Journal of *Imaging*

MDPI

*Article*

# ECRU: An Encoder-Decoder Based Convolution Neural Network (CNN) for Road-Scene Understanding

**Robail Yasrab** [1,2]

[1] Computer Vision Laboratory, School of Computer Science, University of Nottingham, Nottingham NG8 1BB, UK; robail.yasrab@nottingham.ac.uk

[2] School of Computer Science and Technology, University of Science and Technology of China, Hefei 230000, China

Received: 24 June 2018; Accepted: 29 September 2018; Published: 8 October 2018


**Abstract:** This research presents the idea of a novel fully-Convolutional Neural Network (CNN)-based model for probabilistic pixel-wise segmentation, titled Encoder-decoder-based CNN for Road-Scene Understanding (ECRU). Lately, scene understanding has become an evolving research area, and semantic segmentation is the most recent method for visual recognition. Among vision-based smart systems, the driving assistance system turns out to be a much preferred research topic. The proposed model is an encoder-decoder that performs pixel-wise class predictions. The encoder network is composed of a VGG-19 layer model, while the decoder network uses 16 upsampling and deconvolution units. The encoder of the network has a very flexible architecture that can be altered and trained for any size and resolution of images. The decoder network upsamples and maps the low-resolution encoder's features. Consequently, there is a substantial reduction in the trainable parameters, as the network recycles the encoder's pooling indices for pixel-wise classification and segmentation. The proposed model is intended to offer a simplified CNN model with less overhead and higher performance. The network is trained and tested on the famous road scenes dataset CamVid and offers outstanding outcomes in comparison to similar early approaches like FCN and VGG16 in terms of performance vs. trainable parameters.

**Keywords:** convolutional neural network (CNN); ReLU; encoder-decoder; CamVid; pooling; semantic segmentation; VGG-19; ADAS

## 1. Introduction

In recent years, numerous technology systems have come out that can understand the surroundings (e.g., automatic driving). Therefore, scene understanding turns out to be a significant research area for analysing the geometry of scenes and object support associations. In this scenario, CNNs or ConvNets turned out to be a most powerful vision computing tool for image recognition and scene understanding [1–4]. CNNs offer huge support intended for understanding the scenes that normally differ in pose and appearance. CNN technology leads to numerous outstanding and successful results, and now, it is extensively used in several vision problems like object detection [5], image classification [6], human pose estimation [7], visual tracking [8], semantic segmentation [9] and action recognition [10]. To understand a given scene to its pixel level, it is really important to make some critical decisions in an automated environment. In this scenario, some recent studies [11] became the crucial solutions for scene understanding. Most importantly, these methods have set the benchmark on numerous popular datasets [12,13]. In terms of understanding a scene, semantic segmentation turned out to be one of the popular methods for assigning labels to individual pixels in given input images. Semantic segmentation involves a wide variety of applications, i.e., object support relationship analysis, scene understanding, autonomous driving, etc. A number of researchers

proposed numerous ways for semantic pixel-wise labelling [14,15]. Some approaches tried deep architectures for pixel-wise labelling [14] for category prediction. The outcomes are impressive [16], as the max pooling method minimizes the resolution of feature maps. In this scenario, SegNet [17] model is a remarkable methodology that is based on utilizing low-resolution feature mapping. This kind of mapping offers features for perfect boundary localization. The Fully-Convolutional Network (FCN)-based [18] approach has been successfully employed for sparse features to classify the input's local regions. Deconvolution is also incorporated as a bilinear interpolation for labelling at the pixel-level. To get an output map for fine segmentation, the Conditional Random Field (CRF) is applied optionally [19].

As pixel level segmentation is offering better results, researchers started to use these methods for real-time automated systems. Recently, driving assistant systems have become top research areas that provide numerous tools to improve the driving experience. By employing proposed CNN techniques in the Advance Driving Assistance System (ADAS), the driver performance could be enhanced in different road conditions. ECRUis an attempt to implement the powerful CNN for driving assistance in a more trustworthy and productive way. In the last few years, a number of researchers [20–22] have offered outstanding solutions that set the standard for the proposed model. The proposed architecture is a novel idea to implement a CNN for road scene understanding using the semantic segmentation architecture. It is deep FCNN for classification and semantic pixel-wise segmentation. The encoder is based on the VGG-19 network [23] with only the first 16 convolutional layers. The decoder is based on Badrinarayanan et al. [17], who upsampled the lower resolution input feature map(s) produced by convolutional layers in the encoder section. A number of experiments have been performed for the analysis of the proposed architectures (listed in Section 6). It is assessed that proposed network architecture is efficient and offers better performance as compared to earlier benchmark results.

The rest of the paper is arranged as follows: the first part is the literature review and proposed technology overview. The next section presents the algorithm and network architecture with a table- and picture-based explanation. The next sections outline the experimentation, results and an examination of the proposed system.

## 2. Related Work

Since their emergence, CNNs have successfully been employed for numerous vision-based areas including classification and object detection [24,25] (e.g., pose estimation [26], 3D and stereo-depth [27] analysis, instance segmentation [28] etc.). The FCN model proposed by Long et al. [18] emerged as one of the most powerful tools for vision-based tasks. FCNs augment the strength of traditional CNN models by offering arbitrarily-sized inputs. The proposed idea was applied to the classic LeNet [29] to enhance its ability to recognize digit strings. Later, Matan et al. used the Viterbi decoding methodology to extract results from input strings that were limited to one-dimensional. The same idea was applied by Wolf et al. [30] to expand CNN results using two-dimensional maps. The same architecture was later employed to recognize postal address blocks by detecting their four corners. FCN application to such diverse areas for detection and recognition leads to further developments in CNNs. Semantic pixel-wise segmentation is one of the biggest examples of such wide-ranging applications of CNNs [31–34]. In order to get more accurate results and improve the efficiency, additional aids (e.g., pairwise or higher order CRF [35]) were added to classical CNNs and FCNs. These approaches offered better results; however, they added an additional load of processing and memory storage. As vision-based systems emerged, the need for more accurate systems with lower processing and storage also emerged. In this scenario, the aforementioned additional aid-based systems were not suitable. There is a need for more smart and efficient systems with lower system requirements. In the CNN paradigm, encoder-decoder-based neural networks emerged as a replacement of additional aids. The support vector machine [36] and Gaussian process regression [37] methods are also assessed in this literature review.

Huang et al. [38] initially proposed the idea of the encoder-decoder-based neural network. It was initially proposed for unsupervised feature learning. Since then, encoder-decoder-supported neural networks have turned out to be an emerging idea in order to replace additional aids. Recently, Badrinarayanan et al. proposed some impressive designs like SegNet [17] and Bayesian-SegNet [39] to demonstrate the power of these networks in semantic segmentation. There are also a number of other approaches that have emerged using encoder-decoder-supported NNs for pixel-wise segmentation of the whole input [40,41]. The main characteristic of the proposed designs is a single end-to-end training pipeline, which offers easy training and enhanced performance. A similar approach was offered by [42] that was aimed at simplifying the traditional encoder-decoder pipeline. The key idea in all aforementioned network architectures is to employ the decoder CNN to map the low resolution features (extracted from pooling layers) to the final feature-maps of high resolution. The key aspect of encoder-decoder design is to provide efficiency and accuracy [17,39,42].

Recently there have emerged CNNs offering extensive efficiency and performance that can also be employed for scene segmentation to recognize all the pixels in a given image. The key objective of scene segmentation is less about recognizing specific pixels than it is about recognizing the boundaries among the different types of objects in the image. Knowing where the road is compared to other objects in the scene offers a great deal of advantages to a vehicle's navigation and brings us one step closer to autonomous vehicle technology. Embedded CNNs currently offer the efficiency that is required to empower real-time analysis of the road scenes from multiple cameras mounted on a vehicle. CNNs can also help to determine what to react to and what to ignore. Eventually, CNNs will contribute a great deal to changing the future of the Advanced Driver Assistance Systems (ADAS). This technology can improve automotive safety as active safety measures to avoid any dangerous road situation. The proposed technology can actively provide assistance to the driver in many ways (e.g., blind spot detection, pedestrian detection, obstacle detection, etc.). An interesting CNN-based approach for ADAS proposed by Jung et al. [20] was a pedestrian collision warning system. However, it is just limited to pedestrian detection and warning. Another great attempt to understand the road scene was presented by Xie et al. [21]. It is a two-stage cascaded CNN architecture that is trained for traffic sign classification. Attribute-supervisory signals are one of the features of the proposed network that helps to improve network accuracy. In the initial stage, the network is trained using the class label as supervised signals, while in the next stage, the network is trained using superclasses individually. This separate training is performed using auxiliary attributes of traffic signs for additional refinement purposes. A variety of experiments outlined that the hierarchical cascade architecture is able to extract the deep information of related categories, enhance discrimination of the model and escalate traffic signs' classification accuracy. However, a network with wide-ranging road scene understanding is still missing. Some approaches [43,44] used radars and sonar sensor systems for efficient road scene understanding. However, these solutions are expensive and still lacking wide-ranging application. Deng [45] proposed a new method for understanding the road surroundings from vehicle cameras. It is a 360-degree road scene semantic segmentation method that makes use of the surrounding view cameras, with the fisheye images' view. The proposed method works well for short distances. However, fisheye images are not able to segment images, especially on edges. New and more improved methods [3,46–49] are offering more enhanced segmentation, though our research's key objective is to reduce the size of a segmentation network without losing feature details. Recently, a very promising CNN architecture was released named DeepLabV3 [4]. It is the third version of the DeepLab network initially released in 2014. The proposed architecture offers a considerable application diversity. The network extension is also possible through duplicating the last ResNet [50] block module. It makes use of the unique type of pooling methodology known as Atrous Spatial Pyramid Pooling (ASPP) that offers improved features extraction. CRF is not used to reduce network size. However, in comparison to the proposed ECRU, the DeepLabV3 is too complicated and huge in size.

Another efficient CNN architecture released recently is PSPNet [2]. It is designed for pixel-level prediction tasks. It makes use of the pixel-level feature for efficient segmentation in a specially-designed global pyramid pooling architecture, where local and global clues are concatenated together to produce the outcomes. PSPNet is an extensively complex architecture that requires plenty of computational resources and GPU capabilities for training and testing tasks. A very recent study about pixel-wise segmentation presented a new idea of panoptic segmentation (PS) [1]. Panoptic segmentation combines the instance segmentation and semantic segmentation to perform a wide-ranging segmentation task. It offers a vibrant set of feature maps for the final scene segmentation. However, it builds a considerable amount of feature map load that leads to a large-sized network architecture. It shows comparatively good performance as compared to other VGG-based networks, though the crucial weak factor of this design is size. The VGG-based designs are already pretty big, and adding additional layers make the whole system too enormous.

## 3. Algorithm Overview

The ECUR network is an encoder-decoder network. As previously mentioned, the encoder has the initial 16 conv.layers of the VGG-19 network. This network is based on two decoupled CNNs. The first network is intended to perform a convolution process, while the second performs deconvolution. The network performs semantic pixel-wise segmentation. The proposed algorithm for the encoder (Algorithm 1) is initiated by an input image. Later, the image maps are processed through a set of conv-layers, Batch Normalization (BN) and Exponential Linear Unit (ELU) activation function units.

**Algorithm 1** CNN encoder training: Training starts with $X_1 \times H_1 \times D_1$. Additional parameters required in training are $F$ spatial extent, $K$ number of filters, $P$ zero padding and $S$ stride rate. The network outputs an image of $X_2 \times H_2 \times D_2$.

**Require:** $Batch - Norm()$ operations are performed on given parameters; later, $ELU()$ activations are carried out. The $Conv.()$ function carries out the convolution operation by a constant rate of $S = 1$ (stride_rate), kernel_size $= 3$ and $P = 1$ (zero_padding). Next, the $Pl()$ function performs the pooling, using the max pooling layer with hyper-parameters: $P = 0$ (zero_padding), kernel_size $= 2$ and $S = 2$ (stride_rate).

**Ensure:** weights updating $W^{t+1}$, Batch-Normparameters updating $\theta^{t+1}$ and learning_rate updating $\eta^{t+1}$.

1. Encoder CNN:
1.1. Generating feature-maps:

Input $X_1 \times H_1 \times D_1$ size image
**for** $k$ = 0 to $L - 1$ **do**
  $W_{bk} \leftarrow BN(W_k)$
  $W_{ek} \leftarrow ELU(W_k)$
  $W_{ck} \leftarrow Conv.(W_{ek}, W_{bk})$
  $Pl_{kmask}, Pl_k \leftarrow Pl(W_{ck})$
  **if** $k \leq L$ **then**
    $W_k \leftarrow Pool_k$
    **return** $Pl_{kmask}$
  **end if**
**end for**

Next is the decoder (Algorithm 2) model that is composed of upsampling, de-convolution, BN and ELU activation units. The decoder upsamples the encoder's convolution layer indices produced from pooling layers. Each decoder upsamples the activations generated by the corresponding encoder. This is called non-linear upsampling for input feature-maps. Batch Normalization (BN) [51] helps to accelerate the training and also appears to reduce the overall effect of the weight's scale. To achieve better results, we have used the shift-based batch normalization method. Algorithm 3 offers a detailed step-by-step procedure regarding the batch normalization procedure on a given set of inputs.

**Algorithm 2** CNN decoder training: Input pooling mask size of $X_2 \times H_2 \times D_2$. The out volume is $X_D \times H_D \times D_D$.

**Require:** The encoder network extracts a mini-batch of feature maps $X_2 \times H_2 \times D_2$. BN parameters $\theta$, weights $W$, weights initialization-coefficients $\gamma$, and encoder's learning_rate $\eta$ are initialized using He et al.'s method [6].

**Ensure:** weights updating $W^{t+1}$, Batch-Norm parameters updating $\theta^{t+1}$ and learning-rate updating $\eta^{t+1}$.

2. Decoder CNN:
2.1. De-Conv.and upsampling:

Input $Pl_{kmask}, Pl_k$
**for** $k = L - 1$ to $k > 0$ **do**
  $W_{dk} \leftarrow Up\_sample(Pl_{kmask}, Pl_k)$
  $W_{dbk} \leftarrow BN(W_{dk})$
  $W_{dek} \leftarrow ELU(W_{dk})$
  $W_{dck} \leftarrow Conv.(W_{dek}, W_{dbk})$
  **if** $k <= L\ AND\ k \neq 0$ **then**
    $Pl_k \leftarrow W_{dck}$
  **end if**
**end for**
**return** $W_{dck}$

**Algorithm 3** Batch normalization [51]: In the algorithm, let the normalized values be $\widehat{x_{1...m}}$ and their linear transformations be $y_{1...m}$. The transformation is stated as: $BN_{\gamma,\beta} : x_{1...m} \rightarrow y_{1...m}$. This algorithmic procedure states that $\epsilon$ is a constant incorporated for numerical stability.

**Require:** $x$ is the value of a mini-batch: $B = x_{1...m}$; $\gamma, \beta$ are the learning parameters.

**Ensure:** $y_i = \mathrm{BN}(\mathrm{x_i}, \gamma, \beta)$

$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i$ //mini-batch mean

$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2$ //mini-batch variance

$\widehat{x_i} \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}}$ //normalize

$y_i \leftarrow \gamma \widehat{x_i} + \beta \equiv \mathrm{BN}(\mathrm{x_i}, \gamma, \beta)$ // scale and shift

Figure 1 shows the flowchart of the proposed algorithms. Our proposed upsampling idea was inspired by [38], which is intended for unsupervised feature learning. The fundamental aspects of the proposed encoder-decoder network are the decoding process, which has numerous practical advantages regarding enhancing boundary delineation and minimizing the total network size for enabling end-to-end training. The key benefit of such a design is an easy to modify encoder-decoder architecture that can be adapted and changed with very little modification. This encoder offers low-resolution feature mapping for pixel-wise classification. The feature maps produced through the convolution layer are sparse, those later convolved using the decoder filters to generate detailed feature maps.

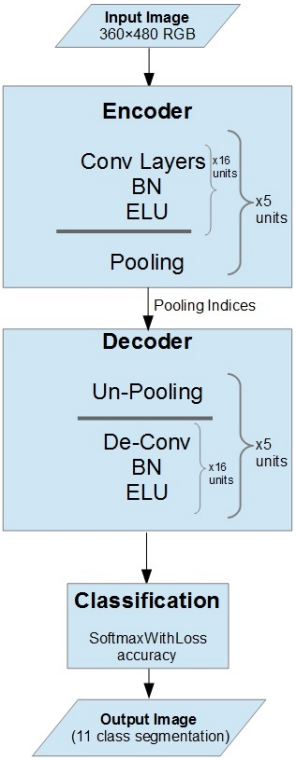


**Figure 1.** Flowchart of the algorithms. BN, Batch Normalization.

The proposed model and algorithms contain the following key features:

- A novel VGG-19 network-based encoder-decoder network for semantic segmentation.
- Simplified training of encoder and decoder networks simultaneously.
- An enhanced encoder-decoder model that improves the overall segmentation performance.
- The encoder architecture feeds the feature map(s) to the decoder to upsample. The proposed core segmentation engine has reduced the amount of the encoder (∼30 M) and decoder (∼0.5 M) parameters. In this way, the overall network size is reduced.
- Implemented the latest enhanced activation function (PReLU, ELU) to increase network efficiency.
- Flexible CNN network to adapt to any size of input.

## 4. ECRU Architecture

The proposed CNN architecture is a pixel-wise semantic segmentation network with encoder-decoder network architecture. The decoder network results are fed to a final pixel-wise classification layer, which offers final pixel-wise image segmentation. The CNN architecture is illustrated in Figure 2. The VGG-19 [23] is used as a base architecture. To retain higher resolution feature maps for the upsampling process in the decoder network, the fully-connected layers are discarded. In this way, the resultant pixel-wise semantic segmentation CNN encoder is smaller in

size as compared to other similar architectures [18,52]. In the proposed network, every encoder has a convolution layer supported by a filter-bank. These convolution layers generate a set of feature maps. Convolution layers are followed by the batch normalization [51] layer. Later, the Exponential Linear Unit (ELU) [53] is employed to speed-up the network learning process.

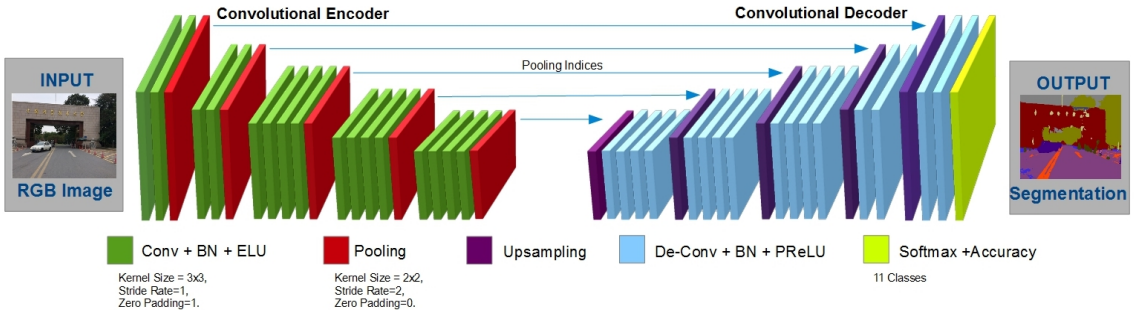


**Figure 2.** ECRUarchitecture.

The ECRU network starts with a class-specific input image $\mathbf{x}_i$ and activation map $\mathbf{g}_i^l$. The input feeds data to the proceeding CONV and pool layers. Lastly, the decoder unit is fed with segmentation map $M(\mathbf{g}_i^l;\theta_s)$, afterwards implementing the softmax (where $\theta_s$ is the segmentation network's parameter). $M(\mathbf{g}_i^l;\theta_s)$ has the back and foreground channels. These channels are represented as $M_b(\mathbf{g}_i^l;\theta_s)$ and $M_f(\mathbf{g}_i^l;\theta_s)$, respectively.

$$\min_{\theta_s} \sum_i e_s(\mathbf{z}_i^l, M(\mathbf{g}_i^l;\theta_s)), \tag{1}$$

Later on, ELU units are applied for activations. There are many experiments performed with different activation functions, and ultimately, it is discovered that the ELU performs the best. It is employed to enhance the speed of the learning process. It is also helpful to minimize the chances of the vanishing gradient through the identity for positive values. ELU converges to the highest point till the completion of the training and testing procedure. The ELU function can be shown as $0 < \alpha$:

$$f(i) = \begin{cases} i & \text{if } i > 0 \\ \alpha(\exp(i) - 1) & \text{if } i \leq 0 \end{cases}, \tag{2}$$

$$f'(i) = \begin{cases} 1 & \text{if } i > 0 \\ f(i) + \alpha & \text{if } i \leq 0 \end{cases}.$$

The hyper-parameter $\alpha$ deals with the saturation of negative ConvNet inputs. The max-pooling units are applied with a $2 \times 2$ window and a stride rate of two. These max-pooling indices are produced in the encoding sequence and later on upsampled in the decoder using upsampling layers. This method offers a great deal of benefit for retaining class boundary details in the segmented images, as well as minimizes the total amount of model parameters. This end-to-end network training is performed by Stochastic Gradient Descent (SGD). The Batch Normalization (BN) unit is applied to every convolutional unit. BN statistics are calculated from the training dataset and utilized at test time and dropout sampling.

The dropout technique inspired by Srivastava et al. [54] is used in the proposed encoder/decoder architecture. The dropout is used as an approximate inference in our CNN and to perform probabilistic inference over the segmentation model. The dropout layer probabilities $(pi)$ could be optimized. In the proposed architecture, it is fixed to the standard probability of dropping a connection as 50%, i.e., $pi = 0.5$.

The decoder model is inspired by the Bayesian-SegNet of Kendall et al. [39] and the unsupervised feature learning model by [38]. The first encoder feeds the corresponding (closest) decoder with a three-channel (RGB) image. This is different from the other encoder/decoder models, which produce the same number for the size and channel feature maps according to the inputs of the feeding encoder. Finally, dense filter maps are sent for pixel-wise segmentation. This process is carried out with the softmax classifier. Finally, the network offers $K$ (No. of classes) channels of probability for a given dataset.

## 5. Training and Experiments

### 5.1. Training

We have used Caffe (Berkeley Vision Library) [55] for the proposed network training and testing tasks. For network initialisation, we have used the proposed method of network initialization of He et al. [6]. The base learning rate for network training is set at 0.1, and weight-decay is set to 0.0005. We have trained the network for 500 epochs. These parameters are set after initial experiments to check that the CNN is converging during training. The buffer overflow is a critical issue in network training. Therefore, we set the batch size to 12 images to avoid such issues. Images were shuffled to ensure the diversity in each batch. The cross-entropy loss [56] was used to sum up the entire losses of the mini-batch pixels.

### 5.2. Experiments

#### 5.2.1. Implementation Details: BMSS (Base Model)

It is difficult to train a big-sized network, especially for experimental purposes. To avoid such a time-consuming situation, we initially performed experiments on a reduced version of the proposed model named Base Model for Semantic Segmentation (BMSS). Figure 3 shows the proposed base model for experiments. It offers a detailed depiction of the layers' configuration. The colour code is used to isolate and depict the layers' architecture separately. The figure shows two models, encoder and decoder. It also shows the transaction of pooling indices from the encoder module to the decoder module. It is smaller in size and convenient to alter for different experiments. It is inspired by the SegNet-Basic architecture, which is founded on 4 encoder-decoder units. The BMSS architecture is also based on a similar architecture as conv, batch normalization, pooling and activation layers. It is used to test different activation functions, pooling method and regularizing methodologies. Table 1 shows the proposed BMSS's experimental architecture. It shows the configuration of the layers and the encoder-decoder architecture. It offers the filter size, pooling and BN layers' configuration. The whole network is divided into four main blocks where the first and last blocks are input and output blocks, respectively. The middle blocks are composed of the encoder and decoder four-layered architecture that is small enough to alter and train for different experiments.

In many basic level tests, we have assessed different aspects of our proposed network. Initially, we have performed tests to find out the most suitable activation function. Experiments were performed to test the TanH [57], ReLU, PReLU [6], ELU [53] and Maxout [58]. The final performance assessment was performed for the aforementioned activation functions in the final implementation stage.

Later, we tested to seek the best performance among all pooling methods. Experiments were performed to test average, stochastic [59] and max pooling methodologies. The network was trained with each of these pooling units and ultimately checked for better performance. Next, some experiments were carried out to test the network without BN layers. The main hypothesis was to build a CNN with lower inference time and training parameters.

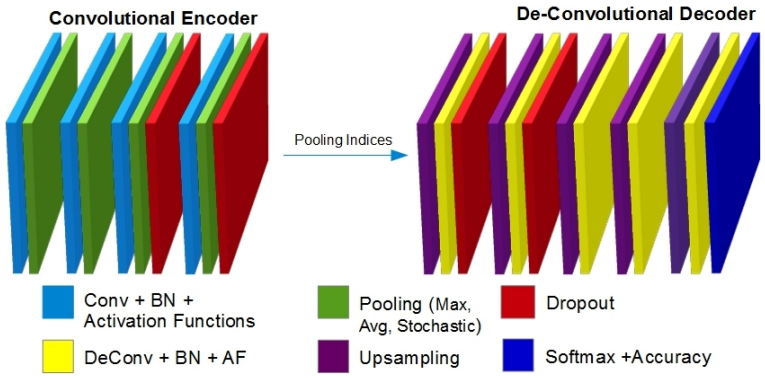


**Figure 3.** Architecture of the Base Model for Semantic Segmentation (BMSS).

**Table 1.** BMSS. ELU, Exponential Linear Unit.

| Input Block | Input 360 × 480 + Norm |
|---|---|
| Encoder Block | 7 × 7Conv.64+ BN + 2 × 2 Pooling (max) + ReLU/ELU/MaxOut/PReLU+ Dropout<br>7 × 7Conv.64 + BN + 2 × 2 Pooling (max) + ReLU/ELU/MaxOut/PReLU + Dropout<br>7 × 7Conv.64 + BN + 2 × 2 Pooling (max) + ReLU/ELU/MaxOut/PReLU + Dropout<br>7 × 7Conv.64 + BN + 2 × 2 Pooling (max) + ReLU/ELU/MaxOut/PReLU + Dropout |
| Decoder Block | Upsample + 7 × 7 De-Conv.64 + BN<br>Upsample + 7 × 7 De-Conv.64 + BN<br>Upsample + 7 × 7 De-Conv.64 + BN<br>Upsample + 7 × 7 De-Conv.64 + BN |
| Classification Block | Convolution Classifier 11 |
| Output Block | SoftmaxWithLoss<br>Accuracy |

#### 5.2.2. Implementation Details: ECRU

After analysing the results from the base models (BMSS), we have trained the complete version of the proposed network. It uses VGG-19 as a base design along with a deconvolution-based decoder. The encoder's pooling indices are fed to the corresponding upsampling layers. It is designed using the most efficient units (tested earlier) (e.g., ELU, max-pooling and BN units). The final tests were performed over the final version of CNN. Figure 2 illustrates the comprehensive model of the ECRU. The proposed design has been assessed for semantic segmentation of road-scenes (for 11 different classes). The principal focus was an enhanced and efficient network with better performance and high accuracy while designing and training the proposed CNN. This system is a significant step towards the driving assistant system toolkit implementation at common levels. The proposed architecture ECUR is demonstrated in Table 2. The table shows a complete encoder-decoder configuration of the network, where both consist of 5 modules each. The encoder Modules 1 and 2 are small modules based on two convolution layers, while Modules 3–5 are bigger modules based on four convolution layers each. Similarly, the decoder mirrors the encoder layer architecture; however, it is composed of deconvolution layers to upscale the pooling feature maps.

**Table 2.** ECRU's encoder-decoder layer's model.

| Encoder CNN | | | Decoder CNN | | |
|---|---|---|---|---|---|
| Input 360 × 480 + Norm | | | Output, softmax with Loss + Accuracy | | |
| Conv 1 | 3 × 3, 64<br>3 × 3, 64 | B-N, ELU, Pool | DeConv 1 | 3 × 3, 64<br>3 × 3, 64 | Upsample, B-N, ELU |
| Conv 2 | 3 × 3, 128<br>3 × 3, 128 | B-N, ELU, Pool | DeConv 2 | 3 × 3, 128<br>3 × 3, 128 | Upsample, B-N, ELU |
| Conv 3 | 3 × 3, 256<br>3 × 3, 256<br>3 × 3, 256<br>3 × 3, 256 | B-N, ELU, Pool, Dropout | DeConv 3 | 3 × 3, 256<br>3 × 3, 256<br>3 × 3, 256<br>3 × 3, 256 | Upsample, B-N, ELU, Dropout |
| Conv 4 | 3 × 3, 512<br>3 × 3, 512<br>3 × 3, 512<br>3 × 3, 512 | B-N, ELU, Pool, Dropout | DeConv 4 | 3 × 3, 512<br>3 × 3, 512<br>3 × 3, 512<br>3 × 3, 512 | Upsample, B-N, ELU, Dropout |
| Conv 5 | 3 × 3, 512<br>3 × 3, 512<br>3 × 3, 512<br>3 × 3, 512 | B-N, ELU, Pool, Dropout | DeConv 5 | 3 × 3, 512<br>3 × 3, 512<br>3 × 3, 512<br>3 × 3, 512 | Upsample, B-N, ELU, Dropout |
| Encoder transfer pooling indices to decoder-CNN | | | | | |

#### 5.2.3. Dataset

A well-known dataset, CamVid, is used for the training and testing. There are "11" different classes (such as building, person, vehicle, person, road, trees, etc.) for classification and segmentation. The dimension of the images is 360 × 480 pixels for training the proposed architecture. The dataset is divided into a training and a validation dataset with a rule-of-thumb ratio of 80/20. These datasets are divided into these categories for better cross-validation. Local contrast normalization [60] is carried out for every RGB input.

#### 5.2.4. Hardware Implementation

The Caffe deep learning library was used for implementing the overall CNN architecture. NVIDIA Tesla K40c Graphical Processing Unit (GPU) by NVIDIA Corporation Ltd., Santa Clara, CA, USA, with 12 G memory was used for CNN training and testing. The training takes nearly 12 h (40,000 iterations) for the segmentation network. The network training is carried out until loss of convergence and there is no significant growth or reduction in accuracy and loss, respectively.

## 6. Results and Analysis

The network performance was analysed with Class Average Accuracy (CAA) and Global Accuracy (GA) units, which are the well-known performance measures for segmentation and classification. The GA represents the total percentage of the dataset's pixels classified, and the CAA is the predictive accuracy of classes. To analyse different aspects and models suitable for developing a more enhanced network, we have experimented with different network architectures. The aforementioned BMSS reduced model was used for the experimentations. The base architecture was transformed and experimented with different kinds of layers, activation functions and pooling architectures. The proposed BMSS network was trained for a maximum of 10,000 iterations to train and test for the critical decrease in loss and augmentation in accuracy. However, the 16-layer version was trained for 40,000 transactions to ensure efficient network convergence.

Figure 4A illustrates the outcomes of the initial test carried-out on BMSS. The proposed network was tested for different activation functions, ReLU, PReLU [6] and ELU [53]), to select the most appropriate one. The results graph shows that the ELU (blue line) was converging very fast and attaining the constant high accuracy rate very early during training. This learning rate kept constant throughout training and analysis. It outperformed the ReLU and PReLU activations.

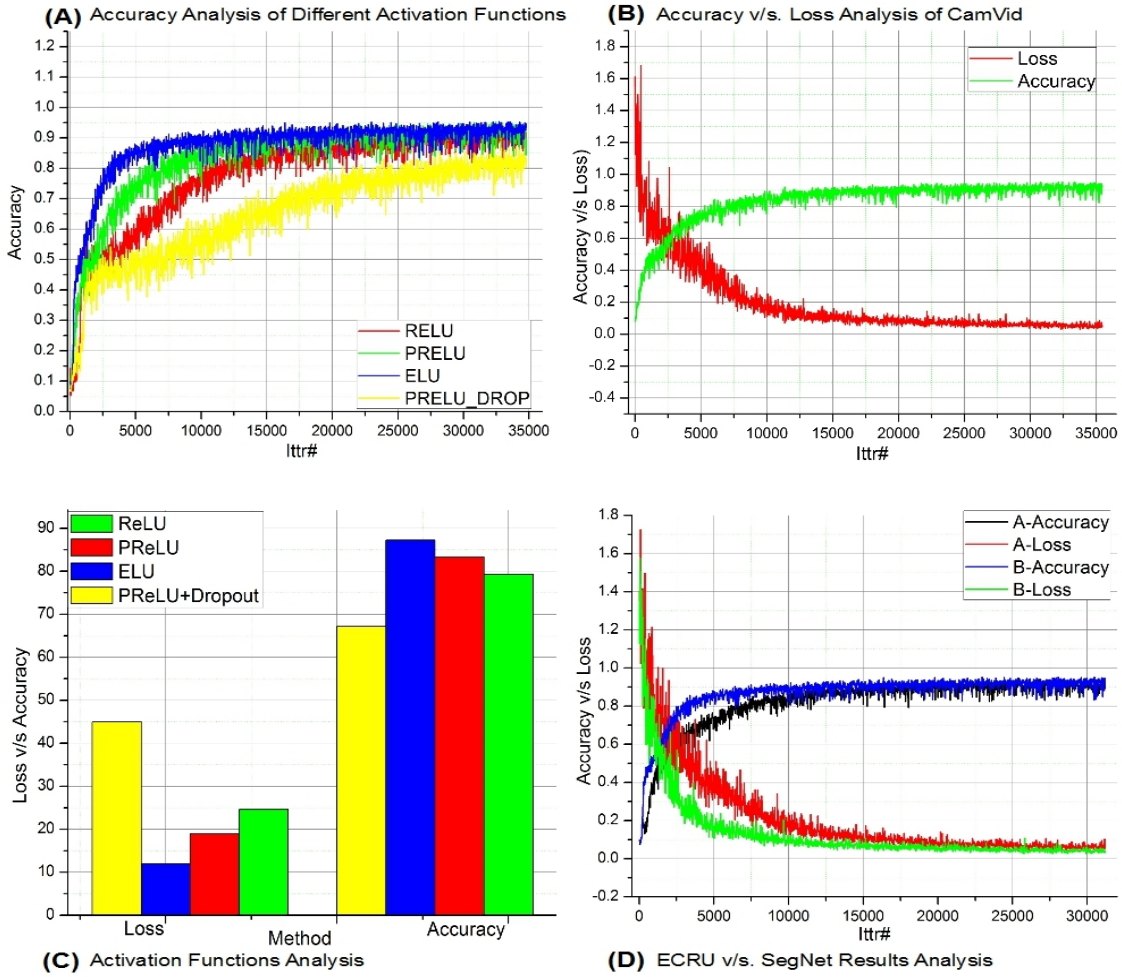


**Figure 4.** ECRU experiments results.

Figure 4C is also a continuation of the BMSS tests where it shows "loss vs. accuracy" analysis of the different activation functions and dropout units. It is assessed that ELU [53] offered more enhanced performance. The blue bars in the graph show the ultimate performance of ELU units. ELUs lowered the probabilities of the vanishing gradient in comparison to ReLUs and leaky-ReLUs. Different from ReLUs, ELUs offered $-ve$ values that could be pushed by the activations very close to a zero value. Pushing mean activations closer to zero offers a faster learning rate. Figure 4B shows the accuracy versus loss analysis. After analysis of the best performing units, we trained the final version of the proposed network and conducted ultimate training. The "x- and y-axis" illustrate the total number of iterations and the percentage of accuracy and loss, respectively. It is assessed that CNN attained an efficient learning level, while the "green" curve converges efficiently and ultimately reaches a constant rate. A similar behaviour can be seen at the "red" loss curve. Loss gradually decreased, and accuracy was improved. This is the ultimate performance of the ECRU.

Figure 4D demonstrates the benchmark results analysis of the proposed network and SegNet (both trained on the CamVid dataset). The ECRU final vision was designed for best performing

max-pooling, ELU and the BN layer. Ultimately, ECRU performed better than early benchmark approaches. Figure 4D demonstrates the accuracy versus loss comparison. The "x- and y-axis" signify the total number of iterations and the percentage of accuracy and loss respectively. Here, "B" denotes ECRU and "A" is the SegNet (as the benchmark). It could be seen that the proposed CNN offers a better learning rate and lower loss. These experiments have shown that the ECRU performed fine with less loss and a higher accuracy rate in contrast to earlier counterparts.

Table 3 illustrates the quantitative accuracy outcomes on CamVid that consists of 11 road scene classes. It outlines the individual class and accumulative accuracy of CamVid dataset classes. Experiment results presented above have shown that the proposed network has greater improvement in each category of segmentation in different road scenes. There was a considerable improvement in the number of categories. ECRU CNN design outperformed all the earlier benchmark designs, including those using additional aids like depth, video and/or CRFs. As ECRU estimations were precise in almost every category, it delineated a great deal of improvement in CAA rate when the network was trained using bigger dataset, and this set a new benchmark for the majority of the individual classes. Particularly, there was a substantial accuracy improvement for the smaller/thinner classes.

Table 3 presents results of the high-quality segmentation of the ECRU network. The proposed network had outperformed in eight classes (out of 11 categories) of the CamVid. There was a great deal of higher segmentation partition and consistent patterns among network output figures. ECRU offered the highest accuracy (98.3%) on pedestrian detection. We also got a very higher accuracy percentage on building (95.1%) segmentation. There were also higher accuracy percentages observed for vehicles, symbols, fence poles and bicyclist. These were all the most significant road objects that need to be segmented while designing any ADAS. The other objects' accuracy can be improved by developing and adding more encoder units in the network; however, this can lead to a more massive and computationally-extensive network architecture.

Figure 5 outlines the input and output results of the proposed network. This figure shows the class-based semantic segmentation of a given input image. The output image had a sharp boundary line that demonstrates the high efficiency of the proposed CNN. The resultant CNN offered the same quality results compared to the established benchmark result. A clear boundary delineation was also evidence for ECRU's high-quality segmentation. The figure shows the description of each identified object through a colour code given at the bottom of the image. The test images were efficiently segmented into different coloured objects. There was a clear boundary delineation among objects, which depicts that the model was not predicting any wrong labels. It also shows that the model uncertainty for object detection was pretty high. This high level of certainty offered reduced ambiguity in labels' description. There were some objects that could be occluded or that were not identified due to their distance from the camera. Another reason for object detection uncertainty could be the lighting conditions. One of the key reason we have used CamVid is that it includes diverse road scenes with different lighting conditions. Therefore, the proposed ECRU offers higher certainty levels in any lighting conditions.

**Table 3.** Quantitative results on CamVid.

| Network/Model | Building | Tree | Sky | Car | Sign/Symbol | Road | Pedestrian | Fence | Column-Pole | Side-Walk | Bicyclist | CAA. | GA. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Brostow et al. [61] | 46.2 | 61.9 | 89.7 | 68.6 | 42.9 | 89.5 | 53.6 | 46.6 | 0.7 | 60.5 | 22.5 | 53.0 | 69.1 |
| Sturgess et al. [33] | 61.9 | 67.3 | 91.1 | 71.1 | 58.5 | 92.9 | 49.5 | 37.6 | 25.8 | 77.8 | 24.7 | 59.8 | 76.4 |
| Zhang et al. [62] | 85.3 | 57.3 | 95.4 | 69.2 | 46.5 | 98.5 | 23.8 | 44.3 | 22.0 | 38.1 | 28.7 | 55.4 | 82.1 |
| Kontschieder et al. [63] | | | | | | - | | | | | | 51.4 | 72.5 |
| Bulo et al. [64] | | | | | | - | | | | | | 56.1 | 82.1 |
| Yang et al. [65] | 80.7 | 61.5 | 88.8 | 16.4 | - | 98.0 | 1.09 | 0.05 | 4.13 | 12.4 | 0.07 | 36.3 | 73.6 |
| Tighe et al. [66] | 87.0 | 67.1 | 96.9 | 62.7 | 30.1 | 95.9 | 14.7 | 17.9 | 1.7 | 70.0 | 19.4 | 51.2 | 83.3 |
| Sturgess et al. [33] | 70.7 | 70.8 | 94.7 | 74.4 | 55.9 | 94.1 | 45.7 | 37.2 | 13.0 | 79.3 | 23.1 | 59.9 | 79.8 |
| Sturgess et al. [33] | 84.5 | 72.6 | 97.5 | 72.7 | 34.1 | 95.3 | 34.2 | 45.7 | 8.1 | 77.6 | 28.5 | 59.2 | 83.8 |
| Ladicky et al. [35] | 81.5 | 76.6 | 96.2 | 78.7 | 40.2 | 93.9 | 43.0 | 47.6 | 14.3 | 81.5 | 33.9 | 62.5 | 83.8 |
| Kendall et al. [39] | 75.0 | 84.6 | 91.2 | 82.7 | 36.9 | 93.3 | 55.0 | 37.5 | 44.8 | 74.1 | 16.0 | 62.9 | 84.3 |
| Badrinarayanan et al. [17] | 80.6 | 72.0 | 93.0 | 78.5 | 21.0 | 94.0 | 62.5 | 31.4 | 36.6 | 74.0 | 42.5 | 62.3 | 82.8 |
| Seg-Net [17] | 88.0 | 87.3 | 92.3 | 80.0 | 29.5 | 97.6 | 57.2 | 49.4 | 27.8 | 84.8 | 30.7 | 65.9 | 88.6 |
| Bayesian-SegNet-Basic [39] | 75.1 | 68.8 | 91.4 | 77.7 | 52.0 | 92.5 | 71.5 | 44.9 | 52.9 | 79.1 | 69.6 | 70.5 | 81.6 |
| Bayesian-Full-Segnet [39] | 80.4 | 85.5 | 90.1 | 86.4 | 67.9 | 93.8 | 73.8 | 64.5 | 50.8 | 91.7 | 54.6 | 76.3 | 86.9 |
| ECRU | 95.1 | 84.0 | 84.5 | 94.9 | 93.3 | 87.4 | 97.3 | 88.3 | 93.3 | 94.1 | 90.2 | 88.4 | 91.1 |


**Figure 5.** ECRU final results.

The proposed ECRU system is a technically efficient solution intended for driving assistance and will give a much better performance. The inference time is the time taken to recognize an object from a trained model. The inference time matters very much when we want real-time object detection. In this scenario, one of the fundamental objectives of the proposed research was to lessen the inference-time to strengthen the real-time detection. In this scenario, early approaches required additional aids (e.g., CRF, region proposals) to facilitate training. These CRFs add more computational complexities to the proposed network; which leads to additional time and resources. Our proposed network does not use any additional CRF, which leads to a small size. Encoder and decoder network architectures offer better performance and better results. The ECUR core segmentation engine has reduced the number of trainable parameters in the encoder network ($\sim$134 M to $\sim$30 M), whereas a very small amount on the decoder network ($\sim$0.5 M). In this way, the overall size of the network is reduced regarding trainable parameters. The inference time is the time taken to recognize an object from a trained model. The inference time matters very much when we want real-time object detection. However, the proposed CNN makes use of a lightweight segmentation engine (encoder-decoder network) without other aids (also discarding the fully-connected layers), which leads to improved inference time.

Table 4 shows the CamVid segmentation results and their analysis with earlier similar approaches with established benchmark results for size and inference time analysis. The network offers very efficient performance with fewer parameters and inference time. This table outlines a quantitative analysis of the proposed network regarding its small size and reduced inference time. As compared to early benchmark results, the system has a minimal size, which shows its prospective use for small devices. The second and most important aspects that the proposed network shows is the small inference time (55 ms), which is one of the key requirement for real-time NN processing.

**Table 4.** ECRU size and inference time analysis.

| Method | Total Size (Million) | Additional Aids Used | Inference Time |
|---|---|---|---|
| DeepLab [16] | <134.5 | validation set | n/a |
| FCN-8 [18] | 134.5 | multi-stage training | 210 ms |
| DeconvNet [52] | 276.7 | object proposals | 92 ms |
| Proposed CNN | <32 | no CRF, no multi-stage training | 55 ms |

Computer vision and machine learning are very vibrant fields of research. We are seeing new and more powerful network models emerging each year. New networks are adding more and more layers to enhance the network accuracy and performance. However, this also adds up to network size and computational workload. The key idea of the proposed ECRU is to cut down the network size and offer lower inference time. In terms of reducing the number of layers, we saw some performance downfalls, in a number of object segmentation classes. However, this performance can be improved. The proposed ECUR is a very dynamic architecture, which can depend on adding additional convolution blocks. The current network is using a five-block configuration; we believe that adding additional blocks can further improve network performance. Wang et al. [67] used a similar approach named "s-FCN-locwhich", which made use of two streams to process the original RGB images and contour maps. It offered higher performance, however exerting more load on an embedded unit of processing, which led to lower resource performance. As we stated at the start of this article, the fundamental objective of the proposed model was to offer a small and compact end-to-end network that provides better performance and fast processing. Additional layers can slow down and increase the size of the system. This problem can be settled with [1–4] network design features.

## 7. Conclusions

The research paper presented the idea of ECRU, a deep CNN model for semantic segmentation. The key idea to designing ECRU is to build a productive CNN for road-scene understanding. Another key objective is to offer a better CNN model, which is improved both in terms of trainable parameters and storage requirements. We have analysed ECRU and assessed its performance with other important benchmark variants in terms of design and efficiency. It is assessed that ECRU offers a great deal of enhanced performance over early benchmark results. It makes use of the special method of re-using pooling indices, which leads to fewer computation parameters and helps to reduce inference time. The proposed network model is well suited for scene understanding applications. It could be employed for driving assistance to offer enhanced vehicle safety and more generally road safety. This network is conceptually suitable to train heterogeneous data with pixel-wise segmentation annotations and/or image-level class labels. Recently, ultrasonic and some other kinds of radar technologies have been employed in smart vehicles. ADAS also needs similar radar technologies, which can be applicable in different kinds of weather and road conditions. The application of vision-based CNN ECRU along with an automobile radar system could be helpful, particularly in different road and weather conditions (e.g., ultrasonic signals cannot detect objects in blind-spots). The design of ECRU offers a deep learning-based automatic road scene understanding model that can operate in any lighting or weather conditions. It also provides a minimal and compact design that can operate in a real-time environment (as shown from the "higher inference rate" in the results above). Overall, the proposed solution is a cost-effective and better performing system for futuristic auto driving systems.

**Funding:** This research received no external funding

**Conflicts of Interest:** The author declares no conflict of interest.

## References


1. Kirillov, A.; He, K.; Girshick, R.; Rother, C.; Dollár, P. Panoptic Segmentation. *arXiv* **2018**, arxiv:1801.00868.
2. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
3. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
4. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arxiv:1706.05587.
5. Everingham, M.; Winn, J. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Development Kit. In *Pattern Analysis, Statistical Modelling and Computational Learning*; Tech. Report; European Commission: Brussels, Belgium, 2011.
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Washington, DC, USA, 7–13 December 2015; pp. 1026–1034.
7. Li, S.; Chan, A.B. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 332–347.
8. Giusti, A.; Cireşan, D.C.; Masci, J.; Gambardella, L.M.; Schmidhuber, J. Fast image scanning with deep max-pooling convolutional neural networks. *arXiv* **2013**, arxiv:1302.1700.
9. Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 898–916. [CrossRef] [PubMed]
10. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231. [CrossRef] [PubMed]
11. Shotton, J.; Sharp, T.; Kipman, A.; Fitzgibbon, A.; Finocchio, M.; Blake, A.; Cook, M.; Moore, R. Real-time human pose recognition in parts from single depth images. *Commun. ACM* **2013**, *56*, 116–124. [CrossRef]
12. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
13. Couprie, C.; Farabet, C.; Najman, L.; LeCun, Y. Indoor semantic segmentation using depth information. *arXiv* **2013**, arxiv:1301.3572.
14. Farabet, C.; Couprie, C.; Najman, L.; LeCun, Y. Learning hierarchical features for scene labelling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1915–1929. [CrossRef] [PubMed]
15. Höft, N.; Schulz, H.; Behnke, S. Fast semantic segmentation of RGB-D scenes with GPU-accelerated deep neural networks. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*; Springer: Cham, Switzerland, 2014; pp. 80–85.
16. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arxiv:1412.7062.
17. Badrinarayanan, V.; Handa, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv* **2015**, arxiv:1505.07293.
18. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
19. Koltun, V. Efficient inference in fully connected crfs with gaussian edge potentials. Advances in Neural Information Processing Systems. In Proceedings of the Twenty-Fifth Conference on Neural Information Processing Systems (NIPS 2011), Granada, Spain, 14 December 2011.
20. Jung, H.; Choi, M.K.; Soon, K.; Jung, W.Y. End-to-End Pedestrian Collision Warning System based on a Convolutional Neural Network with Semantic Segmentation. *arXiv* **2016**, arxiv:1612.06558.
21. Xie, K.; Ge, S.; Ye, Q.; Luo, Z. Traffic Sign Recognition Based on Attribute-Refinement Cascaded Convolutional Neural Networks. In *Pacific Rim Conference on Multimedia*; Springe: Cham, Switzerland, 2016; pp. 201–210.

22. Huval, B.; Wang, T.; Tandon, S.; Kiske, J.; Song, W.; Pazhayampallil, J.; Andriluka, M.; Rajpurkar, P.; Migimatsu, T.; Cheng-Yue, R.; et al. An empirical evaluation of deep learning on highway driving. *arXiv* **2015**, arxiv:1504.01716.
23. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arxiv:1409.1556.
24. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014; pp. 580–587.
25. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
26. Tompson, J.J.; Jain, A.; LeCun, Y.; Bregler, C. Joint training of a convolutional network and a graphical model for human pose estimation. Advances in Neural Information Processing Systems. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2014), Montréal, QC, Canada, 8–13 December 2014; pp. 1799–1807.
27. Zbontar, J.; LeCun, Y. Computing the stereo matching cost with a convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1592–1599.
28. Gupta, S.; Girshick, R.; Arbeláez, P.; Malik, J. Learning rich features from RGB-D images for object detection and segmentation. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 345–360.
29. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
30. Wolf, R.; Platt, J.C. Postal address block location using a convolutional locator network. In *Advances in Neural Information Processing Systems*; Advances in Neural Information Processing Systems: Denver, CO, USA, 1994; p. 745.
31. Ning, F.; Delhomme, D.; LeCun, Y.; Piano, F.; Bottou, L.; Barbano, P.E. Toward automatic phenotyping of developing embryos from videos. *IEEE Trans. Image Process.* **2005**, *14*, 1360–1371. [CrossRef] [PubMed]
32. Shotton, J.; Johnson, M.; Cipolla, R. Semantic texton forests for image categorization and segmentation. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
33. Sturgess, P.; Alahari, K.; Ladicky, L.; Torr, P.H. Combining appearance and structure from motion features for road scene understanding. In Proceedings of the 23rd British Machine Vision Conference (BMVC 2012), London, UK, 3–7 September 2009.
34. Brostow, G.J.; Fauqueur, J.; Cipolla, R. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognit. Lett.* **2009**, *30*, 88–97. [CrossRef]
35. Ladickỳ, L.; Sturgess, P.; Alahari, K.; Russell, C.; Torr, P.H. What, where and how many? Combining object detectors and crfs. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2010; pp. 424–437.
36. Kang, F.; Han, S.; Salgado, R.; Li, J. System probabilistic stability analysis of soil slopes using Gaussian process regression with Latin hypercube sampling. *Comput. Geotech.* **2015**, *63*, 13–25. [CrossRef]
37. Kang, F.; Xu, Q.; Li, J. Slope reliability analysis using surrogate models via new support vector machines with swarm intelligence. *Appl. Math. Model.* **2016**, *40*, 6105–6120. [CrossRef]
38. Huang, F.J.; Boureau, Y.L.; LeCun, Y. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
39. Kendall, A.; Badrinarayanan, V.; Cipolla, R. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv* **2015**, arxiv:1511.02680.
40. Yasrab, R.; Gu, N.; Zhang, X. An Encoder-Decoder Based Convolution Neural Network (CNN) for Future Advanced Driver Assistance System (ADAS). *Appl. Sci.* **2017**, *7*, 312. [CrossRef]
41. Yasrab, R.; Gu, N.; Xiaoci, Z.; Asad-Khan. DCSeg: Decoupled CNN for Classification and Semantic Segmentation. In Proceedings of the 2017 IEEE Conference on Knowledge and Smart Technologies (KST), Pattaya, Thailand, 1–4 February 2017; pp. 1–6.

42. Yasrab, R.; Gu, N.; Xiaoci, Z. SCNet: A Simplified Encoder-Decoder CNN for Semantic Segmentation. In Proceedings of the 2016 5th International Conference on Computer Science and Network Technology (ICCSNT), Changchun, China, 10–11 December 2016; pp. 1–6.
43. Zolock, J.; Senatore, C.; Yee, R.; Larson, R.; Curry, B. *The Use of Stationary Object Radar Sensor Data from Advanced Driver Assistance Systems (ADAS) in Accident Reconstruction*; Technical Report, SAE Technical Paper; SAE: Warrendale, PA, USA, 2016.
44. Kedzia, J.C.; de Souza, P.; Gruyer, D. Advanced RADAR sensors modeling for driving assistance systems testing. In Proceedings of the 2016 10th European Conference on Antennas and Propagation (EuCAP), Davos, Switzerland, 10–15 April 2016; pp. 1–2.
45. Deng, L.; Yang, M.; Li, H.; Li, T.; Hu, B.; Wang, C. Restricted Deformable Convolution based Road Scene Semantic Segmentation Using Surround View Cameras. *arXiv* **2018**, arxiv:1801.00708.
46. Laugraud, B.; Piérard, S.; Van Droogenbroeck, M. LaBGen-P-Semantic: A First Step for Leveraging Semantic Segmentation in Background Generation. *J. Imaging* **2018**, *4*, 86. [CrossRef]
47. Zhang, X.; Chen, Z.; Wu, Q.J.; Cai, L.; Lu, D.; Li, X. Fast Semantic Segmentation for Scene Perception. *IEEE Trans. Ind. Inform.* **2018**. [CrossRef]
48. Kalith, A.S.; Mohanapriya, D.; Mahesh, K. Video Scene Segmentation: A Novel Method to Determine Objects. *Int. J. Sci. Res. Sci. Technol.* **2018**, *4*, 90–94.
49. Darwich, A.; Hébert, P.A.; Bigand, A.; Mohanna, Y. Background Subtraction Based on a New Fuzzy Mixture of Gaussians for Moving Object Detection. *J. Imaging* **2018**, *4*, 92. [CrossRef]
50. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*; Google Inc.: Mountain View, CA, USA, 2017; Volume 4, p. 12.
51. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arxiv:1502.03167.
52. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1520–1528.
53. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv* **2015**, arxiv:1511.07289.
54. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
55. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; ACM: New York, NY, USA, 2014; pp. 675–678.
56. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; MIT Press: Cambridge, MA, USA, 2012.
57. Drucker, H.; Le Cun, Y. Improving generalization performance in character recognition. In Proceedings of the 1991 IEEE Workshop on Neural Networks for Signal Processing, Princeton, NJ, USA, 30 September–1 October 1991; pp. 198–207.
58. Goodfellow, I.J.; Warde-Farley, D.; Mirza, M.; Courville, A.C.; Bengio, Y. Maxout networks. *ICML* **2013**, *28*, 1319–1327.
59. Zeiler, M.D.; Fergus, R. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv* **2013**, arxiv:1301.3557.
60. Jarrett, K.; Kavukcuoglu, K.; Lecun, Y. What is the best multi-stage architecture for object recognition? In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 2146–2153.
61. Brostow, G.J.; Shotton, J.; Fauqueur, J.; Cipolla, R. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2008; pp. 44–57.
62. Zhang, C.; Wang, L.; Yang, R. Semantic segmentation of urban scenes using dense depth maps. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2010; pp. 708–721.
63. Kontschieder, P.; Bulo, S.R.; Bischof, H.; Pelillo, M. Structured class-labels in random forests for semantic image labelling. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2190–2197.

64. Rota Bulo, S.; Kontschieder, P. Neural decision forests for semantic image labelling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014; pp. 81–88.
65. Yang, Y.; Li, Z.; Zhang, L.; Murphy, C.; Ver Hoeve, J.; Jiang, H. Local label descriptor for example based semantic image labelling. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2012; pp. 361–375.
66. Tighe, J.; Lazebnik, S. Superparsing. *Int. J. Comput. Vis.* **2013**, *101*, 329–349. [CrossRef]
67. Wang, Q.; Gao, J.; Yuan, Y. Embedding structured contour and location prior in siamesed fully convolutional networks for road detection. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 230–241. [CrossRef]

© 2018 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).