DOI: 10.21917/ijivp.2021.0336

PATCH BASED STEREO MATCHING USING CONVOLUTIONAL NEURAL NETWORK

Rachna Verma and Arvind Kumar Verma
[1]*Department of Computer Science and Engineering, Jai Narain Vyas University, India*
[2]*Department of Production and Industrial Engineering, Jai Narain Vyas University, India*

Abstract

The paper presents a new Convolutional Neural Network (CNN) architecture, called stacked stereo CNN, for computing disparity map from stereo images. In stacked stereo CNN, left and right image patches are stacked back-to-back and fed to a single tower CNN. This is in contrast to Siamese network where two towers are used, one for the left patch and other for the right patch. The proposed network is trained on a large set of similar and dissimilar image patches, which are generated from stereo images and their ground truth images from Middlebury stereo datasets. The network returns a dissimilarity score for a pair of image patch which is used to compute the cost volume. The cost volume is further refined using post processing steps before generating the final disparity map. The proposed network is evaluated on Middlebury datasets and achieves comparable results to the state-of-art algorithms.

Keywords:

Stereo Vision, Patch Matching, Disparity Map, CNN

1. INTRODUCTION

Stereo vision is a widely used technique to estimate depth of various scene points from two images of the scene obtained from two slightly different viewpoints. The basic principle of stereo vision is that the depth of a scene point is inversely proportional to its disparity in the left and right images. However, efficient and accurate implementation of this simple principle, specifically calculation of disparity, is eluding researchers for the last three decades due to the presence of occlusion, repetitive patterns, reflections, textureless areas. Thus, the stereo vision is still a very active research area of computer vision. It has many applications, such as 3D scene reconstruction, autonomous driving, obstacle avoidance and robotics.

Over three decades of research in stereo vision, many methods have been reported in literature. These methods can be grouped into two categories: traditional approaches and machine learning based approaches. Traditional disparity estimation methods start by first computing dissimilarity cost at each disparity level for each pixel and generates a cost volume followed by cost volume filtering for smoothing and noise removal and finally, disparity selection by selecting index of the lowest cost. Commonly, some measures of a small window around the pixel of interest in one view, called the reference image, is used to locate the most similar window in the other view called the target image. The commonly used handcrafted similarity measures are: absolute difference or squared difference of pixel intensities, normalized cross-correlation. However, due to occlusion, repetitive patterns, reflections, textureless areas, a large number of wrong matches are generated by direct matching.

To improve the accuracy of the disparity map by traditional methods, a typical pipeline of stereo disparity computation, using other clues from the stereo pair images and neighbourhood continuity, consists of the following steps: (1) matching cost computation, (2) cost aggregation, (3) optimization, and (4) disparity refinement [1]. For more details about these steps, refer [1]. The major drawback of traditional methods is inability of handcrafted similarity measures to generate accurate initial disparity map leading to inaccurate final disparity map, even after applying all the steps of the stereo pipeline.

In machine learning based approaches, neural networks are used in one or more steps of the traditional stereo pipeline. Due to the availability of advanced GPU hardware and deep learning libraries, researchers in stereo vision have reported many deep learning based approaches to calculate disparity maps. These methods outperform traditional methods in terms of accuracy and sometimes speed. Based on how convolutional neural networks (CNN) is used in stereo pipeline, deep learning stereo matching methods are classified into patch based [2][3] or end-to-end [4] [5].

In patch based methods, stereo matching problem can be modelled as a binary or multi class classification problem or a regression problem where a network is trained to categorise similar and dissimilar patches or to generate a similarity/dissimilarity score for the input patches. In a rectified stereo image pair, given a patch from the left (reference) image, the task of the network is to locate the best matching patch in the right (target) image. In patch based methods, the hand crafted feature to generate matching score is replaced by a trained network. Similar to traditional methods, to further refine the disparity map, extensive post processing steps are used. On the other hand, in end-to-end deep learning methods, hourglass shaped deep convolutional neural networks are used to generate the final disparity map in one go and no further post processing steps are used. Currently, the end-to-end networks are capable to handle domain specific problems. Researchers are unable to train a generic stereo matching network due to non-availability of suitable labelled datasets. Further, the end-to-end methods are very expensive in terms of memory requirements.

In this paper, a new CNN architecture, called stacked stereo CNN (SS-CNN), for the patch based stereo matching is proposed. In contrast to commonly used two tower Siamese network [2] [3] for the patch based stereo matching, the proposed network is a single tower CNN where the two input patches are stacked together along the colour plane axis before feeding to the network. The network directly generates the dissimilarity score for the input patches which can be used as dissimilarity cost in downline processing steps. The dataset to train the proposed network is generated automatically from publically available stereo images datasets with ground truth from Middlebury [6]. The raw disparity map obtained from the proposed network is further refined using semi global matching and left-right (LR) consistency check. Further, a new method is proposed to fill inconsistent disparities generated after LR check. Typically, inconsistent disparities are filled by any interpolation method. The proposed method uses superpixels for filling inconsistent disparities and assumes that all the pixels of a superpixel have close disparity values.

The main contributions of this paper are as follows: (i) A new architecture for patch matching using deep convolution network is proposed for computing dissimilarity score from similar and dissimilar image patches, (ii) A new method based on superpixels is proposed for filling inconsistent disparities.

The remaining part of this paper is organized as follows. Section 2 discusses the related work focusing on patch matching using stereo images. The proposed stacked network architecture is discussed in section 3. Section 4 explains how the proposed architecture is used to generate disparity map. The experimental results are presented in section 5 and section 6 concludes the paper.

2. RELATED WORK

A large number of methods has been proposed in literature for solving stereo matching problem ranging from traditional methods to deep learning based methods. For the detailed review of the traditional methods, [7] can be referred. For the general overview of the deep learning based stereo vision, interested readers may refer [8]. The focus of this paper is on patch based convolutional neural network approach for computation, therefore, the methods related with this approach are presented in this section.

Pioneering work of Zbontar [2] uses a Siamese convolutional neural network for computing matching score for the reference patch in the left image and the target patch in the right image, in place of hand crafted image features used by the traditional methods. The initial disparity map generated by this method is considerably better than the state-of-the-art traditional methods. In a Siamese neural network, two identical neural network with identical weights work in parallel on two different input vectors to compute comparable output vectors. The output vectors are then combined together and fed to another neural network to generate matching score for the two input vectors (here the left and right image patches).

The Fig.1 shows an architecture of a Siamese neural network. Zbontar [2] used many traditional disparity refinement post processing steps on the initial disparity map and the final disparity map was the state of the art. Luo et al. [9] modified Zbontar's work and used inner product for computing similarity score from the output of two branches of Siamese network, in place of an additional network as used by Zbontar. Further, they passed a wider image patch in one of the Siamese branch to obtain matching cost for all possible disparities with one pass of the CNN. Park and Lee [10] proposed a per-pixel pyramid pooling module to the baseline architecture of Zbontar [2] to increase the receptive field of the network. Ye et al. [11] also used pooling module in ensemble network architecture to produce good results in weakly textured areas.

Brandao et al. [12] used a Siamese architecture similar to Luo et al. [9] focusing on the types of features used for correspondence matching. Zagoruyko and Komodakis [3] explored and studied various types of networks that learns similarity function for patches which implicitly considers various transformations, such as wider baseline and illumination. Different from a typical Siamese network, Chen and Jung [13] proposed a patch based stereo matching using 3D CNN which considers spatial colour and disparity features simultaneously for smooth disparity map while preserving edges.

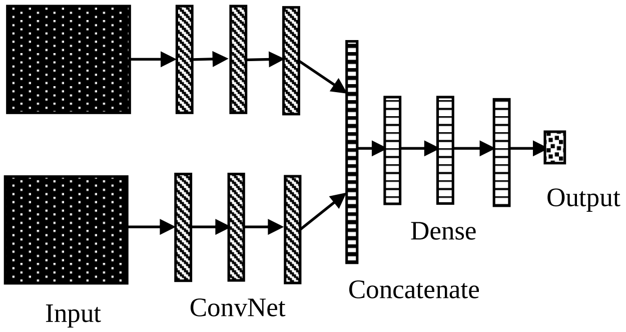


Fig.1. A Siamese Neural Network Architecture

3. PROPOSED METHOD

Similar to the most of the patch based approaches, the proposed method also consists of two main steps: initial cost volume generation for computing disparity map using a network and post processing steps for disparity refinement. However, the proposed method uses a new CNN architecture, called stacked stereo CNN (SS-CNN), for the patch based stereo matching, in contrast to commonly used Siamese networks in most of the previous patch based stereo matching approaches. Fig.2 shows the proposed architecture, which is a single tower CNN where the two input patches are stacked together along the colour plane axis before feeding to the network. The network directly generates the dissimilarity score for the input patches which can be used as matching cost in downline processing steps. The network requires low GPU memory and does not require a very large dataset of stereo images for training. Though time consuming, the proposed network has been successfully trained and used to generate disparity map on a normal CPU computer.

3.1 ARCHITECTURE OF THE PROPOSED NETWORK

The Fig.2 shows the proposed architecture of the SS- CNN, which is used to generate the dissimilarity score between two image patches.

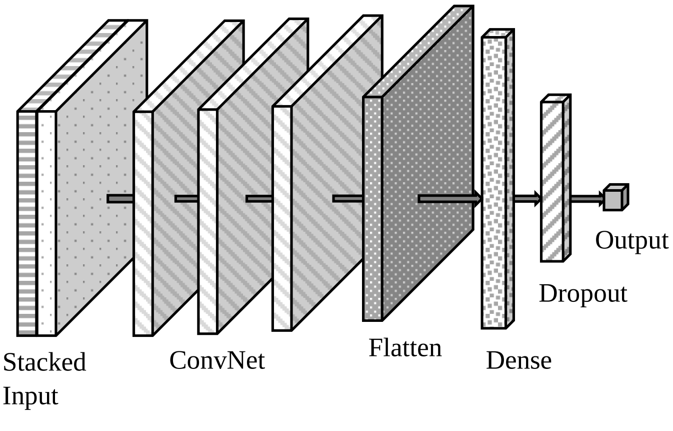


Fig.2. Stacked Stereo CNN architecture

The input to the network is a tensor of $W\times W\times W$, where $W\times W$ is the size of the left and right patches and C is the total number of colour channels: 2 for grey images and 6 for coloured images (3 for the left image patch and 3 for the right image patch). The input tensor is generated by stacking two patches along the colour channel axis. The colour images are normalised before stacking and feeding to the network. The proposed network consists of, in sequence of, three convolutional layers, one flatten layer, one dense layer, one dropout layer and finally the output layer. The Table.1 gives the meta parameters of the proposed network.

The network is constructed by adding three layers of 2D convolution with rectified linear units (ReLU). The final layer of 2D convolution is flatten and is connected to fully connected NN layer with 64 units and ReLU non-linearities. This layer is finally connected with the output layer with units=1, and activation function as Sigmoid which outputs the final dissimilarity score.

The convolution kernel size used in all the layers are of size 3×3. The number of filters used in the first layer is 8, 16 in the second layer and 32 in the third layer. The network is trained by minimizing cross-entropy loss and optimizer used is Stochastic Gradient Descent (SGD). Learning rate is set to 0.001 and decay is set to decay=1e^{-6}. The Table.1 gives the model summary of the proposed network and its parameters.

Table.1. Model Summary

Layer	Output Shape	No of Parameters
Conv2D	(11, 11, 8)	440
Conv2D	(11, 11, 16)	1168
Conv2D	(11, 11, 32)	4640
Flatten	3872	0
Dense	64	247872
Dropout	64	0
Dense	1	65
Total Parameters		254,185
Trainable Parameters		254,185
Non-trainable Parameters		0
Model Type		Sequential

3.2 TRAINING DATASET GENERATION

Twenty-five Middlebury stereo images and their ground truth [6] are used to generate binary classification (similar and dissimilar patch pairs) dataset for training and validation purpose. Two types of image patch pairs are generated: similar patch pairs and dissimilar path pairs. For generation of similar image patch pairs, a random pixel location, say (x,y), is picked from the left image and a square patch of size 11×11 pixels centred around (x, y) in the left image is selected as the left patch.

Using the disparity value, say d, of the location (x, y) obtained from the ground truth image, the right image patch is obtained by selecting pixels of a square window of size 11×11 centred around $(x - d,y)$ in the right image. For generation of dissimilar image patch pairs, the left patch is a square of size 11×11 obtained from the left image and is generated as described above. The right image patch, of size 11×11 in the right image, is generated with centre location as $(x\text{-}z, y)$, where z does not lie between the range of $(d\text{-}3)$ to $(d+3)$. The value of z is selected randomly between 1 to maximum disparity range.

4. DISPARITY MAP GENERATION

The Fig.3 shows the block diagram of the steps used to generate the final disparity map: initial cost volume generation, semi-global matching, Initial disparity map, left-right consistency check, hole filling and final disparity map. These steps are essential to improve the accuracy of the raw disparity map generated by applying only the trained SS-CNN. In the following sections, these steps are described.

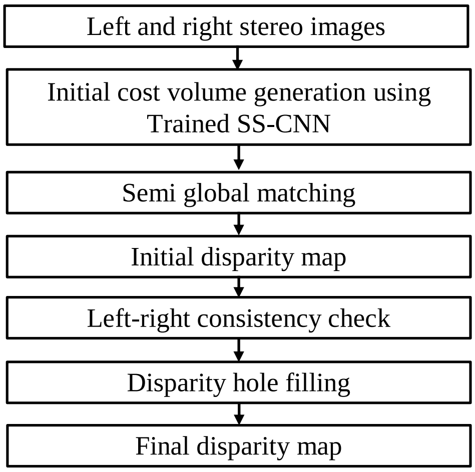

Left and right stereo images
Initial cost volume generation using Trained SS-CNN
Semi global matching
Initial disparity map
Left-right consistency check
Disparity hole filling
Final disparity map

Fig.3. Steps for generation of disparity map

4.1 INITIAL COST VOLUME GENERATION

This is the first step for generation of the disparity map in the patch based approach. For each pixel location in the left image, the left image patch is compared with the right image patches in the right image for all the possible disparity values and corresponding dissimilarity scores are computed. This process, when applied to all the pixels of the left image, results into a 3D array of size $H\times W\times D$, where H is the height, W is the width of the left image and D is the disparity range. This cost volume stores the raw dissimilarity scores for all possible disparities as generated by the trained SS-CNN. These raw scores are not enough for generating disparity map because of poor computation in textureless areas, occlusion, repetitive patterns and reflections, therefore initial cost volume is refined further.

4.2 SEMI GLOBAL MATCHING

It can be observed easily that, except at the region boundaries, disparity varies smoothly in a region. The semi global matching attempts to enforce the above observed smoothness constraints on the disparity image by defining an energy function that penalises based on their neighbourhood disparity values. In this paper, we have used the semi global matching scheme developed by Zbontar [2], which is based on the original function developed by Hishurmulle [14]. All the parameters used for minimizing the energy functions are adopted as it is from Zbontar [2] and the final cost in semi global matching is computed by taking the average across all four directions as done in [2] to avoid streaking effect.

4.3 INITIAL DISPARITY MAP

The semi global matching step refines the cost volume. At this stage, the disparity $D(p)$ at each pixel, p, of the reference image is obtained using winner-takes-all strategy. This strategy selects disparity with minimum cost and is given as follows:

$$D(p)=\arg\min_{d}\left(C(p,d)\right) \tag{1}$$

where, d is the maximum disparity range. $C(p,d)$ is the dissimilarity cost computed from SS-CNN for pixel p at each disparity value. The *argmin* function returns the index corresponding to the minimum matching cost in the cost volume at each pixel p. This step creates the initial disparity map which can be displayed as an image. Using the above steps, the initial disparity maps for the left and right images (taking right image as the reference image) are computed.

4.4 LEFT RIGHT CONSISTENCY CHECK

Initial disparity maps of both the left image and the right image are computed separately. Left-Right consistency check is applied to detect consistency between both the disparity maps obtained. Let D_L and D_R denote the disparity maps obtained for left and right images, respectively. For each pixel at (x,y), let d_l is the disparity at $D_L(x, y)$ in the left disparity map. Let d_r is the disparity at $D_r(x, y-d_l)$ in the right disparity map. The following scheme is used to classify the left image disparity values in two classes: consistent and inconsistent.

- *Consistent Disparity*: If $|(d_l-d_r)|\leq 1$, i.e. the disparity for a scene point differ by only one pixel in the left and right disparity maps.
- *Inconsistent*: Otherwise.

Finally, the inconsistent disparities are discarded in the left disparity map and are set to a negative value. The regions of negative disparity values are called holes in the disparity map and the scheme described in the next section is used to fill with the most appropriate values using the reference image details.

4.5 HOLE FILLING SCHEME

Segmentation is one of the techniques to cluster pixels based on similarity of image features, such as colour. It has been used in many computer vision tasks including disparity filling [15]. In this paper, we propose a new hole filling algorithm that is based on superpixel based segmentation of the reference image. A superpixel is a small image region where all pixels are similar in terms of some image features, such as colour, texture, etc. (See Fig.4). We have used Simple Linear Iterative Clustering [16] (SLIC) algorithm for image segmentation. The SLIC algorithm segments the image into superpixels, i.e. it groups pixels with similar colour values and generates a number of regions, where each region is a superpixel. This superpixel image is used to assign suitable disparity values to negative values (holes) in the disparity map. The following steps are used for hole filling:

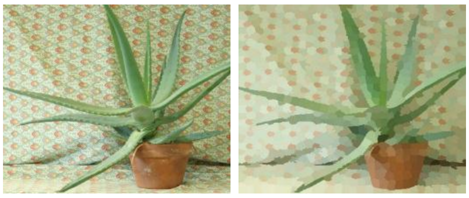
Step 1: Convert the reference image into superpixel image segments using SLIC algorithm.

Step 2: For each row of the disparity map image, scan it from left to right until a pixel (x,y) with the negative value is found. Consider a small window of size 5×5 around (x,y). Experimentally, we found that a 5×5 window gives good results.

a. Select all the pixels with non-negative values of the above window which are also in the super pixel region of the pixel (x,y) and calculate their median value.

b. Assign this median value to the pixel (x,y).

Step 3: Finally, a median filter of 5×5 is used to smoothen the hole filled disparity map to generate the final disparity map.



(a) (b)

Fig.4(a). Left image, (b) Superpixel segments

5. IMPLEMENTATION AND RESULTS

The above discussed algorithms are implemented in Python using Tensorflow and Keras packages. The developed patch based SS-CNN model is trained and evaluated on publically available Middlebury stereo dataset [6]. We used the quarter resolution stereo images of the dataset for training and evaluation to reduce the memory requirements. The developed model can be run on a simple CPU based workstation, however it takes more time for training as compared to GPU based system. The proposed network is trained using SGD to minimize the cross-entropy loss and used a learning rate of 0.001.

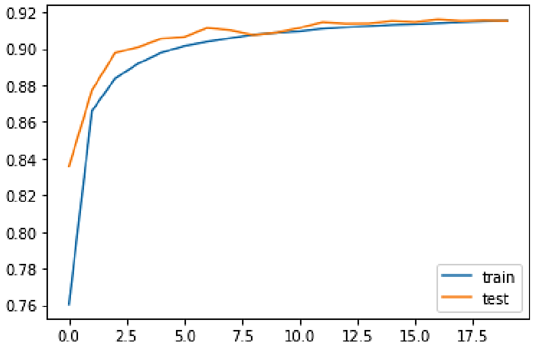


Fig.5. Plot of accuracy achieved after training SS-CNN

From 25 stereo image pairs, a total of 7.28 lakh image patch pairs of size 11×11 are extracted for training, in which half image patch pairs are similar and other half image patch pairs are dissimilar. The image pixel values are normalised before generating the training dataset. After many experimentations, the final batch size of 16 and 20 epochs are used for the network training. The network is trained on Google Colab with GPU setting on and it took around 30 minutes for training. The training accuracy and validation accuracy achieved for the network is 91%

(see Fig.5). For segmentation of the reference image, the SLIC [16] algorithm is used. The segmentation parameters compactness is set to 20 and number of segments is set to 800.

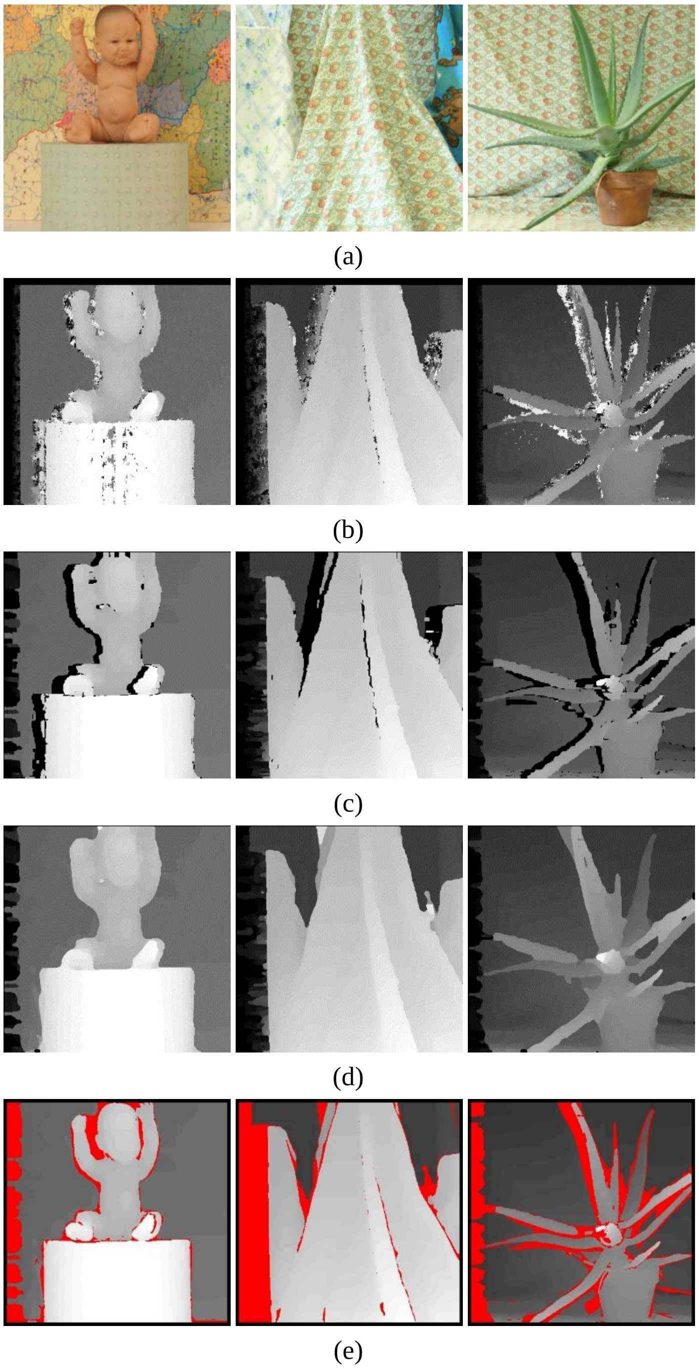


Fig.6(a). Left image, (b) raw disparity map after applying SS-CNN, (c) disparity map after LR check,(d) Final disparity map and (e) final disparity map with error region shown in red colour (for all region)

The Fig.6 shows the results obtained at various stages of processing for generating final disparity map. The results obtained with the proposed network (SS-CNN) is shown in Fig.6(b) i.e. SS-CNN output. The Fig.6(c) shows the left disparity map after LR check, and Fig.6(d) shows the final disparity map after filling inconsistent disparities using the proposed hole filling scheme. The Fig.6(e), the last row, shows 1-pixel error computed from the ground truth. The error regions are marked in red for 1-pixel error. As can be seen from the bottom row of Fig.6(e), the most error occurred in the left side of the image is because this region in the reference image is not visible in the right image i.e. no corresponding pixels exist in the right image. These image regions could be filled using interpolation.

Error is computed for the final disparity map using ground truth. The error percentage of the proposed method on different images [6] for SS-CNN output (Initial error) and after refinement (Final error) are given in Table.2 and Table.3 for different image regions. The results are reported for 1-pixel error and 2-pixel error for two image regions: region1 and region 2.

Region 1 is the complete reference image area except the boundary pixels of half of the size of the window, i.e. region 1 is considered in Fig.6(e).

Region 2 is the area of the reference image which is visible in both the images of a stereo image pair i.e., excluding the left region (up to maximum disparity) of the image.

The percentage errors for region 1 and region 2 for various test images are shown in Table.2 and Table.3, respectively. These regions show error for all region [6], i.e. including the occluded region. The accuracy achieved by the proposed method is comparable to the state of the art methods.

Table. 2. Error in Percentage for Region 1

Image Name	**Initial Error (CNN output)**		**Final Error (After Refinement)**	
	≤1	**≤2**	**≤1**	**≤2**
Baby1	25.6	13.1	11.6	10.3
Aloe	23.9	19.8	18.3	17.2
Cloth4	18.4	16.7	17.6	16.9

Table. 3. Error in Percentage for Region 2

Image Name	**Initial Error (SS-CNN output)**		**Final Error (After Refinement)**	
	≤1	**≤2**	**≤1**	**≤2**
Baby1	22.6	9.2	6.0	4.6
Aloe	21.9	17.6	14.6	13.2
Cloth4	8.3	6.4	4.9	4.0

6. CONCLUSION

In this paper, we have reported a new patch based stereo matching convolutional neural network, stacked stereo CNN, and a new algorithm for hole filling to estimate disparity values in occluded and boundary regions. The major advantage of the proposed architecture is its low memory requirement and it can be trained and used on simple CPU based computer. Similar to other patch based stereo matching networks, the proposed network uses only the local information due to which it fails in textureless areas and repeated patterns. Further, many post processing steps are used to get better results, which make overall computation costlier. Experimental results demonstrate the effectiveness the proposed architecture and results are comparable to other methods in its class. In future, training on larger dataset will be attempted, which might reduce error even further and give better disparity map.

REFERENCES

[1] K.Y. Kok and P. Rajendran, “A Review on Stereo Vision Algorithms: Challenges and Solutions”, *ECTI Transactions on Computer and Information Technology*, Vol. 13, No. 2, pp. 134-150, 2019.

[2] J. Zbontar and Y. Le Cuny, “Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches”, *Journal of Machine Learning Research*, Vol. 17, No. 2, pp. 1-32, 2016.

[3] S. Zagoruyko and N. Komodakis, “Learning to Compare Image Patches via Convolutional Neural Networks”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4353-4361, 2015.

[4] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia and L. di Stefano, “Real-Time Self-Adaptive Deep Stereo”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-13, 2019.

[5] X. Song, X. Zhao, L. Fang and H. Hu, “EdgeStereo: An Effective Multi-Task Learning Network for Stereo Matching and Edge Detection”, *International Journal of Computer Vision*, Vol. 128, pp. 910-930, 2020.

[6] Middlebury Stereo Datasets, Available at https://vision.middlebury.edu/stereo/data/ last, Accessed at 2020.

[7] R.A. Hamzah and H. Ibrahim, “Literature Survey on Stereo Vision Disparity Map Algorithms”, *Journal of Sensors*, Vol. 2016, pp. 1-23, 2016.

[8] K. Zhou, X. Meng and Bo Cheng, “Review of Stereo Matching Algorithms Based on Deep Learning”, *Computational Intelligence and Neuroscience*, Vol. 2020, pp. 1-18, 2020.

[9] W. Luo, A.G. Schwing and Raquel Urtasun, “Efficient Deep Learning for Stereo Matching”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5695-5703, 2016.

[10] H. Park and K.M. Lee, “Look Wider to Match Image Patches with Convolutional Neural Networks”, *IEEE Signal Processing Letters*, Vol. 24, pp. 1788-1792, 2017.

[11] X. Ye, J. Lib, H. Wang and X. Zhang, “Feature Ensemble Network with Occlusion Disambiguation for Accurate Patch-Based Stereo Matching”, *IEICE Transactions on Information and Systems*, Vol. 100, No. 12, pp. 3077-3080, 2017.

[12] P. Brandao, E. Mazomenos and D. Stoyanov, “Widening Siamese Architectures for Stereo Matching”, *Pattern Recognition Letters*, Vol. 120, pp. 75-81, 2019.

[13] B. Chen and C. Jung, “Patch-Based Stereo Matching using 3D Convolutional Neural Networks”, *Proceedings of IEEE International Conference on Image Processing*, pp. 3633-3637, 2018.

[14] H. Hirschmuller, “Stereo Processing by Semiglobal Matching and Mutual Information”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 2, pp. 328-341, 2008.

[15] R. Verma, H.S. Singh and A.K. Verma, “Depth Estimation from Stereo Images Based on Adaptive Weight and Segmentation”, *Journal of the Institution of Engineers (India): Series B - Electrical, Electronics and Telecommunication and Computer Engineering*, Vol. 93, No. 4, pp. 223-229, 2013.

[16] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, “SLIC Superpixels Compared to State-of-the-Art Superpixel Methods”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 11, pp. 2274-2282, 2012.