

**Project Spot Check:**

**Developing a Vision-Based Program for Identifying Individual Snow Leopards from Camera Trap Photographs**

**User Guide to ECE 18.7’s updated version of HotSpotter**



**Introduction**

The following User Guide details both the manual automated processes within HotSpotter for use by Panthera biologists in Snow Leopard identification.

DISCLAIMER: Whether the user is working in Linux or Windows, be sure to keep the file structure intact the way it is when installing from the Linux shell file or the Windows executable. The script files expect a specific file structure and therefore the source files must be kept in the state they are in upon installation.

DISCLAIMER: This user guide does not detail installation of HotSpotter. If you have not yet installed HotSpotter, those instructions can be found on our GitHub setup repository:

<https://github.com/SU-ECE-18-7/ece187setup>

**Manual Process**

1. Open a project. ECE 18.7 has saved all of their test image sets in /hotspotter/hstest/testdb . We recommend that you do this or opt to store your project files in a file directory that you trust.
2. Import Image(s). There are two options for this step: Importing specific image files or importingan entire image directory. After importing images, you can then interact with them by clicking on an image index.
3. “Chip out” your image. After clicking and opening an image, you can then define regions of interest within that image in which HotSpotter can use to compare snow leopards throughout the data set. Begin this process by opening an image and then pressing “A”. This will allow you to click on two opposite points of your rectangular region of interest for HotSpotter to use as reference. Repeat this process as many times in one image throughout the whole dataset until you feel that there is a sufficient amount of chips.
4. Query a chip. Once all of the desired chips have been created, you can then continue on to querying. Querying is the process in which HotSpotter will compare an individual chip across the entire chip database and return the best chip matches. Start this by selecting a chip in your chip table and then pressing “Q”. This will launch the Querying process and once finished, the results from that process will be visible in the “Query Results Table”.

**Automated Process**

1. Open/Create a project. ECE 18.7 has saved all of their test databases in /hotspotter/hstest/testdb . We recommend that you do this or opt to store your project files in a file directory that you trust.
2. Import Image(s). ECE 18.7 recommends that for the automated process, you import an image directory. The correct format for the image directory to be imported requires a subdirectory named “templates” that contains all of the templates that correspond to the images in the selected folder. Our AutoChipping feature expects a subdirectory of bitmap templates for the image set within your image directory. This is required in order to run AutoChipping.
3. AutoChip. This process automatically chips out an optimized number of chips from each image in your image table. Simply press the AutoChipping button and HotSpotter will chip each image. This process will take approximately 20 seconds per image in your project. The parameters for AutoChipping can be edited by going to Options>Edit Preferences > AutoChipping. Description of the two parameters can be found in the AutoChipping Dev Guide.
4. AutoQuery. This process runs a query for every chip in the database and obtains recognition scores for every chip. Simply press the “AutoQuery” button. Instead of displaying results in the Query Result table, all recognition scores are saved in a .csv file in your \_hsdb folder, which is created in the selected project folder. This .csv file is then used in the next process: Clustering.
5. Clustering. This process will take the recognition scores obtained from AutoQuerying and use a disjoint clustering algorithm to associate very strong matches with each other and group those matches into individual cats. Simply press the Clustering button. The output from this process is another.csv file containing cat identities with corresponding image names.
6. Sort into Folders. This process will take the outputs from Clustering and sort database images into folders based on their identification. For instance, all images labeled as “cat\_1” would be put in the “cat\_1” folder and so on. Images which have been given multiple labels are put in an “Unknown” folder for a biologist to identify.