**Task c** – Add image classification capabilities and evaluate it: You will train,  
save and test a CNN model that can detect the contents of some images related to your  
chatbot topic. The programs for training/saving the model are not parts of the chatbot,  
but should be submitted separately. In your main chatbot program, you will import the  
pre-trained model and use it for serving user’s query on image classification. If the user  
inputs a question such as “What is in this image?”, the chatbot should show a file  
selection dialog box (or asks for a URL) to locate the image, then should invoke the pre-  
trained CNN to provide an answer about its content. You decide yourself which CNN  
architecture/parameters and training data set to use. Different source of images, e.g.,  
files, Python libraries or camera input can be used for training/testing. You may also  
decide to trigger this feature in other ways. There are web sites that serve as a  
repository for many good data sets such as Kaggle.com or the UCI Machine Learning  
Repository. Do not worry too much if it doesn’t fit that well with the topic of chatbot. A  
weak connection between the image topics and your chatbot topic is acceptable. Once  
trained, you should save the model (e.g., as a h5 file) and include in your submission.  
Report and explain the model’s performance using a measurement of your choice. If the  
classification does not perform well discuss the causes and/or limitations.

Your aim is task c is to train a CNN model to identify the content(s) of an image, and integrate this model into your chatbot. Therefore, you will need to develop two programs: An independent program for training and saving your model, and the other one being your chatbot programmed for loading and using the saved model as a part of its conversations with the user.

The way you integrate the classifier with the chatbot is up to you. As a simple approach, the user may write a command (e.g., “what is this image?”), then the user is asked to locate a file, then the chatbot answers with a text (e.g., “this is a cat”). Sample conversation logs are provided.

You first need to identify or develop a dataset of images related to your own topic. If you struggle to find a good match, a partial relevance to your topic is also acceptable. Identify a limited number of classes (2 or more) to be the output of your classifier. For example, dogs and cats for a picture dataset, if your topic is about animals. Kaggle.com or the UCI ML Repository could be used to start searching. The dataset should have:

* A good number of images: there is no golden number – it depends on the images and your target. You need to try and see the accuracy.
* Images are better to be clear and sharp enough, intermediate size and resolution, with not much background items.
* The target classes to be well identifiable, e.g., do not try to identify kitties from cats!

To train your model;

* Find out how to load your own dataset, instead of the FASHION\_MNIST dataset.
* Adapt the code to train a CNN model from the loaded dataset. You may need to adapt the network’s parameters and layers, to make sure it converges to an acceptable accuracy.
* Once the model is trained, save the trained model as a data file (e.g., .h5). There are different possible ways to achieve that. It may be easiest to save the whole model, and use Keras, not plain TensorFlow.
* Edit your chatbot program: Insert the code for restoring the saved network into your chatbot python script, and integrate it into the remainder of your chatbot via the AIML file and/or any other suitable changes of code.

If you choose to do this task as one of your 3 coursework tasks, you will need to include in your zipped submission:

* A program that trains/saves the model (not to be a part of your chatbot program)
* The saved model data file (.h5 or similar format)
* Sample image files for testing (up to 10 images - do NOT include your whole training or testing datasets)