Common instructions for stage - 1 for LSM projects under Vivek:  
1. Run the connections.m (generating the network weights), preprocessing.m (generating the input from dataset), RR.m (generating the reservoir response) and TT.m (training and testing for the given dataset) for all 10 digits of the TI digits dataset. Report the training and testing accuracy.  
2. Plot the output from preprocessing.m (plot the rasters for a few input digits to compare them), plot the output from reservoir response (raster of the spikes generated by reservoir neurons), you may also use these plots to debug in case the accuracy seems too low for a 125 neuron reservoir and 10 output classes (based on your literature reading i.e. adjusting the reservoir activity using weights).

Specific stage - I instructions for people working with Low SNR LSM:

1. Listen to the audio signals in the dataset (look up sound command in matlab or figure out a way). Calculate the typical SNR of these signals, look at the signal power when the digit is spoken compared to the signal power when the speaker is silent (noisy background). Plot the audio signals and zoom in on noise levels to observe their amplitude.

2. Add noise to the audio signal to control the SNR (generate a few realistic cases - discernible spoken digits vs total garbage signal). Listen and plot the new audio signals.

3. Focus on understanding the preprocessing.m. Plot the preprocessing signal chain for say 3 channels (low, mid, high frequency): Input -> Filtered Outputs -> Rectified Outputs -> AGC Outputs -> Differ Output -> Decimation Output -> BSA output -> Total preprocessed raster

4. Now you are ready for your stage - 2 activities. How can you improve the performance of the network in the presence of noise by - (a) controlling preprocessing, (b) training in absence of noise, testing in presence of noise, (c) training in presence of noise, testing in presence of noise, (d) something else.

Specific stage - I instructions for people working on pitch insensitive LSM:

1. Generate a dataset of pitch-shifted speakers. This can be done in a few ways: (1) by artificially changing the pitch of a few speakers from the TI-digits dataset, the uploaded dataset has 5 female speakers. (2) Or it can be done by recording your own samples! Develop a recording method to get multiple utterance of 1s samples of different spoken digits from your friends (with different pitches). (3) Or taking samples from the ti46 full dataset uploaded in the LSM materials folder. It has different male and female speakers.

2. Observe the generated preprocessed rasters from the pitch shifted instances of the same spoken content using preprocessing.m. Report the differences in the rasters. Can you distinguish different digits from their rasters? Can the LSM network distinguish? Run the preprocessing -> RR -> TT on the new dataset and report its accuracy. Ensure that the dataset you build/choose has the same structure as TI46-IF.mat (the original dataset of TIdigits), i.e. the consecutive samples cycle through the output classes incrementally (0 to 9, then 0 to 9 and so on).

3. You are ready now to deal with stage - 2 activities. Can the LSM train on one pitch and test well on another pitch? Can the LSM train on mixed pitches and test well on any of these pitches? Does scaling the network help with storing more pitch information and train/test better? Is there a way for the network to focus on the preprocessed raster shape rather than its frequency offset?

Plan to conduct your stage-2 study in a systematic manner, the same way we are developing a systematic methodology for stage -1.