Robotics Image Classification Inference

Amay Yadav

Abstract—Robots are automated systems with moving parts and they need to infer their surrounding and detect and classify objects in real-time from vision systems. This paper presents a multi-functional neural net model for real-time object detection and classification for use on resource-limited hardware systems, such as the Nvidia Jetson platform. This paper will display the findings of running the neural net for image classification on two different datasets and evaluating it's performance.

Index Terms—Robot, IEEEtran, Udacity, LaTeX, deep learning.

1 Introduction

The Real-time inference of vision systems in robotics has recently acquired significant prominence in self-driving cars and drones. These systems require fast and accurate interpretations of the vision data as delays and inconsistencies can have fatal consequences. With robotic applications becoming more advanced and no longer being confined to controllable laboratory environments, robots now require self-contained on-board systems that are light, reliable and fast. Recent advancements in both hardware and model architectures have decreased the processing time of a model while simultaneously increasing the accuracy of the predictions. Current state of the art systems involve large computer hardware to perform time-critical image classification, but as robots move into more complex and remote environments, network communications can be limited as well as their size and shape limitations can limit the on-board systems. This paper will explore robust methods for training a network-based inference model that can be deployed remotely onto a Nvidia Jetson TX2 embedded platform that provides above 75% classification accuracy at over 100Hz update rate.

There are two parts of this image classification project:

- P1 Data is the supplied images of candy boxes, belts and nothing (empty conveyor belt).
- Acquired images of different breeds of dogs.

2 Background / Formulation

The LeNet is a seven-layer convolutional neural network (CNN) that takes handwritten numbers and outputs a probability map for each number class. The AlexNet is an extension of the LeNet principle and it was the first deep neural network (DNN) to win the ImageNet classification challenge, with a top-5 error of 15.4%. Current approaches have improved further on object classification in computer vision by employing various forms of the CNN architecture out of which the most notable of are ResNet, GoogLeNet [1] and VGG.

2.1 Hardware and GPU analysis

One of the most primary factors that we need to consider while choosing the neural net is the performance as we

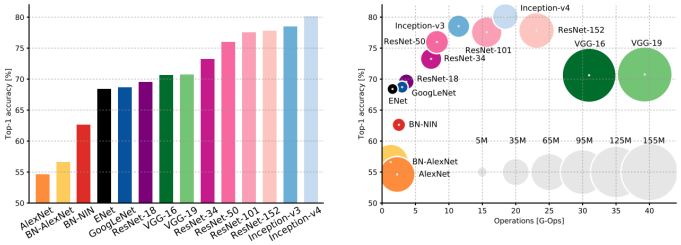

Fig. 1. Accuracy Graphs for different neural net models. [2]

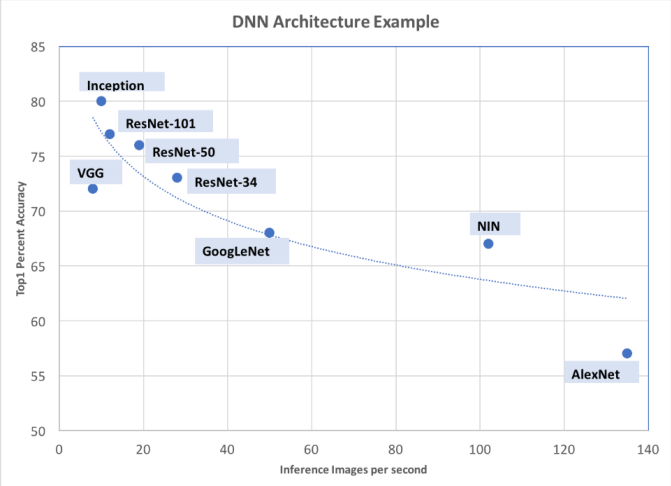

Fig. 2. Accuracy % Graphs for different neural net models. [3]

are constrained by GPU and limited hardware and time. The improvements on the aforementioned neural nets has definitely seen an increase in accuracy but at the expense of computational resources and time. A computational analysis of state-of-the-art DNNs, trained on ImageNet and deployed on a Jetson TX1[6], compared the top1 accuracy, operations, parameters, inference time, power and memory usage. The following table and graph demonstrate the comparison between the different neural nets.

Nvidia has released the TX2 platform that has increased clock speed a whole lot more and also decreased the power

DNN	Top1 Accuracy	Operations	Parameter	Inference	Power	Memory	Info Density
Network name	%	G-Ops	M	fps	W	MB	%accuracy/ Mparams
AlexNet-BN	57	2	60	50	10.9	310	1.0
GoogleNet	68	3	7	33	10.7	200	9.7
VGG-16	71	31	135	6	11.8	850	0.5
ResNet-101	76	16	45	10	12.9	300	1.7
Inception-v3	78	12	25	12	11.6	200	3.1

Sampling of metrics for architectures running ImageNet inference on a Jetson TX1

Fig. 3. Power conusmption table for different neural net models. [3]

consumption by 30%. Furthermore, a second improvement has been made on the software in question i.e. the latest JetPack 3.1 and associated TensorRT 2.1 increases the processing speed by approximately twofold. Please refer to the diagrams below for reference.

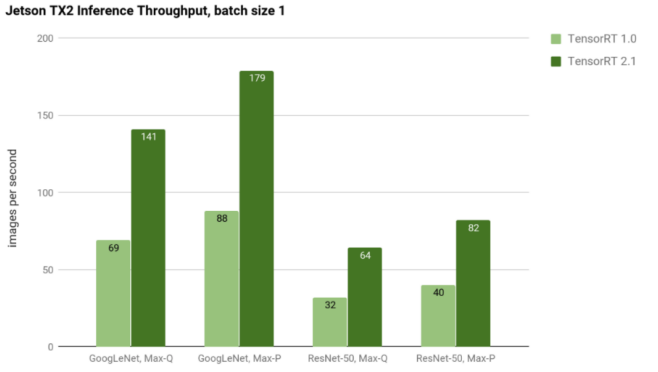

Figure 2: Inference throughput of GoogLeNet and ResNet-50 using Jetson TX2 Max-Q and Max-P power profiles. TensorRT 2.1 provides twice the inference throughput on GoogleLeNet and ResNet.

Fig. 4. Jetson 1 vs Jetson 2 comparison on different GoogLeNet models. [3]

2.2 Neural Net Model

As you can notice from the above charts and graphs that GoogLeNet shows the most amount of promise in object detection and classification on limited hardware and there's studies proving that. Hence, the chosen model in this paper is the GoogLeNet and we'll closely study it's effectiveness towards real-time image analysis on two different datasets referenced above.

DNN.png

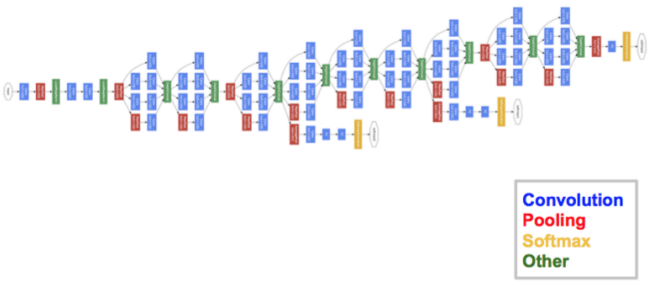

Fig. 5. GoogLeNet network with all the bells and whistles. [1]

A lot of software is built around business problem considerations and we must always strive for spending as little as possible to come up with a minimum viable product so that extra resources can be used for further scaling issues. In this case we just have a single AWS account with some GPU time or otherwise spend more money on more bulkier hardware. Hence, the resources are limited and the business problem is to infer the objects on a conveyor belt which means we need to classify the item when it's passing by and the minimum accuracy required is 75%. The GoogLeNet is well-suited for the inference model because it offers the smallest resource utilization with a high inference rate that has comparable accuracy to other resource intense neural net.

2.3 Parameters and Training

The Digits platform has the different neural nets like LeNet and GoogLetNet already loaded into it. I just used the GoogLeNet as it came with 30 epochs and learning rate of 0.01 which was divided by 10 after every 10th epoch. 25% of the data was used as the validation set in both the datasets. All of the other parameters are defaults provided by the Digits platform.

3 Data Acquisition

The first data set is already provided in the project and it contains candy boxes, belts and nothing. Nothing are just blank images with no objects i.e. the conveyor belt with no items in the view of the camera. This dataset contains 10094 RGB images which are squashed during creating the dataset as depicted in the image below. The image size is 256x256 which is ideal for this experiment.

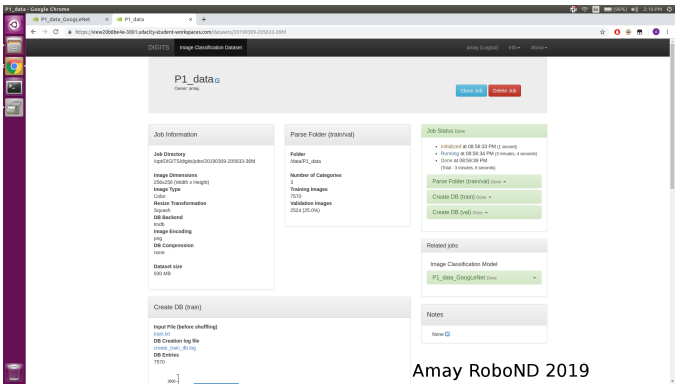

Fig. 6. P1 dataset on Digits.

The other dataset are the images of different dog breeds. I got this from an older project of mine https://github.com/Amay22/Deep-Learning-Dog-Classifier. I selected 8 different breeds of dogs are demonstrated in the images below namely Akita, Beagle, Bulldog, German shepherd, dog, Golden retriever, Great dane, Greyhound and Poodle. There are around 60 images per dog breed. This dataset contains 559 RGB images which are squashed during creating the dataset as depicted in the image below. The image size is 256x256 which is ideal for this experiment.

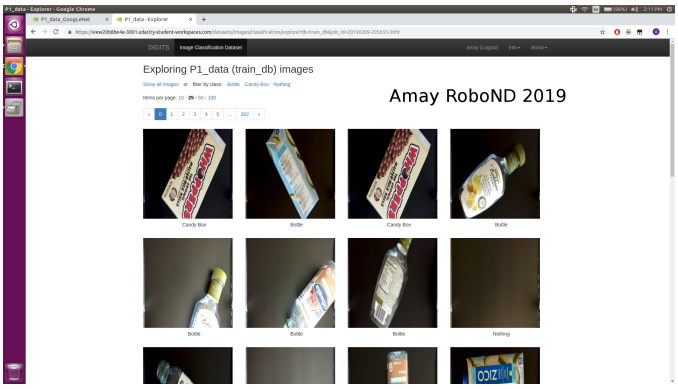

Fig. 7. P1 dataset candy boxes and bottles.

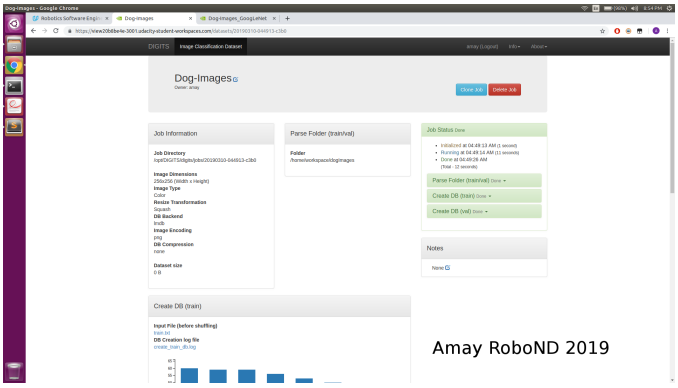

Fig. 8. Dog Breed dataset on Digits.

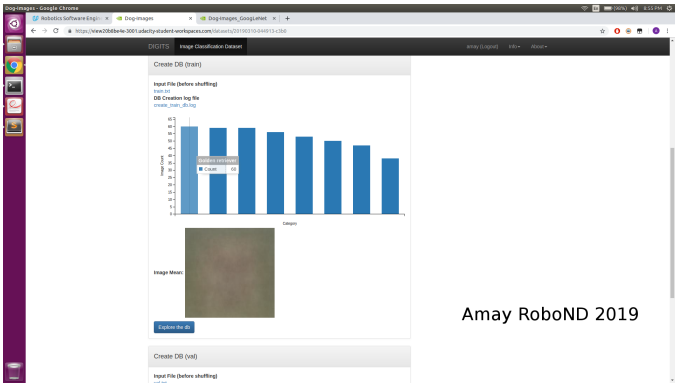

Fig. 9. Dog Breed dataset on Digits.

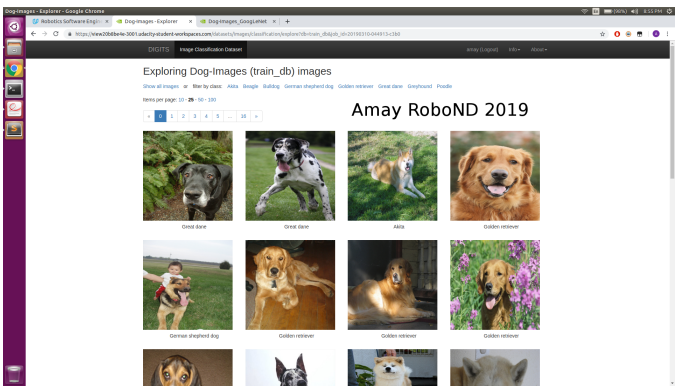

Fig. 10. Dog Breeds for Image Classification.

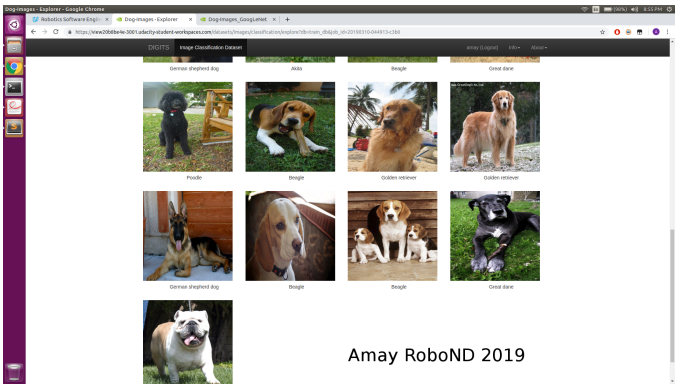

Fig. 11. Dog Breeds for Image Classification.

4 Results

4.1 P1 Dataset Result

The bottle-candy box dataset contains 10094 labelled images of candy boxes, bottles and nothing. The data has been divided into 7570 images for training and 2524 images for validation. The results of the training obtained an overall accuracy of 99.98%. The model was trained for 30 epochs but we the accuracy has reached 95% for top-5 values within the first 5 epochs.

The model's inference speed is averaged over five batches of 10 runs of data. The max inference speed was 5.62901 ms with an accuracy of 75.4098360656%.

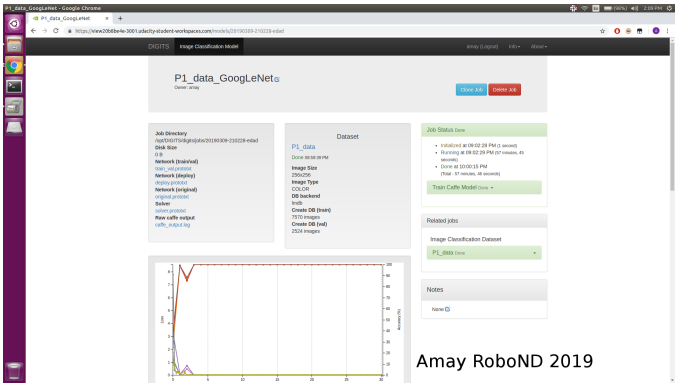

Fig. 12. P1 Data GoogLeNet Model running on Digits.

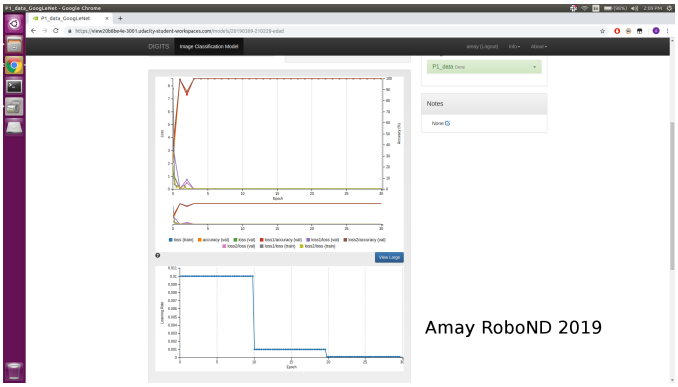

Fig. 13. P1 Data GoogLeNet Model Accuracy.

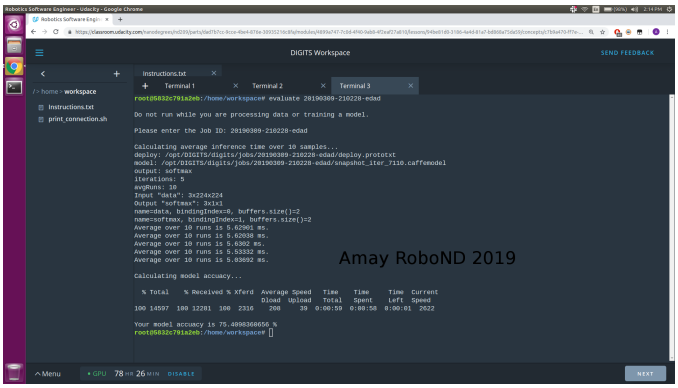

Fig. 14. P1 Data evaluate command result.

4.2 Dog Breed Dataset Result

The Dog breed dataset contains 559 images for 8 different dog breeds. The data has been divided into 442 images for training and 137 images for validation. The results of the training obtained a top-5 validation accuracy of 86.8056%. The overall accuracy was poor but that was expected due to low number of images for each dog breed and no real hyerparameter tuning. I tuned the learning rate to be 0.5 and that did show better results but but then again due to lack of images that wasn't very high.

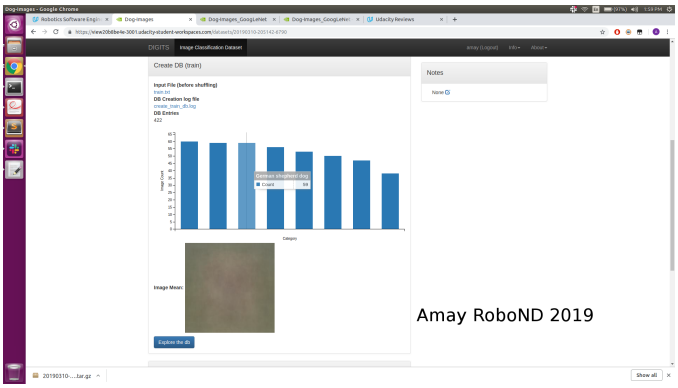

Fig. 15. Robot Revolution.

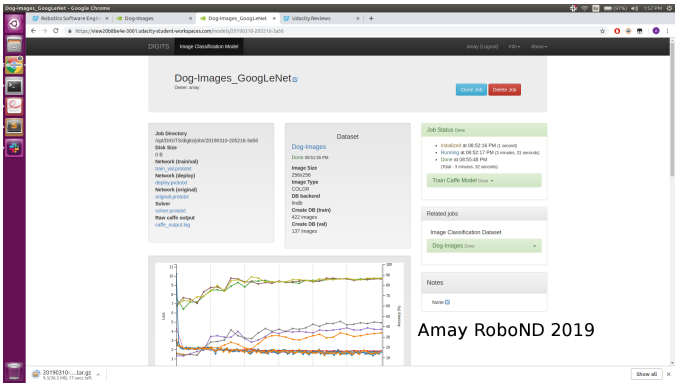

Fig. 16. Robot Revolution.

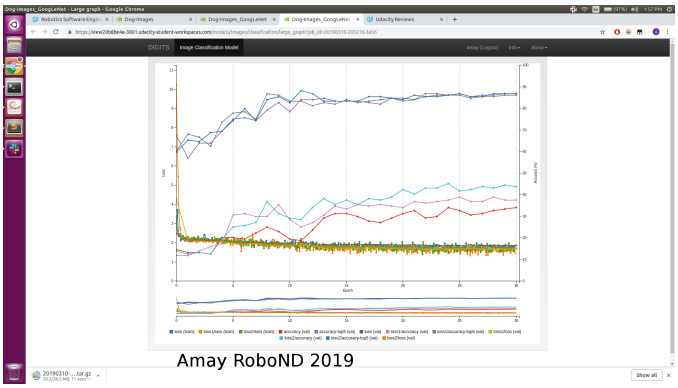

Fig. 17. Robot Revolution.

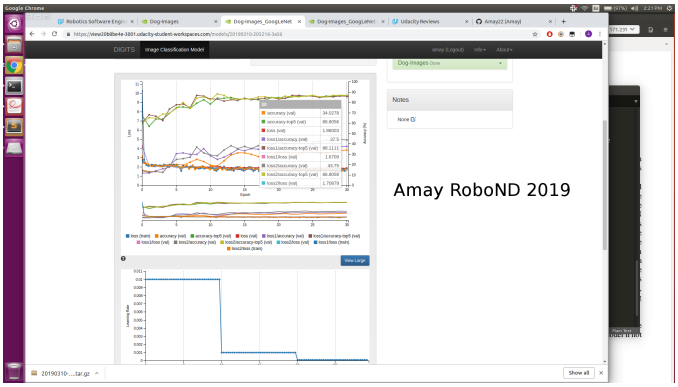

Fig. 18. Robot Revolution.

5 Discussion

5.1 P1 Dataset Result Discussion

The results for P1 dataset is incredible and it proves that with correct training and a sufficiently large dataset, image classification can be achieved in real-time. Moreover, we have proves that embedded hardware with complex CNN architectures is well-suited for this kind of problem. The model has successfully demonstrated this with the bottle, candy box and nothing dataset. 75% accuracy is pretty good I am sure this can be increased by changing the hyperparameters and adding more images and some data preprocessing.

5.2 Dog Breed Result Discussion

The dog breed dataset has shown how image classification can improve. We have experimented on this dataset earlier using VGG model and it performed very well but in this case we just used default Digits hyperparameter values with no real image preprocessing. Moreover, the images were squashed and the number of images are most certainly lesser than the supplied dataset.Based on the top-5 accuracy of 86% it has shown that despite limited data, less than 70 per class, placed in a wide variety of environments, it can still produce remarkable results with only 30 epochs of training. The overall accuracy is at a staggering low of 34%. There are certain issues in the entire process of the dog breed dataset and they are listed down as follows:

- The dog breed dataset has very less number of images. Also the images have objects in the background leading to a lot of noise. The extreme variation of colors can hamper classification heavily. A small dataset of a class can have substantial variation in colours, backgrounds, positioning, texture, shape and size. This is a major contributing for the low inference accuracy.
- Dog Breed images don't have consistent background environments that can cause the model to learn the environment rather than the desired object. Images can contain multiple classes as well.
- Large images with a smaller sized object can lose the detail of the object when resized to 256x256, causing miss-classification.

We can improve the accuracy of this model by adding more images to the dataset and then preprocessing the images to remove some of the background objects. We can also tune the hyperparameters like increase the learning rate and increasing the epoch. The 34% accuracy can be increase by a lot more if we take these steps.

6 Conclusion / Future work

The paper explores the problem statement of Robotics Inference on Image Classification Neural Net model advancements to train and accurately classify small user-defined datasets. We have also discussed various CNN architectures and compared it's accuracy and speed, the GoogLeNet architecture was used to produce an accuracy of 75% at an inference speed of 189 fps. The network was trained on two different datasets. The dataset provided had a lot images and fewer classes and hence it was extremely effective; not to mention the images with objects had black background making it simpler to train. Whereas the supplied dataset with the dog breeds was smaller and it's not easy to classify one dog from the other and the background also had different things like the ground or people which added too much for the GoogLeNet to handle and resulted in not so good accuracy. Some preprocessing of the images would have helped but the results for top-5 accuracy was spectacular.

The future work for this project would be to gather better dataset and then preprocess the images for specific objects only and then train the model effectively with hyperparameter tuning. The results are promising and if we apply some more techniques of Computer vision we can attain better results probably with even lesser hardware.

References

[1] Y. J. P. S. S. R. D. A. D. E. V. V. A. R. Christian Szegedy, Wei Liu, *Going deeper with convolutions*. arXiv:1409.4842, 2014.
[2] E. C. Alfredo Canziani, Adam Paszke, *An analysis of deep neural network models for practical applications*. arXiv:1605.07678, 2017.
[3] *RoboND Nvidia Digits Inference lecture*.