Appendix 10. Code for Batch-processing Images.

Fig. A10.1. Example IDL (Exelis Visual Information Solutions, Boulder, CO, USA) code to batch-process multiple aerial image files for automated detection of Lesser Snow Geese (*Chen caerulescens caerulescens*). The script instructs ENVI (Exelis) to sequentially open all TIFF-format raster image files in a specified directory, including in subdirectories. For each file, ENVI is instructed to ignore pixels with a data value of 0, and query the file's metadata to retrieve the image's filename, spatial dimensions and list of spectral bands. A low-pass filter is then applied to the image, and the resulting smoothed image is saved to a specified output directory with the identifier "smoothraster" appended to the end of the original filename. The smoothed image is then passed along to the 'Rule-Based Feature Extraction' procedure, which is instructed to process the image's full spatial extent and all spectral bands. The established image segmentation parameters are then applied (note that IDL identifies spectral bands 1, 2 and 3 as 0, 1 and 2, respectively) and classification of the resulting image objects is executed in accordance with the specified rule set file. Finally, a segmentation raster file as well as a polygon vector file (with the identifier "classifiedsnowgeese" appended to the end of the original filename) delineating all classified geese are saved to the output directory. Since the segmentation raster is four times the file size of the original image and its only purpose is to serve as the input for classification following segmentation, to save disk space it was not assigned a unique filename for each processed image file. This way, the segmentation raster generated by each successive image simply overwrites the file generated by the previous image.

```
PRO batch_process_5cm_originals
  COMPILE_OPT IDL2

  e = envi(/HEADLESS)
  files = File_Search('F:\Snow_goose_imagery\5cm_GSD\', '*.tif')
  FOR i = 0, N_Elements(files)-1 DO BEGIN
    raster = e.OpenRaster(files[i], DATA_IGNORE_VALUE = 0)
    fid = ENVIRasterToFID(raster)
    envi_file_query, fid, sname = sname, dims = dims, nb = nb
    task = ENVITask('LowPassFilter')
    Task.Input_Raster = raster
    Task.Output_Raster_URI = 'F:\Snow_goose_imagery\5cm_GSD\Classification_output\'+sname+'_smoothraster.dat'
    Task.Execute
    smooth_raster = Task.Output_Raster
    s_fid = ENVIRasterToFID(smooth_raster)
    envi_doit, 'envi_fx_rulebased_doit', $
      fid = s_fid, dims = dims, pos = Lindgen(nb), $
      segment_bands = [1, 2], scale_level = 90.0, $
      merge_algorithm = 'fast lambda', merge_bands = [1], merge_level = 90.0, $
      kernel_size = 5, $
      rule_filename = 'F:\Snow_goose_imagery\5cm_GSD\5cm_ruleset.rul', $
      segmentation_raster_filename = 'F:\Snow_goose_imagery\5cm_GSD\Classification_output\segmentationraster.dat', $
      vector_filename = 'F:\Snow_goose_imagery\5cm_GSD\Classification_output\'+sname+'_classifiedsnowgeese.shp'
  ENDFOR

END
```