**Bit Classification**

* Nice that you included computer specs
* Great use of augmentation
* Nice reference to interactive workflow
* It would be nice to print true labels and prediction labels together with the image along with the accuracy. Example: <https://www.mathworks.com/help/deeplearning/ref/alexnet.html>
* The same as above bullet for the cracks
* Overall, great script
* Jon: any discussion on accuracy?
* Jon: any validation data?
* Jon: maybe some pictures using deep network designer

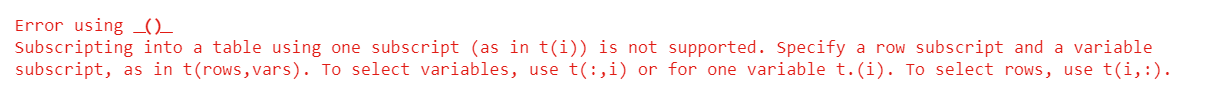
**Classification Overview**

* The digits dataset is called MNIST data set, maybe you can mention that.
* If you can combine prediction label with the image, that would be great, see above comment
* You can add a few words about different links you gave above line 53 (the same in the following sections)
* Nice further reading material
* Jon: this hangs up on me
* Throw a disclaimer if the computation time is long

**Draw\_Classification**

* Nice use of built-in functions and mention of classification learner app
* There wasn’t a confusion matrix under supervised learning classifier, was there supposed to be? If yes, could you please add the code and the matrix?
* Mention classification learner app for tree models and how they can be trained in parallel
* Include general tree info: <https://www.mathworks.com/help/stats/decision-trees.html>
* Line 78, error Warning: Solver failed to converge. Try standardizing the predictor data to avoid scaling issues. Please resolve that.
* Jon: same warning

**kNN Classifier**

* Instead of calling it predict method, I’d say “predict function”
* Nice mention of Live Tasks
* Jon: I get an error: (seems like it might have been resolved) 

**kNN Regression**

* Nice use of controls in Live Script
* Jon: Nice!

**Lithium-Ion Batteries**

* Good comparison of different techniques
* Jon: do you need all the commented out code?- clean the unnecessary parts

**LSTM Networks**

* Mention interactive ways of building and training LSTMs with Deep Network Designer
* Jon: a lot of big code chunks, maybe unavoidable though

**Neural Network Regressor**

* Nice mention of deep network designer and training courses
* Jon: I feel like the MSE should be more with the tanh activation...?
* You can add clc clear all at the beginning

**Soil Classification**

* Nice use of Deep Network Designer
* It would be nice to have labels right under the image
* Nice that you indicated computation times
* Jon: Would you want to show everything in deep network designer
* You can also mention that you can export code from deep network designer or export the model to workspace

**Steel Plate Defects**

* Good
* Jon: what is the “play” button for? Maybe explain
* Jon: nice loss plot at the end

**Sean’s hand-tracking and Reinforcement Learning Scripts from Summer Folder**

**Hand\_tracking\_modified**

* Add table of contents to easily navigate between different approaches
* Description before the scripts could be a bit more specific. For example, you can write something like this: “To detect objects or regions of interest in images and videos, there are several approaches relying on pure image processing and computer vision or data-driven techniques such as deep learning. In this script, we will explore both approaches. Objects can be identified in images and they can be tracked over a sequence of images (i.e. live or recorded videos). In this script, we will explore how to detect and track objects using color-based segmentation by acquiring live video feed from a webcam. Approaches below are derived from MATLAB documentation examples and the original examples are also linked”
* There are superpixels lines at line 8 and 26, are both of them needed? If yes, maybe you can explain why both are needed.
* You may want to indicate which toolboxes and hardware support packages are needed to run these examples. If any previous set up is needed, explain it before the code.
* If you didn’t modify anything in the documentation example, you don’t need to copy paste the entire code, you can hyperlink the documentation you are using to the text, in the code section, you can use openExample('vision/FaceTrackingUsingKLTExample') or whatever the the command is to open documentation (you can get the correct code by clicking on “Copy Command” in the documentation). Please note that in the latest releases, installing documentation became optional. You can add a line “If documentation is not installed on your computer, follow these steps to download it: <https://www.mathworks.com/help/matlab/matlab_env/install-documentation.html>)
* At Line 171, I got this error “Error using imaq.internal.VideoDeviceInternal  
  winvideo: The device associated with device ID 1 is already in use. A new videoinput object cannot be created for this device while it is in use.Error in imaq.VideoDevice” make sure to add lines to clear video object before these lines.
* Did you write “Method 2” code or is it from documentation? If it is from documentation, indicate which documentation piece that is. If you haven’t modified anything in the code, instead of copying, use “openExample” and give the hyperlink in the text.
* Method 3 – you can give the original example, say that the original example uses recorded video and you modified it to take live video feed from a webcam. You can explain what you modified and tell reader to do the necessary set up before running the code (refer back to my e-mails where I explain what is modified)
* You can mention that CascadeObject Detector is following a data-driven approach and link trained object detectors in MATLAB.
* You can add further reading section and add the links that I shared with you when you were putting together the script such as:
  + Object Detection in MATLAB: <https://www.mathworks.com/help/vision/object-detection.html>
  + Getting Started with Object Detection Using Deep Learning

<https://www.mathworks.com/help/vision/ug/getting-started-with-object-detection-using-deep-learning.html>

* Tracking and Motion Estimation: <https://www.mathworks.com/help/vision/tracking-and-motion-estimation.html>
* To get started with Image Processing: Image Processing Onramp self-paced online training <https://matlabacademy.mathworks.com/details/image-processing-onramp/imageprocessing>

**Reinforcement Learning**

* If you haven’t modified anything in the code, use “OpenExample” instead of copying the code and give the hyperlink of the example in the description text.
* You can add a further reading section and include these:
  + Reinforcement Learning Toolbox Documentation: <https://www.mathworks.com/help/reinforcement-learning/index.html>
  + Reinforcement Learning Onramp (free, self-paced tutorial): <https://matlabacademy.mathworks.com/details/reinforcement-learning-onramp/reinforcementlearning>
  + Reinforcement Learning Tech Talks (short videos): <https://www.mathworks.com/videos/series/reinforcement-learning.html>
  + Process Control with Reinforcement Learning (short video): <https://www.mathworks.com/videos/process-control-with-reinforcement-learning-1610006017506.html>
  + Playing Pong with Deep Reinforcement Learning (example): <https://www.mathworks.com/matlabcentral/fileexchange/87939-playing-pong-with-deep-reinforcement-learning>