Minerals Classification

Prinanda Rahmatullah, prinanda50@gmail.com, 085297600471

ANN/DL Architecture

The MobileNetV3 large is used as an architecture with a transfer learning approach. This network was chosen due to it being fast for mobile usage [1]. The output layer block follows the concept of the recent CNN classification layer which includes pooling, batch normalization, dropout, and fully connected layer to normalize network and faster training time [5, 6, 7, 8]. The output layer has 7 nodes as the number of mineral classes.

Positive/Negative Sample

Data annotation is preferred for generating positive and negative samples from each image because we don't have any single pre-trained object detection and classification engine to determine which object is the sample that we need. But for this task, I assumed the sample is correctly divided to each directory name as a positive and negative sample.

Training

Data

Trained dataset provided by the team is divided into 90% training and 10% validation for training purposes. ImageDataGenerator is used as a data loader which supports geometry augmentation techniques.

Hyperparameter

Batch size 16, Epoch 50, and learning rate of $1x10^{-4}$. The optimization function in the final layer uses Adam as the most stable for general purpose training as shown in the research of CNN [3]. The loss function uses categorical cross entropy for multi class classification tasks. And also, the metric uses multi class metrics [2,4]

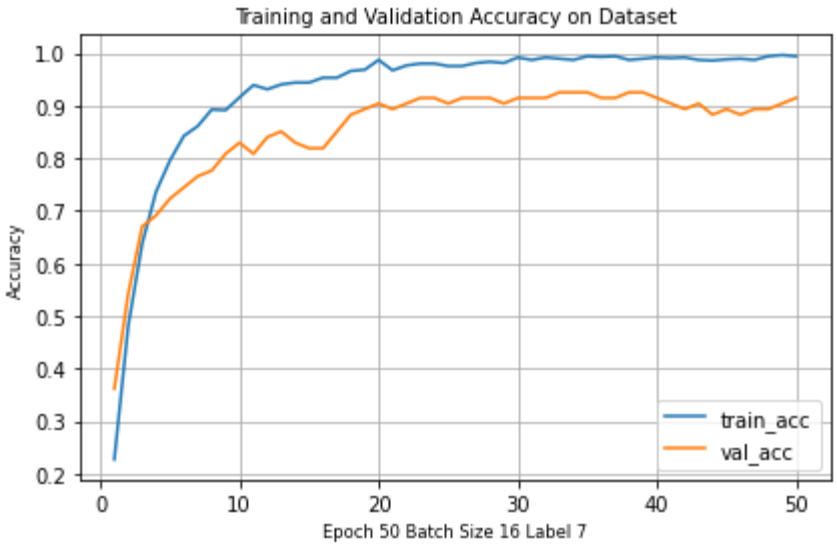
Callback

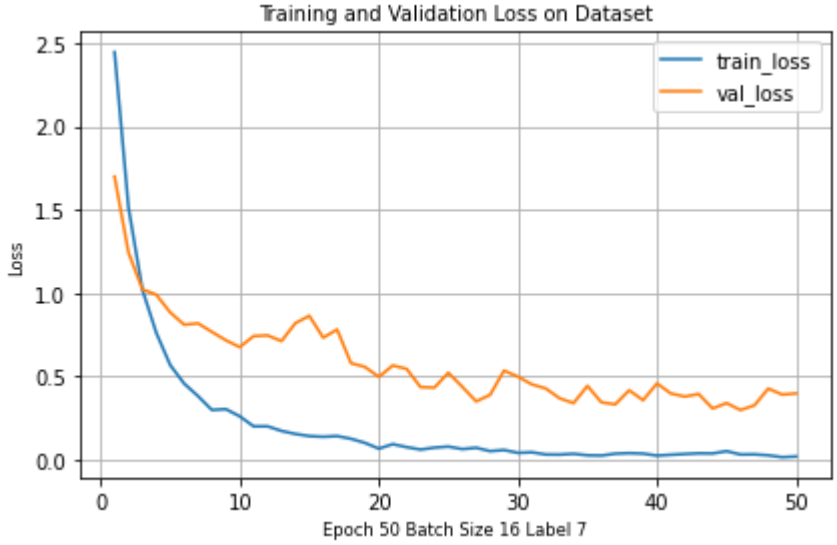
This research use some callbacks, EarlyStopping to stop training when the network in some next epochs isn't learning anymore, CSVLogger to save log each epoch of

training, validation, accuracy and loss, and ModelCheckpoint to save best model of the highest accuracy and the lowest loss.

Evaluation

The best model of lowest loss is chosen as it has the minimum number of false predictions. The validation loss and accuracy respectively are 0.298 and 0.883. And also, the testing loss and accuracy respectively are 0.16 and 0.929.





Inference

The best inference strategy I think is when the model is possibly embedded running on a mobile device or has the least prediction time if using API. In this research, I chose an approach of looping over the directory, then showing the image with its label and confidence score. It is likely the method we use when using API with the function is pretty much the same.

Snapshot Result

Reference

[1] A. Howard et al., "Searching for mobilenetv3", International Conference on Computer Vision, pp. 1314-1324, 2019.

[2] Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for Multi-Class Classification:

an Overview. 1–17. http://arxiv.org/abs/2008.05756

[3] Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 1–15.

[4] Opitz, J., & Burst, S. (2019). Macro F1 and Macro F1. 2, 1–12. http://arxiv.org/abs/1911.03347

[5] https://cs231n.github.io/convolutional-networks/

[6]https://towardsdatascience.com/batch-normalization-in-3-levels-of-understanding-14c2da90a338

[7]https://stackoverflow.com/questions/39691902/ordering-of-batch-normalization-and-dropout

[8]https://pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/