

# Snow Depth Measurement

## What is it?

This is a tool that allows a user to perform automated measurements of snow depth using outdoor cameras. The application is built on Python 3.6 and can be run by any user if they have the necessary dependencies installed.

## The Field Setup

The required setup consists of an outdoor camera fixed in position and pointing at a series of stakes with coloured reference markers on them. The algorithm measures the snow depth along the stakes using the reference markers to generate a tensor (conversion from pixels to standard units of measurement such as mm). An image of the setup can be seen below.



# Installing

To install the application on your machine, following the steps below:

1. Download a zip of the latest release here on Github  
(<https://github.com/NRCANTerry/snow-depth/releases>)
2. Install Python 3.6 (preferably in a virtual environment)
  - You can use Miniconda to create a virtual environment for this project
  - You can download Miniconda here (<https://conda.io/miniconda.html>)
  - Create a Python 3.6 environment using `conda create -n envName python=3.6`
3. Install the following Python packages (using pip, conda, etc.)

## Option 1: Using requirements.txt file

Run the following commands (in the repository directory):

- `conda install matplotlib`
- `pip install -r requirements.txt`

## Option 2: Individual Imports

Run the following commands:

- `conda install matplotlib`
- `pip install tqdm`
- `pip install opencv-python`
- `pip install Pillow`
- `pip install numpy`
- `pip install scipy`
- `pip install keras`
- `pip install xlswriter`
- `pip install PollyReports`
- `pip install reportlab`
- `pip install scikit-learn`
- `pip install tensorflow`
- `pip install scikit-image`

You have installed all the required dependencies for this project. Please see the next section on how to run the application.

# Project Structure

Before I discuss how to run the application, I would like to provide a quick overview of the project structure.

- The main Python script that is executed by the user is snow\_depth.py
- The remaining Python files contain functions and classes used to perform individual functions (e.g. equalization)
  - They are stored inside the "./Include" directory
- At the root level (directory where snow\_depth.py is saved), two folders will be created by the application
  - AppData: contains deep learning models, registration templates, preferences and training images that are used by the application
  - this folder should not be modified
  - Results: contains the output for each individual "run"
  - this folder can be deleted along with individual runs inside it

The full project structure can be seen in the diagram below:

```
| - - - snow-depth (project folder)

|     | - - - AppData (generated by the algorithm)

|     |     | - - - models (Deep Learning Models)

|     |     | - - - templates (Templates saved by the algorithm)

|     |     | - - - training (Deep learning training images are stored here)

...

|     | - - - Include

|     |     | - - - DL (contains Deep Learning algorithms)

|     |     |     | - - - classify.py

|     |     |     | - - - lenet.py

...

|     |     | - - - GUI (contains GUI code)

|     |     |     | - - - datePicker.py

|     |     |     | - - - main.py

|     |     |     | - - - template.py

...

|     |     | - - - calculate_depth.py

|     |     | - - - colour_balance.py
```

```
|      |      | - - - equalize.py
|      |      | - - - filter_night.py
|      |      | - - - generate_report.py
|      |      | - - - get_tensor.py
|      |      | - - - intersect.py
|      |      | - - - order_points.py
|      |      | - - - overlay_roi.py
|      |      | - - - register.py
|      |      | - - - update_dataset.py
|      |      | - - - validate_stakes.py
...
|      | - - - snow_depth.py (main script)
```

Further information about the individual files as well as explanations of the algorithms used can be found in the Wiki here on GitHub.

## Running the Application

To run the algorithm run the following command when you are in the project (snow-depth) folder.

```
python snow_depth.py
```

## First Steps

- To measure the snow depth, the algorithm requires several files/preferences
- It will automatically generate these files

### 1. Generate a Template

- To get a snow depth measurement, a template is required
- The template should be a daytime image with no dust/dirt on the camera lense
- The template can have snow or be a no snow image

1. Launch the script
2. In the top menu bar select "Templates" then "Create Template"
3. Select the template image using the pop-up window
4. Indicate whether the image has snow in it
5. Fill in the required parameters
  - i. Tensor STD Dev - how much the tensor of an input image can vary from the average (5 is a good starting point)
  - ii. Register STD Dev - how much the registration matrix of an input image can vary from the average (5 is a good starting point)
  - iii. Maximum Transformation - how much an image can be rotated, translated and scaled before the translation is considered invalid (used to filter out poor registrations)
6. Click Continue
7. Follow the GUI instructions
  - i. Select all the stakes in the image

Good selection:



- ii. Select blobs on stake
8. If the template has no snow, after selecting the blobs, you will be asked to identify the intersection point
- ii. If it is visible, you may click on it with the mouse
  - iii. If it is not visible, you may click on any point along the stake and provide the GUI with the height of that point. It will automatically calculate where the base of the stake should be.

## 2. Create a Profile

- A profile contains the majority of the settings that are required to run the algorithm
1. With the script running and the main GUI window open select "Profiles" then "Create Profile"

Measure Snow Depth

**Preferences**

**Menus**

Basic

Registration

Upper Border  px

Intersection

Lower Border  px

Other

Template

Clip Limit

Tile Size (  ,  )

☒ Use Deep Learning

Create Profile



2. Fill in the empty boxes in the preferences window
3. Filled boxes can also be modified but contain default values and should only be changed after running the algorithm on the default settings
4. Click "Create Profile" and follow the on-screen instructions

## 2.1 Profile Options

### ***Basic***

<b>Name</b>	<b>Explanation</b>
Upper Border	Upper crop factor to remove metadata
Lower Border	Lower crop factor to remove metadata
Template	Template to be used with profile
Clip Limit	Limit for contrast enhancement (prevents oversaturation)
Tile Size	Number of rectangular regions the image is divided into for histogram equalization (e.g. (8, 8) produces 8 rows and 8 columns)
Use Deep Learning	Option to use deep learning within profile

### ***Registration***

<b>Name</b>	<b>Explanation</b>
ORB Features	Number of features for ORB algorithm
ORB Threshold	How aggressively the ORB matches are discarded (lower threshold means less matches but higher quality)
ECC Params (w/ ORB)	ECC parameters if feature-based registration was successful

Name	Explanation
ECC Params (no ORB)	ECC parameters if feature-based registration failed
Threshold (1e-)	Termination threshold for ECC algorithm (higher values i.e. 1e-10 will give better accuracy but will add significant computation time)
Iterations	Maximum number of iterations ECC algorithm will run for before exiting (if it doesn't meet the termination threshold)
Use SSIM Selection	Option to use structural similarity index to compare ORB and ECC registered images and choose the image that is most similar to the template. Should only use in cases where ECC is causing poor registrations.
Use Robust Registration	Option to use registration dataset which tracks historical registrations and discards those that do not fit the trend. Should only be enabled if numerous poor registrations are not being detected.
Mode	Registration mode: Either ORB + ECC, ORB Only, or ECC Only

### ***Intersection***

Name	Explanation
Peak Height	Minimum intensity for a peak in the intensity signal
Peak Threshold (%)	Used to calculate <code>snow_threshold</code> in peak analysis ( <code>snow_threshold = peak_intensity * params[1]</code> )
Min Threshold	Minimum <code>snow_threshold</code> used in peak analysis
Stake Coverage (%)	How much of the signal before the peak should be identified as stake ( <code>intensity &lt; snow_threshold</code> )



Name	Explanation
Snow Coverage (%)	How much of the signal after the peak should be identified as snow ( <code>intensity &gt; snow_threshold</code> )
Use Robust Intersection	Option to use stake edge detection to refine intersection point estimation. Highly recommended unless intersection results become erroneous.

### **Other**

Name	Explanation
Diameter	Diameter of each pixel neighbourhood used during filtering ( <a href="https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html?highlight=bilateralfilter#bilateralfilter">https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html?highlight=bilateralfilter#bilateralfilter</a> )
sigmaColour	Filter sigma in the colour space ( <a href="https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html?highlight=bilateralfilter#bilateralfilter">https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html?highlight=bilateralfilter#bilateralfilter</a> )
sigmaSpace	Filter sigma in the coordinate space ( <a href="https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html?highlight=bilateralfilter#bilateralfilter">https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html?highlight=bilateralfilter#bilateralfilter</a> )
Use Parallel Pool	Option to use a parallel pool. If this option is selected, the application will automatically use a parallel pool for computation to reduce processing time.

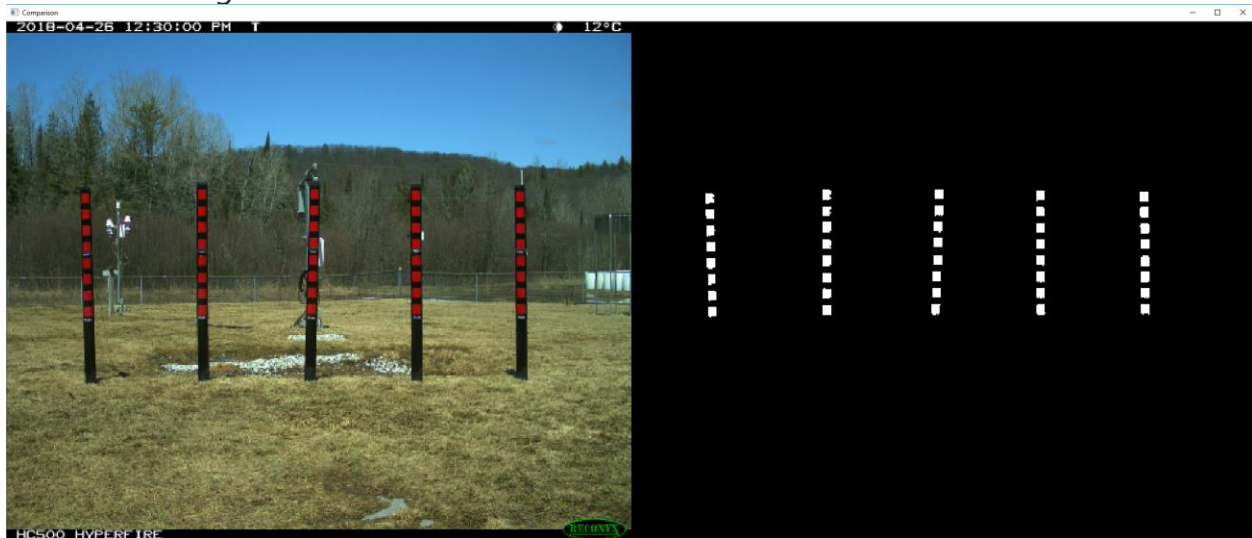
## **3. Determine the HSV Range**

- In order to identify blobs in an input image, the algorithm must know what colour the reference blobs are
- After multiple runs the algorithm will initialize its own deep learning algorithm to help identify blobs but it is still recommended that you provide it with an HSV range

1. Fill in the boxes in the main GUI window under "HSV Range"

2. If you are not sure what the HSV Range for your image is, go to "File" then "Load Preview Tool"
  - i. This will open the algorithm's built in tool to identify blob HSV ranges
  - ii. Select one of the images that you will be running through the algorithm
  - iii. Use the sliders to experiment with different HSV ranges
  - iv. The input image is shown on the left and the binary (thresholded) image on the right
  - v. When you find an HSV Range(s) that results in only the blobs being included in the binary image, click "Save Values" to have them inputted into the main GUI window
3. Once you are happy with your HSV Range, save it by clicking "Colours" then "Save HSV Range"

Good HSV Range:



## 4. Run the Algorithm

1. If you have followed the above steps you have successfully created a template, profile and an HSV range that will be used by the algorithm to determine the snow depth of your input images
2. To run the algorithm first select the folder containing the input images by clicking "Select"
3. Then either input an HSV range into the text fields or select a saved field under "Select HSV Range"

4. Select a profile under "Settings"
5. **Optional** To specify output and date ranges press the arrow on the right to open the advanced menu

i. The advanced menu looks like this:

- ii. **Selecting a Date and Time Range:**
- iii. To run images taken between two dates, specify the start and end dates in the FROM and TO sections
- iv. To run images taken between two dates and times, specify the start and end dates and times in the FROM and TO sections
- v. To run images taken at a certain time (but with no start and end date) specify the time in the FROM and TO sections

6. If you would like the algorithm to output images at each stage (i.e. equalization, registration, etc.) then select "Debug"
7. Click "Run"
8. The algorithm will provide progress bars with time remaining estimates
9. Once it has completed, your results can be found in the "Results" folder generated by the algorithm
10. The results are sorted by date so open the most recent folder