Fusing Sensor Data from both Depth Maps and Visual Imagery to Improve 3D Computer Vision

Anonymous Authors[1]

Abstract

Many robotics platforms that feature depth mapping technologies also sport visual cameras for taking in RGB data. Processes for utilizing sensor fusion between the two modalities for improved computer vision is currently rather immature, but methods are developing quickly. This paper aims to review four recent publications on the topic and re-implement one of them with tweaks to the performance to fit available computational resources and data sets. A modified convolution operation is tested on a recent aerial image dataset, but no significant difference is found between it and the traditional convolution operation.

1. Introduction

Computer vision is a vitally important tool for many modern devices that interact with physical objects in the real world. While there exist effective solutions for reliable vision processing in controlled environments, next-generation technologies such as autonomous vehicles require even better performance in even more challenging and unstructured environments. Many mobile robotic platforms that are being used to develop these technologies include depth sensors (such as LIDAR, time-of-flight, structured light, or stereo camera sensors) (Lopes et al., 2022) as well as digital cameras (to capture visual light in red, green, and blue ranges). These two modalities–RGB and depth, often abbreviated together as "RGB-D"—provide complementary information, as many features that are indistinguishable with only depth information are much easier to identify with visual cues, while many features that are indistinguishable with visual cues are much easier to identify with depth information (Wang & Neumann, 2018). However, there has been limited research on how to most effectively fuse information from both modalities.

This project aims to explore current methods for fusing depth data with visual imagery in order to better perform computer vision tasks in three dimensions. Three publications are considered: one that performs semantic segmentation, one that performs object recognition, and one that performs salient object detection. Semantic segmentation is a computer vision task that aims to divide a scene into separate segments and give each segment a meaningful label. Salient object detection is a similar task, which aims to pick out only the most prominent segment in a scene.

Ideally, techniques from these papers will help with improving real-time scene understanding for robotic systems, such as the one that this author is developing at Purdue University.

2. Literature Review

2.1. Depth-aware CNNs

The first paper of those reviewed here was published in 2018 at the IEEE/CVF Conference on Computer Vision and Pattern Recognition, or 2018 CVPR (Wang & Neumann, 2018). This publication discusses the use of Convolutional Neural Networks (CNNs) for semantic segmentation of RGB-D data.

2.1.1. Strengths

The novel approach in this publication bakes the depth information from a depth map into the convolution operations performed on the associated RGB image, rather than adding parameters to the model. This means that the technique has minimal performance losses compared to a traditional CNN. Another intrinsic strength to that approach is that its modified CNNs are all two-dimensional affairs, rather than "2.5D" or even 3D operations, which keeps memory and computational requirements low; two-dimensional CNNs are

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

highly parallelizable, which means that they are very suitable for GPU acceleration.

In addition, the algorithms described in this paper are very flexible and can be combined easily with other CNN techniques, as the authors did in their tests with the HHA (Horizontal disparity, Height above ground, and norm Angle) two-stream approach they replicated from (Gupta et al., 2014).

2.1.2. Weaknesses

While restricting itself to only two dimensions is wise in terms of reducing the computational cost of this technique, this compromise makes it more challenging to export the results to a more complete model of the physical space processed. As a result, the output is still only a two-dimensional projection of the space, rather than a three-dimensional segmented model of the space. Additionally, their model can only work with depth maps; some 3D sensors output directly to a point cloud, which would need to be projected down to two dimensions in order to create such an input.

2.2. ImVoteNet

The next paper to discuss was published two years later at the same conference, the 2020 CVPR (Qi et al., 2020). This one, rather than performing semantic segmentation, attempts three-dimensional object recognition instead. It builds very directly on a previous paper (Qi et al., 2019), which introduced a novel algorithm for performing object detection directly on geometric data from a point cloud, without any visual imagery inputs at all.

2.2.1. Strengths

This approach works directly in three-dimensional space, which allows it to generate a three-dimensional model. This allows better planning for tasks such as moving robotic systems through physical space. Also, the inclusion of color information allows the detection algorithm to work much more reliably with sparser depth data, which gives it more flexibility to work with lower-cost, less-capable depth estimation technologies–another common feature of robotic systems.

In addition, this approach allows using inputs from two-dimensional recognition algorithms, which means that it can both benefit from any future developments in that subject, as well as leverage existing two-dimensional systems that are generally much more mature than their three-dimensional counterparts. A key feature of this strength is that two-dimensional inputs are considered simultaneously with three-dimensional cues. Many similar systems that attempt to include two-dimensional information in a three-dimensional detection scheme do so in a cascaded fashion, which means that any features missed in the first level are impossible to recover in the second. By considering both at the same step, this technique allows both modalities (three-dimensional depth data and two-dimensional RGB data) to work in tandem, such that any shortcomings in one will not negatively impact cues from the other.

2.2.2. Weaknesses

As this technique relies on a more sophisticated pipeline of three-dimensional processes, it is more computationally intensive than one that works directly on two-dimensional data. This may somewhat counterbalance its strength in working with lower-cost depth estimation technologies, as it requires more expensive computing hardware in turn.

2.3. BBS-Net

The third paper was published at the 2020 European Conference on Computer Vision, or 2020 ECCV (Fan et al., 2020). This publication describes a system for salient object detection (SOD), which is related to image segmentation (like the paper in Section 2.1) but seeks to identify the most visually prominent objects, rather than just divide the image into separate objects. Much like that previous paper, it also relies on CNNs to perform this task, but it does so in a specialized cascading fashion.

2.3.1. Strengths

This approach uses a "bifurcated backbone strategy" or "BBS" to split work between two sets of CNNs, a "teacher" decoder and a "student" decoder. One is more specialized in working with fine details, which allows it to refine boundaries of salient objects more effectively, while the other is more specialized in working with larger features, which helps the system reject low-level noise in the input. By chaining feedback from both decoders into one another, the system can iterate through multiple cycles in order to converge on a more accurate output. This gives the model a significant advantage in both noise reduction and boundary precision, tasks which are usually somewhat mutually exclusive.

The backbone structure of this technique is somewhat modular as well, which means that pieces of it can be replaced with other algorithms to produce outputs for related but different tasks in computer vision. Such

programs would gain the benefits of this bifurcated system, in that they could draw on both RGB and depth data iteratively in order to gain the benefits of both modalities, while requiring relatively little new work in order to adapt to those new tasks.

2.3.2. Weaknesses

This system utilizes CNNs much like the 2018 paper discussed earlier. However, it loses some of those CNNs' computational efficiency by cascading feedback loops between the two decoders; the first (teacher) decoder must finish its output before the second (student) decoder can even begin work. This prevents parallelizing the workload as completely as could be done under a non-cascaded scheme. That said, this technique deserves credit for limiting the effects of this tradeoff, as it requires many fewer iterations through those feedback cycles than comparable cascaded CNNs.

2.4. Shape-Aware CNNs

The fourth and most recent paper was published in 2021 at the International Conference on Computer Vision, or 2021 ICCV (Cao et al., 2021). This paper is similar to that in section 2.1 above, as it also discusses the use of Convolutional Neural Networks (CNNs) for semantic segmentation of RGB-D data. However, their modified CNN operation is slightly simpler, reducing to an ordinary CNN in certain cases.

2.4.1. Strengths

The operation described in this publication uses a modified CNN algorithm to better utilize the unique information that depth data brings to computer vision in order to better learn the weights of the kernel. However, once those weights have been determined during the training phase, they remain constant during evaluation, which means that CNNs trained with this method are completely equivalent to vanilla, traditional CNNs once trained. This allows models to remain more lightweight and portable than those requiring specialized modified operations. Additionally, these CNNs are two-dimensional, which keeps their memory and computational requirements lower and more suitable for GPU acceleration.

2.4.2. Weaknesses

Just as noted in section 2.1 above, a crucial weakness of this technique is that it segments only a two-dimensional projection of the space, rather than a full three-dimensional model of the real world. While this wisely reduces computational requirements, it also prevents creation of a more realistic model of the physical space as a whole. Likewise, it is incompatible with point cloud depth information, as it requires two-dimensional image inputs.

This technique has one weakness that is unique among those discussed above. In order to integrate the depth information into the modified CNN, the depth information is simply channelwise concatenated to the RGB information. In previous studies (Hazirbas et al., 2016) (Ma et al., 2017) (Wang et al., 2016), this has been shown to be better than simply ignoring the depth information, but by treating it the same as RGB information there is often missed potential (Wang & Neumann, 2018). This paper's technique has traded its ability to fully utilize the unique perspective that the depth domain offers in order to maintain backwards compatibility with traditional CNNs. That said, it still manages to utilize at least some part of that unique information with its specialized training process, so the tradeoff is not so severe as it may first appear.

3. Implementation

3.1. Implementation Motivation and Goals

This study originally intended to replicate the operations described in section 2.1 above with a newer dataset that was published a few years afterwards. This author spent a significant amount of time porting a codebase that implemented the depth-aware convolutions from a deprecated software stack (Mauceri, 2021) to a more modern system. While updated CUDA operations were successfully compiled–what this author naively expected to be the most challenging part of the process–the supporting code used to implement those CUDA operations in a modern neural network was unable to be repaired or successfully replaced. As a result, the project underwent a late shift from an attempt to impelment the depth-aware convolution (Wang & Neumann, 2018) to an attempt to use the same general model architecture while using the shape-aware convolution (Cao et al., 2021) in its stead. As a result, some parts of this experiment were based on the depth-aware paper, while others were based on the shape-aware paper.

The general goal for this experiment was to test the efficacy of the shape-aware convolution relative to a traditional ordinary convolution for semantic segmentation of RGB-D data. The dataset used to train and test the network consisted of synthetic aerial imagery (Chen et al., 2020). This was deemed a novel choice for the experiment because both the original

depth-aware and shape-aware papers tested their algorithms primarily on indoor imagery. The outdoor imagery from this new aerial dataset thus provides an interesting change of domain which tests models' abilities to generalize to a new setting, and, if successful, would provide a useful tool for mobile robotic platforms that utilize computer vision for situational understanding: another research interest of this author. Additionally, this dataset comes with pixel-accurate labels for semantic segmentation, allowing the network to be trained using supervised learning, intended to simplify the training process.

3.2. Methods

The program created for this study can be broadly divided into three pieces. First, the VALID dataset (Chen et al., 2020) is loaded from a subdirectory VALID2020 based on the metadata files included by the creators. Next, a neural network is constructed based generally on the architecture that was originally used to test the depth-aware convolution (Wang & Neumann, 2018). Finally, the network is trained and tested using the training and testing images from the dataset loaded previously.

3.2.1. Dataset Loading

VALID provides several different sets of images so that the dataset can be used for many different purposes. In this experiment, two main parts are used: VALID-Depth, which includes both RGB and depth imagery, and VALID-Seg, which includes ground truth images for both panoptic segmentation (unused in this experiment) and semantic segmentation. (Panoptic segmentation is a different type of segmentation that segments different instances of the same class of object as separate individuals, in addition to segmenting the scene by class of objects generally. It can be considered a combination of semantic segmentation and instance segmentation (Kirillov et al., 2019).)

The splits folder that comes with VALID provides several different lists of VALID scenes, split into training, testing, and validation sets. To load the dataset, these lists are read in order to create a list of metadata files for each VALID scene in the appropriate set. Each of those metadata files contains a wealth of information about the associated scene, but for this experiment the only pieces extracted were the identities of the RGB, depth, and ground truth semantic segmentation images: 3345 of each for training and 2230 of each for testing. A final metadata file is read at the end, which provides a map between the 31 different colors used in the ground truth semantic segmentation images associated categories.

As the goal for the segmentation is to divide the RGB-D image into segments by category, not color, the ground truth images are first converted from their RGB encoding (3-channel images) to a more direct 1-channel array of category values (ranging from 0 to 30). The targets used for training are use in this 1-channel encoding.

Once a set of RGB, depth, and ground truth segmentation images are loaded in from the VALID folder, they are then preprocessed in order to synthetically augment the effective sample size. First, each one is cropped by a random amount (up to 75%) centered on a random location on the original image. (All three images in the set are cropped in the same way such that features remain aligned between them.) Next, there is a random chance (50%) that the scene may be mirrored. A small amount of random color jittering is performed on the RGB image, and then all three images in the set are scaled down equally to the final size of 256 pixels square.

3.2.2. Model Architecture

The model used in this experiment is primarily based on the one used in the paper that introduced the depth-aware convolution (Wang & Neumann, 2018). Originally, this experiment attempted to start from the Pytorch 1.5 modernization (Mauceri, 2021) of the codebase published with that paper. However, that model was structured in a way that made it difficult to read, modify, and maintain, the same structure was rebuilt, loosely based on a different codebase (Lei, 2020) but primarily constructed from scratch. The core architecture used in the paper comes from an older network called DeepLab (Chen et al., 2014). Some of the traditional CNNs used in that architecture were replaced with shape-aware convolutions, in the same places that depth-aware convolutions were used for the depth-aware paper. The shape-aware convolution operations were imported from a codebase published by one of the authors of the original shape-aware paper (Leng, 2021).

The model worked in two phases. First, RGB and depth information were channelwise concatenated and fed through a DeepLab-inspired encoder, which output a 2048-channel latent representation of the input. This architecture is summarized in Figure 1. The output from that encoder was then put through three transpose convolutions in order to decode the latent representation back into a full-resolution estimate of the segmentation.

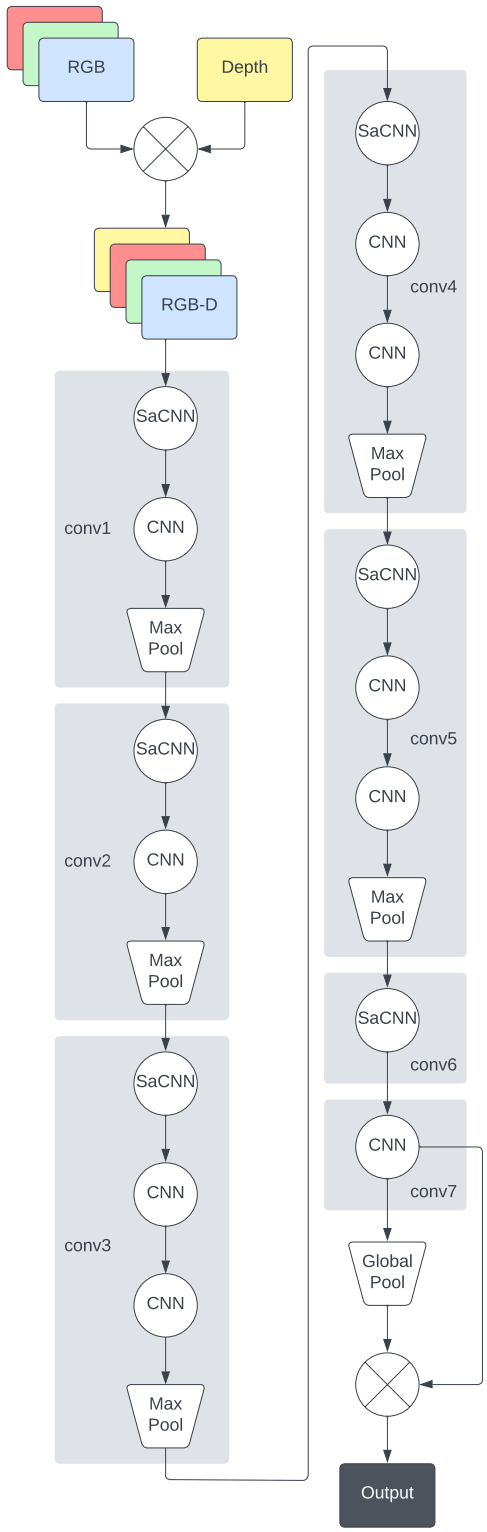

Figure 1. A block diagram of the encoder architecture, based on an adaptation of DeepLab v1 from (Wang & Neumann, 2018).

Model	Pixel Accuracy (%)
Shape-aware (ours)	22.82
Control	22.74
State-of-the-art	92.70

Table 1. Both the shape-aware and the control model fared extremely poorly against the state-of-the-art.

3.2.3. Training and Testing

Images were loaded into the model in batches of 10 while training and 20 while testing. Cross entropy was used to quantify loss, and the adam optimizer (Kingma & Ba, 2014) was used for gradient descent. The shape-aware and the control model were each trained for seven epochs on an nVidia RTX 3080 GPU, taking approximately 90 minutes apiece. After training and testing was complete, a few sets of images were saved for manual review. Segmented images were converted from their 1-channel indexed form to false color using the same RGB mapping that VALID uses for their ground truth segmentation images.

All code used in this study is available on GitHub (Author, 2022).

4. Results

The performance of the shape-aware and control networks are given in Table 1. Pixel accuracy refers to the total percentage of pixels estimated to belong to the correct category. As shown, neither method trained for this experiment fared very well compared to the state-of-the-art for this dataset (Zhao et al., 2017). Several example images are shown in Figure 2, which quite obviously scored poorly.

5. Discussion

Both models from this experiment had very poor accuracy. They tended to assume that everything in the image was a road, rather than segmenting different portions of the image as belonging to different categories as intended. As a result, it is difficult to suggest that one is better or worse than the other; the 0.08% difference in their pixel accuracy is negligible.

There are several likely reasons for the rather severe shortcomings of the models trained for this experiment. Both of them were trained from scratch, rather than beginning from a pre-trained model. It may have been wise to try to adapt pre-trained weights from another DeepLab-inspired model to provide a better starting point for training the models. This is a common approach in the literature (Lopez-Campos &

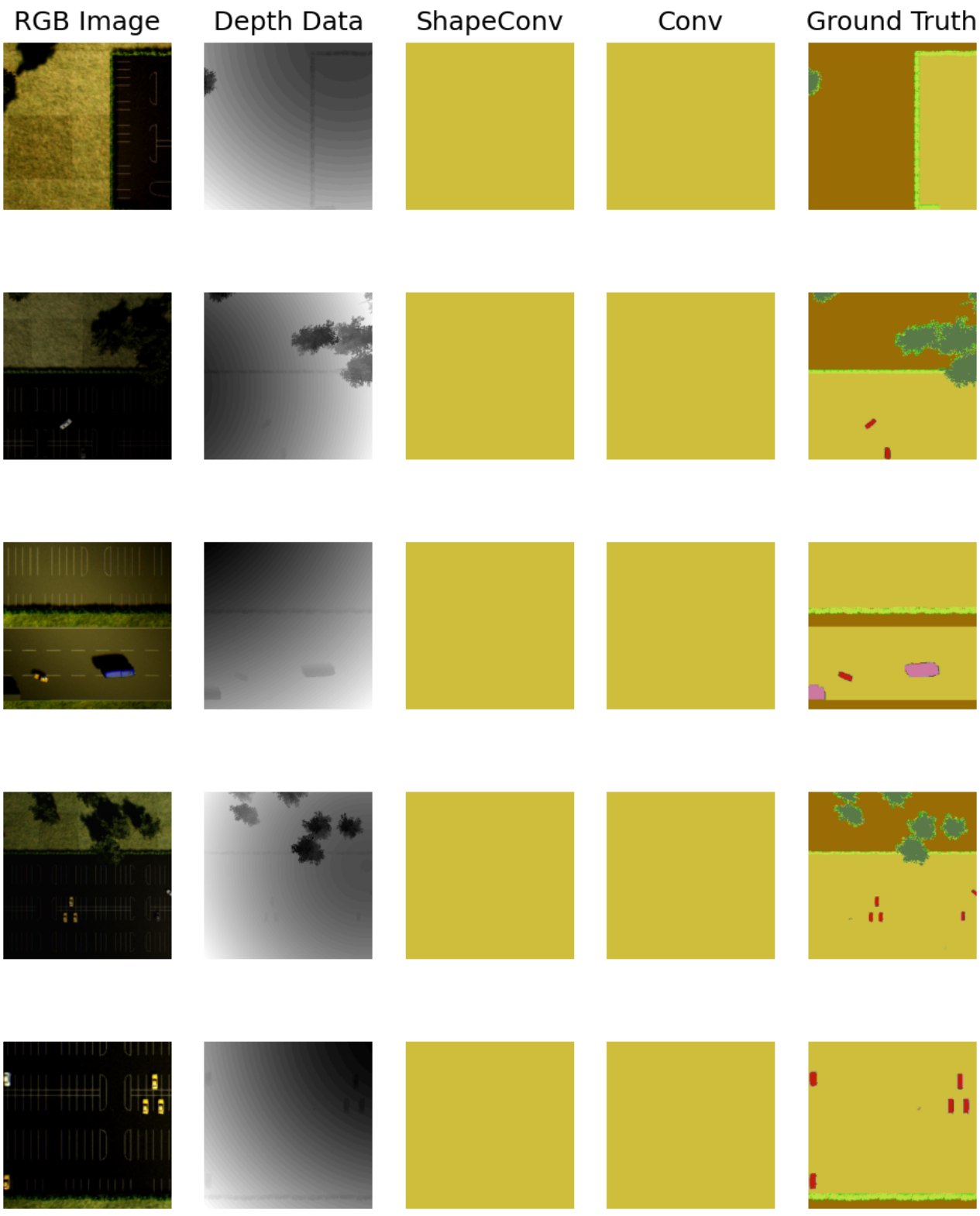

Figure 2. Five sample images from the experiment. From left to right: the RGB input image (after random cropping, jitter, etc), the depth input image (after random cropping), our shape-aware model's segmentation estimate, our control (traditional CNN) model's segmentation estimate, the ground truth segmentation.

Martinez-Carranza, 2021) (Wang & Neumann, 2018) but not universal (Mehta et al., 2022).

Another likely reason the models in this experiment failed to converge on any accuracy is probably due to a simple lack of training. Seven epochs on over 3000 images initially seems like a lot of training, but for a model with many millions of parameters like this one, it is not enough. Unfortunately this author did not have enough time or computational resources to train the models significantly more than was done for this study.

A final concern about the models used in this experiment lie in the decoder architecture. A few simple transpose convolutions were used for this experiment, as the primary goal for the decoder module was just to convert the very small-resolution latent representation of the segmentation up to a full 256 pixels square, so that it could be compared against the ground truth labels. However, that is hardly the most elegant architecture. A more thoughtful design, perhaps similar to that used in DeconvNet (Simonyan & Zisserman, 2014) or the U-shaped decoder from FuseNet (Hazirbas et al., 2016), may significantly improve the performance of these models without compromising the DeepLab-inspired encoder portion or greatly increasing the computation cost of the model.

6. Conclusion

Unfortunately, the results from this experiment are inconclusive. There is a negligible difference between the accuracy of the shape-aware model and the control model. However, there are enough doubts about whether they had truly converged during training that such a result cannot be construed to mean that the shape-aware model is no better than the control.

If another attempt at a project like this one were to be made, it is recommended that one chooses to either (i) use a smaller and less challenging dataset and appropriately simple model for the experiment, or, perhaps, (ii) start from a pre-trained model in order to converge on better solutions more quickly.

References

Author, A. Using shape-aware convolution on the VALID imageset, 2022. URL Anonymous.

Cao, J., Leng, H., Lischinski, D., Cohen-Or, D., Tu, C., and Li, Y. ShapeConv: Shape-aware convolutional layer for indoor RGB-D semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7088–7097, 2021.

Chen, L., Liu, F., Zhao, Y., Wang, W., Yuan, X., and Zhu, J. VALID: A comprehensive virtual aerial image dataset. In 2020 IEEE international conference on robotics and automation (ICRA), pp. 2009–2016. IEEE, 2020.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062, 2014.

Fan, D.-P., Zhai, Y., Borji, A., Yang, J., and Shao, L. BBS-Net: RGB-D salient object detection with a bifurcated backbone strategy network. In European conference on computer vision, pp. 275–292. Springer, 2020. URL https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123570273.pdf.

Gupta, S., Girshick, R., Arbeláez, P., and Malik, J. Learning rich features from RGB-D images for object detection and segmentation. In European conference on computer vision, pp. 345–360. Springer, 2014.

Hazirbas, C., Ma, L., Domokos, C., and Cremers, D. Fusenet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. In Asian conference on computer vision, pp. 213–228. Springer, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2014.

Kirillov, A., He, K., Girshick, R., Rother, C., and Dollár, P. Panoptic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9404–9413, 2019.

Lei, W. Pytorch 1.2 implementation of DeepLab v1, 2020. URL https://github.com/wangleihitcs/DeepLab-V1-PyTorch.

Leng, H. ShapeConv: Shape-aware convolutional layer for pytorch 1.7, 2021. URL https://github.com/hanchaoleng/shapeconv.

Lopes, A., Souza, R., and Pedrini, H. A Survey on RGB-D datasets. Computer Vision and Image Understanding, 222:103489, 2022.

Lopez-Campos, R. and Martinez-Carranza, J. Espada: Extended synthetic and photogrammetric aerial-image dataset. IEEE Robotics and Automation Letters, 6(4):7981–7988, 2021.

Ma, L., Stückler, J., Kerl, C., and Cremers, D. Multi-view deep learning for consistent semantic mapping with RGB-D cameras. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 598–605. IEEE, 2017.

Mauceri, C. Pytorch 1.5 implementation of depth-aware CNN for RGB-D segmentation, 2021. URL https://github.com/crmauceri/DepthAwareCNN-pytorch1.5.

Mehta, D., Mehta, A., Narang, P., Chamola, V., and Zeadally, S. Deep learning enhanced uav imagery for critical infrastructure protection. IEEE Internet of Things Magazine, 5(2):30–34, 2022.

Qi, C. R., Litany, O., He, K., and Guibas, L. J. Deep hough voting for 3d object detection in point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9277–9286, 2019.

Qi, C. R., Chen, X., Litany, O., and Guibas, L. J. ImVoteNet: Boosting 3D object detection in point clouds with image votes. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4404–4413, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/papers/Qi_ImVoteNet_Boosting_3D_Object_Detection_in_Point_Clouds_With_Image_CVPR_2020_paper.pdf.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

Wang, J., Wang, Z., Tao, D., See, S., and Wang, G. Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks. In European Conference on Computer Vision, pp. 664–679. Springer, 2016.

Wang, W. and Neumann, U. Depth-aware CNN for RGB-D segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 135–150, 2018. URL https://openaccess.thecvf.com/content_ECCV_2018/papers/Weiyue_Wang_Depth-aware_CNN_for_ECCV_2018_paper.pdf.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2881–2890, 2017.