

# MATLAB Code Breakdown

Jelena's Code	Matt's Code	VLFeat
Classify.m	K_Means_Clustering.m	phow_caltech_18.m
StandardizeImage.m	HMM_Train.m	
getImageDescriptor.m	Find_Features.m	
makeVideo.m	Find_Training_Hists.m	
mask.m		
numRucksandMauls.m		
numScrums.m		
HSV_Threshold.m		
maskingImage.m		
videoIn.m		
makeVideo.m		
Offscreen_video_eg.m		
text_image.m		

HSV\_Threshold.m

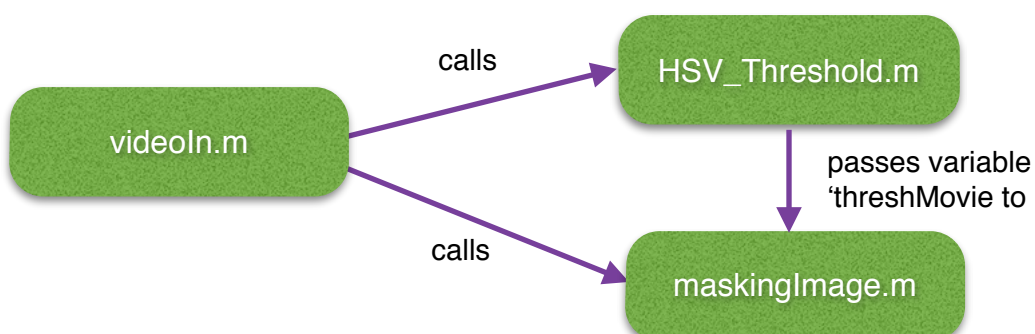
- Removes green from and image.

maskingImage.m

- Removes spectators from the image.

videoIn.m

- Reads in video.
- Counts number of frames in video.
- Calculates height and width of video.
- Uses MatLab function 'struct' - which creates or converts information into a structure array - to create a struct array 'im1' with fields 'cdata' and 'colormap'.
- Reads every frame one at a time and converts the frame into HSV.
- Calls HSV\_Threshold to remove the green.
- Calls maskingImage to remove the spectators.



phow\_caltech\_18.m

- Outputs 5 files
  - tiny-vocab.mat - histogram
  - tiny-hists.mat - histogram
  - tiny-model.mat - info about things learned.
  - tiny-result.mat - results
- If effect - teaches the machine about images and tests the accuracy, giving a confusion matrix of accuracy at the end.

standardizeImage.m

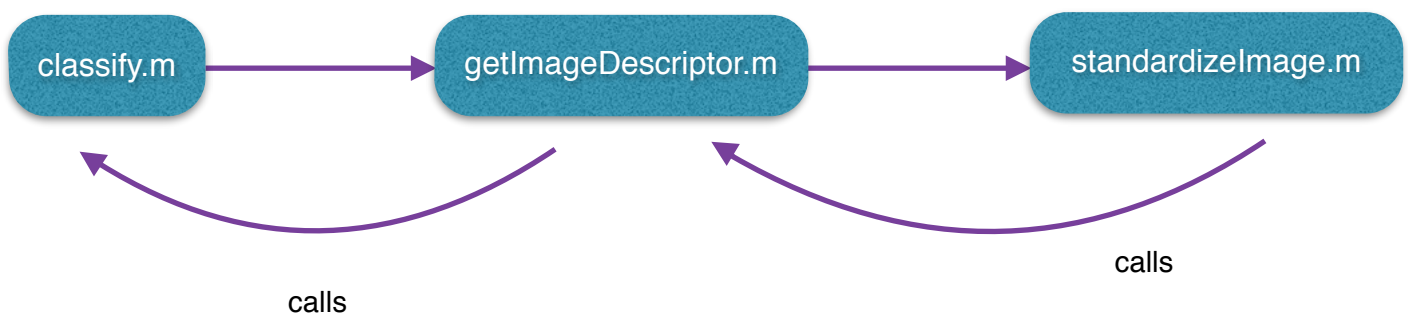
- resizes the image by converting the image into single precision.
- imresize(im, [480 NaN]) - resizes the image to have 480 rows.

getImageDescriptor.

- Accepts a model and an image as arguments.
- Calls standardizeImage, calculates the height and the width of the image.
- Calculates the size of the model.vocab.
- Extracts the 'phow' features from the image.
- Quantises the appearance of the image.

classify.m

- Calculates whether or not an image is one of the set the machine has been taught.
- Accepts the model of what the machine has learned and an image to be compared with the model.



numRucksandMauls.m & numScrums.m

- Parses an XML file of a match containing tagged information on events looking for Rucks, Mauls, Scrums and Lineouts.
- Finds the start and end times of these events.
- Removes the events from the original video, by taking individual frames and saving to file - after using HSV\_Threshold to remove the pitch.

makeVideo.m

- Accepts a file of images (frames from a video) and generates a video from these frames, and outputs it.

Offscreen\_video\_eg.m and tex\_image.m

- Adding words to the top left hand corner of the frame.

K\_Means\_Clustering.m

- Needs a video of an event.
- Reads in frames of video.
- Calculates the height and the width of a video.
- Creates a struct array with cdata and colormap values.
- Removes pitch and spectators from each frame.
- Generates K-Means Data using VLFeat's vl\_kmeans
- Plots results.
- Outputs 3 files:
  - -assign.mat
  - -centre.mat
  - -histogram.mat
- In order to make the program work I had to use Jelena's masking methods as Matt's were causing errors.
- Does take around an hour to run.

HMM\_Train.m

- Need to be pointed to a file containing data.
- Loads the data -assign.mat
- Makes an initial guess of parameters using  $Q = 7$  and  $O = 7$ .
- Uses file mk\_stochastic.m from [http://www.cs.columbia.edu/~jebara/code/dst/src/mk\\_stochastic.m](http://www.cs.columbia.edu/~jebara/code/dst/src/mk_stochastic.m) to ensure that the argument is a stochastic matrix (a stochastic matrix is used to describe the transitions of a Markov chain).
- Uses files dhmm\_em.m (finds the ML/MAP of parameters of an HMM with discrete outputs using EM) and dhmm\_logprob.m (computes the log-likelihood of a dataset using a discrete HMM) to improve the guess of parameters and compute likelihood - files come from a HMM Toolbox found at - [http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm\\_download.html](http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm_download.html)
- It then plots the values.

Find\_Features.m

- Classifies each frame of a video based on the cluster training data and compares it to the HMM model for that cluster data.
- Need a video of a match, and to be pointed to a file of data.
- Loads in video file and calculates number of frames and height and width of video.
- Generates a Histogram using the number of frames.
- Uses Jelena's function getImageDescriptor to populate the Histogram.
- Loads a 'Frame centres.mat' file.
- Computes KNN - using the cvKnn function from <http://sourceforge.net/projects/cvprtoolbox/>.
- Takes a sample of histogram data for every frame.
- Uses the function dhmm\_logprob.m again from the aforementioned HMM Toolbox.
- Was originally using the knnclassify.m function from the BioInformatics Toolbox but had to change to another KNN classifier.
- Does take a few minutes to run.

Find\_Training\_Hists.m

- Classifies each frame of a video based on the cluster training data.
- Needs to be given a video of a match.
- Loads in video file and calculates number of frames and height and width of video.
- Generates a Histogram using the number of frames.
- Uses Jelena's function getImageDescriptor to populate the Histogram.
- Ran over night and still hadn't finished.