TECHNISCHE UNIVERSITÄT
CHEMNITZ

Depth estimation from stereo camera based on Convolutional Neural Networks

Seminar Report

Chair of Computer Engineering
Dept. of Computer Science

Submitted by: Ranjitha Subramaniam
Matrikel Nr.: 516518
Submission date: 10.03.2019

Supervising tutor: Prof. Dr. Wolfram Hardt
Dipl. Inf. Rene Schmidt
Dipl.-Ing. Philip Parsch
Prof. Dr. Uranchimeg Tudevdagva

Abstract

Depth estimation from the two-dimensinal images has several applications and this report emphasizes the significance of accurate depth perception in the recent technological advancements. The report is organized with an introduction to the stereo image pairs followed by which different terms concerning the stereo geometry are explained. Stereo depth perception was carried out using conventional algorithms before the Convolutional Neural Networks gained their significance. These stereo matching algorithms could not provide efficient depth perception under specific scenarios. As we are in the era of Assisted and autonomous driving, accurate depth estimation for the real world scenario becomes essential. The recent popularity of deep learning resulted in the proposal of numerous stereo depth perception architectures using Convolutional Neural Networks. A few recently proposed CNN architectures for the real time depth perception of stereo images are discussed in this report. Additionally, an evaluation of these architectures is documented by comparing the different aspects of each network including their pros and cons.

Keywords: Stereo depth perception, disparity,Epipolar Geometry,Rectification, Convolutional Neural Network (CNN), Convolution, Max-pooling, Loss function, Optimisation.

Contents

List of Figures

List of Tables

List of Abbreviations

CNN	Convolutional Neural Network
2D	Two-Dimensional
3D	Three-Dimensional
ReLU	Rectified Linear Unit
FC	Fully Connected
MLP	Multi Layer Perceptron
CC	Cross Correlation
MSE	Mean Squared Error
AdaGrad	Adaptive Gradient
Adam	Adaptive moment estimation
SGD	Stochastic Gradient Descent
RMSE	Root Mean Squared Error
SSD	Sum of Squared Differences
GPU	Graphics Processing Unit

1 Introduction

Since few decades, estimation of depth information from an image is an important factor and an inevitable challenge in computer vision. Depth provides knowledge about the position of an object in the real world. Depth Estimation, in principle, refers to the extraction of spatial structure of a scene from an Image. An image is generally formed by the projection of a real world scene into a two-dimensional (2D) image plane. This projection results in the reduction of one dimension which basically holds the depth information of the scene. Many applications, especially, the recent advancements of automotive domain like autonomous and assisted driving scenarios, demand the extraction of this depth information so that the distance of stationary and moving obstacles from the vehicle could be predicted. Also, this impels the need for faster and precise depth estimation as immediate actions are to be triggered if in case any obstacles are detected. Depth perception is possible with monocular as well as stereo images. This study focuses on the depth perception of stereo images using Convolutional Neural Networks (CNN) as they provide much accurate and faster depth perception. Fig.1.1 represents a general model for stereo depth perception using CNN[11].

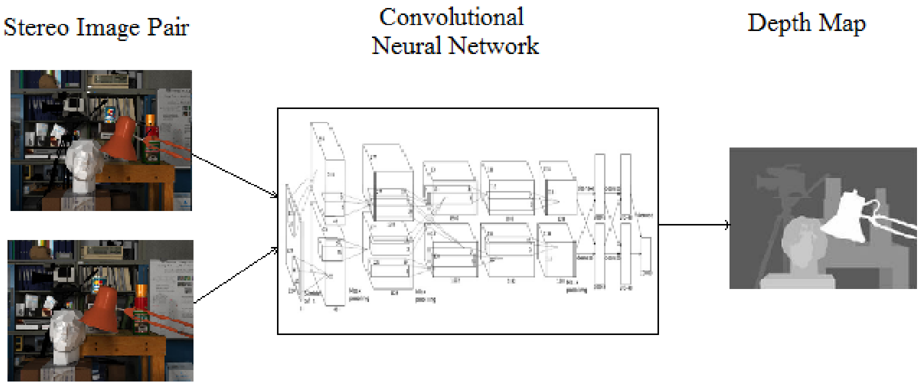

Figure 1.1: A model for Stereo depth perception using CNN

2 Stereo Depth Estimation

2.1 Introduction to stereo Images

Fig2.1 represents a stereo image pair. Stereo images symbolize a pair of images that are captured simultaneously using a stereo camera and they both represent the same scene. A stereo camera is composed of two lenses that are displaced by a known distance. Stereo cameras are inspired by and comparable with the Human Binocular Vision, where the two eyes are used to capture images of the same scene and with these images, the depth information is perceived for any object in a scene[13]. Camera calibration also plays a significant role in the accuracy of depth perception.

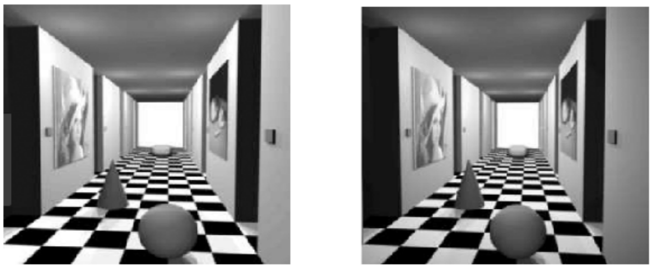

Figure 2.1: Stereo Image Pair [14]

2.2 Geometry of stereo Image pairs

In stereo vision, the two cameras are ideally expected to be horizontally displaced from each other. This is referred as the standard geometry of the stereo cameras. The image planes are symmetric with this geometry. But in reality, the two cameras mostly have a general geometry, where the two image planes will have different optical directions [12]. Due to this general geometry, the stereo image pairs are to be preprocessed before the depth perception. Stereo vision thus necessitates the familiarization of terms like Correspondence Problem, Disparity, Epipolar Geometry and Image Rectification.

2.2.1 Correspondence Problem

Considering a stereo image pair, for any pixel in one image, finding its equivalent position in the other image is referred to be the correspondence problem.

2.2.2 Disparity

Disparity is a measure of difference in coordinates for the same feature in a stereo image pair. In general, objects that are close to the camera tend to have larger disparities than those that are located far away from the image planes. This implies that, the distance is inversely proportional to the disparity. For the standard Geometry, it is given by the relation [16],

$$\text{Distance} = (\text{B} * \text{F})\ /\text{disparity}$$

where B is the Baseline distance and F is the Focal length

2.2.3 Epipolar Geometry

Epipolar Geometry is a special geometry of stereo images which constraints the correspondence search between the image pair [14]. This geometry relates the two camera centers, a point in three-dimensional(3D) space and its projection on the image planes. One needs to consider the baseline, epipolar plane, epipolar lines and epipoles for this geometry. Baseline refers to the line which connects the two camera centers. Epipolar plane is a plane that contains the baseline and a point in the real world. Epipolar lines are are formed by the intersection of epipolar plane with the image planes [6]. Any 3D point that lies on the epipolar plane will have its projection on the corresponding epipolar lines of the image planes. Epipole is the point of intersection of the baseline with the image plane. All the epipolar lines of an image intersect at the epipole. When the image planes are symmetric, the epipolar lines are parallel to each other and hence the epipoles are at infinity.

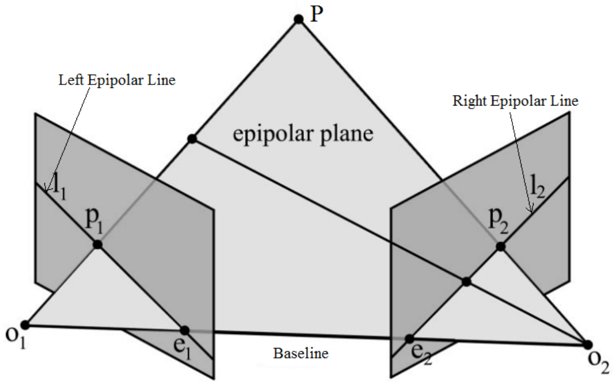

Figure 2.2: Epipolar Geometry [6]

In the Fig.2.2, O1 and O2 correspond to the camera centers. P is the point under consideration where P1 and P2 refer to its respective projection on the left and the right image plane. l1 and l2 represent the epipolar lines whereas e1 and e2 are the epipoles. Thus, the search for the correspondence of point P1 on the right image plane can be restricted only to the epipolar line l2, instead of an exhaustive search on the right image plane.

2.2.4 Image Rectification

Image Rectification is introduced to make the correspondence problem much easier. Here, we warp the image pairs such that the image planes become parallel to the baseline. By doing so, the epipoles are at infinity and also the epipolar lines become horizontal. This constraints our search to a single dimension which is only the horizontal shift [5]. Rectified image pairs are thus always preferred for stereo depth perception

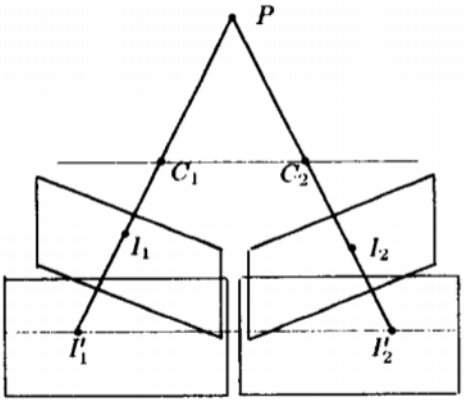

Figure 2.3: Image Rectification [1]

2.3 Advantages over Monocular Depth perception

Depth estimation is possible even using Monocular cameras. But to have accurate depth estimation from monocular images more complex algorithms are essential whereas with stereo vision, we use much simpler algorithms on the image pair to generate a disparity map, which contains the depth information. Also, absolute Depth estimation requires at least one known distance in the world [13], which could be even the baseline distance between the two camera centers in stereo vision. But such a data is missing in monocular vision. Also, stereo depth perception could work better for unfamiliar scenes as it only considers the difference between the two images for estimating depth. Considering the above factors, stereo vision stands to be a better option than monocular depth perception.

3 Limitations of Conventional Stereo Algorithms

3.1 Conventional stereo matching algorithms

Before deep learning gained its significance, the stereo depth perception was carried out using different learning algorithms. The most common steps followed in such algorithms are, computation of matching cost; cost aggregation and disparity estimation [12]. The broad classifications are local and global algorithms.

Generally, in local algorithms, a window is defined over a pixel region of the left image, which is slided over the corresponding epipolar line of the right image. At each position, the Sum of Squared Differences (SSD) is calculated. The region with minimum SSD value is considered as the best match. Here, the matching cost refers to the squared differences of the pixel intensities; Cost aggregation is the summation of matching cost over the defined window; Disparity estimation is done by selecting the minimal aggregated cost value [17].

Global algorithms have explicit smoothness assumptions, where two terms are considered for the optimization problem: Match quality and Smoothness [12]. Match quality term tries to find for every pixel in the first image, the best match in the second image. Smoothness term is concerned about the fact that, if two pixels are adjacent, they are generally subjected to the same amount of displacement which means that they should have similar disparities. Such algorithms are referred as Energy minimization algorithms and they have two cost factors, a match cost and a smoothness cost. Different such algorithms exist based on the minimization procedures used.

3.2 Disadvantages of conventional stereo matching

Conventional stereo matching algorithms stated above may not work well in situations with Poor image resolution, partial occlusions, large motions, noisy images and texture-less regions. This may result in inaccurate, unstable and slower depth perception [16]. However, utilizing deep learning for stereo depth perception adds more robustness to the network and hence could survive the above stated issues. Employing CNN to stereo depth perception also results in faster and accurate depth perception.

4 Overview of Convolutional Neural Networks

4.1 Introduction

A CNN is a class of deep neural networks which has several hidden layers in its architecture. The different layers of a deep network extract increasingly complex features[9]. CNNs are basically designed to process data which are in the form of multiple arrays like images. As the name indicates, the network employs a special kind of linear operation called convolution. It's because, When we deal with images, which have very high input dimension, we cannot use Fully Connected(FC) layers throughout the network as like Multi Layer Perceptron (MLP). This would result in explosion of the number of weights to be learned which could lead to poor generalization. Also, for images, the early features are usually local to a particular region and thus these features could be extracted locally with the weights being shared between the different locations. This motivates to use convolution in the early CNN layers. The general architecture of CNN has a series of convolution and pooling layers followed by a few FC layers. The layers of a CNN is explained in the following subsections. A CNN is generally trained using backpropagation which is also discussed further.

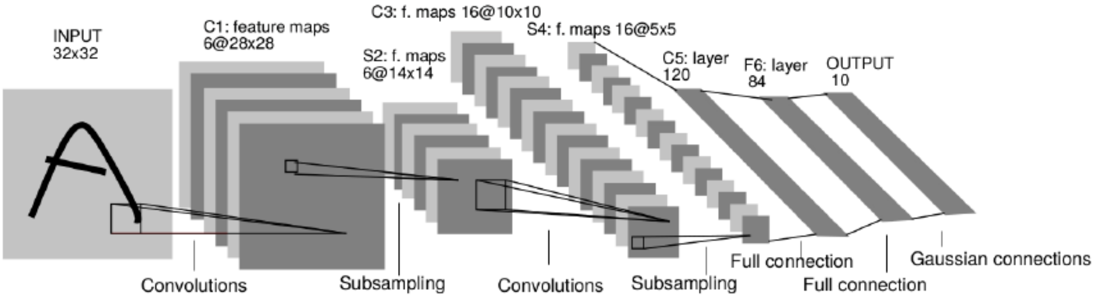

Figure 4.1: A classic CNN Architecture [9]

4.2 Layers of a CNN

4.2.1 Convolution layers

In this layer, we extract the local features in an image using a bank of filters. Main advantages of this layer are parameter sharing and sparse connections. Parameter sharing implies that the same filter can be used throughout an image. Sparse connection means that, each output depends only on a smaller number of inputs. Hence, a convolution layer has only few weights to learn. Usually the convolution is followed by a nonlinear activation function like tanh or Rectified Linear Unit(ReLU)[8], in order to introduce non-linearity. Otherwise, no matter how deep the network is, it won't be possible to learn complex nonlinear models.

4.2.2 Pooling layers

Pooling layers are introduced in CNNs to reduce the spatial dimension since the number of elements at the output of the convolution layer is still high. Pooling thus helps to speed up the computation and in addition, it could make the features more robust. This is possible because, in pooling only one value which retains the important information among a local neighborhood will be considered [8]. Different pooling methods namely max-pooling, mean-pooling and sum-pooling exist among which the max-pooling which considers the largest element within the neighborhood is the commonly used one.

4.2.3 Fully Connected layers

The last layers in a CNN are FC layers where each neuron in the current layer will contribute to every neuron in the next layer. Thus, a FC layer has more parameters to learn compared to the convolution layers. The final output layer could be a softmax or a simple FC layer based on the application.

4.3 Training a CNN

For CNN, large number of labelled datasets are required for training. A measure of deviation between the network's prediction and the ground truth information is referred as the loss function and the network aims to attain the minimum of the loss function. To achieve this, optimization methods like gradient descent are iteratively used to modify the free parameters until convergence. Here, the gradient of the loss function is back propagated through every layer of the network and the weights are updated based on their contribution to the error term. This method is called as backpropagation [9]. During backpropagation, every layer adds its contribution to the back propagated gradient. Once the network is trained, it should be tested using a separate test set data to evaluate its performance. A trained network should always have a better generalization for new unseen inputs.

5 CNN architectures for stereo depth perception

5.1 Overview

There exist several CNN architectures for stereo depth perception and a few popular architectures among them are discussed in this report. Any architecture in CNN would require a lot of input data along with labelled ground truth depth information for training as well as testing.For stereo depth perception, most of the networks generate their own synthetic datasets along with depth map whereas only a few networks use the commonly available datasets for stereo vision like KITTI, Middlebury dataset etc,.

5.2 Encoder-Decoder Architecture

Encoder-Decoder architecture uses an encoder part which takes the raw inputs and extracts features of different resolutions through down sampling. A decoder network processes these feature representations at different levels by upsampling and produces an output which has the same resolution as that of the input. Generally, the different layers of decoder network utilizes the information from corresponding encoder layers as well.

The research work [16] considers two CNNs with Encoder-Decoder architecture. These networks are particularly designed for autonomous driving scenarios. Around 14,000 image pairs are generated using an open source application called Blender. For training the network, pavements and sidewalks are excluded from the distance map so that more emphasis can be kept on the surrounding objects than the pavements. Both the networks are validated using 500 separate test set data and their performance is compared.

5.2.1 Simple Architecture

Fig. 5.1 represents the simple Encoder Decoder architecture from [16] where the input image pairs are concatenated and fed to the Encode network. Each encoder layer is comprised of a convolution layer followed by a ReLU activation function and then max-pooling. As mentioned earlier, convolution layers extract features in the image and max pooling layers reduce the spatial dimension at each level. Decoder uses bilinear upsampling to increase the resolution at each stage. This data is

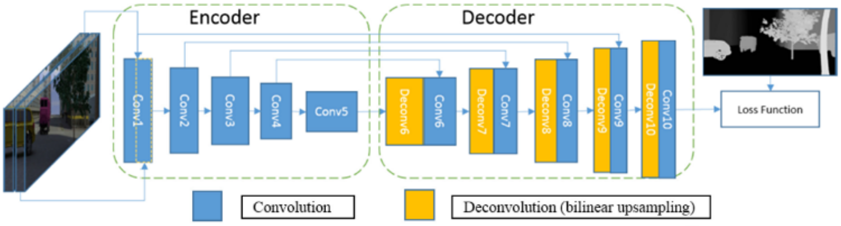

Figure 5.1: Simple Encoder-Decoder Architecture [16]

then concatenated with the features from corresponding encoder layer through skip connections [15] which is then followed by convolution. This results in the extraction of distance information at different resolutions along with the feature data.

5.2.2 Modified Architecture

Fig. 5.2 represents a modified architecture from [16] specifically designed to extract more specialized features for stereo matching rather than the generic ones as in the first architecture. This network computes similarity between the feature maps of different resolutions using Cross Correlation(CC) layers and generates various distance maps which are then aggregated to a single distance map. In contrast to the first network, the left and right images are fed separately to two encoder structures and the weights are shared between them. By doing so, similarities between the corresponding features in the two images are preserved.

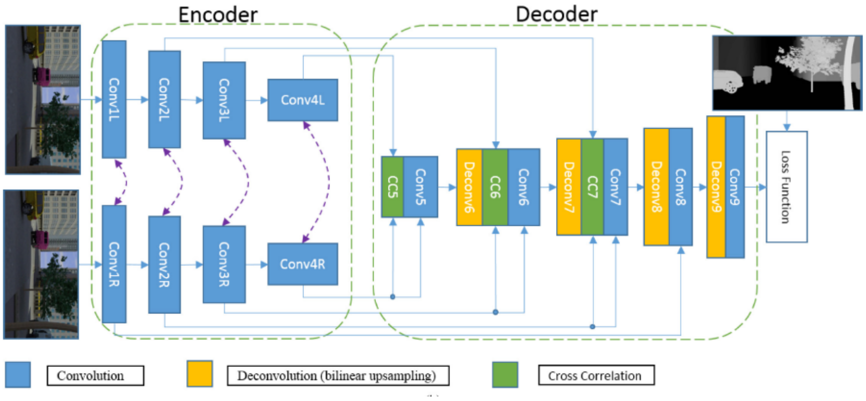

Figure 5.2: Modified Encoder-Decoder Architecture [16]

Decoder architecture employs CC layers at multiple resolutions to extract the Pairwise similarities between the two feature maps. This provides a prior distance map for the next finer layer and hence, the network is constrained to stereo matching features. The similarity output from the CC layer is then concatenated with the features from the corresponding encoder layer as in the first architecture. Bilinear upsampling is carried out to increase the resolution at every layer of decoder.

5.2.3 Loss function and Training

A common loss function is considered for both the architectures. Considering the scenario of depth perception in autonomous vehicles, closer objects should have more accurate distance estimation than those that are far away in the scene. So, the loss function is taken as a weighted sum of squared error between the estimated depth and the ground truth. Larger weights are assigned for the closer pixels when compared with the farther ones [16]. Optimization is carried out using Stochastic Gradient Descent(SGD) with mini-batches. Learning rate is an important factor in the optimization which decides the speed and the stability of the convergence.

5.2.4 Evaluation and Comparison

Once trained, the networks are evaluated using 500 test images using the Mean Square Error (MSE) function. The table.5.1 compares the MSE results of the two network architectures including their execution time. As the features extracted in second network are more specific to depth information, it has a better accuracy than the first network. It can also be inferred that the second network has fewer parameters to learn than the first one, but a bit slower due to the inclusion of CC layers. Still this negligable difference in computational speed won't have a noticeable impact. Though post processing and smoothing steps could lead to better results for occluded and texture less regions, they will demand much larger execution time [16]. Since for hard real time scenarios like autonomous driving, faster depth perception is also substantial those steps are not considered.

Name	Training data	Test data	Time(ms/frame)
First Network	5.60	8.91	47
Second Network	2.76	7.10	63

Table 5.1: Performance comparison between the Encoder-Decoder Networks [16]

5.3 Modified AlexNet and Fully Convolutional Architectures

5.3.1 Modified AlexNet

Alexnet [8] is a CNN designed basically for the image classification using supervised learning. It has eight layers among which five are convolutional layers (convolution + ReLU nonlinearity + max-pooling) and three are FC layers. A softmax function uses the output of the last FC layer and provides the probability of an image belonging to a particular class. The aim of the network is to maximize the log likelihood for the true label. Since there are huge weights to be learned, to avoid overfitting, methods like data augmentation and dropout is employed [8]. By data augmentation, the number of training data is increased which leads to better generalization. Dropout is used to drop out few neurons while training each input based on a random probability . The network is trained on the huge ImageNet dataset using Graphics Processing Units(GPU) and this resulted in very high accuracy.

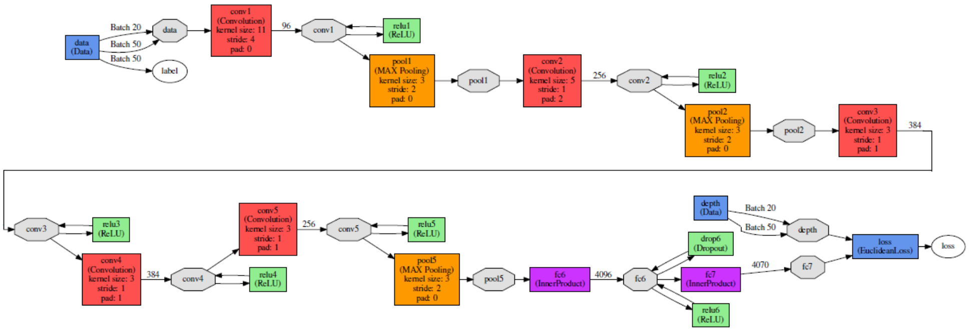
data (Data)
Batch 20
Batch 50
Batch 50
data
label
conv1 (Convolution) kernel size: 11 stride: 4 pad: 0
96
conv1
relu1 (ReLU)
pool1 (MAX Pooling) kernel size: 3 stride: 2 pad: 0
pool1
conv2 (Convolution) kernel size: 5 stride: 1 pad: 2
256
conv2
relu2 (ReLU)
pool2 (MAX Pooling) kernel size: 3 stride: 2 pad: 0
pool2
conv3 (Convolution) kernel size: 3 stride: 1 pad: 1
384
conv3
relu3 (ReLU)
conv4 (Convolution) kernel size: 3 stride: 1 pad: 1
384
conv4
relu4 (ReLU)
conv5 (Convolution) kernel size: 3 stride: 1 pad: 1
256
conv5
relu5 (ReLU)
pool5 (MAX Pooling) kernel size: 3 stride: 2 pad: 0
pool5
fc6 (InnerProduct)
4096
fc6
drop6 (Dropout)
relu6 (ReLU)
fc7 (InnerProduct)
4070
fc7
depth (Data)
Batch 20
Batch 50
depth
loss (EuclideanLoss)
loss

Figure 5.3: Modified AlexNet Architecture [7]

Inspired by the architecture of [8], the authors of [7] developed a similar network for stereo depth perception which is represented in Fig.5.3. This network has 5 convolutional layers (Convolution + ReLU+ max-pooling) and 2 FC layers. As in [8], Dropout is carried out to avoid overfitting. The output of the last FC layer is compared against the labelled depth map and an L2-norm loss function is derived. It is basically the SSD of the two values which is nothing but the MSE term. Two modern optimizers, SGD with momentum and Adaptive Gradient (AdaGrad) [3] are considered for training and SGD with momentum provided better results. Both these optimizers could adapt the learning rate during training while SGD cannot. Taking into account the need for huge datasets for training a CNN, around 60,000 synthetic datasets along with their depth maps are generated using few available frameworks. Using Manhattan world assumption [4], a cubic room with plane aligned cubic geometry is considered for it and diverse randomisations are carried out.

5.3.2 Fully Convolutional Architecture

A second architecture designed in [7] is a fully convolutional network. The max-pooling and the FC layers are discarded in this architecture.It has 7 convolution layers along with ReLU activation function as in Fig.5.4 . Padding is introduced to retain the spatial dimension throughout the network. However, the number of outputs for every layer is limited considering the memory restrictions. The output of the last convolution layer is considered for predicting the loss function. However,

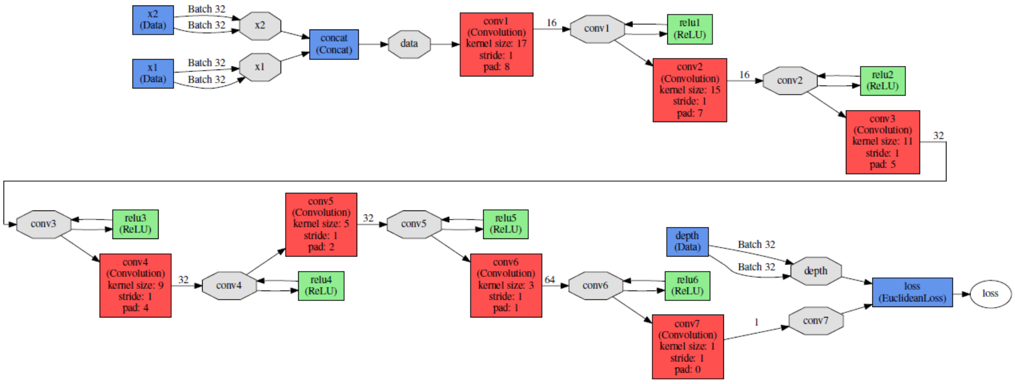

Figure 5.4: Fully Convolutional Architecture [7]

similar loss function and optimisation techniques as that of the AlexNet based architecture is considered. Training and testing are carried out using the same synthetic datasets.

5.3.3 Results and Comparison

Both the networks are trained using 50,000 generated synthetic datasets and then validated with the rest 10,000 datasets. The Root Mean Squared Error (RMSE) for both the networks are compared in the table 5.2. It is evident from the table that the fully convolutional network resulted in better performance than the AlexNet based architecture. Though the first network has close to accurate global depth estimates,it works poorly for complex geometries [7]. Also, due to max-pooling layers, the fine details which contain some useful depth information are removed. Thus, in the second network, max-pooling is avoided to preserve such details. Also, this architecture incorporates larger filters in the early stages to find larger disparities and then the filter sizes are gradually reduced at the successive layers to assess smaller disparities. This resulted in more accurate depth perception. However, the outputs of this network have additional artifacts which are the effects of zero padding.

Architecture	RMSE
Modified AlexNet	0.098
Fully Convolutional	0.033

Table 5.2: RMSE comparison between the Modified AlexNet and Fully Convolutional Network [16]

5.4 Semi-Supervised Approach

Fig.5.5 represents an architecture from [13] which employs a semi supervised deep learning to estimate the disparity of the stereo image pairs. The features of the left and the right images are individually extracted using ResNet-18 [15] based architectures. ResNet is a popular deep learning network which included skip connections to minimise the vanishing gradient problem which is a major issue with deep networks.

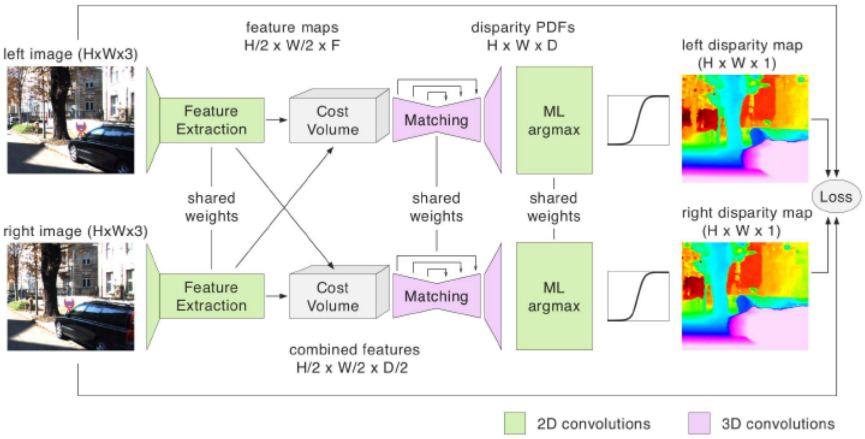

Figure 5.5: Semi Supervised Architecture [13]

The weights are shared between the two feature extractors. Using the extracted features, two cost volumes(one for left-right and another for right-left matching) are created by sliding each tensor along the epipolar lines of the other tensor in a particular direction up to maximum disparity [13]. The two cost volumes are then fed to the encoder-decoder like 3D convolution-deconvolution networks that does stereo matching. At the last layer, we get two tensors which have the pixel matching costs between the images. These tensors are then fed to a sequence of 2D convolutions that construct a Machine Learned-argmax function which when followed by a sigmoid function results in the disparity estimate for each pixel.

5.4.1 Loss function, training and results

For training the network, a loss funcion with a supervised term along with unsupervised terms is defined. Supervised term compares the network's disparity estimate to the ground truth sparse LIDAR data. The unsupervised terms ensure the following: consistency of the two disparity maps, photometric consistency and smoothness of the disparity map[13]. The network is trained using KITTI dataset with 29,000 training images. The optimisation function used is Adaptive moment estimation (Adam) where the learning rate gets adapted during learning. The network is tested using 200 images from KITTI and compared against different similar architectures. This architecture outperformed others and in addition when the same network is fine tuned, it resulted in much less error for the non-occluded regions. This semisupervised approach is able to extract even the fine details in the images resulting in much accurate depth estimation.

6 Observations from different stereo architectures

Architecture	**Encoder Decoder**	**Modified AlexNet**
Training data	14,000 image pairs generated using an open source application called Blender	Trained using 50,000 synthetic datasets generated using the available frameworks
Testing data	500 separate test set data generated from Blender	Tested with 10,000 datasets generated similarly as training data
Loss function	MSE weighed by the distance	L2-norm loss function which is the RMSE term
Optimization	SGD	SGD with momentum and AdaGrad
Advantages	Focuses on autonomus driving and gives more importance to the interesting objects in the scene. In addition computational speed is given importance	Trained on numerous datasets and adaptive optimizers are used
Disadvantages	Not trained on enough datasets; SGD cannot adapt the learning rate during training; Artifacts in the simple architecture; No post processing carried out considering the execution time	Not easily comparable with other architectures because of the unique dataset; Due to maxpooling fine details are missed out in the depth map

Table 6.1: Comparison of different CNN Architectures(1)

Architecture	**Fully Convolutional**	**Semi supervised Network**
Training data	Trained using 50,000 synthetic datasets generated using the available frameworks	Trained using KITTI dataset with 29,000 training images
Testing data	Tested with 10,000 datasets generated similarly as training data	Validated with 200 images from KITTI
Loss function	L2-norm loss function which is the RMSE term	Loss function with supervised and unsupervised terms
Optimization	SGD with momentum and AdaGrad	Adam optimizer
Advantages	Trained on numerous datasets; adaptive optimizers are used; Could preserve fine depth information as well	Trained on the available KITTI datset and hence easily comparable; Preserves fine details in the depth estimate using unsupervised terms; Adaptive optimizer is used
Disadvantages	Not easily comparable with other architectures because of the unique dataset; Additional artifacts and black borders due to padding effects	Not enough data for testing; Poor performance around the object boundaries due to the lack of dense ground truth data

Table 6.2: Comparison of different CNN Architectures(2)

The above tables list and compare various aspects of the different algorithms considered including their advantages and disadvantages. Some of them are designed to have very good architecture but still are not trained on enough datasets. In few networks, modern adaptive optimizers are used which helps in fast learning. Also, different loss functions are considered accounting various scenarios. Since different datasets are considered for training each architecture, an exact comparision between the performance of all these networks is not feasible. However, with the above comparison, we can reach a decision that, for real world scenarios, the Encoder-Decoder architecture or the semi supervised network could result in accurate and efficient depth perception. This is because, these networks are trained and designed concerning the real world factors like computational speed. However, we should also consider the disadvantages of these networks and derive some improvement methods.

7 Conclusion

While there are different architectures available for stereo depth perception, deep learning methods are much efficient and could result in accurate depth perception. They have better performance even with texture less regions and partial occlusions. Also, with the help of GPUs, these methods could compute depth maps in a faster way which is highly significant for the assisted and autonomous vehicles. However, this is still a research topic with high scope for improvements and hence many network architectures are continuously being proposed. So, there are progressive improvements than the already existing networks. The networks discussed in this report are quite a few among the recent architectures which employed CNN for stereo depth perception. There are other existing architectures as well like [2] [10] which are modeled for stereo depth estimation. Since, deep learning itself is yet to overcome few known limitations and also it's a domain with lot of innovations, there are high possibilities that the discussed networks could also overcome their existing limitations that are presented.

Acknowledgment: The report is drafted under the direction of the staff of Automotive Software Engineering, Chemnitz University of Technology, Germany. Special say to the Prof. Dr. Wolfram Hardt, Dipl. Inf. Rene Schmidt, Dipl.-Ing. Philip Parsch and Prof. Dr. Uranchimeg Tudevdagva for ceaseless input, amid addresses and activities which enhanced the nature of the examination.

Bibliography

[1] Ayache, N., Hansen, C.: Rectification of images for binocular and trinocular stereovision. In: [1988 Proceedings] 9th International Conference on Pattern Recognition. pp. 11–16. IEEE (1988)

[2] Du, L., Li, J., Ye, X., Zhang, X.: Weakly supervised deep depth prediction leveraging ground control points for guidance. IEEE Access 7, 5736–5748 (2019)

[3] Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12(Jul), 2121–2159 (2011)

[4] Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Manhattan-world stereo. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1422–1429. IEEE (2009)

[5] Fusiello, A., Trucco, E., Verri, A.: A compact algorithm for rectification of stereo pairs. Machine Vision and Applications 12(1), 16–22 (2000)

[6] Jurjević, L., Gašparović, M.: 3d data acquisition based on opencv for close-range photogrammetry applications. In: ISPRS Hannover Workshop: HRIGI 17–CMRT 17–ISA 17–EuroCOW 17 (2017)

[7] Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 66–75 (2017)

[8] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

[9] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)

[10] Loni, M., Majd, A., Loni, A., Daneshtalab, M., Sjödin, M., Troubitsyna, E.: Designing compact convolutional neural network for embedded stereo vision systems. In: 2018 IEEE 12th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC). pp. 244–251. IEEE (2018)

[11] Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3061–3070 (2015)

[12] Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision 47(1-3), 7–42 (2002)

[13] Smolyanskiy, N., Kamenev, A., Birchfield, S.: On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1007–1015 (2018)

[14] Sun, C.: Fast stereo matching using rectangular subregioning and 3d maximum-surface techniques. International Journal of Computer Vision 47(1-3), 99–117 (2002)

[15] 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. pp. 1–9 (2015)

[16] Taimouri, V., Cordonnier, M., Lee, K.M., Goodman, B.: Distance map estimation of stereoscopic images using deep neural networks for autonomous vehicle driving. Tech. rep., SAE Technical Paper (2017)

[17] Ye, X., Li, J., Wang, H., Huang, H., Zhang, X.: Efficient stereo matching leveraging deep local and context information. IEEE Access 5, 18745–18755 (2017)