

Capstone Project Proposal: Quebec Iron Ore 1

Jingyun Chen, Socorro Dominguez, Milos Milic, Aditya Sharma

Executive summary

Quebec Iron Ore (QIO) is a Quebec-based mining company founded in 2015 specializing in iron ore extraction. They currently gather 100 Gigabytes of data per day and would like to leverage Machine Learning in order to improve their yield from 6 to 7 million tonnes of iron ore per year. We will be working on an image segmentation project which will allow segmenting up to 8 different rock types from images of blasted rock faces. Ore is collected from blasted faces and accurate prediction will reduce waste or ore contamination thereby, reducing operational costs. The major issues in this project include proper pre-processing of the images and creating an accurate model with a limited number of training examples.

Introduction

Being young in the industry and with a strong passion for innovation, QIO believes that applying Machine Learning will help increase their operational efficiency. For this purpose, we aim to develop a prototype called “Image Region Optimizing Neural Network (IRONN)” that can segment different rock types in a blasted rock face image through physical characteristics such as colour and texture of the rocks.

Identifying rocks types correctly reduces the chances of them getting contaminated with waste. It also reduces the chances of throwing ore away into the Waste Dump. Sending contaminated rocks to the Crusher or iron ore to the Waste Dump is expensive for QIO. A proper blend of certain rock types is essential during the crushing stage as it will avoid unnecessary transport costs. Optimizing the fleet is not only a question of where to assign the next trucks to be loaded but a balancing act between what resources are available and what material is required by the mill for the blend. IRONN would help the mine operators to not always depend on geologists’ expertise to identify different rock types from a blasted face. This would allow them to “haul” an appropriate blend of rocks to the mill.

Our project entails developing a proof of concept for this segmentation task. In later phases (beyond the scope of this project), operators will have a product that will enable them to identify blasted rocks accurately in real-time. With an easy tool at hand, they will be able to choose which rocks to send to the stockpile, waste dump or crusher faster and with an appropriate blend composition.

The problem will be broken down into the following **tangible objectives**:

1. **Exploratory Data Analysis:** Going through the images manually and programmatically to find useful patterns to provide summary statistics for model building and help in test-driven development.
2. **Building a model:** Building a machine learning model that will be able to identify different rock types based on the features from the original images.
3. **Prediction:** Predicting the rock type from a new image taken by an operator.
4. **Visualization:** Visualizing and displaying the prediction superimposed on the original image for the operators.

The **deliverables** of this prototype would include the following:

- a Python package for segmenting and predicting different rock types using blasted rock face images
- proper documentation of the code in the package along with clear instructions on how to use the package
- all of the above including any other relevant information in a GitLab repository

The package will be built with the overall pipeline of the project in mind so that any change in the process (for example, adding new, clearer and more consistent images) doesn’t involve a lot of tweaking and is easy to incorporate.

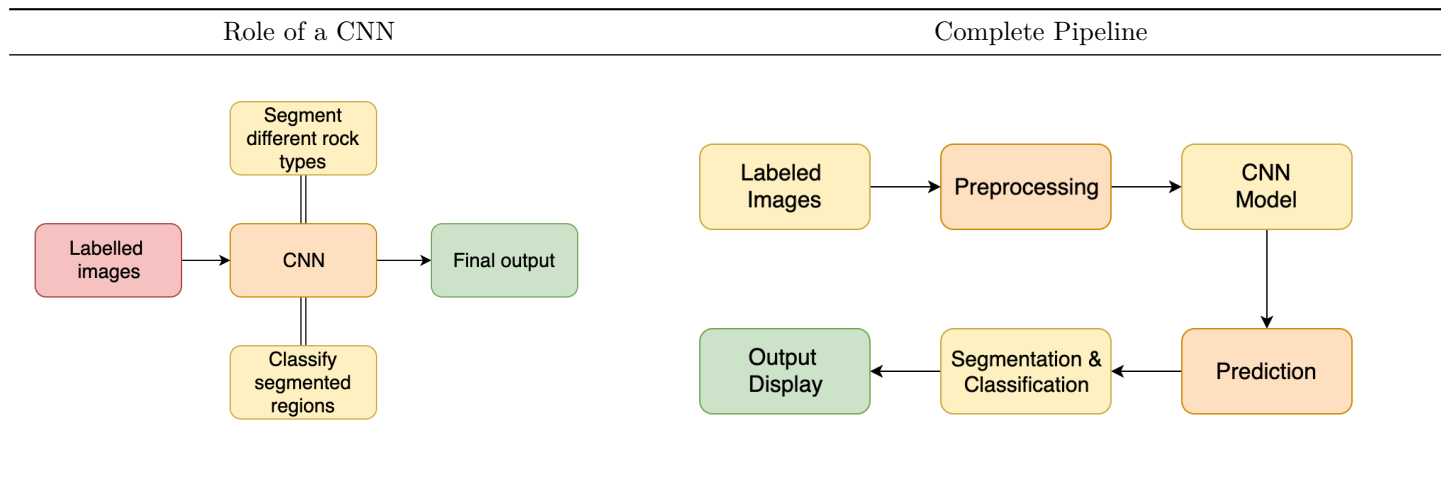
Data Science Techniques

- **Data preprocessing:**
 - Image reading, resizing, transforming (OpenCV - open source image processing library)
 - Labeling images (polygons) using LabelMe (an open source image annotation tool)
 - Noise removal (removing sky, ground and other background noise if required)
 - Manual removal of bad examples and outliers that distort model performance

- **Model building**
 - Extracting properties such as roughness, saturation and pixel intensity from images to study patterns in rocks
 - Using statistical tests to check for significant differences between rock types using the above features
 - Using Convolutional Neural Networks for image segmentation and classification
 - Researching techniques to handle fine-grained texture differences between rock types
- **Model improvement:**
 - Collecting more data to improve model performance
 - Using state of the art algorithms to perform the tasks

What are Convolutional Neural Networks?

A Convolutional Neural Network (CNN) is a Machine Learning algorithm that can be applied to images to extract useful local spatial information such as edges and textures. They use filters that are “learned” during the training process which are then applied to new images to perform the same task that they were trained to perform. They are used in almost all kinds of automation tasks involving images and videos such as self-driving cars, image classification, object detection, etc.



Challenges & Difficulties:

- **Image preprocessing phase:**
 - Labeled images might require segmentation to remove any noise elements while preserving the rock face.
 - Images have different lighting conditions along with other outdoor weather conditions (snow, rain).
 - Angles at which the pictures are taken are not consistent.
 - The source of images (camera) is not consistent.
- **Model training phase:**
 - It will be difficult to detect fine-grained differences between the different rock types.
 - We may not have enough images to differentiate between all of the rock types.
 - Most image segmentation tasks have been performed on objects with clear boundaries. We need to research on how to apply those techniques to rock types without a clear distinction.
 - We are constrained by time which could limit how much we can fine-tune our model.

Timeline

Date	April 26th	May 3rd	May 10th	June 7th	June 18th	June 21st	June 27th
Task	EDA & Presentation	Proposal document	Preprocess Data	Build, train, evaluate model	Finish product	Polish Product	End of project