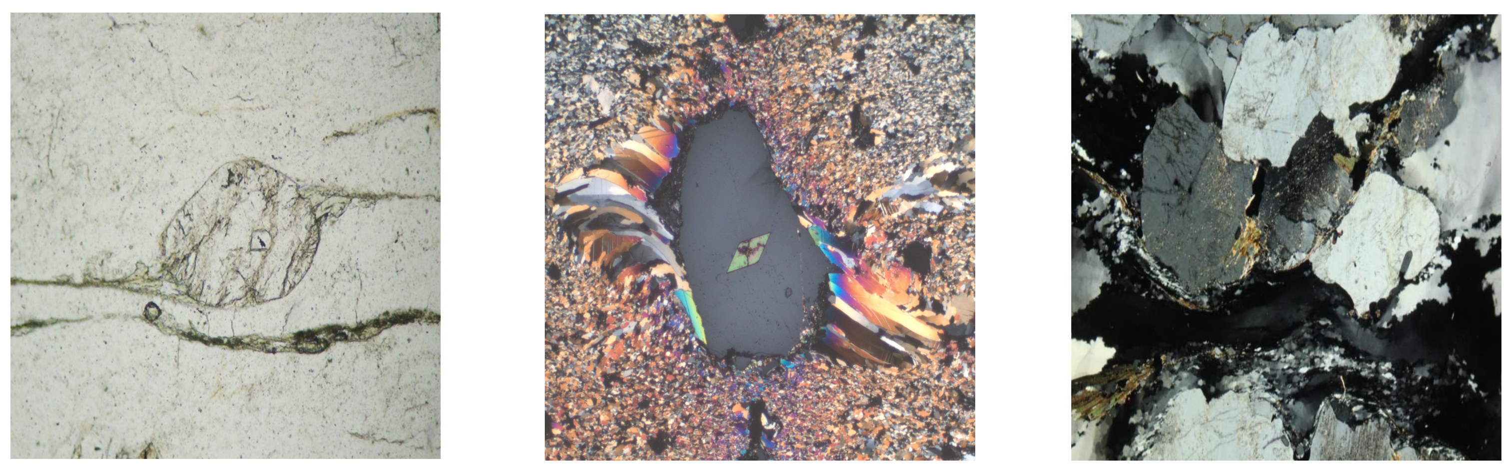
Sigma Clast Recognition using Transfer Learning with a Convolutional Neural Network

Joseph Robinson, Catherine Meyer, Dr. Gurman Gill
Sonoma State University Computer Science Department

Introduction

- High demand for cyber-infrastructure which enables geological image data to be shared.
- Develop a system which accurately identifies whether an input image contains a sigma-clast, even with limited data.

Dataset

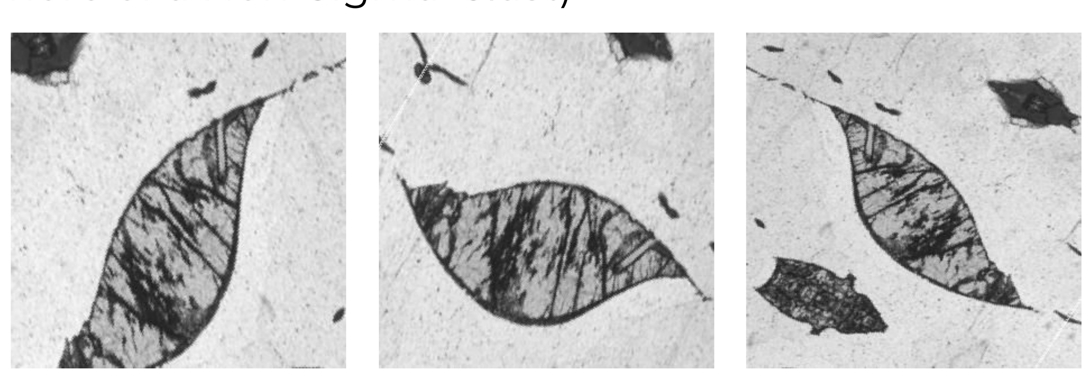


Sigma-Clast — Sigma-Clast — Non-Sigma-Clast

Data Augmentation

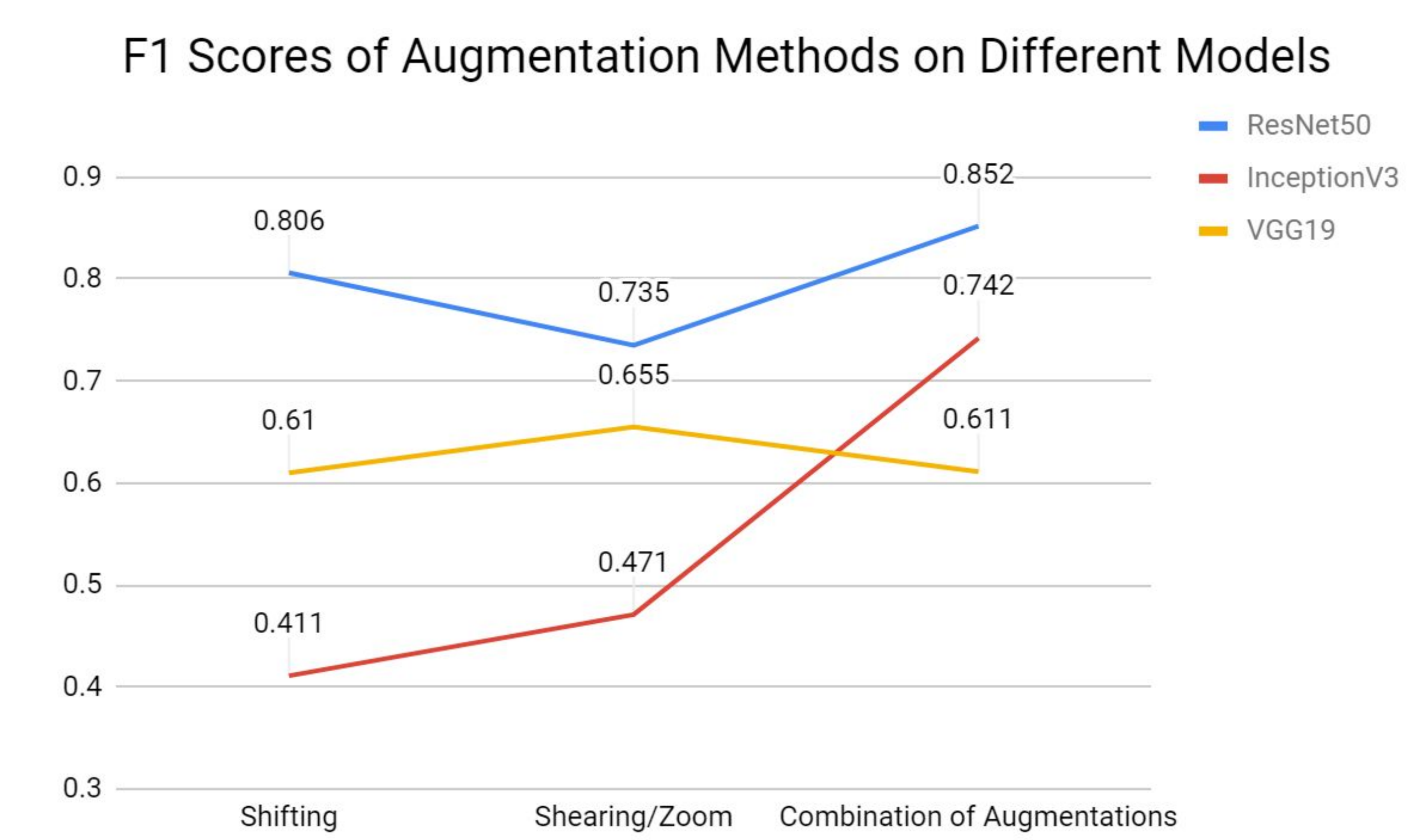
- Allows for smaller datasets to become larger, as long as data is not augmented to a different classification(ex. Distorting Sigma-Clast image to become more of a Non-Sigma-Clast)

Effects Utilized:
Rotation range: 40˚
Height/width shift: 10%
Shear range: 0.2 rad ccw
Zoom: 30%
Horizontal flipping: true
Fill mode : "reflect"



Counterclockwise **R**otation **Z**oom — **He**ight Shift Width Shift **Z**oom — **Ho**rizontal Flip Clockwise **R**otation **F**illed

Testing Data Augmentations ($n_{non\text{-}sigma} = 879$; $n_{sigma} = 100$)

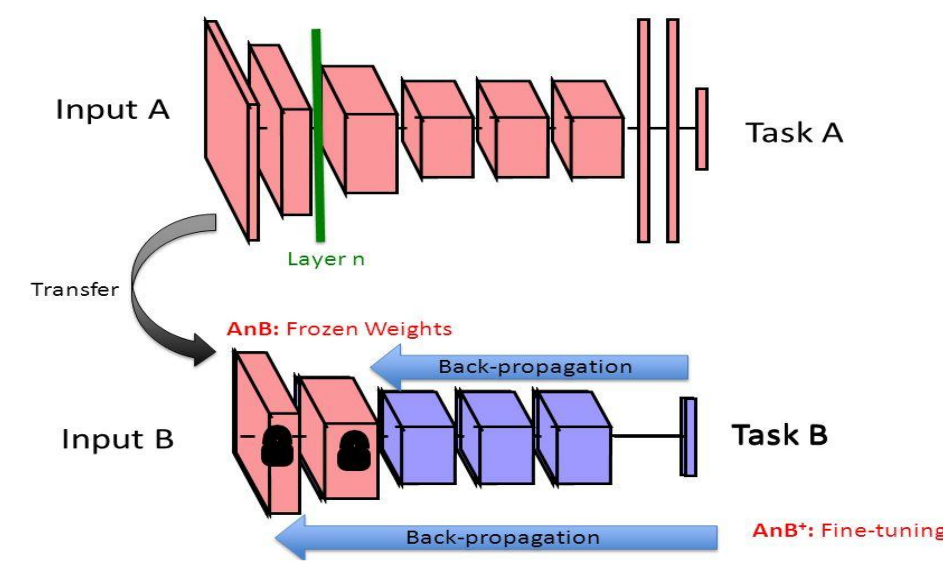


*Combination of Augmentations refers to the augmentations listed under *Effects Utilized* in the above frame

*F1 score is an evaluation metric that is a trade-off between the recall (true positive rate) and precision (fraction of relevant classifications)

Transfer Learning and Fine Tuning

- Allow a Convolutional Neural Network to correctly learn over a very large dataset
- Take the Convolutional Neural Networks and utilize its features instead of creating our own.
- Fine tuning modifies last few layers to ensure we look for the complex structures we want.

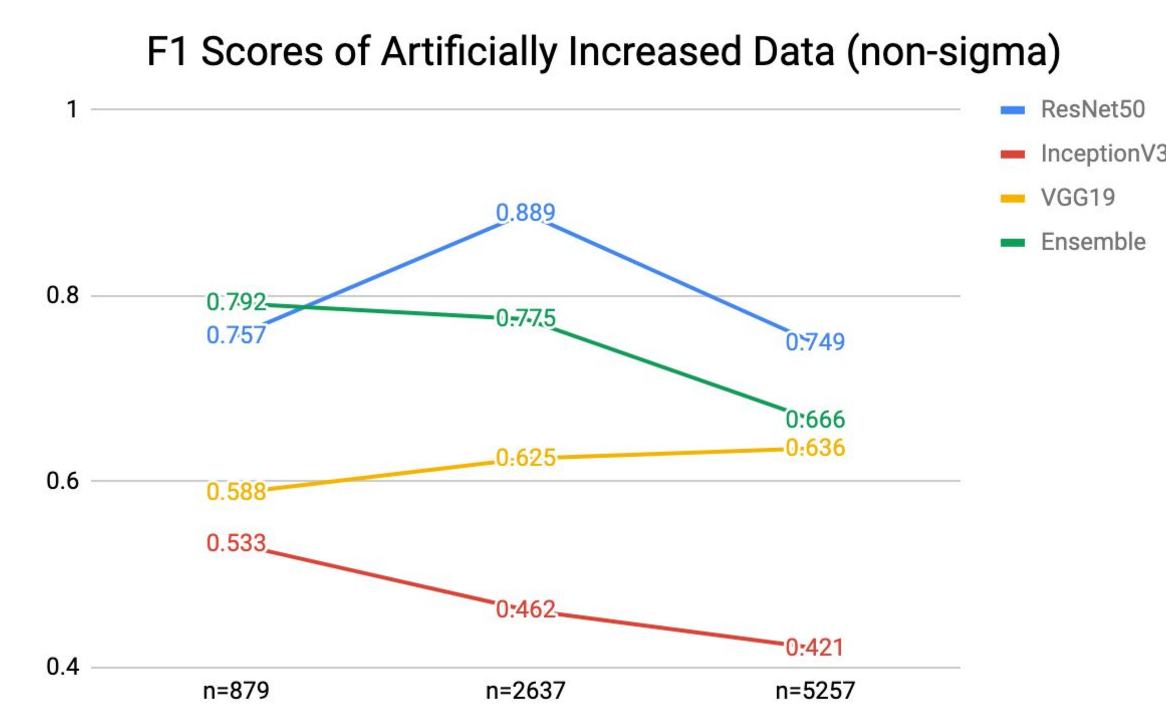


Neural Network Ensembles

- Takes Multiple Convolutional Neural Networks and combines their output.
- This method ensures that we reduce the variance between multiple models by taking the average of each model to compute the classifications

Artificially Increasing the Dataset with Oversampling ($n_{non\text{-}sigma}$= 879, 2637, 5257; n_{sigma} = 100)

- We can artificially increase the largest class of data we have, then use oversampling to create copies of the smaller dataset to match



Best Combination

- Used Methods: Oversampling with n~3000 and Fine Tuning, on ResNet50
- F1 Score: 0.919

Confusion Matrix	Non-Sigma	Sigma
Non-Sigma	791	0
Sigma	5	25

Full Stack Architecture for Web-based Classification

Front-End: React JS
Handles the user interface, sets necessary objects to be sent to the backend.

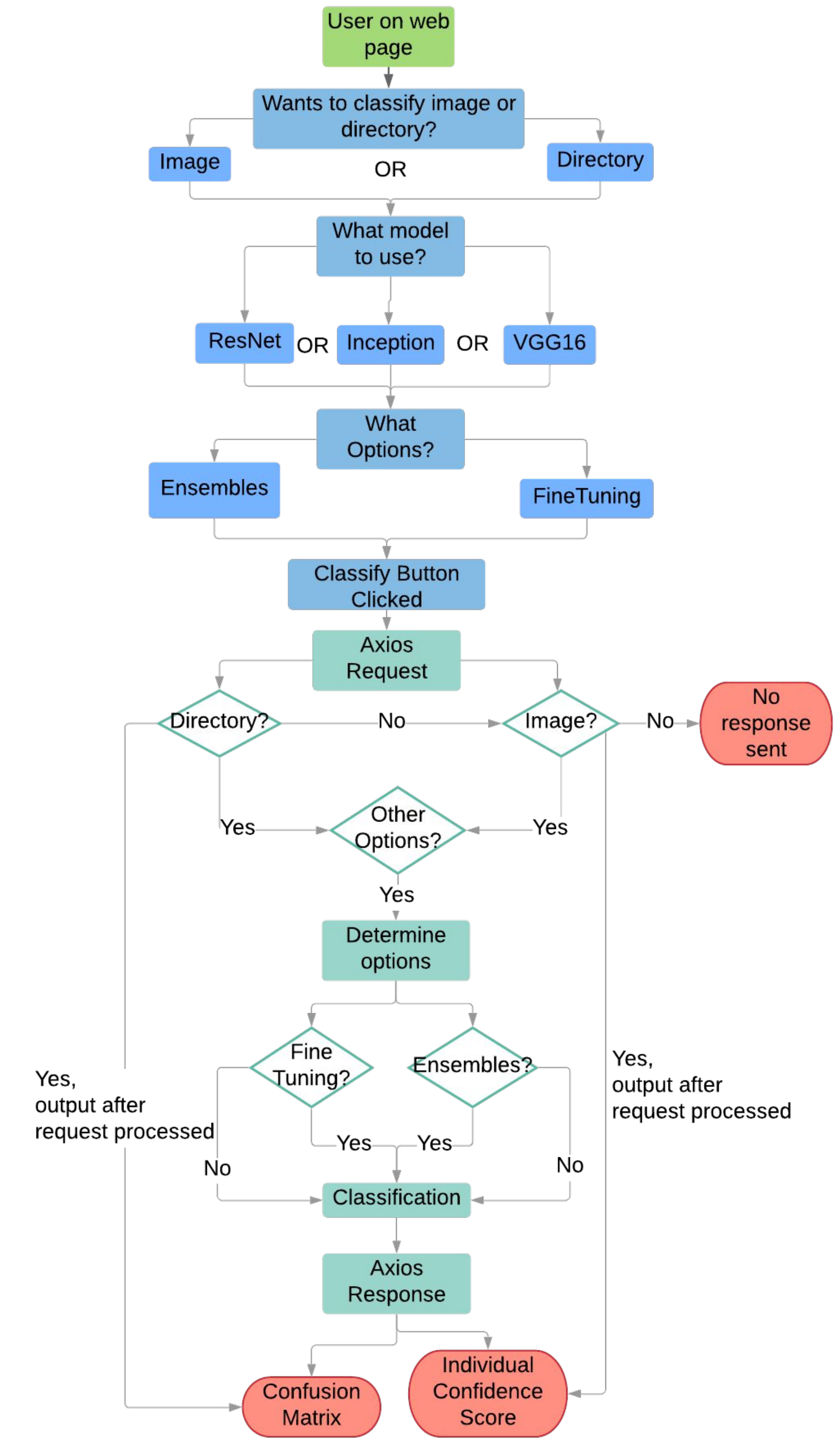
HTTP Client: Axios
"Connects" the front-end to the back-end using HTTP.

Microframework: Flask
Designed for Python; handles the HTTP requests that come from Axios.

Back-End: Python
Creates functionality to resize and classify images

Classification: Keras and Tensorflow
The libraries that allow for the prediction and classification of images.

Process Flow for Web-based Classification



Conclusion and Future Research

- Significant improvement of accuracy can be seen among the system of networks.
- Being able to classify the rotation of a Sigma-Clast (i.e. Counter-Clockwise/Clockwise)
- Allow for K-Fold cross validation evaluation
- Generalize the model for multiclass classification.
- Web Application
  - Enhance UI -- color scheme, fill whitespace effectively
  - Allow for prediction options such as ensembles and fine-tuning
  - Host application on a live site

Acknowledgements

Special thanks to Earthcube NSF Grant for funding this research, as well as Dr. Gurman Gill for guiding the team