LinkNet

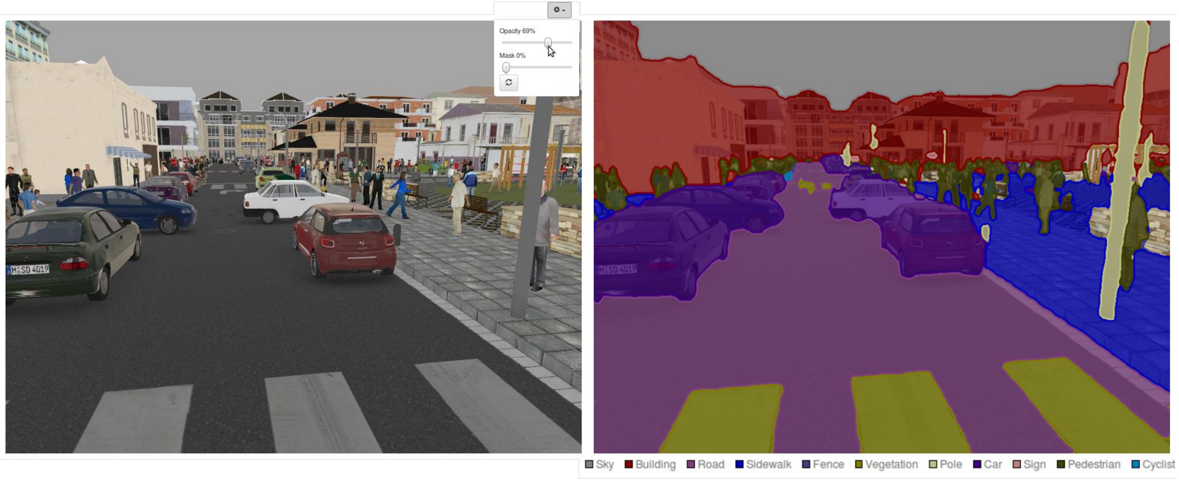
Exploiting Encoder Representations for Efficient Semantic Segmentation

NNFL Term Project
Project ID - 32

Ujjwal Gandhi - 2017A7PS0143P
Atmadeep Banerjee - 2017A7PS0101P
Mayank Jain - 2017A7PS0179P

Semantic Segmentation

It is way of grouping pixels in a semantically meaningful way, so that **every pixel** in the image is labeled with the class (*for eg., person, road, building*) of its enclosing object.

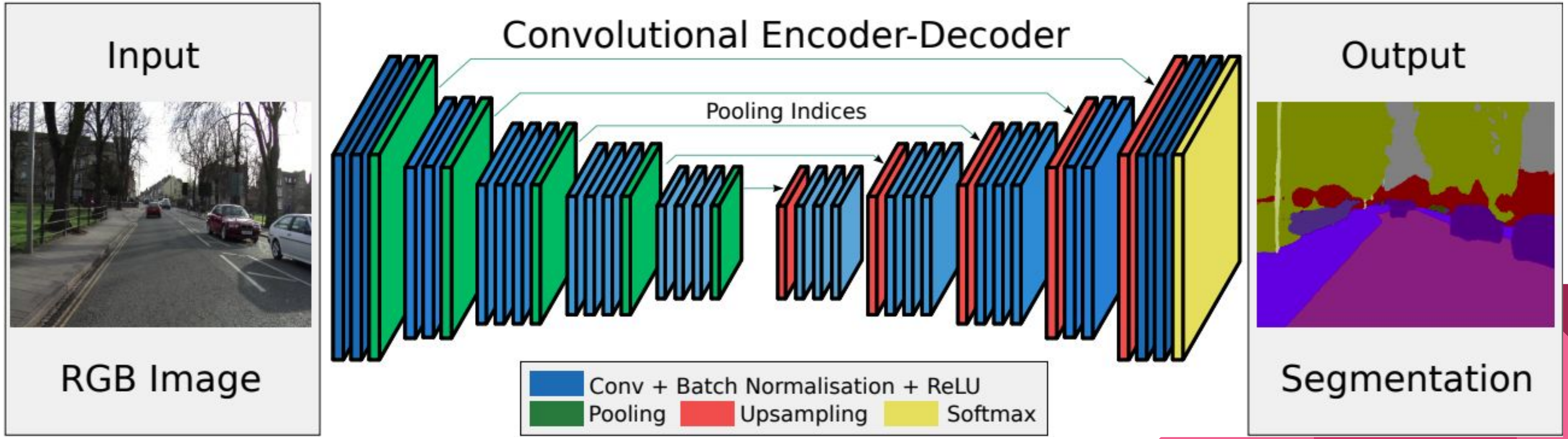


Opacity 69%

Mask 0%

Traditional encoder-decoder approach

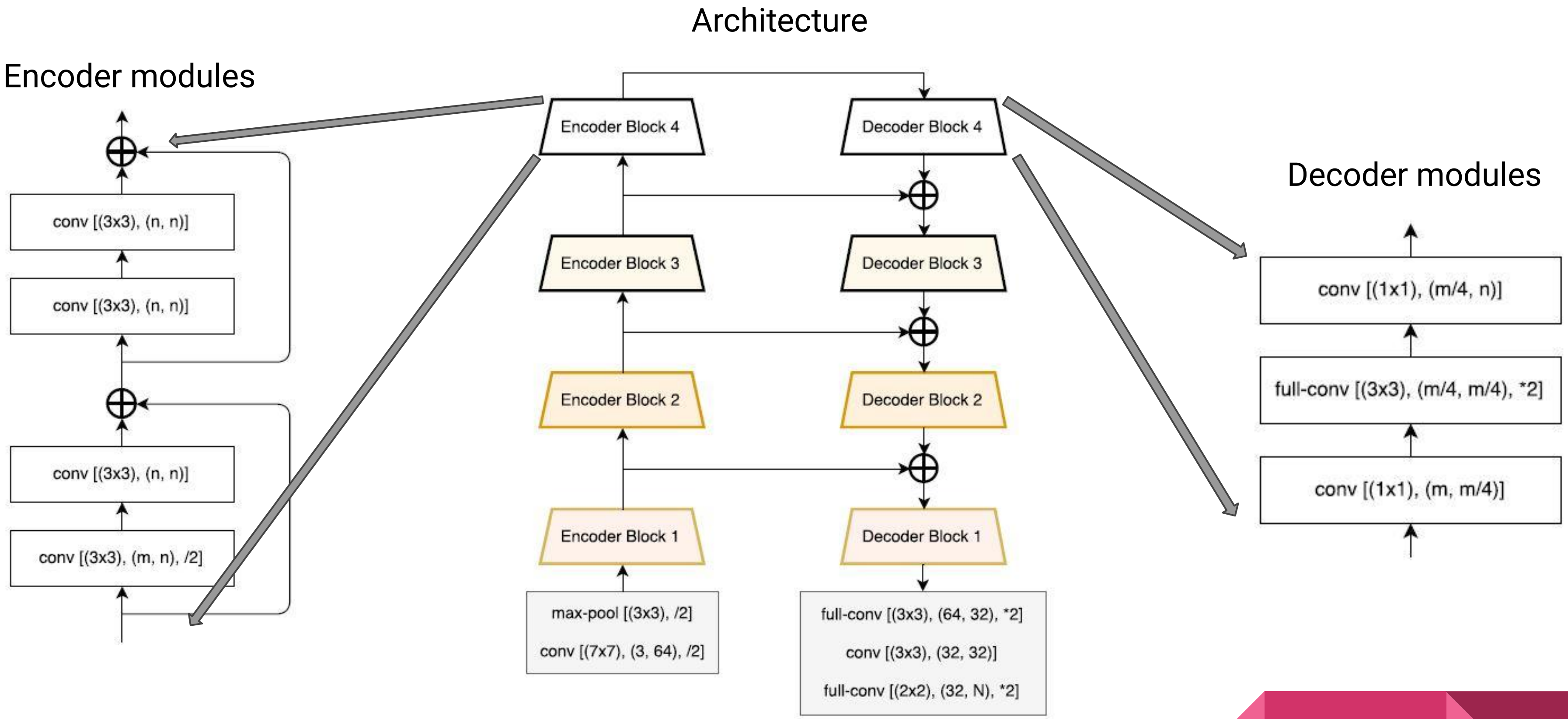
Existing techniques for semantic segmentation are based on the principle of auto-encoders, wherein encoder-decoder pairs are used to **encode** image information into a feature space, followed by **decoding** into spatial categorization.


Input
RGB Image
Convolutional Encoder-Decoder
Pooling Indices
🟦 Conv + Batch Normalisation + ReLU
🟩 Pooling 🟥 Upsampling 🟨 Softmax
Output
Segmentation

What is LinkNet? Why LinkNet?

Existing techniques try to make the network deeper and deeper. Evidence reveals that network depth is of crucial importance, but it leads to vanishing gradients, loss of information and degrading accuracy.

LinkNet uses **residual**/bypass connections instead, bypassing spatial information directly from the encoder to the decoder, thus preventing information loss at each level of the encoder and achieving a significant decrease in running time, making real-time semantic segmentation accurate and efficient.



Network architecture

ResNet 18

LinkNet uses ResNet-18 (11M parameters) as its encoder, which is a fairly lighter network as compared to VGG16 (138M parameters) and ResNet-101 (45M parameters) used in contemporary segmentation models.

With the network depth increasing, accuracy gets saturated and then degrades rapidly. This indicates that not all systems are similarly easy to optimize. To overcome this ResNets use residual mapping, rather than the original, unreferenced mapping, which adds residues from earlier layers to improve optimization.

Bypass connections

By performing multiple downsampling operations in the encoder, some spatial information is lost. It is difficult to recover this lost information by using only the downsampled output of encoder.

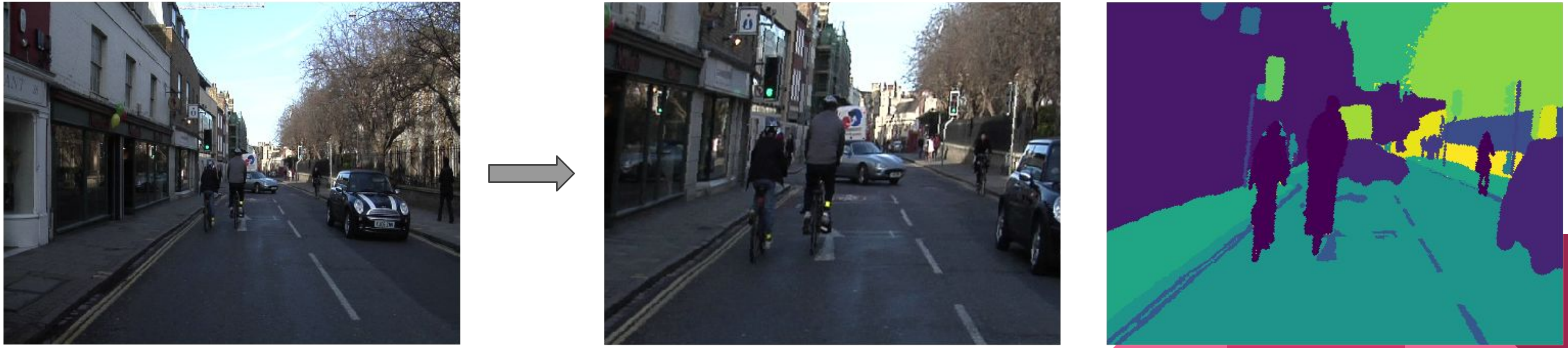
Therefore, input of each encoder layer is also bypassed to the output of its corresponding decoder. By doing this we recover lost spatial information that can be used by the decoder and its upsampling operations. In addition, since the decoder is sharing knowledge learnt by the encoder at every layer, the decoder can use fewer parameters. This results in an overall more efficient network when compared to the existing segmentation networks, and thus real-time operation

Dataset

- The model was trained on the Camvid and Cityscapes dataset. Both these datasets consist of video frames depicting urban areas, annotated pixel wise.
- The original Camvid dataset has 376 training images, 101 validation and 233 test images. Each image has a size of 960x720 pixels with 32 discrete classes. The authors fused similar classes to reduce the number to 12. They also reduce the size of images to 768x576 to make it easier to fit the dataset.
- The cityscapes dataset has 2975 training and 500 validation images. Each image has a size of 2048x1024 with 34 discrete classes. The authors reduce the number of classes to 19 and image size to 1024x512. *We were unable to find this 19 class version of the dataset and used the original 34 class version.*

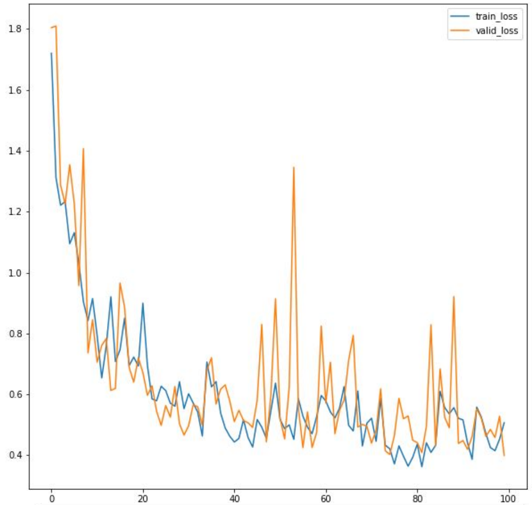
Preprocessing

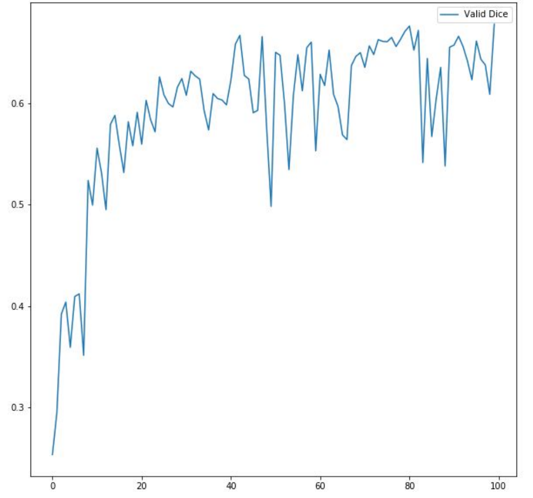
- We apply shear, affine and scale transforms on the training images for augmentation.
- We also normalize the images with imagenet statistics before feeding them to the model.



Training

- We train the model using weighted cross-entropy loss. This loss function assigns higher weights to less frequently occuring classes in the dataset. It is helpful in dealing with the heavy class imbalance in the dataset.
- The paper mentions to set the class weights as 1/ln(p + 1.02) where p is the frequency of a particular class.
- The Camvid model is trained for 100 epochs. Due to resource constraints we were only able to train the Cityscapes model for 20 epochs.



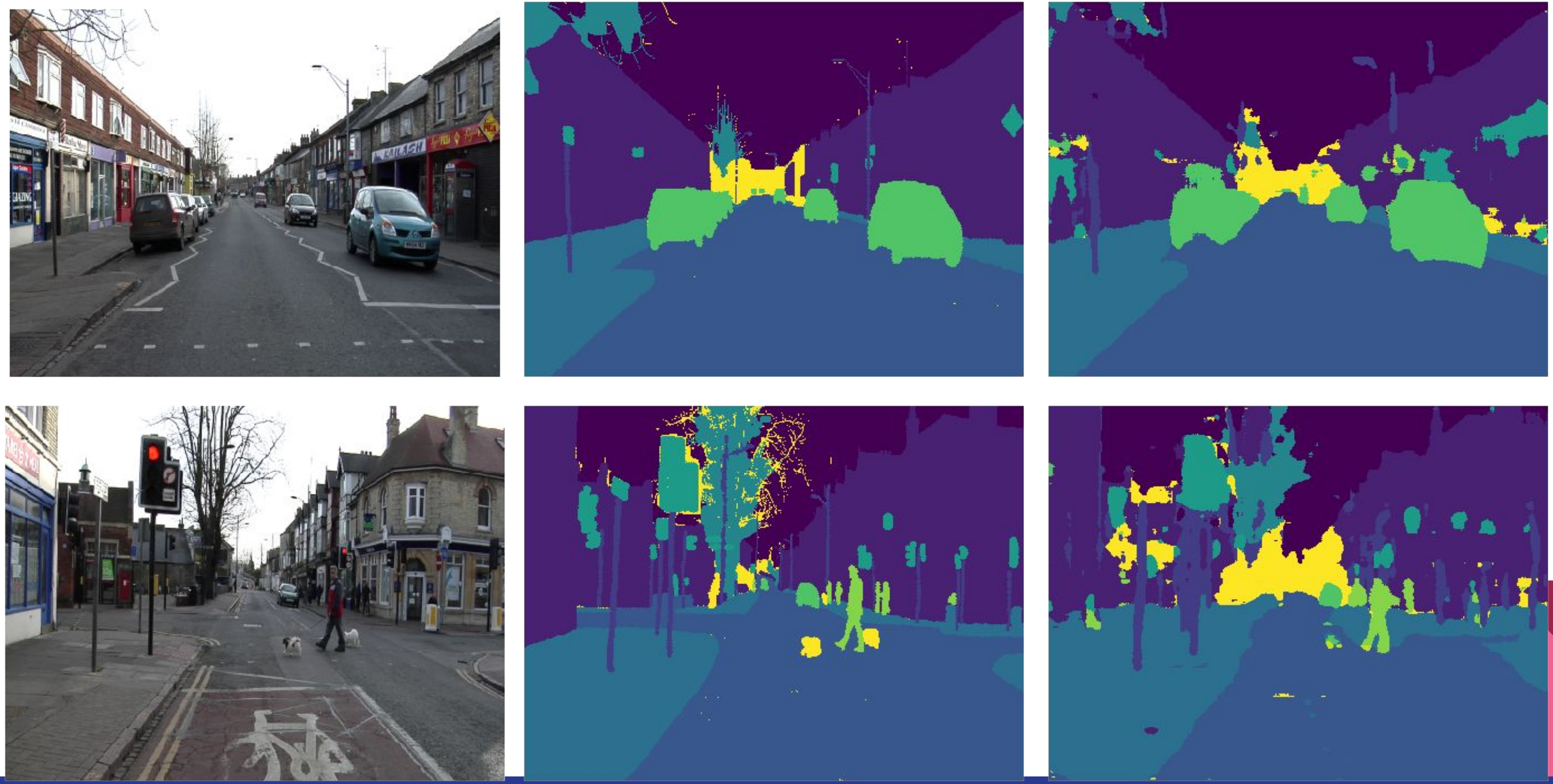


Results

Model		Class IoU
1	SegNet	65.2
2	ENet	68.3
3	LinkNet(Original)	68.3
4	LinkNet (Our implementation)	67.63
5	LinkNet with modifications	65.17

Table 1: Camvid test set results

Results



Input

Ground Truth

Result

Modifications to the original paper tried

We tried the following modifications to the original paper:

- **AdamW with One Cycle learning rate schedule.** AdamW is a modified version of Adam that implements weight decay for regularization. One cycle is a learning rate schedule which consists of starting with a moderate lr, going to a high lr following a half-cosine curve and coming back down to a very low lr. This cyclic learning rate has been attributed to model convergence with lesser amount of training epochs(super-convergence).
- **Sub-pixel convolutions.** We swapped the deconvolution layers in the decoder with sub-pixel convolution layers. These layers are mathematically equivalent to deconvolution, but are computationally much faster.

The modified network had 10% lower training time but reached a 3% lower Dice score of 65.17

Thank You