Detecting Meteors Within Spectrographs With Machine Learning

Kirin Patel
12/11/19
AE 457

What do I Propose to do?

- My project proposes to introduce a machine learning model to identify and mark meteors within spectrographs that can be continuously trained with new data so that accuracy can be continually improved.

Why am I Proposing this?

- There is no accurate automated method for detecting meteors within spectrographs that are generated by BRAMS (Belgian RAdio Meteor Stations).
    - There is an OpenCV implementation that was created but it has not been maintained for over 2 years and its accuracy and effectiveness are unknown (it is also not well known).
- Large amounts of data are collected by BRAMS daily, roughly one spectrograph per minute per location, with roughly 30 locations within Belgium.
    - This data needs to be processed so that analysis can be conducted. By decreasing data processing times, more analysis can be done.

What are Some Expected Roadblocks?

- Lack of data
- Data format
- Time
- Lack of knowledge regarding Keras/Tensorflow
- Motivation

What has Already Been Done?

- This problem has already been solved! (Calders, 2019)
  - A feature detection algorithm has already been made, this can help guide my machine learning model implementation.
- Color salience has been shown to improve accuracy or feature detection. (Weijer & Bagdanov, 2006)
  - By attributing weight values to different colors, specifically those representing activity within spectrographs, meteors can be more accurately detected.
- Extensive research has been conducted on low end hardware efficiency for running real-time image processing. (Rosten & Drummond, 2006; Khan, Bennamoun, Sohel & Togneri, 2014)
  - This research can help propel a machine learning model's efficiency when run on cost effective hardware.

Alternative Solutions

- Another solution would be to implement an OpenCV based feature detection algorithm
  - This would not provide a new medium for detecting meteors within spectrographs
  - Large improvements in efficiency can be made over the current implementation that was created by Stijn Calders (Calders, 2019)

Proposed Solution

- Create a machine learning model that is based on feature recognition to detect meteors within spectrographs

Requirements

Item Name	Minimum Expected State	Desired Expected State	Future State of Project After Senior Design
Machine Learning Model	Model code exists - Written with the Keras framework - Is integratable with Kubeflow Pipelines and/or a standalone training environment	Model is generated - Is able to identify some meteors within a spectrograph - Utilizes color salience to aid in feature detection	Model is generated - Model has 80% accuracy of identifying meteors within a spectrograph - Model is setup within Kubeflow Pipelines for CI/CD
			- Model is continually trained to improve robustness

Requirements

Data Set	A data set of at least 500 spectrographs exists - Data is already in a JPG format - Data has meteors identified so that training can occur	A data set of at least 5000 spectrographs exist - Same requirements as minimum expected state Data is obtained from RMZ	Have a system that automatically converts a spectrograph into a JPG upon data collection within a BRAMS operated location
Documentation	Have design doc created before implementation begins - Details requirements - Details implementation phases - Details timeline Have documentation that details how to use model	Have documentation for how to initialize project for other users Take screenshots of model and other portions of system to further enhance documentation	Create wiki where documentation can be easily searched
Financial Resources	N/A, project will be conducted using resources accessible for free	N/A, project will be conducted using resources accessible for free	N/A, project will be conducted using resources accessible for free

Budget

- Data collection: $0
  - Obtaining JPG/image formatted spectrographs from BRAMS
- Model creation: $0
  - Creating a machine learning model with Keras/Tensorflow
- Model training: $0
  - Use Kubeflow Pipelines to train the machine learning model to detect meteors within the JPG/image formatted spectrographs
    - Provided by Google from previous Internship
- Model application: $0
  - Using the trained model to detect meteors within the JPG/image formatted spectrographs
  - A terminal/command line based application will be used, can be run on existing systems

Supports

- Personal computer
  - Can be used for development and machine learning model training
- GCP Compute Engine cluster (provided from manager at previous internship)
  - Can be used for machine learning model training
  - All costs are waived
- Dataset (provided by BRAMS)
  - Used to train machine learning model
  - Openly available if an account is made with BRAMS

Project Schedule

- It’s too long to include here, so go here or type https://docs.google.com/document/d/1KOFcpaqxCSPieydm9xnnAcbaCvvaYOZ-rVl7yCs2uio/edit?usp=sharing into your browser

Detecting Meteors Within Spectrographs With Machine Learning

Kirin Patel

- kirinpatel@gmail.com
- @kirinpatel

References

- Calders, S. (2019, January 29). calders/RadioMeteorZoo. Retrieved from https://github.com/calders/RadioMeteorZoo.
- Khan, S. H., Bennamoun, M., Sohel, F., & Togneri, R. (2014). Automatic Feature Learning for Robust Shadow Detection. 2014 IEEE Conference on Computer Vision and Pattern Recognition. doi: 10.1109/cvpr.2014.249
- Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images. Machine Learning, 30(2/3), 195–215. doi: 10.1023/a:1007452223027
- Rosten, E., & Drummond, T. (2006). Machine Learning for High-Speed Corner Detection. Computer Vision – ECCV 2006 Lecture Notes in Computer Science, 430–443. doi: 10.1007/11744023_34
- Weijer, J. V. D., Gevers, T., & Bagdanov, A. (2006). Boosting color saliency in image feature detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(1), 150–156. doi: 10.1109/tpami.2006.3

References Review

- It’s too long to include here so go here or type https://docs.google.com/document/d/1Nx2FslU_0uNe8vgmFnrkekPbf1DkoMV3jFx6QgDSOKk/edit?usp=sharing into your browser