Indian Driving Dataset – A Multi-label Problem

Individual Final Report

Author: Ishan Kuchroo

12/13/2022

Overview of Project

Semantic image segmentation is a computer vision task in which we label specific regions of an image according to what's being shown. Mostly used in Autonomous vehicles and Medical Image diagnostics, the goal of semantic image segmentation is to label each pixel of an image with a corresponding class of what is being represented. Because we're predicting for every pixel in the image, this task is commonly referred to as dense prediction. One important thing to note is that we're not separating instances of the same class: we only care about the category of each pixel. In other words, if you have two objects of the same category in your input image, the segmentation map does not inherently distinguish these as separate objects.

IDD is a dataset for road scene understanding in unstructured environments used for semantic segmentation and object detection for autonomous driving.

Roles and Responsibility

Team Member	Area of Work	Shared Responsibility
Varun Shah	Model exploration	Fine-tuning
Hemangi Kinger	Model interpretation	Model exploration
Ishan Kuchroo	Data Preprocessing and Fine-tuning	Model interpretation

What is my responsibility?

I have taken the primary responsibility of data collection and preprocessing for the Indian Driving Dataset. I'll be using the processed data and proceed as follows":

- Apply image transformations
- Apply different pre-trained models and fine-tune them

In addition to this:

- I'll be proof-reading and making changes in the summary report created by team
- Consolidating the code of data-preprocessing and modelling and creating a pipeline to ensure the code runs smoothly.

Data Preprocessing

Overview:

IDD consists of images, finely annotated with 16 classes collected from 182 drive sequences on Indian roads. The label set is expanded in comparison to popular benchmarks such as Cityscapes, to account for new classes. It also reflects label distributions of road scenes significantly different from existing datasets, with most classes displaying greater within-class diversity. Consistent with real driving behaviors, it also identifies new classes such as drivable areas besides the road. The dataset is inspired from the one used in the 2018 paper IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments

Source:

http://idd.insaan.iiit.ac.in/dataset/details/

1. **Split the data into training and validation sets**:

   The dataset of 9860 images has been split into train, validation and test sets with each set having 6,991, 1912 and 957 images respectively.

2. **Convert labels to target class**

3. **Image Transformation**:

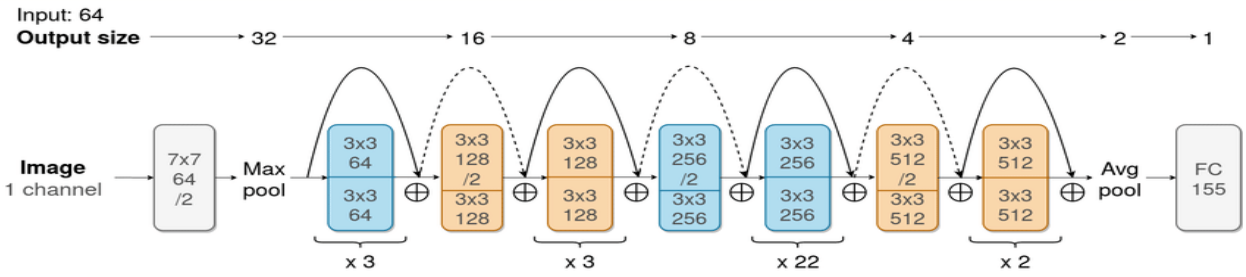
   Using cv2 and torch vision library, we've transformed the images as follows:

   - Resize into 480, 480
   - Normalize image (mean=[0.485, 0.456, 0.406] and std=[0.229, 0.224, 0.225])
   - Rotate image

Model Training and Fine-Tuning

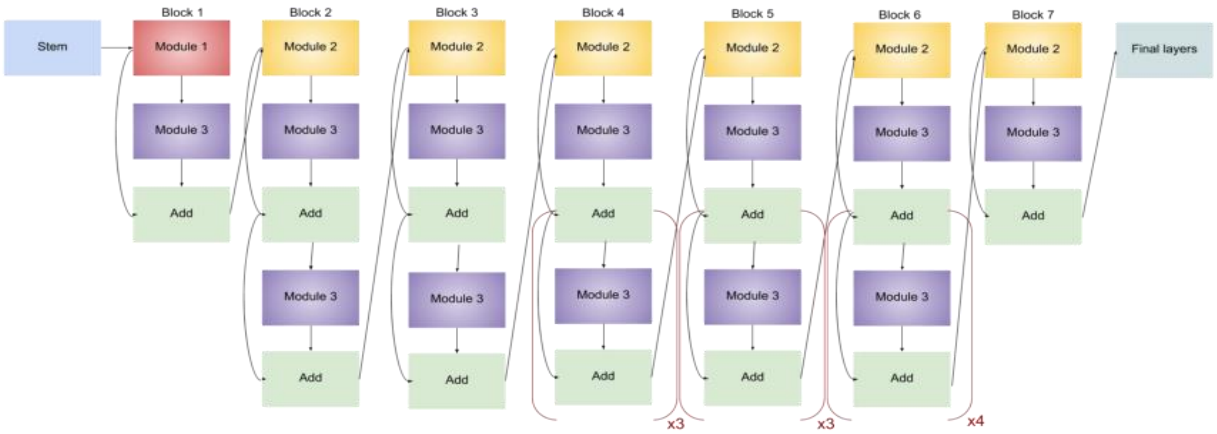
Multiple models were trained and fine-tuned (including AlexNet, VGG16, NFNet, MobileNet, ShuffleNet, PNASNet etc.) but here we'll talk about out top 3 networks (based on accuracy).

1. **ResNet-101**:
   ResNet-18 is a convolutional neural network that is 101 layers deep. Because of being pre-trained, the network has learned rich feature representations for a wide range of images.



2. **EfficientNet-B3:**

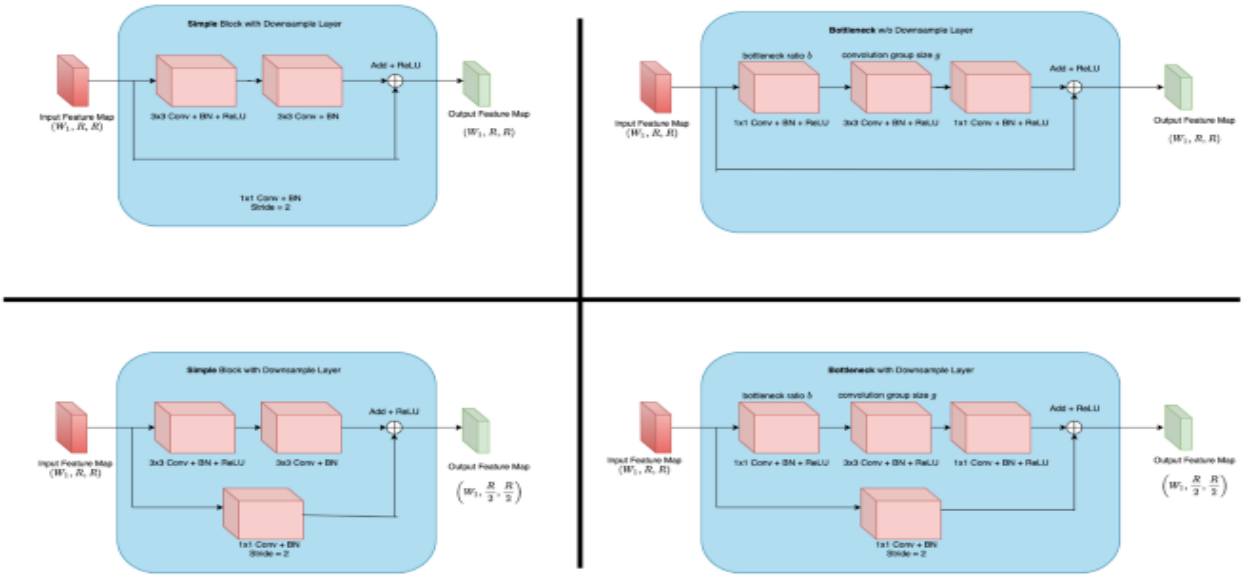
   Efficient Net is a convolutional neural network architecture and scaling method that uniformly scales all dimensions of depth, width, or resolution using a compound coefficient. The compound scaling method is justified by the intuition that if the input image is bigger, then the network needs more layers to increase the receptive field and more channels to capture more fine-grained patterns on the bigger image.



3. **RegNet**:

   Reg-Net is not a single architecture, it is a design space defined as a regulatory module for ResNet. Based on concept of using a Bottleneck Residual Block

   - A variant of the residual block that utilizes 1x1 convolutions to create a bottleneck.
   - Reduces the number of parameters and matrix multiplications to increase depth



RESULTS

EPOCH = 40, Learning Rate = 0.0001, Optimizer = Adam, Batch Size = 32, Image Size = 224					
Change No.	Change Details	Test_Accuracy	hlm	Sum_Metric	Keep Changes
1.1	ResNet101	0.48326	0.06629	0.54956	Yes
1.2	AlexNet	0.06067	0.14697	0.20764	No
1.3	VGG16	0.07531	0.14769	0.223	No
1.4	EfficientNet_B3	0.4662	0.06681	0.54171	Yes
1.5	MobileNet_V3	0.38	0.07538	0.45822	No
1.6	MobileNet_V2	0.42	0.085	0.505	No
1.7	ResNet152	0.44	0.07048	0.51085	No
1.8	NFNet	0.33	0.09336	0.42286	No
1.9	RegNet_x_800mf	0.45534	0.06623	0.53694	No
2	RegNet_y_800mf (epochs = 40)	0.43	0.06956	0.50366	No
2.1	RegNet_x_400mf (epochs = 40)	0.4341	0.06832	0.50242	No
2.2	MLMIxer	0.25	0.11199	0.36617	No
2.3	ViT	0.06	0.1607	0.21613	No

Conclusion

From my analysis of different neural networks, we can conclude that ResNet module performs the best on our data.

Referenced Code %

(534 – 224)/ 534 + 112 * 100 = **47%**

References

https://machinelearningmastery.com/how-to-implement-major-architecture-innovations-for-convolutional-neural-networks/

https://github.com/rishikksh20/MLP-Mixer-pytorch/blob/master/mlp-mixer.py

https://towardsdatascience.com/self-driving-car-on-indian-roads-4e305cb04198

https://medium.com/mlearning-ai/vision-transformers-from-scratch-pytorch-a-step-by-step-guide-96c3313c2e0c

https://github.com/labmlai/annotated_deep_learning_paper_implementations

https://rwightman.github.io/pytorch-image-models/models/vision-transformer/

https://github.com/BrianPulfer/PapersReimplementations/blob/main/vit/vit_torch.py

https://medium.com/the-owl/imbalanced-multilabel-image-classification-using-keras-fbd8c60d7a4b