A Deep Learning Model for Estimating Mars Rover Power Consumption Based on Terrain Images

Xiang Cao
Institute for Aerospace Study
University of Toronto
Toronto, ON, M3H5T6
tonyxiang.cao@mail.utoronto.ca

Abstract

This report presents a deep learning approach to estimate the power consumptions of Mars Rover to driver over different type of terrains. The training dataset is based on the Canadian Planetary Emulation Terrain Energy-Aware Rover Navigation Dataset (CPET), collected at Canadian Space Agency's Mars Emulation Terrain (MET). The deep learning model is trained on 12099 color images, collected from Rover onboard camera (Occam Omni Stereo) during 5 runs of the dataset, and tested on 2133 images in Run2 of the dataset. The deep learning model uses transfer learning from the Xception model to extract visual features, and then adds LSTM layers to capture sequence information and TimeDistributed Dense layers to add time dimension of output power data. The model achieved a mean absolute error (MAE) of 63.3 on testset, which deviates 24.5% from the average power ground truth in Run2 of 257.4 Watt.

1 Project Problem Definition

1.1 Mars Rover Power Awareness

Energy awareness is a crucial concern for Mars rovers, as it's almost impossible to provide additional power source to rovers running on Mars. The Spirit and Opportunity rovers relied on solar power, which is an unreliable power source during occasions such as dust storm or winter seasons. The Curiosity rover and the recently launched Perseverance rover prevented solar power's unpredictability by replacing the power source with radioisotope thermoelectric generator, but they are not rechargeable. The main motivation of this project is improving Mars rovers' power awareness to help it have a longer range to achieve more scientific explorations. If a correlation model between terrain image and power usage can be generated, it will provide valuable information on rovers' path planning. The CPET dataset provided an excellent resource with both onboard image and power data logged at 5Hz [1].

1.2 The Computer Vision Approach for Terrain Analysis

Different type of terrains, such as loose sand, bedrock, small rocks, and wheel tracks, pose different challenge in friction, vibration and wheel slip for Mars rover and can result in drastically different energy consumption. The knowledge of terrain surface can help Mars rovers to optimize path planning by choosing a path that travel over terrains requiring less power.

Several previous research have used deep learning based computer vision approaches to acquire terrain knowledge. Rothrock et.al from Jet Propulsion Laboratory, built a deep learning classifier to classify terrains, trained with images captured by the Curiosity rover Navcam [2]. Their deep learning model is a fully-convolutional neural network (CNN), and the front end structure is identical with VGG architecture [3]. They trained on 700 images and achieved a 90.2% classification accuracy. Manderson et.al from Mcgill University, fused onboard camera images and aerial images for a ground mobile robot, and fed to a deep learning model combining convolutional layers and recurrent (LSTM) layers [4]. They used the terrain prediction outcome to further train a reinforcement learning model, and achieved over 90% ratio of driving on smooth terrain.

This project aims to build a novel deep learning model to analyze terrain images captured in CPET dataset, and corelate them with the power consumption logged on the rover. The scope of the project remains the same with the project proposal.

2 Project Methodology

2.1 Utilizing CPET Dataset

The CEPT dataset is collected on a four-wheeled rover based on Clearpath Husky UGV. It has 2 types of on board camera for computer vision analysis: a monidirectional stereo camera(Occam Omni Stereo), which composed of 10 individual RGB camera at 750 x 480 pixel resolution; a monochrome camera (Point Grey Flea3) at 1280 x 1024 pixels. In this project, the RGB images captured with omni camera are used, since 3 channel color images performs better on transfer learning model, which were trained on ImageNet dataset. Among 10 omni cameras, camera 5 is selected as the input image source, as it faces forward and located at the lower position, focusing on the terrain in front of the rover. The CPET dataset contains14246 RGB images recorded from Omni camera 5, and 16214 entries of power data.

2.2 Data Pre-processing

Both the omni camera and drive power monitors are recorded at 5hz rate. The power draw on left motor and right motor are added together for total power prediction. However the actual logged RGB images are about 12% less than the amount of power data, due to lagging of processing and stitching panorama images. To match the sequence and sample amount of input images and output powers, we removed 1 power sample among every 9 power data, and also removed several power samples at both the beginning and end of each runs, when the robot are standing still with minimal power consumption. The goal of the project is to predict power consumption while robot is moving, so zero power consumption will affect the prediction accuracy. The first 14 images of Run1 are also deleted due to poor exposure as the camera was initializing. This pre-processing method doesn't achieve a perfect instance to instance matching, but the error of capture time is typically within 1 second. The time error impact is reduced because the deep learning model loads 8 sequenced power data in a batch. The overall distribution of the total power value is plotted in Figure 1 histogram, which generally follows a normal distribution.

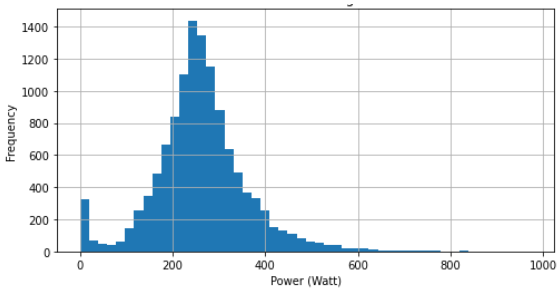

Figure 1 Power Data Histogram

The pre-processing of input images were done using OpenCV library. Images were firstly converted to a numpy array, and then cropped to the region of 52 to 752 pixel in width (removing left and right edge), and 281 to 480 pixel in height (remove top half). The remaining region (600 x 200 pixel) after cropping focus on the area where rover will traverse across in the following seconds. To feed the images with an appropriate aspect ratio for Convolutional layers, they are shrinked by half horizontally, resulting in a batch of 300 x200 pixels images. The last step of image pre-processing is normalizing images across the entire sample. A sample comparison of image preprocessing, the 538th image of Omni camera 5 in Run 3, is shown in Figure 2.

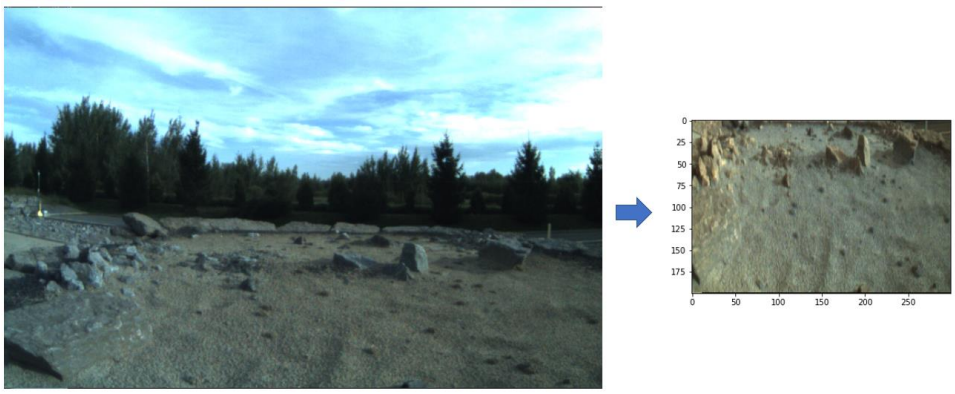

Figure 2 Image Pre-processing Sample

2.3 Deep Learning Model

The pre-processed image data and power data are fed into a novel, customized deep learning model for predicting sequential correlation between terrain images and power usage. The deep learning model consists of 3 sections: Transfer learning CNN based Xception architecture for visual feature extraction; Recurrent Neural Network (RNN) architecture for sequential relation analysis; Time Distributed wrapped output architecture for corelating several output predictions. The overall deep learning model structure used in this project is shown in Figure 3.

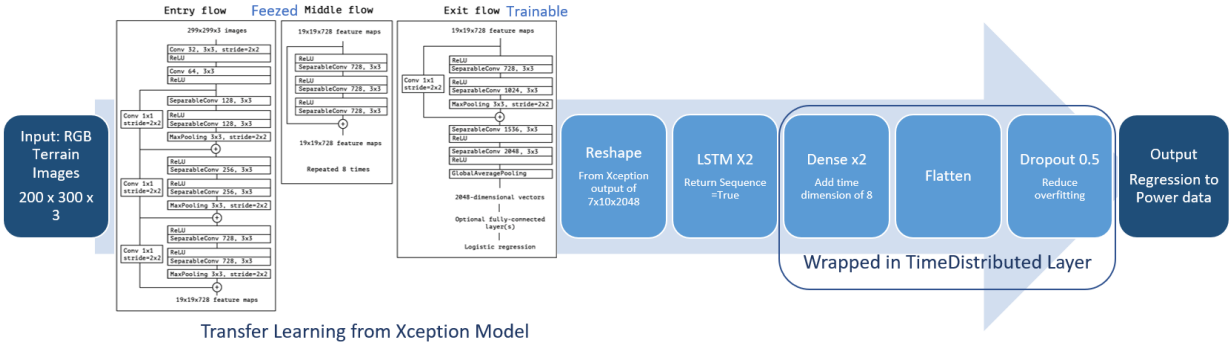

Figure 3 Deep Learning Model Structure for Terrain-Power Correlation Prediction

In the last decade, CNN models had tremendous growth in its capability and performance. The ILSVRC ImageNet is one of the most widely used benchmark to compare performance in object detection. The top 5 error rate for classification feel from 26% to 2.3% in just 6 years. This project took advantage of transfer learning which applied prior knowledge trained on the huge dataset of ImageNet(350 million images) to a relative smaller image dataset (~10K images). The transferred model used is the Xception architecture developed by Francois Chollet in 2016 [5]. This model's development is partially based on Inception V3 model, and merged the idea from GoogLeNet and ResNet. It outperformed the Inception V3 with top-5 classification accuracy of 94.5% and are relatively fast to train and require less memory with 22 million parameters (7 times less parameters than VGG16). Xception architecture uses depthwise separable convolution layer, and can separately model spatial patterns and cross-channel patterns. Among Xception's 126 total layers, the last 5 layers (in Exit Flow) are set as trainable in this project, which will update the weights and bias parameter based on the terrain images. The model structure of the Xception architecture is shown in Figure 4.

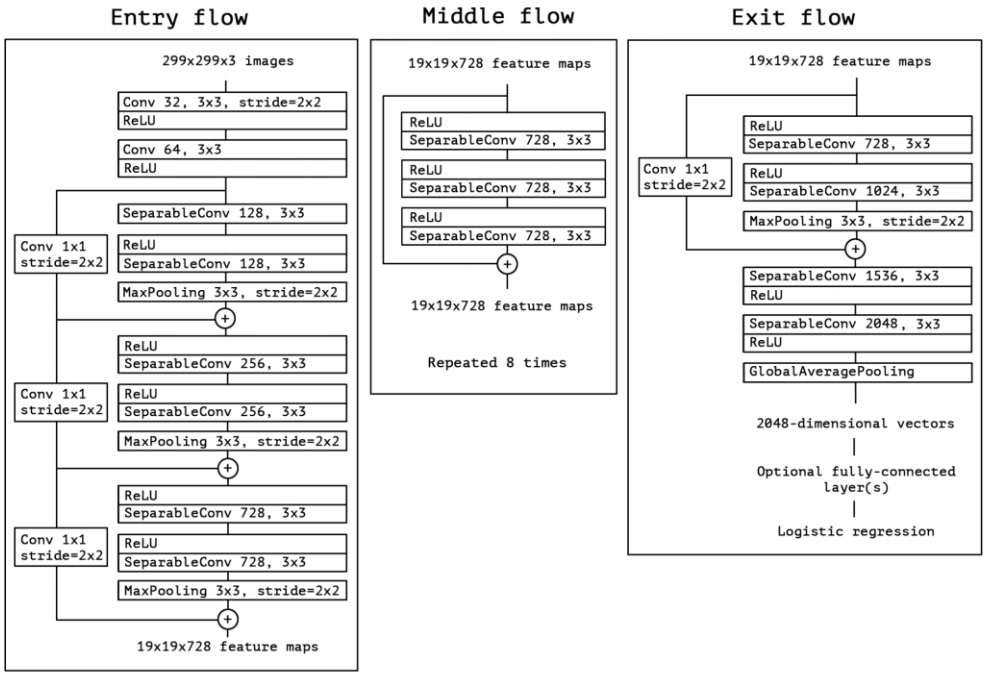

Figure 4 Xception Model Structure [5]

The challenge of only using CNN model in this project is that CNN can effectively extract features only on static image but not on dynamic information. Donahue Et al.'s constructed a Long-term Recurrent Convolutional Networks model (LR-CNs) that is able to accept time-varying inputs and outputs and learn temporal dynamics [6]. They added RNN, specifically Long Short-Term Memory (LSTM) layers, after the CNN layers. Inspired by their work, 2 LSTM layers are added after the Xception architecture to learn the sequential relation between input and output. A reshape layer was added between the Xception layers and LSTM layers to map the output shape.

We further wrapped TimeDistributed layers around the last several Dense layers, which added an additional time dimension to the original dense layers to handle the sequential output from LSTM layers. With the input tensor shape chosen as 8, the TimeDistributed wrapped Dense layer takes a batch of 8 inputs, and carries this additional time dimension of 8 through the following Dense layer, Flatten layer, Dropout layer and the final output Dense layer for regression prediction. With such construction of adding the time dimension, the sequence information trained from LSTM layers can be inherited to output layers. During forward propagation and backpropagation in training, the model processes varied length of previous inputs, and calculates the error based on a sequence of data.

2.4 Training Process

The original project proposal aimed to only use Run1 for training/validation/testing, but thanks to efficient model and powerful GPU, we were able to expand the scope of data and used all 6 runs in the CPET dataset. This project used Run # 1,3,4,5,6 as training set and Run #2 as the holdout test set. The proportion of the test set is 2133/14232 = 15%. Among the 12099 training samples, 20% are separated as the cross-validation set. The training is conducted in a mini batch size of 32, and used Mean Square Error(MSE) as loss function, and Adam optimizer with a finer learning rate of 0.0001. The training is run on a Laptop equipped with Nvidia RTX 2060 GPU, and the entire training of 12099 images took 17 minutes to reach optima.

To avoid over-fitting or under-fitting, the training code added *earlystopping* fucntion will monitor the MAE value on the cross validation set, and will stop training when the validation MAE no longer decreased for 3 continuous epochs. The *savebest* function will save the best performing model (lowest validation MAE) as a Keras model file during training. The best performing model is at the 10^{th} epoch with a validation MAE of 67.71. The learning curve with MAE metric is shown in Figure 5 .

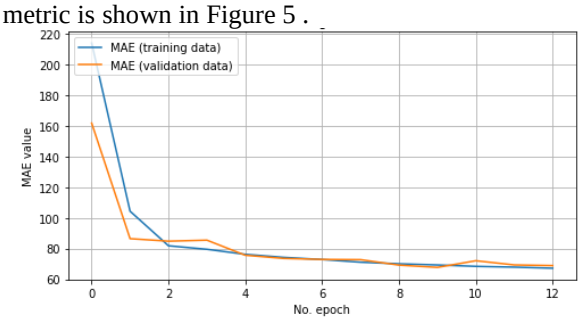

Figure 5 Model Training Learning Curve

The code implementation in this project used the following Python library packages: Numpy, Tensorflow, Keras, OpenCV, Matplotlib. The submitted code in *rob501_project.py* and *Pre_process.py* are relatively short as they only contain the pre-processing and evaluation functions for testset. The full code with training functions is written in *Xception_LSTM_TimeDistributed-Model.ipynb*, which include sample and training outputs for reader-friendliness. A HTML format of the IPython notebook is also saved for reading, in case there is difficulty opening the ipynb.

3 Project Evaluation and Result

3.1 Evaluation Criteria

As defined in project proposal, Mean Absolute Error (MAE) is chosen as the metric to evaluate the performance of the model, which measures the average difference between the predicted power and actual power consumption. The loss function uses MSE function, which significantly penalizes larger error value to help the model converge faster. However MAE is a better candidate than MSE as the final evaluation metric, as it can be directly interpreted as real world engineering unit: Watts of rover power consumption. As there is no prior benchmark that directly corelates terrain images with power output, we defined the target performance as having an error within 25% from the ground truth average power. The testset of Run2 has an average total

power of 257.40 Watt, and the goal is to reach a MAE within 64.35.

3.2 Evaluation Results

The best model at Epoch 10 is saved in the src folder and packaged as part of the submission. The evaluation code loaded the saved model, and generated prediction of power data based on the 2158 input images in Run 2. The MAE calculate on 2158 samples of Run 2 is 63.28, which reached the target of MAE smaller than 64.35. This is equivalent to a 24.5% deviation from the average power. To better understand the data distribution and predictive model behavior, Run2's prediction compared with ground truth are plotted in Figure 6. The red color predictions generally follow the trending well, especially when the actual power value is close to the mean. However it doesn't predict well when the power is extremely high or low, which overlapped with the region of steep slope in Run2. Adding slope information oto the model is a very promising next step for future project.

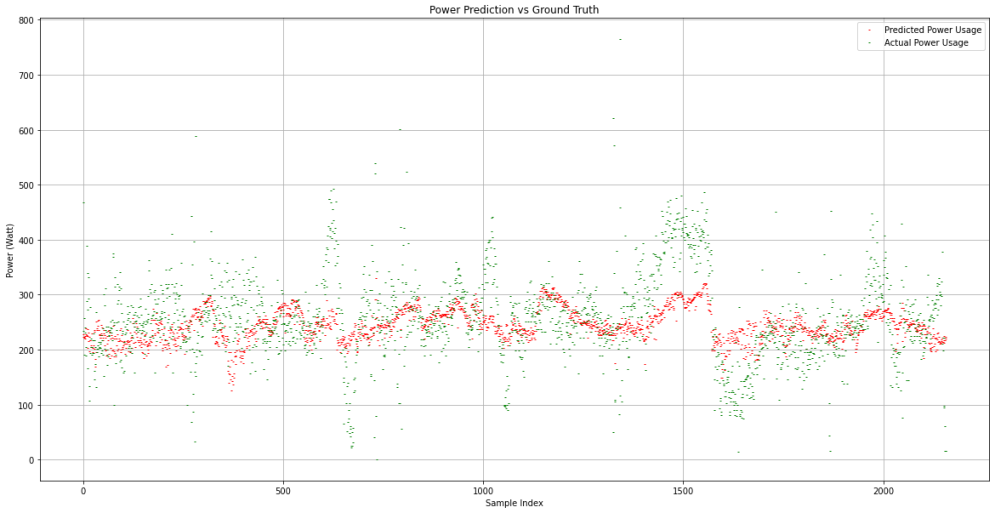

Figure 6 Run 2 Actual Power Data vs Model Prediction

This final evaluation result came after testing several deep learning models, as previous attempts are reocrded in Table 1.

Table 1 Model Performance Improvement History

Models Structures	Validation MAE	Implementation Details
3 CNN layers + 2 LSTM + Dense	136.1	Constructed and trained model from scratch
VGG16+ Dense layers	99.7	Last 4 layers are trainable
VGG16+ 2 LSTM +Dense	89.4	Add LSTM after VGG
VGG16+ 2 LSTM + TimeDistributed Dense	76.1	Wrapped dense layers with TimeDistributed layers
Xception+ 2 LSTM + TimeDistributed Dense	67.7	Change VGG model to Xception

3.3 Result Discussion

Overall, the deep learning method proposed in this project achieved an acceptable result of MAE 63.28 as a proof of concept study. The result is very close to the initial target performance in project proposal, and it wasn't a surprise for us. Machine learning model often tends to predict value closer to the mean and median value of the samples to minimize the error value, as shown in Figure 6. On the promising side, the model is able to handle flexible storing and processing that is especially useful in this project, as the robot will traverse through different portion of images at a variable speed. The instant power draw is actually representing the terrains photo taken several frames ago. The mobile robot's PID controller applied additional temporal complexity and delay to the correlation between terrain surface roughness and power draw. With these challenges, it's almost an impossible task to explicit a computer vision model that extract terrain image information and predict a power consumption value.

These challenge are mitigated with the addition of LSTM recurrent neural network layers and TimeDistributed wrapped layers, which significantly improved the performance as demonstrated in Table 1.
There are 3 major challenge of analyzing terrain solely with computer vision approach. Firstly, terrain image doesn't have distinctive geometry features for algorithm to recognize the terrain type. When we initially constructed a deep learning model using CNN layers with common parameters such as layer numbers, kernel size and stride, the model had a poor performance of 136.1 MAE, doubling the final result. Terrains like sands don't have distinctive geometry features such as lines, edges, corners, and are thus difficult for computer vision to differentiate.

Meanwhile, the power consumption are affected by other factors, especially wheel slip and slope of the terrain. Unlike the first challenge, these two challenges can be potentially solved with additional engineering which can be scope for future works. One solution for wheel slip detection is constructing a visual odometry system and compare its value with the wheel encoder. The terrain slope information is also available in the multi-sensor CPET dataset, as it can be calculated as a quaternion orientation angle. The slope data can then be trained together with previous deep learning model's output, and form another simple multi-layer perceptron models with just 2 inputs to predict the output of power consumption.

3.4 Conclusion

In the final project of ROB 501 Computer Vision for Robotics, a novel deep learning model is constructed to predict the power consumption of a simulated Mars Rover driving over Mars Emulation Terrain. The CPET dataset provides an energy-awareness dataset with various sensors' input. The learning model used a 68%/17%/15% training/ validation/ testing split, and trained on 12099 RGB images collected from onboard Omni Camera #5. The final performance on the holdout testset Run2 data achieves a Mean Absolute Error of 63.28, which is 24.5% deviation from actual average power of 257.40 Watt. The model performance can be further improved by adding slope data into the model and wheel slip detection, as well as better pre-processing. Overall the final project completed scope of work listed in the project proposal and achieved performance target at evaluation. This proof of concept study will hopefully provide insight to improve rovers' energy awareness and path planning.

References

[1] O. L. F. M. J. K. Olivier Lamarre, "The Canadian Planetary Emulation Terrain Energy-Aware Rover Navigation Dataset," *The International Journal of Robotics Research,* vol. 39, no. 6, pp. 641-650, 2020.

[2] J. P. R. K. M. O. M. H. C. Brandon Rothrock, "SPOC: Deep Learning-based Terrain Classification for Mars Rover Missions," in *AIAA Space Forum*, Long Beach, California, 2016.

[3] A. Z. Karen Simonyan, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations (ICLR)* , San Diego, 2015.

[4] T. Manderson, S. Wapnick, D. Meger and G. Dudek, "Learning to Drive Off Road on Smooth Terrain in Unstructured Environments Using an On-Board Camera and Sparse Aerial Images," in *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020.

[5] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017.

[6] L. A. H. M. R. S. V. S. G. S. T. D. Jeff Donahue, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, no. 4, pp. 677-691, 2017.