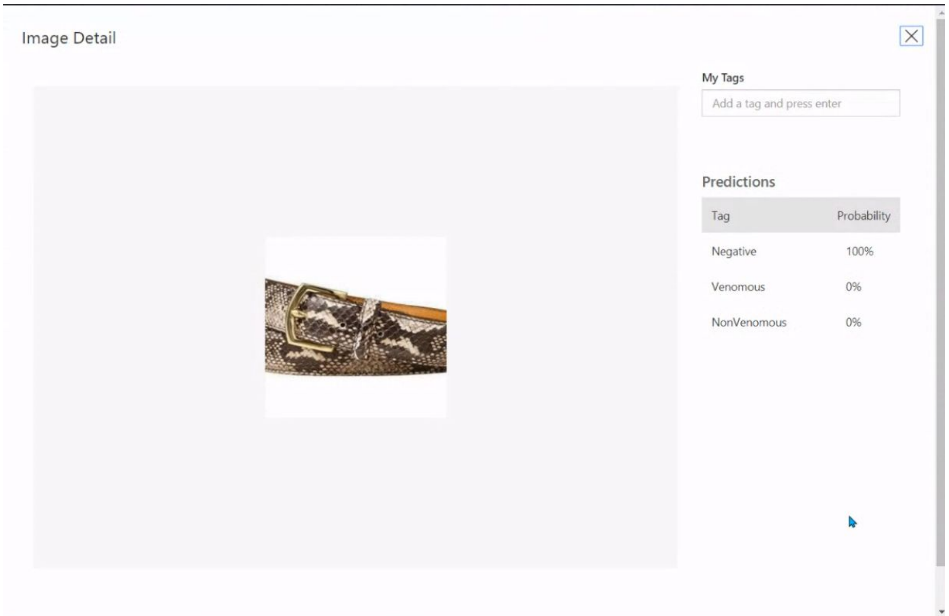
a. **Precision:** Measures how many selected images are relevant to each tag
b. **Recall:** Measures how many relevant images were selected for each tag.

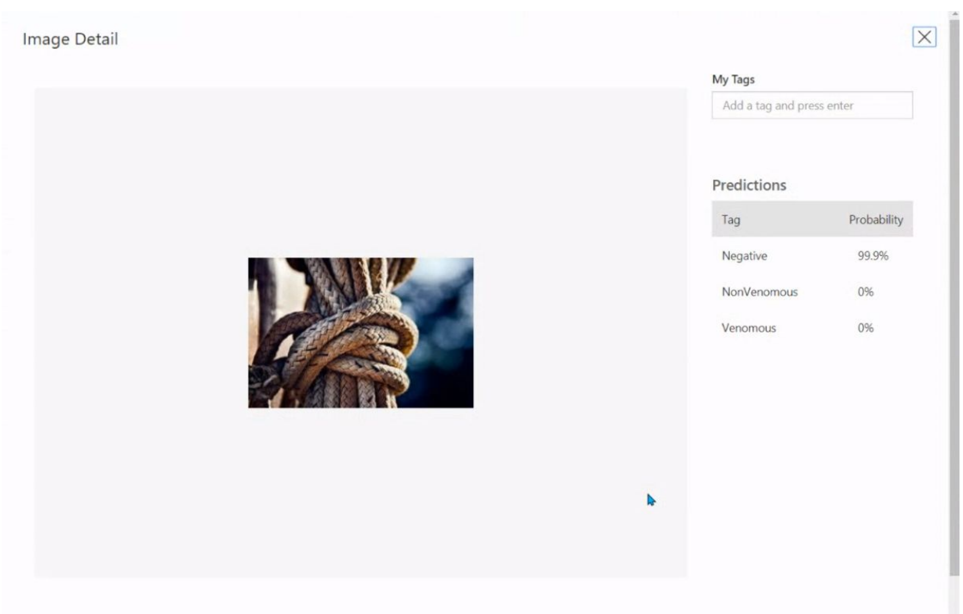


Initial quantitative measures generated by the application before adding test set of images.

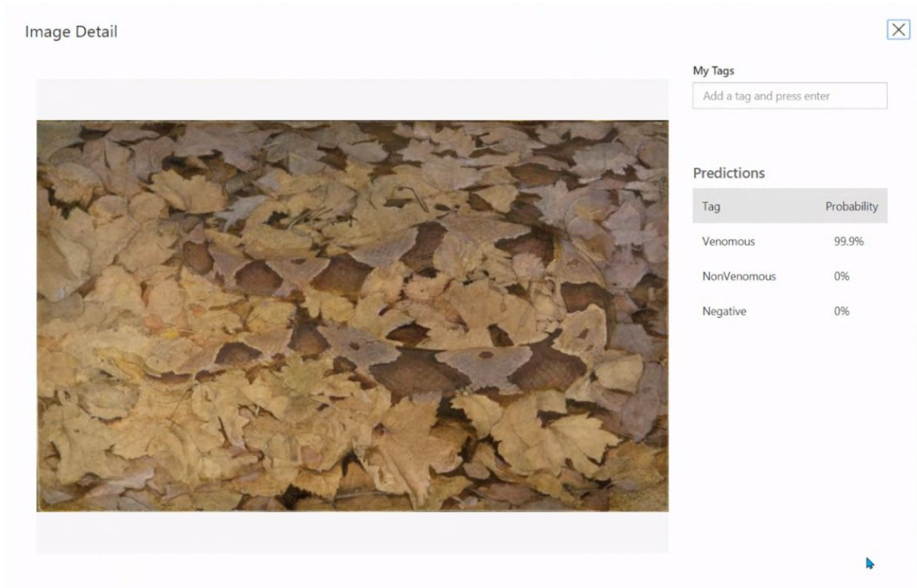


F. **Loading the Test Set:** Unlike Watson, the method of testing the model does not easily lend itself to batch testing. Instead it relies on the use of a "Quick Test" option that allows the developer to load or test one image at a time against the training set. Rather than evaluating a batch of test images at one time, Azure shows metrics for each test image, one at a time. Notably, unlike Watson, Azure would identify non-snake items as negative for snake rather than identifying as venomous or non-venomous.

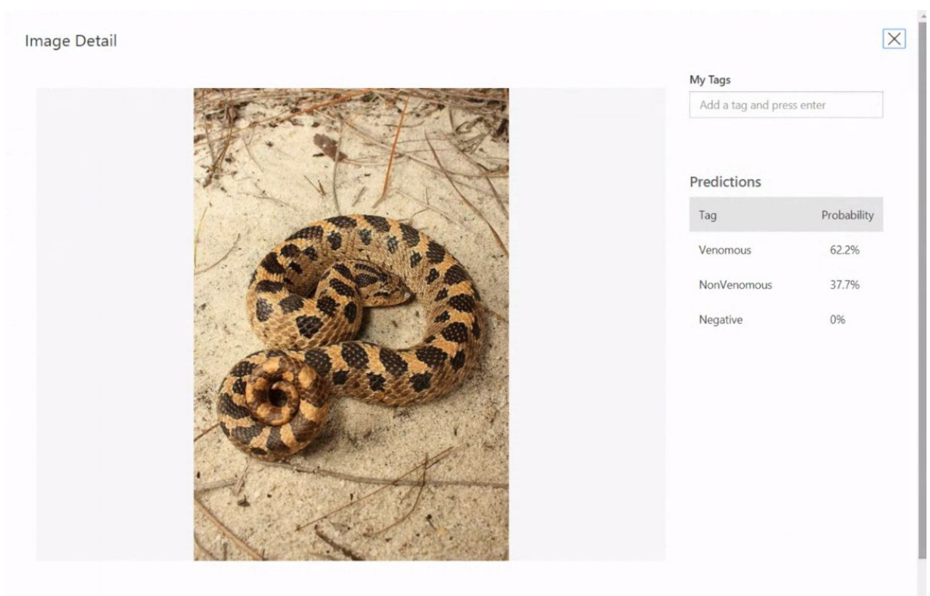




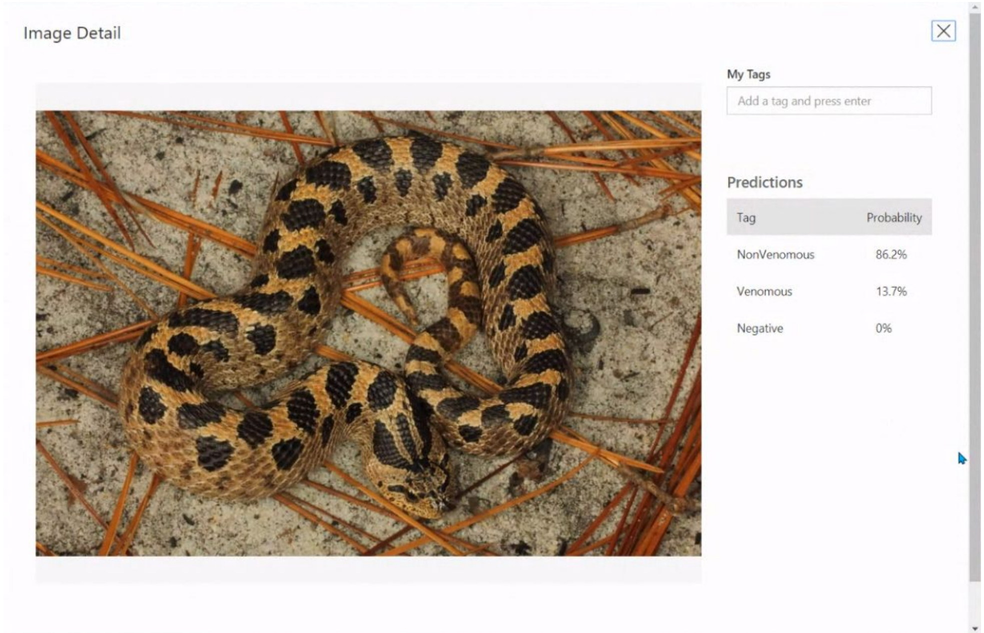
Azure’s ability to identify the venomous copperhead snake, even when camouflaged in leaves was impressive.



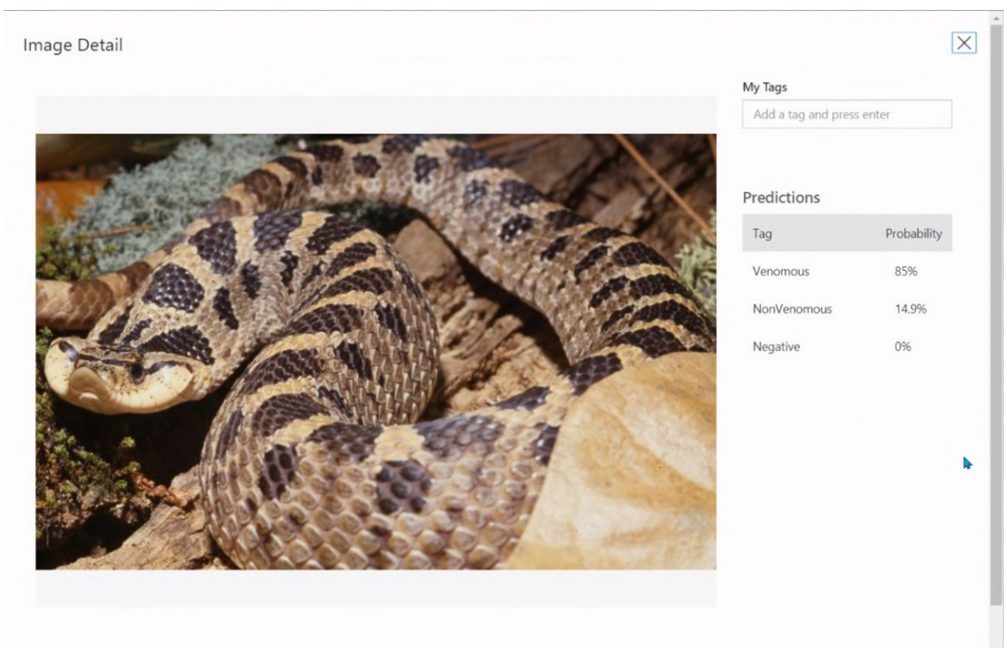
An example of a misclassification (false positive) in Azure:



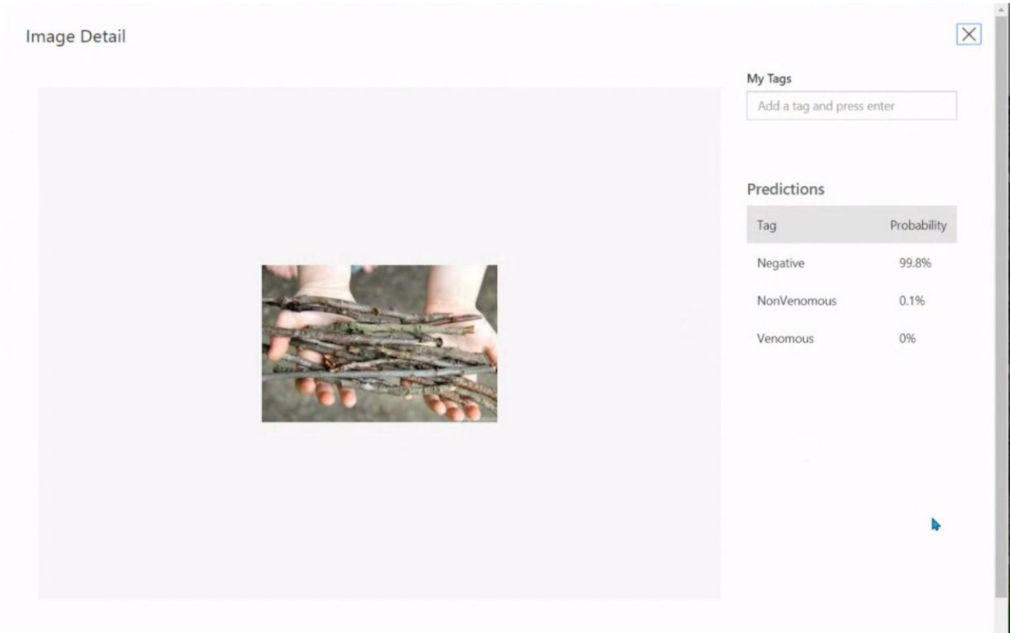
Azure correctly identifying a hognose snake as non-venomous (one of the more difficult snakes to identify.)



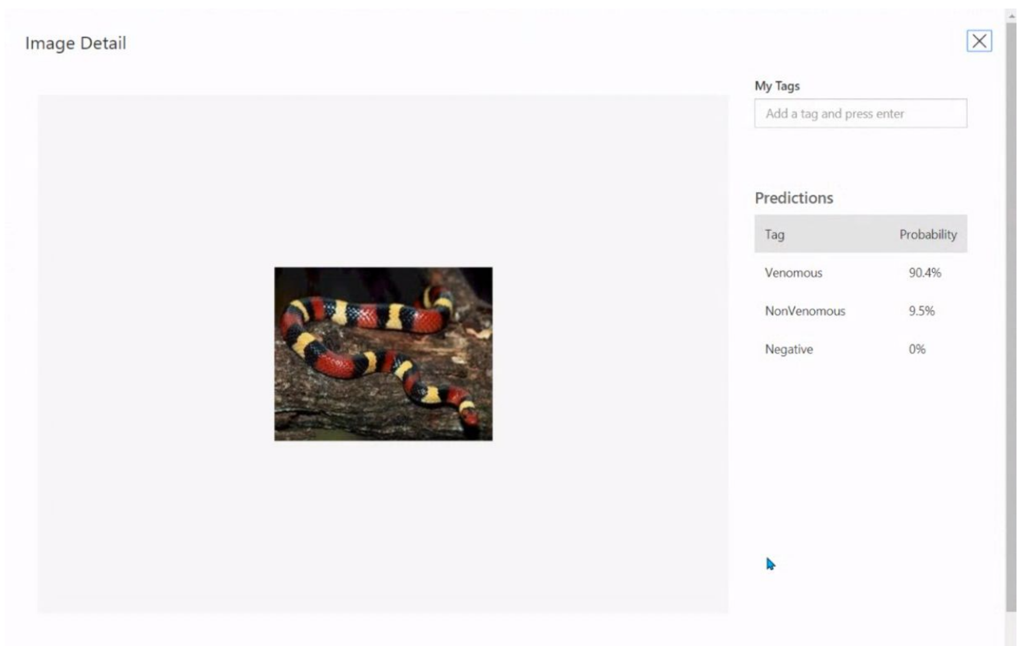
However, Azure did not identify the hognose snake as non-venomous 100% of the time.



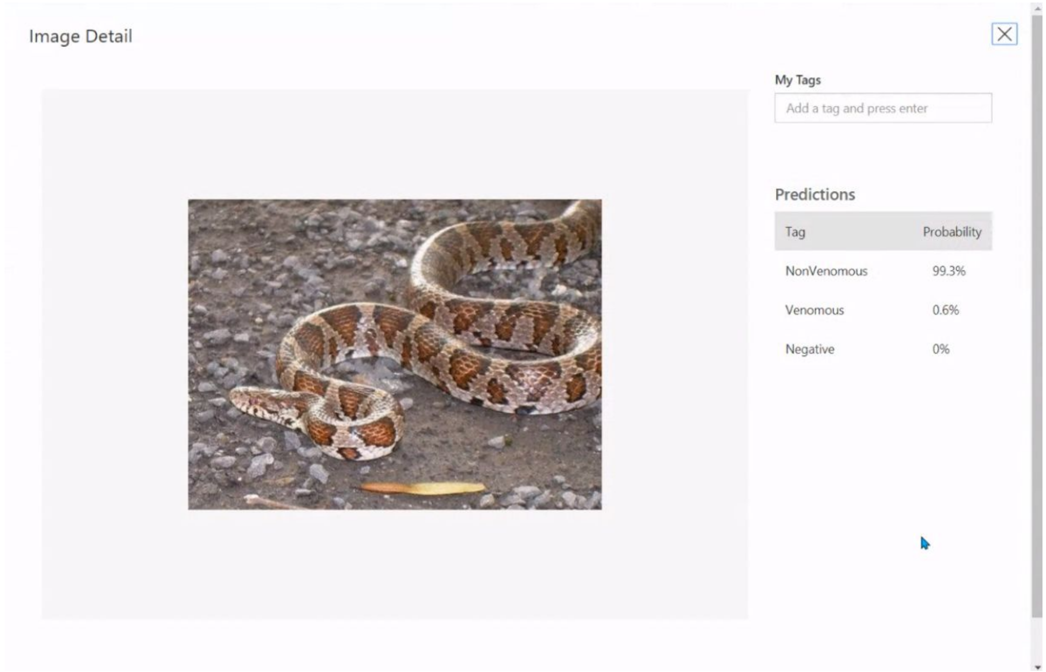
Azure did not mistake these sticks for a non-venomous snake as Watson did:



Azure correctly identifying the venomous coral snake (another easily confused snake):



Azure correctly identifying the non-venomous milk snake (often confused with the venomous copperhead.)



G. **Loading the Validation Set if Applicable:** There is no validation set per se. More images can be loaded and the model re-trained, but Azure only allows for a training set and test compared to that training set.
H. **Deploying the Application:** Deploying the application requires a bit of coding through any number of apps, the most talked about of which is Visual Studio. For that reason, we stopped short of programming that final phase.

Summary

While Azure yielded better results given the set of images that we had to work with, our favorite of the two image classification engines was the Watson Studio. Jeanne was at the helm for both Watson and Azure, but Watson allowed Mark to view, use and test, where Azure was not so straight forward with a multi-user setup. The test results were also better with Watson, because it allowed for the review of several tiles at once with the image name clearly labeled on each tile. It made the job of understanding whether or not Watson correctly classified each of the images. We believe that with a much larger training set, Watson would close the very narrow gap on Azure with respect to accuracy of image classification. Thus overall, Watson is likely the preferred tool, particularly if the images sets were expanded since the manipulation of the training sets and test sets were much easier with this application.

Overall, this activity was informative and as application users, we were impressed with how well both applications were able to perform in terms of accurately identifying venomous and non-venomous snakes. It was a useful demonstration of the effectiveness of cognitive applications and the usefulness of their applications.