AI Programming in Snap!

Abstract (Workshop)

We have added new blocks to Snap! for speech synthesis, speech recognition, image recognition, and machine learning. These run in a Chrome browser without any extensions or installation. An interactive guide has been developed. Ten sample programs are available. During the workshop participants will explore these new blocks. They will have the opportunity to modify the sample programs. Familiarity with Snap! or Scratch will be very helpful but not absolutely necessary. While this software can run on tablets and smartphones we recommend you bring a laptop.

Keywords

Visual programming; machine learning; block languages; Snap!, AI services; Cloud services; speech synthesis; speech recognition; image recognition;

Speech Synthesis and Recognition

New Snap! blocks will be explored that build upon the browser’s Web Speech API (Mozilla 2018). The most complex of these blocks enables control over the rate, pitch, volume, voice and language of the generated speech.

The Web Speech API also supports speech recognition. New Snap! blocks enable programs to respond to speech after it has been recognised. The most complex version enables one to receive partial results as speaking is still ongoing. It also supports the specification of the language being recognised. One can ask for several alternative interpretations of what was spoken and the confidence the system has that they are correct.

Image Recognition

There is no web API standard for image recognition. However, many companies provide web-based vision cloud services. The new Snap! blocks can send image recognition queries to Google, IBM, or Microsoft. While these are commercial services all the providers have free quotas for limited use. In addition to being able to obtain image labels and their confidence scores the most advanced Snap! blocks provide access to other information these services provide such as locations of faces, image properties, and more.

Machine Learning

Snap! blocks could be created to use machine learning web services. We, instead, use the deeplearn.js library (Deeplearn.js 2018) to perform machine learning locally in the browser. This library is able to use the GPU (graphics processing unit) to run very fast. Currently the new Snap! blocks are limited to training to classify images. Programs using these blocks can interleave training (either from the camera or from sprite costumes) and classification of new images. A sample program using these blocks implements a Rock Paper Scissors game where players make moves by configuring their hand in front of the camera.

Conclusion

Creating AI programs in Snap! can be fun. One can learn about perception and machine learning. And one can be creative in making programs that do very impressive and interesting things.

References

Deeplearn.js (2018) https://deeplearnjs.org/.

Mozilla (2018) Speech Synthesis API. https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API