Project
FACT/NIFA

USDA
FAIN:20196702229922

USER GUIDE

Cloud Coverage Prediction from Sky Images

Sponsored by United States Department of Agriculture (USDA), Rutgers University and Siemens Technology collaborated on a project titled "FACT: Deep Learning for Image-based Agriculture Evaluation". Its goal is to develop vision-based algorithms for precision agriculture, focusing on yield estimation and sun irradiance forecasting to assess risks of fruit overheating.

This document presents how to use one module of the proposed solution for cloud coverage estimation from ground-based sky images.

SIEMENS

Contents

Background

The proposed tool for cloud coverage estimation belongs to a larger pipeline for precision agriculture. We provide here some context.

USDA Project

Machine vision for precision agriculture has attracted considerable research interest in recent years.

This work is sponsored by United States Department of Agriculture (USDA) with award number *FAIN:20196702229922*. It is part of a collaboration project titled "*FACT: Deep Learning for Image-based Agriculture Evaluation*" with Rutgers University serving as prime and Siemens Technology providing industrial and scientific expertise.

The goal of this project is to develop an end-to-end cranberry health monitoring system to enable and support real time cranberry over-heating assessment to facilitate informed decisions that may sustain the economic viability of the farm.

Overall Solution

To that end, we proposed two main deep-learning-based modules:

- One module for cranberry fruit segmentation to delineate the exact fruit regions in the cranberry field image that are exposed to the sun
- One module for prediction of cloud coverage conditions and sun irradiance, to estimate the inner temperature of exposed cranberries.

Scientific Background

Both modules rely on deep-learning, a recent field of machine learning that leverages novel artificial neural network architectures to learn complex tasks from data.

The proposed module for cloud coverage analysis performs both cloud segmentation and motion estimation from up-facing sky images, relying on weak supervision techniques (i.e., learning its tasks by leveraging their synergy and, as a result, requiring little human guidance / few annotations).

The tool shared here has been trained on a large dataset of sky images, and the resulting optimized parameters are being shared as part of this code base.

For additional details on the pipeline, please refer to our peer-reviewed scientific paper:

Akiva P, Planche B, Roy A, Dana K, Oudemans P, Mars M. AI on the Bog: Monitoring and Evaluating Cranberry Crop Risk. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision 2021 (pp. 2493-2502) - direct link.

Application

This document introduces a simple tool extracted from the overall pipeline and packaged for low-tech users.

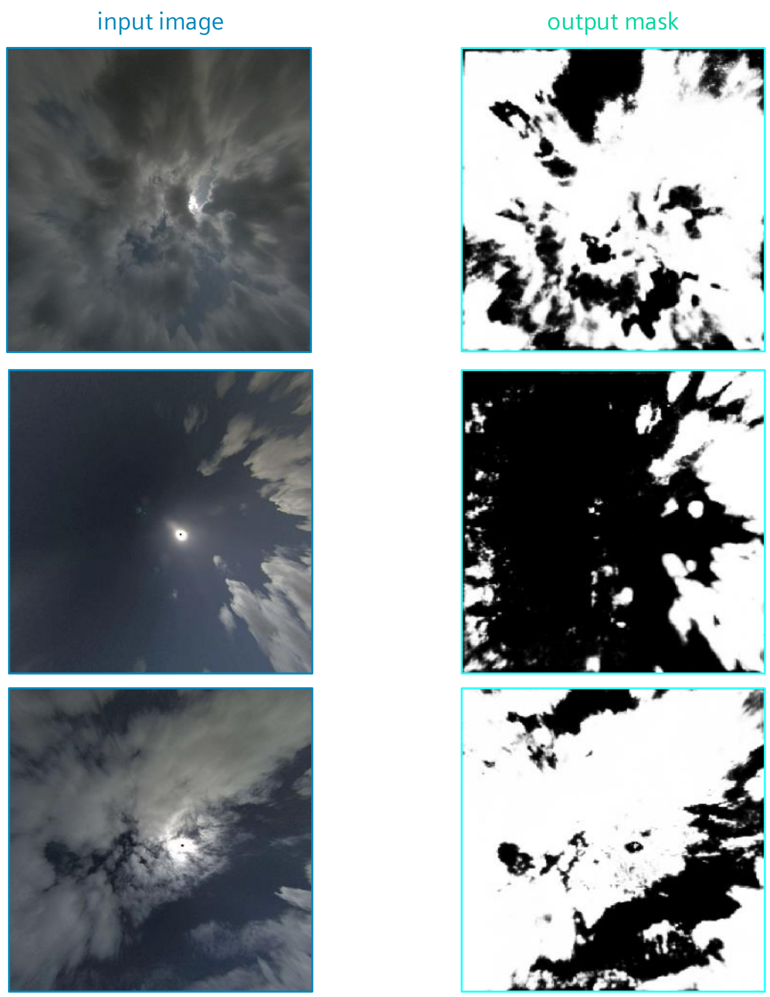
Task Performed

This command-line tool takes an upward-facing sky image as input, and return the predicted mask highlighting the presence of clouds in the image. To achieve this, the tool loads one of our pretrained deep-learning algorithms and feeds it with the provided image.

The resulting mask is saved into a designated file, so that users can then visualize it and use it for downstream tasks (e.g., cloud coverage masks are used within our larger system to forecast sun irradiance values).

Examples

Some examples of input images and results from the proposed tool are presented below:



Usage

We now present the steps to install and use the tool.

Requirements

Implement in the programming language *Python*, the proposed tool relies on the following code libraries:

- `Python >= 3.7`
- `PyTorch >= 1.2`
- `OpenCV == 3.4.2`
- `Pillow >= 7.0.0`
- `tqdm` (optional)

Installation

To install these requirements and have the tool ready, the following steps should be followed.

Direct link to the ZIP archive: https://github.com/siemens/fact/archive/refs/heads/master.zip

Code Download

First, download the code from our GitHub repository: https://github.com/siemens/fact. In the GitHub project page, click on the green "*Code*" button, then on "*Download ZIP*" in the expanded menu.

Once the code archive downloaded, unzip it into a folder.

Python Installation

If not on the machine yet, download and install Python (version > 3.7): https://www.python.org/downloads.

If any problem arose during installation, feel free to contact the authors or open an issue on GitHub: https://github.com/siemens/fact/issues

Installation of Requirements

Open a command terminal in the folder containing the code base, then enter the following command to have all the requirements automatically installed:

```
pip install -r requirements.txt --user
```

If all went well, the tool is now ready to be used.

Run

To use the tool, run the following command:

Command

```
python estimate_cloud_coverage.py \
--image {path_to_image} \
--output {path_for_result}
```

For example, to run the tool on one of the provided test sky images and to save the results in a different folder:

```
python estimate_cloud_coverage.py \
--image
"resources/test_images/Princeton/Princeton_TestSamples/t
est sequence (1).png" \
--output resources/test_images/result.png
```

Parameters

The proposed tool has additional arguments, detailed below. Tool documentation can also be accessed by entering the command:

```
python estimate_cloud_coverage.py –help
```

- -h, --help
  - To show the help message and exit
- -c {value}, --config {value}
  - To provide a different configuration file for the deep-learning model (value = path to file).
- -r {value}, --resume {value}
  - To provide different trained parameters to the deep-learning model (value = path to file).
- -d {value}, --device {value}
  - To specify which GPUs the tool should use (value = GPU indices).
- -i {value}, --image {value}
  - To provide the target sky image (value = path to file).
- -o {value}, --output {value}
  - To specify output path where to save resulting cloud coverage (value = path to file).

License

See *LICENSE* file: https://github.com/siemens/fact/blob/master/LICENSE.

Acknowledgements

- This project uses pytorch-template by Victor Huang as boilerplate: https://github.com/victoresque/pytorch-template.

- The original SegMotionNet model was proposed and implemented by Chen Qin et al., c.f. Joint-Learning-of-Motion-Estimation-and-Segmentation-for-Cardiac-MR-Image-Sequences: https://github.com/cq615/Joint-Learning-of-Motion-Estimation-and-Segmentation-for-Cardiac-MR-Image-Sequences.

- The original PWC-lite was implemented by Liang Liu et al., c.f. ARFlow: https://github.com/lliuz/ARFlow.

Authors

Aditi Roy, Benjamin Planche
Research Scientists
Siemens Technology
Smart Machine Vision & Visualization