IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 08, Issue 9 (September. 2018), ||V (V) || PP 47-51

OpenPOWER Architecture: A Case Study on Semantic Segmentation using ENet Model

Prisilla J[1], Iyyanki V Murali Krishna[2]
[1]*Research Scholar, Hyderabad, India*
[2]*Former Professor and Director R&D JNTUH, India. iyyanki@gmail.com*
Corresponding Author: Prisilla J

Abstract: Semantic segmentation is labelling each pixel for which an image or a class belongs to. The ability to perform pixel-wise semantic segmentation in real-time is of extract more detailed information of the image. Learning about thousands of objects from millions of images, a model with a large learning capacity is required. In this paper, a deep neural network model named ENet is discussed as efficient neural network and for tasks precisely requires low inference value. ENet is nearly 18 times faster, has 79 times less parameters, and provides better accuracy to existing models. The ENet model on OpenPOWER architecture would provide much better results with less speed throughput on the CPU / GPU.

Keywords: OpenPOWER, deep learning, ENet, semantic segmentation

---

I. INTRODUCTION

Deep learning, a multi-layer neural network, has modernized computer vision, and continues to revolutionize IT due to availability of rich data sets, new accelerating neural network training techniques and fast hardware with GPU accelerators. Few machine and deep learning applications can identify high value sales opportunities, detect and react to intrusion or fraud, and suggest solutions to technical or business problems. Despite of using several GPUs in a single server, one computation run of deep learning software can take few days, or sometimes weeks, to run. The OpenPOWER architecture, range of servers thrives on this kind of compute intensity. The POWER processor has much higher compute density than x86 CPUs probably up to 192 virtual cores per CPU socket in Power8.

POWER8, The very best services for Deep Learning

One of the applications of OpenPOWER is the high-performance POWER8, ideal for deep learning and machine learning with large caches, 2x-3x higher memory bandwidth, very high I/O bandwidth and tight integration with GPU accelerators. The high memory POWER8's multi-threaded architecture and I/O bandwidth ensuring that GPUs are used to their fullest potential.

Open Source Biometric Recognition

A biometrics framework supports the development of open algorithms and reproducible evaluations. The face recognition in OpenBR is implemented by the 4SF algorithm. The complete OpenBR implements a complete evaluation harness for evaluating face recognition, face detection, and facial land marking. A consistence environment for the repeatable evaluation of algorithms for the academic and open source communities is available.

OpenPOWER optimizer frameworks like Caffe, Torch and Theano are OpenPOWER pre-built binaries optimized for GPU acceleration. The OpenPOWER management tool is based on open source products, enabling OpenPOWER nodes in a private or public cloud.

The most popular and the fastest growing area in Deep Learning is Computer Vision. Using OpenCV 3.1, the DNN module in the library implements forward pass with deep layer networks, pre-trained using certain deep learning frameworks, such as Caffe. In OpenCV 3.3 the module has been promoted from opencv_contrib repository to the main repository and has been accelerated meaningfully.

One of the powerful tasks in the field of computer vision is Semantic Segmentation. The aim of semantic segmentation is to label each pixel for which an object or class it belongs to. Semantic segmentation can come in with different settings, when the requirement is only to label each pixel according to class; it is very accurately described by pixel-level labeling or pixel-wise classification.

II. PREVIOUS WORK

FCN-32

Yen-Kai Huang (2017) proposed a FCN model with many layers of convolution on the image to extract a multiscale feature representation of the image, with the dimension (Hi; Wi; Ci), where Ci is the number of channels or kernels. The layers are stacked from shallow layers to deep. The dimensions of shallower layers have similar size of original image, while the deeper layers will be much smaller in height and width but have many more channels. The architecture of FCN-32 indicates that in the pipeline it performs five max-pooling to reduce the size of image by 1=32. The architecture is very efficient to compute but can result in losing fine-grained features. The output shows the pre-trained model validity, which allows attempting transfer learning based on this model [1].

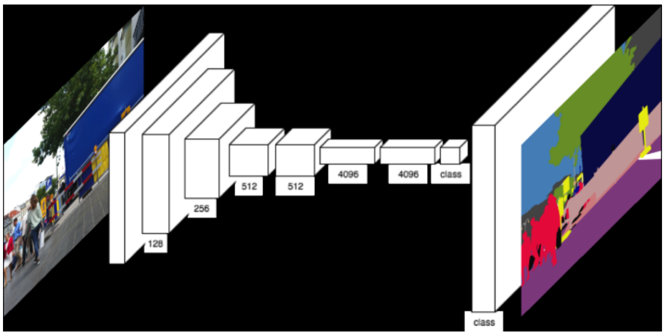


Figure 1: *Fully Convolutional Network -32*

AlexNet

Alex Krizhevsky et. al (2012) proposed a large, deep convolutional neural network that was trained to classify the 1.3 million high-resolution images from ImageNet training set into the 1000 different classes in the LSVRC-2010. The test data achieved top-1 and top-5 error rates of 39.7% and 18.9% respectively much better than the past results. The neural network consisting of 60 million parameters and 500,000 neurons, five convolutional layers, followed by max-pooling layers, and two globally connected layers with a final 1000-way softmax. Non-saturating neurons and a very efficient GPU implementation of convolutional nets were used to make the training. The network's input is 150, 528-dimensional, and the number of neurons in the network's remaining layers is specified by 253, 440–186, 624–64, 896–64, 896–43, 264– 4096–4096–1000 [2].

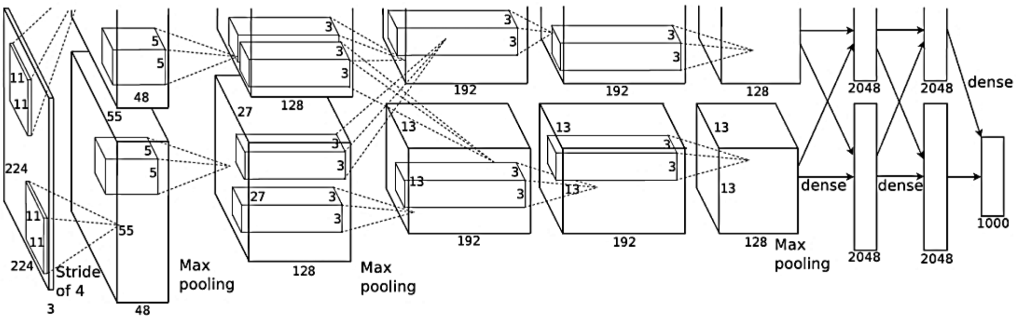


Figure 2: *AlexNet Architecture*

ResNet

Kaiming He et al. (2016) proposed residual networks that were optimize easier, and can gain accuracy from increased depth. On the ImageNet dataset evaluate residual nets with a depth of up to 152 layers—8 times deeper than VGG nets but still having lower complexity. The residual net achieved 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task and also presented analysis on CIFAR-10 with 100 and 1000 layers.

In a deep residual learning framework, desired mapping H(x), the stacked nonlinear layers fit mapping of F(x) = H(x) − x. The original mapping is reorganized into F(x) + x. It is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. If an identity mapping were optimal, it would be easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers. The formulation of F(x) + x can be realized by feedforward neural networks with shortcut connections, the shortcut connections are those skipping one or more layers and referred as skip connections. Identity shortcut connections add neither extra parameter nor computational complexity. The entire network is trained end-to-end by SGD with backpropagation, and implemented using common libraries (Caffe) without the solvers modification [3].

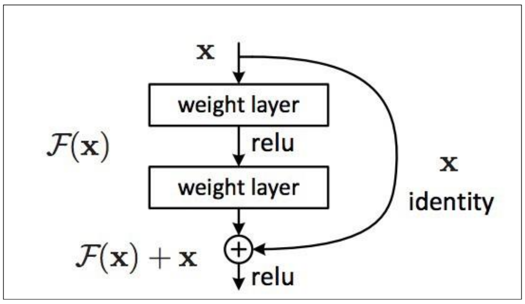
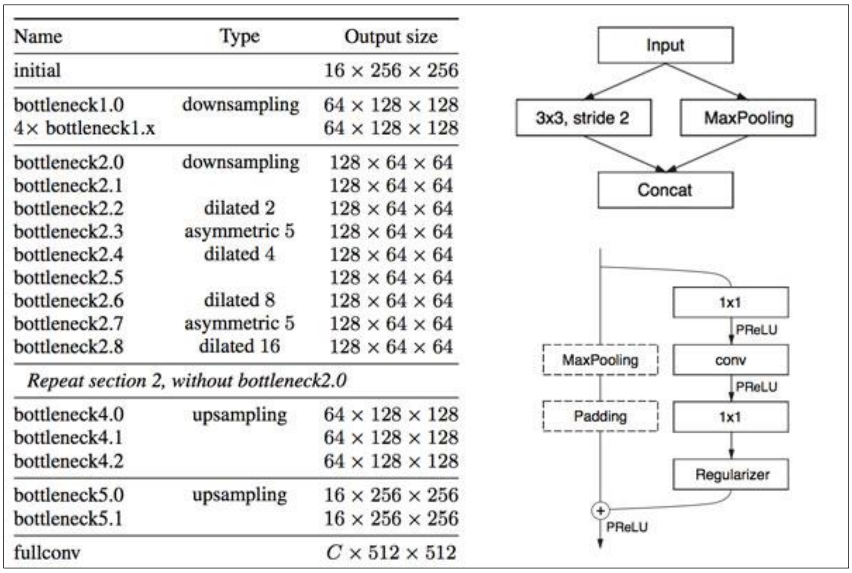

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 3: ResNet building block

III. DISCUSSION AND RESULTS

In this study the implementation of the ENet model for semantic segmentation using OpenCV was carried out on windows 7 64- bit Intel core i5 with python. OpenCV is a cross-platform library using which real-time computer vision applications can be developed. OpenCV focuses on image processing; video capture and analysis including features like face detection and object detection. OpenCV can implement the image segmentation, thresholding and semantic segmentation. ENet architecture is divided into several stages, as shown in figure 4 on the left hand side and the first digit after each block name. Output sizes are reported for an input image resolution of 512 x 512. In figure 4 describes the adoption of ResNet, a single main branch and extensions with convolutional filters that separate from it, and then merging back with an element-wise addition. Each block consists of three convolutional layers: a 1x1 projection that reduces the dimensionality, a main convolutional layer, and 1 x 1 expansions. The batch Normalization and PReLU are placed between all convolutions, and the bottlenecks modules are referred. A max pooling layer is added to the main branch, when the bottleneck is down-sampled. The first 1 x 1 projection is replaced with a 2 x 2 convolution with stride 2 in both dimensions.



Name	Type	Output size
initial		$16 \times 256 \times 256$
bottleneck1.0	downsampling	$64 \times 128 \times 128$
4× bottleneck1.x		$64 \times 128 \times 128$
bottleneck2.0	downsampling	$128 \times 64 \times 64$
bottleneck2.1		$128 \times 64 \times 64$
bottleneck2.2	dilated 2	$128 \times 64 \times 64$
bottleneck2.3	asymmetric 5	$128 \times 64 \times 64$
bottleneck2.4	dilated 4	$128 \times 64 \times 64$
bottleneck2.5		$128 \times 64 \times 64$
bottleneck2.6	dilated 8	$128 \times 64 \times 64$
bottleneck2.7	asymmetric 5	$128 \times 64 \times 64$
bottleneck2.8	dilated 16	$128 \times 64 \times 64$
Repeat section 2, without bottleneck2.0		
bottleneck4.0	upsampling	$64 \times 128 \times 128$
bottleneck4.1		$64 \times 128 \times 128$
bottleneck4.2		$64 \times 128 \times 128$
bottleneck5.0	upsampling	$16 \times 256 \times 256$
bottleneck5.1		$16 \times 256 \times 256$
fullconv		$C \times 512 \times 512$

Figure 4: ENet architecture

The zero padding activates, to match the number of feature maps. The conv is either a regular, dilated or full convolution referred as deconvolution or fractionally strided convolution with 3 x 3 filters. It can be replaced it with asymmetric convolution i.e. a sequence of 5 x 1 and 1 x 5 convolutions. For the regularizer, use of spatial dropout, with p = 0.01 before bottleneck 2.0, and p = 0.1 afterwards. The initial stage contains a single block, stage 1 consists of 5 bottleneck blocks, while stage 2 and 3 have the same structure, with the exception that stage 3 does not down-sampled the input at the beginning. The three first stages are the encoder, Stage 4 and 5 belong to the decoder.

The inputs were given from ImageNet datasets which gave the inference as 1.1700 and 1.2012 seconds whereas for the inputs of own images gave the inferences as 1.1544 and 1.1388 seconds to get the semantic output of the image. The different colors are predefined and labelled accordingly to get the output images as shown in figure 5.

Feature map resolution

The two main disadvantages with down-sampling images during semantic segmentation are firstly, reducing feature map resolution implies loss of spatial information like exact edge shape. Secondly, full pixel segmentation requires that the output has the same resolution as the input. This indicates strong down-sampling will involve equal strong up-sampling, which increases model size and computational cost. The advantage of down-sampling is that the filters operating on down-sampled images have a better receptive field, which allows them to gather more contexts. This helps in differentiating between classes like, rider and pedestrian in a road scene and how people look like. Finally the use of dilated convolutions gives better results.

The first two blocks of ENet heavily reduces the input size, and use a small set of feature maps. The visual information is highly spatially redundant, and can be compressed into a more efficient representation. They act as good feature extractors and only preprocess the input for future portions of the network.

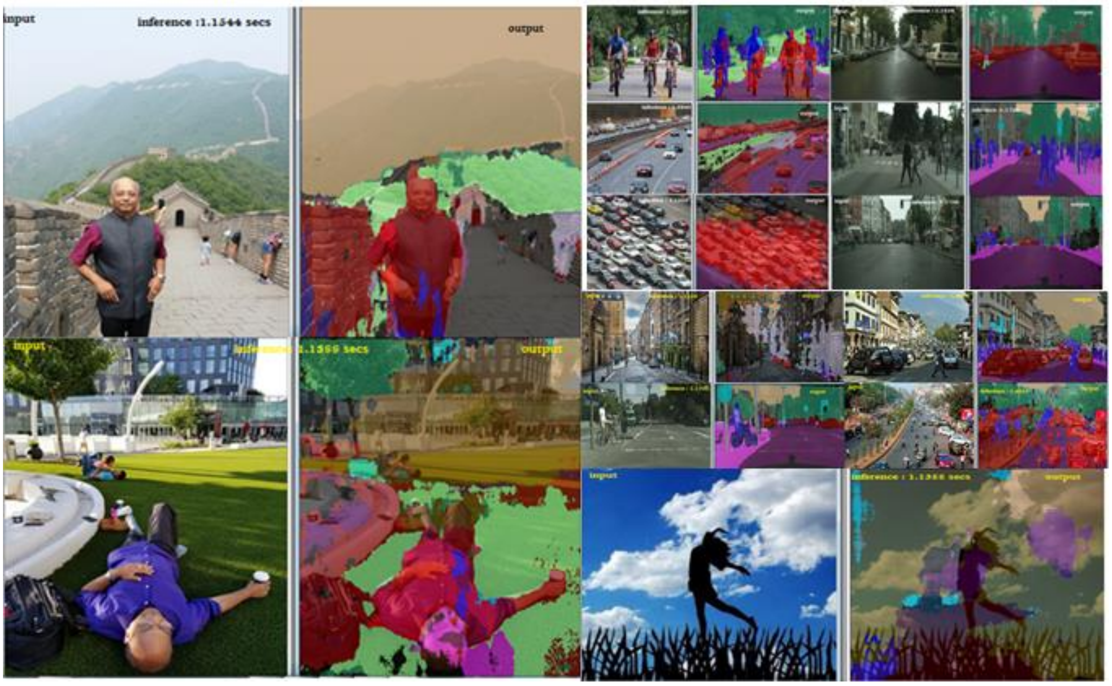


Figure 5: The outputs of Semantic Segmented images using ENet model with the corresponding inference values

Decoder size

Symmetric architecture SegNet has the encoder is an exact mirror of the encoder whereas the ENet architecture consists of a large encoder, and a small decoder, motivating the encoder should be able to work in a similar manner to original classification architectures, and to operate on smaller resolution data and provide for information processing and filtering. Thus, the role of the decoder is to upsample the output of the encoder with fine-tuning details.

Nonlinear operations

Replacing all ReLUs in the network with PReLUs, that uses an added parameter per feature map, with the aim of learning the negative slope of non-linearity. The layers where identity is a preferable transfer function, PReLU weights will have values close to 1, and equal values around 0 if ReLU is preferable. First layers weights show a large variance and are slightly biased towards positive values, while in the later portions of the encoder they settle to a recurring pattern. All layers in the main branch behave nearly exactly like regular ReLUs, while the weights inside bottleneck modules are negative i.e. the function inverts and scales down negative values.

Factorizing filters

The convolutional weights with reasonable amount of redundancy, and each n x n convolution can be decomposed into two smaller ones following each other. And one with n x 1 filter and the other with 1x n filters allows increasing the variety of functions learned by blocks and increasing the receptive field.

Dilated convolutions

Avoiding downsampling the feature maps, and dilated convolutions are used to improve the model.

Regularization

The pixel-wise segmentation datasets are comparatively small, expressive models as neural networks quickly begin to overfit them [4].

IV. CONCLUSION

ENet demonstrated on different datasets gave the inferences values accordingly and were calculated to be accurate for practical applications. Inference time for a single input frame of varying resolution reports number of frames per second that can be processed. ENet is significantly faster and can be provided for high frame rates for real time applications and that permits for practical use of deep neural network models with encoder-decoder architecture. Hence, processing the ENet model on the OpenPower architecture helps in reducing the inference time approximation on a CPU / GPU and more accurate results can be achieved on several datasets with image pixel-wise clarity.

REFERENCES

[1]. Yen-Kai Huang and Vivian Yang, “Street View Segmentation using FCN models”, 2017.

[2]. Alex Krizhevsky et al., “ImageNet Classification with Deep Convolutional Neural Networks” in Advances in neural information processing systems, 2012, pp. 1097–1105.

[3]. Kaiming He et al., “Deep Residual Learning for Image Recognition” in proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[4]. Adam Paszke et al., “ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation” arXiv: 1606.02147v1, 2017.

IOSR Journal of Engineering (IOSRJEN) is UGC approved Journal with Sl. No. 3240, Journal no. 48995.

Prisilla J. " OpenPOWER Architecture: A Case Study on Semantic Segmentation using ENet Model” IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 9, 2018, pp. 47-51.