ENHANCEMENT OF CONSISTENT DEPTH ESTIMATION FOR MONOCULAR VIDEOS APPROACH

Mohamed N. Sweilam[1,2,*] and Nikolay Tolstokulakov[2]

[1]Faculty of Computers and Information, Suez University, Egypt.
[2]Department of Mechanics and Mathematics,
Novosibirsk State University, Russia.

ABSTRACT

Depth estimation has made great progress in the last few years due to its applications in robotics science and computer vision. Various methods have been developed and implemented to estimate the depth, without flickers and missing holes. Despite this progress, it is still one of the main challenges for researchers, especially for the video applications which have more difficulties such as the complexity of the neural network which affects the run time. Moreover to use such input like monocular video for depth estimation is considered an attractive idea, particularly for hand-held devices such as mobile phones, nowadays they are very popular for capturing pictures and videos. Here in this work, we focus on enhancing the existing consistent depth estimation for monocular videos approach to be with less usage of memory and with using less number of parameters without having a significant reduction in the quality of the depth estimation.

KEYWORDS

Monocular video, monocular depth estimation, deep learning, geometric consistency, lightweight network.

1. INTRODUCTION

Depth estimation of a monocular video presents an attractive point of research for computer vision, and is important for Robotics to provide the distance information needed for different applications, 3D reconstruction of scenes, augmented reality, and object detection. Nowadays, most of the research works are focusing on the unsupervised monocular depth estimation as most of the techniques produce a prediction of depth as a supervised problem and it requires a lot of ground truth depth data for training even for the depth estimation for a single image such as Eigen et al.[1,2] their technique as results have dense pixel depth estimation using a two deep neural network have trained on images and their corresponding depth values, Karsch et al. [3] tried to have a consistent image predictions by taking a copy from the whole depth images from a training data set. The problem in that technique is that it requires the whole training set to be available at the test time. All previous methods and the other supervised methods require a high quality, pixel aligned, ground truth depth data at the training time. But here we perform our work using a single depth estimation network and apply it on the video frames but as an unsupervised method as it needs a stereo color image, instead of ground truth depth during the training time. The Deep3D network of Xie et al. [4] is an unsupervised technique aiming to produce the corresponding right view from an input as a left image to be the context of binocular pairs. The

David C. Wyld et al. (Eds): ITCSE, ICDIPV, NC, CBIoT, CAIML, CRYPIS, ICAIT, NLCA - 2021
pp. 105-115, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110910

right image pixels as the results are a combination of the pixels on the same scan line from the left image which was the input, weighted by the probability of each disparity. The disadvantage of this technique is that increasing the number of disparity values leads to increase in memory consumption, which makes it difficult to apply on bigger data or bigger resolution. For depth estimation from a video is a challenging problem recently because of moving objects and camera pose ,that's why the video depth estimation technique suffering from poorly textured areas, occlusions and repetitive patterns. The existing techniques for that purpose rely on motion segmentation and explicit motion modeling for the moving objects like [5]; Moreover now the one of the easiest ways to capture a video is by hand held camera phones which leads to more challenges such as high noise level ,lightening ,motion blur and shaking that's why the existing method produce some errors in the depth estimation such as missing regions which make some white holes in the depth, in addition to it's consider as an inconsistent geometry depth and flickering depth such as in figure [1] ,that's why the geometrically consistent approaches have the best results and accurate ones but suffering from complexity and long test time as the The produced depth is flicker free and geometrically consistent throughout the input monocular video. For these reasons we have decided to enhance the Consistent Video Depth Estimation approach [6] by making that approach use less memory and lighter depth estimation network.

2. CONSISTENT VIDEO DEPTH ESTIMATION

The Consistent Video Depth Estimation approach produces a single image depth estimation and further improves the geometric consistency values of the depth estimation on the videos. It contains two phases :

2.1. Pre-processing

In that phase the approach performs a traditional Structure from Motion (SfM) reconstruction using the open source software COLMAP [7] and using Mask R CNN [8] to detect people segmentation and remove these regions to make it more reliable for keypoint extraction and matching. This phase is important to provide accurate intrinsic and extrinsic camera parameters in addition to a sparse point cloud reconstruction.

2.2. Test-time Training

In that phase what happens is that the approach takes two frames randomly and then have their depth images after processing them to a single depth estimation network and usually the results of that network will have some flickers. Moreover, it calculate the camera pose using COLMAP [7] , then assume we have a point on an image then the approach will find the corresponding point in the other frame using Optical Flow and re project the two point in a 3-D scene and here this process break down into two components, 1) re project the a point to another camera and calculate the distance on image plane and 2) re project a two points along the Z axis and compute the difference. Because of the depth is inconsistent there will be a distance between the two point in the 3-D scene which called geometric losses as Spatial Loss for distance between the two points in the screen space and Disparity Loss for the distance between the two points in the depth space.

For these losses the depth is inconsistent so the approach takes the two losses and fine tune the initial single depth estimation network by back propagation at the test time for all pairs of frames. Finally the approach will estimate a sharper and geometric consistent depth which will be very accurate and better than before.

2.3. Geometric Losses

This approach has two geometric losses as the following if we assume we have a given frame pair(i,j) so:

$$L^{spatial}_{i\rightarrow j}(x) = ||p_{i\rightarrow j}(x) - f_{i\rightarrow j}(x)||_2 \quad (1)$$

Where the flow displaced point is, $f_{i\rightarrow j(x)}$ the depth reprojected point $p_{i\rightarrow j(x)}$. so the image space loss (1) indicates the distance in the image space between the flow displaced point and the depth reprojected point.

$$L^{disparity}_{i\rightarrow j}(x) = u_i|z^{-1}_{i\rightarrow j}(x) - z^{-1}_j(f_{i\rightarrow j}(x))| \quad (2)$$

Where u_i is the frame’s focal length, the z components are scalar z components from a 3D point in the frame's camera coordinate system.

The Total loss is a combination of both losses for all pixels:

$$L_{i\rightarrow j} = \frac{1}{|M_{i\rightarrow j}|} \sum_{x\epsilon M_{i\rightarrow j}} L^{spatial}_{i\rightarrow j}(x) + \lambda L^{disparity}_{i\rightarrow j}(x) \quad (3)$$

Where λ=0.1 is a balancing coefficient.

2.4. Optimization

That approach takes the geometric loss between the frames and fine-tunes the initial depth estimation network using the standard back propagation.

Having the parameters of this network using a pre-trained network for depth estimation allows the approach to transfer the knowledge to produce the depth map on the images that are already considered as challenging for traditional geometric based reconstruction. This approach fine tune using 20 epochs for all experiments.

3. PYD-NET

It's a lightweight network called Pyramidal Depth Network (PyD-Net) [9], when we train it in an unsupervised method, it represents a high accuracy in that field.

If we compare that model after training to the others, this model is about 94% smaller as it can even work on CPUs without reducing the accuracy slightly, it requires limited resources only. Moreover, the PyD-Net can be deployed even on embedded devices, such as the Raspberry Pi 3, allowing to have depth with low number of parameters using less than 150 MB memory available at test time because the other approaches count a huge number of parameters and thus require a large amount of memory For Example with the VGG model [19], counts 31 million parameters, however in the Pyd-net number of parameters reach to 2 Millions of parameters. Which makes Pyd-net more efficient for low power devices or CPUs.

3.1. The Architecture

The PyD-Net architecture in Figure 2, as it contains a pyramid of features coming from the input image and at each level of that pyramid a network build depth. The features which they proceed are sampled to the upper level to refine the estimation, up to the highest one.

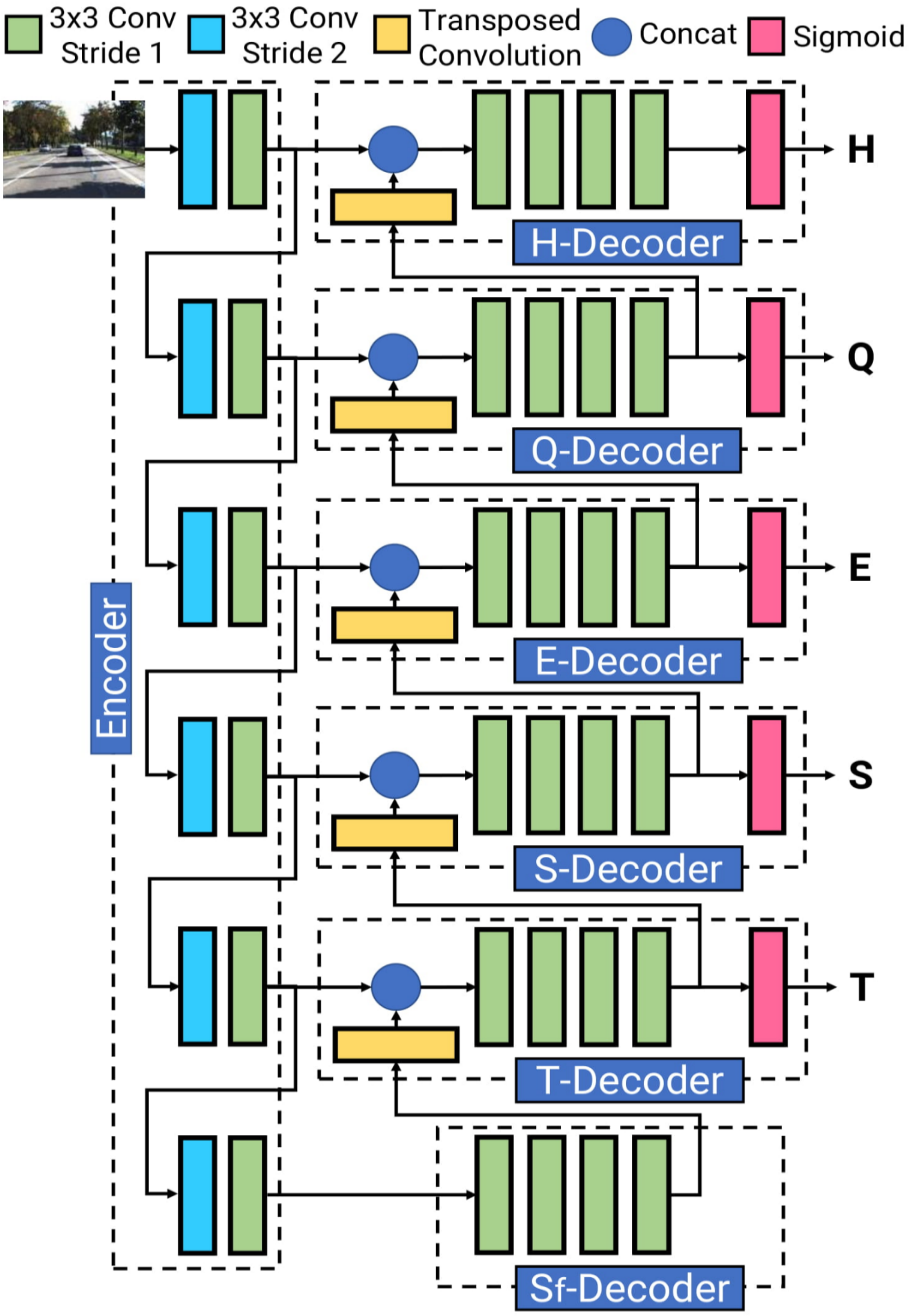

Figure 2. Pydnet architecture

3.2. Pyramidal Features Extractor

It's a small encoder made by [10], made of 12 convolutional layers. At full resolution, the first level of the pyramid is produced by the first layer by applying convolutions with stride 2 followed by a second convolutional layer. By having the same technique for each level the resolution is reached to the lowest resolution at the highest level ,the total number of levels is 6 levels, from L1 to L6, corresponding respectively to the image resolution from 1/2 to 1/64 of the original input size.

Every down sampling module builds a number of extracted features, respectively 16, 32, 64, 96, 128, and 192,and each convolutional layer deploys 3×3 kernels and is followed by a ReLU with α= 0.2.

3.3. Depth Decoders and Upsampling

There is a decoder at the highest level of the pyramid, the decoder made of 4 convolutional layers, producing respectively 96, 64, 32 and 8 feature maps.

This decoder has two purposes: 1) to produce a depth map at the current resolution, by means of sigmoid operator, and 2) to pass the features which processing to the next level in the pyramid, by means of a 2×2 deconvolution with stride 2 which increases by a factor 2 the spatial resolution.

The next level matches the features extracted from the input frame with the features which are sampled and processes them with a new decoder, repeating this procedure up to the highest resolution level.

Each convolutional layer uses 3×3 kernels, leaky ReLU activations, just the last one followed by a Sigmoid activation for normalizing the outputs. This design makes at each scale the PyD-Net to learn to produce depth at full resolution.

4. MONODEPTH

A new training method made by C. Godard, O. Mac Aodha, and G. J. Brostow[11] for enabling the convolutional neural network to learn to make a single image depth estimation , with the absence of ground truth depth data. Using the epipolar geometry constraints, the method generates disparity images by training the network with an image reconstruction loss. That is why it is considered as a new training loss which enforces consistency between the disparities estimated according to the left and right images, which leads to to improve the performance and robustness compared to the existing techniques. This method has state of the art results for monocular depth estimation when it is trained on the KITTI driving dataset, and even better than the supervised methods which have been trained on ground truth depth.

4.1. Sampling Strategies

Sampling strategies for backward mapping as in figure [3] here which originally in [11]. With naive sampling the CNN generates a disparity map of the right image from the left image and the disparity map aligned with the right image which is the target.

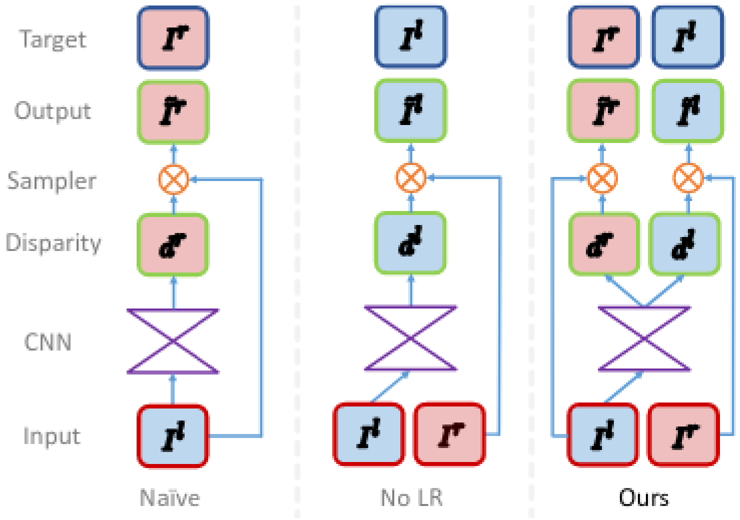

Figure 3

But the output should be the disparity map of the left image which considered as the input therefore the network has to sample it from the right image that's why the second strategy which is No LR is to train the network to produce the left view from the right image and creating a disparity map aligned to left view but at that strategy there are some errors like exhibit 'texture-copy' artifacts at depth discontinuities and as a solution for that the approach has to be trained to produce disparity maps for both right and left views by sampling it from the opposite input image. It requires only one image during the test time as the left view but in training time needs the right view image too .Enforcing the consistency for both disparity maps like that leads to better accuracy.

4.2. The Network Architecture

The architecture here by Disp-Net [12], with some modifications to train with the absence of ground truth data .The network contains two main parts: an encoder and decoder. The decoder uses a technique called skip connections [13] from the encoder's activation blocks, it helps for high resolution details. The output predictions contain four scales (disparity 4 to disparity 1); it is going to be doubled in spatial resolution at every subsequent scale. Moreover, it only takes a single image as input, the network produces two disparity maps at every output scale left to right and right to left.

4.3. Training Loss

The loss C_s here calculated at each outer scale s, the C_s has a three parts

$$C_s = \alpha_{ap}(C^l_{ap} + C^r_{ap}) + \alpha_{ds}(C^l_{ds} + C^r_{ds}) + \alpha_{lr}(C^l_{lr} + C^r_{lr}) \tag{3}$$

Where C_{ap} to make the reconstructed image to similar to the corresponding training input image , ap C_{ds} for the smooth disparities, and C_{lr} to make the produced left and right disparities consistency.

$$C^l_{ap} = \frac{1}{N} \sum_{i,j} \alpha \frac{1 - SSIM(I^l_{ij}, \hat{I}^l_{ij})}{2} + (1 - \alpha) || I^l_{ij}, \hat{I}^l_{ij} || \quad (4)$$

this equation (4) for reconstruction error calculating the difference between the original image *I* and the warped one*I* by using SSIM [14].

$$C^l_{ds} = \frac{1}{N} |\partial_x d^l_{ij}| e^{-||\partial_x l^l_{ij}||} + |\partial_y d^l_{ij}| e^{-||\partial_y l^l_{ij}||} \quad (5)$$

Equation (5) for disparity smoothness, it tries to make the disparities to be locally smooth on the disparity gradients ∂.

$$C^l_{lr} = \frac{1}{N} \sum_{i,j} |d^l_{ij} - d^r_{ij+d^l_{ij}}| \quad (6)$$

Equation (6) represents the Left Right Disparity Consistency Loss as the approach trying to produce more accurate disparity maps, the training of the network to produce both the left and right image maps, however the only input is the left view to the network.

At test time, the disparity for the left image d^l, it has the same resolution as the input image. While estimating the right disparity d^r during training, it is not used at test time. Using the camera baseline and focal length from the training set, the approach converts the disparity map to a depth map.

5. EXPERIMENT

Our Experiment is to enhance Consistent Video Depth Estimation approach by changing the initial depth estimation for single image in that approach to be more lighter and use less memory, so we decided to use a lightweight architecture for network which we chose to be the Pyd-net architecture[9] as it has the ability to enable such an accurate and unsupervised monocular depth estimation with very limited resource requirements, but it needs a framework to train with, therefore we used Monodepth framework[11] for training as it has better results even than the supervised methods that have been trained with ground truth daat. After testing the pretrained network we succeeded to integrate the network in the Consistent Video Depth Estimation approach and run the model in test time and observe the fine tuning process as it used 20 epochs for fine tuning, then we evaluated the results, tested on hand-held videos and compared.

5.1. Dataset

KITTI dataset [15] as it has been recorded from a moving platform while driving in and around Karlsruhe, Germany. Using KITTI Split which contains contain 30,159 images almost 175 GB, we keep 29,000 for training and the rest for evaluation .we used unlabeled stereo pairs of images as according to the approach we are using for training, we need at the training phase to have right and left view, but in test time we need just one image.

Example from the KITTI dataset of a stereo image the upper one is right view and the below one is the left view

5.2. Implementation

We implemented our work in Pytorch [16] ,therefore we had to reimplement the whole architecture of Pydnet into Pytorch as it is official as tensorflow. We also had to implement the framework of Monodepth in Pytorch and train it. We use the specifications of Novosibirsk State University for training, we used a GeForce GTX TITAN X GM200 as GPU, trained using 200 epochs with a batch size of 12 and using Adam optimizer [17] and the loss during training shown in figure [4].

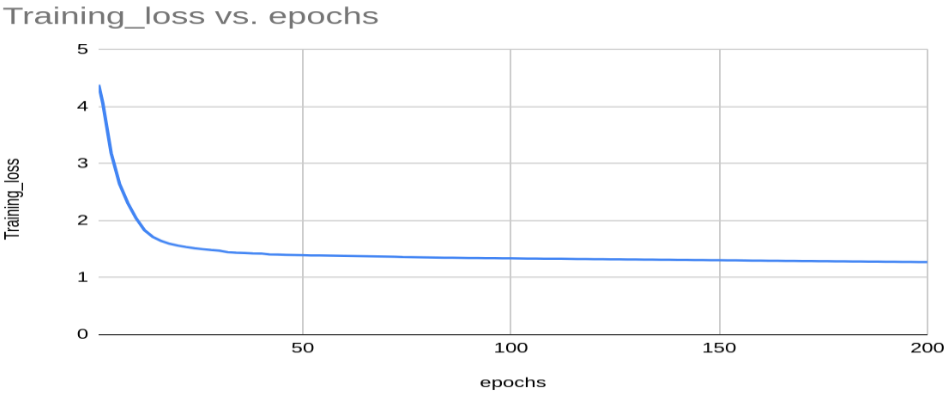

Figure 4. Train loss during training for 200 epochs.

5.3. Evaluation Metrics

To compare with the previous methods we use the popular Evaluation metrics for comparison several errors from prior works such as [18] and we used these metrics: SQ_REL:Relative squared error,ABS_REL: Relative absolute error ,RMSE: Root mean squared error of the inverse depth and RMSE (log)

5.4. Results

For the visual results we didn’t notice a significant difference between our results after the modification and the previous results even with our videos which are shown in figure [5], figure [6].

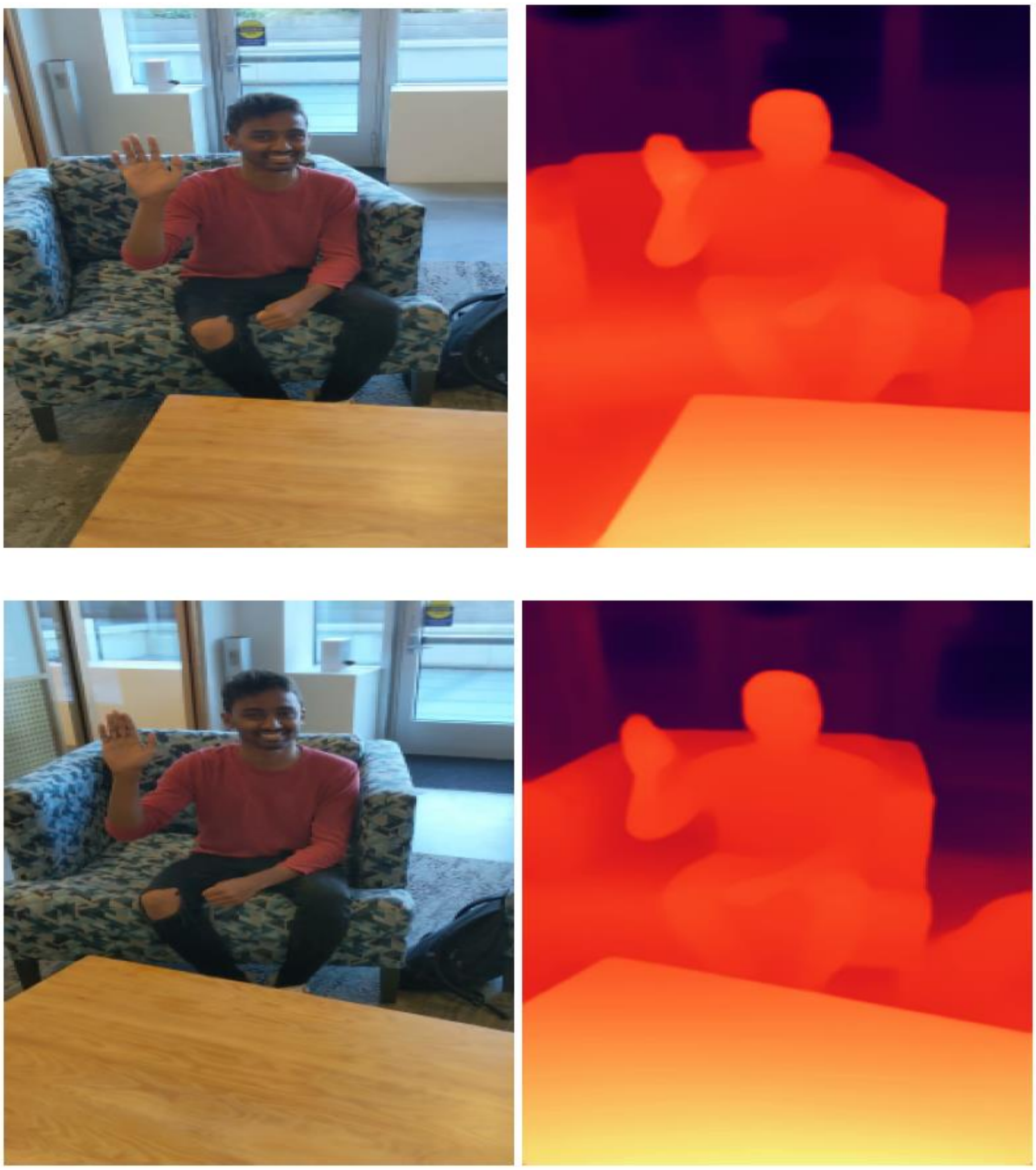

Figure 5. a test video frames from the previous work , the upper frame and its depth is using the existing approach , the below frame and its depth is using the approach after enhancement.

For Quantitative results we compared using the evaluations metrics between our modification and two from the previous models for depth estimation approaches ,shown in table [1]:

Table 1: comparison of depth Models which are all trained and evaluated on KittiRAW

Model	ABS_REL	SQ_REL	RMSE	RMSE (log)
Ours	0.149	1.250	5.464	0.228
Monodepth2	0.132	1.042	5.138	0.210
Fast depth	0.317	13.325	10.207	0.384

We noticed that our modification has very close numbers to the Monodepth 2 [21] the better than fast depth [20], however it have a significant difference in the memory usage as we have done an experiment to compare the memory usage, we have noticed that Monodepth2 using 0.95 GB of memory at the test time and Monodepth without any modifications using 2.14 GB at test time, in the other hand our modification with Pyd-net uses less than 150 MB for all experiments. Which is considered as a one step to make the Consistent Video Depth Estimation faster and lighter so it can be applied on cell phones or low power devices. Moreover we observed the geometric loss during the test time in that approach after modifying it and we noticed that the geometric loss values before and after modifications were in the same range between 0.2 and 0.8, which means that the new initial depth network did not cause more geometric loss than before.

6. CONCLUSION

In this research, we have proposed a new modification for the Consistent Video Depth Estimation approach which uses a huge of memory at the test time, therefore we have reduced that amount by changing the initial depth estimation network for a single image in that approach with a new one enhanced by a lightweight architecture can be used for low power devices and mobile phones which is Pyd-net. After testing, the results showed that there is no significant difference in the depth quality, however there is a significant difference in the memory usage at the test time. The future work is to try to focus more on the geometric consistency to make it less complex and lighter which can make the approach in future lighter to work on mobile phones or low power devices.

ACKNOWLEDGEMENT

We gratefully acknowledge the support of Stream Data Analytics and Machine Learning laboratory at Novosibirsk State University for providing us the specifications we needed to complete this research.

REFERENCES

[1] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. InICCV, 2015

[2] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. InNIPS, 2014.

[3] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling.PAMI,2014.

[4] J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. InECCV, 2016

[5] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth Prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In AAAI Conference on Artificial Intelligence (AAAI),2019.

[6] Luo, Xuan and Huang, Jia Bin and Szeliski, Richard and Matzen, Kevin and Kopf, Johannes. Consistent Video Depth Estimation .InACM Transactions on Graphics (Proceedings of ACM SIGGRAPH),2020.

[7] Johannes L Schonberger and Jan-Michael Frahm. Structure from motion revisited. InIEEE Conference on Computer Vision and Pattern Recognition (CVPR).2016.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. InInternational Conference on Computer Vision.2017.

[9] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "Towards real-time unsupervised monocular depth estimation on cpu," inIROS, 2018.

[10] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume,"arXiv preprintarXiv:1709.02371, 2017.

[11] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monoc-ular depth estimation with left-right consistency," inCVPR, vol. 2,no. 6, 2017, p. 7.

[12] N. Mayer, E. Ilg, P.H. Ausser, P. Fischer, D. Cremers, A. Dosovit-skiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. InCVPR,2016.

[13] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation.PAMI, 2016

[14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality assessment: from error visibility to structural similarity,"IEEETrans. on Image Processing, vol. 13, no. 4, pp. 600–612, 2004.

[15] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. InCVPR, 2012.

[16] Paszke, Adam and Gross, Sam and Chintala, Soumith and Chanan, Gregory and Yang, Edward and DeVito, Zachary and Lin, Zeming and Desmaison, Alban and Antiga, Luca and Lerer, Adam. Automatic differentiation in PyTorch.2017.

[17] D. Kingma and J. Ba, "Adam: A method for stochastic optimiza-tion,"International Conference on Learning Representations, 12 2014

[18] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction froma single image using a multi-scale deep network. InNIPS, 2014.

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

[20] Wofk, Diana and Ma, Fangchang and Yang, Tien-Ju and Karaman, Sertac and Sze, Vivienne. "FastDepth: Fast Monocular Depth Estimation on Embedded Systems", IEEE International Conference on Robotics and Automation (ICRA) .2019.

[21] Godard and Oisin {Mac Aodha} and Michael Firman and Gabriel J. Brostow."Digging into Self-Supervised Monocular Depth Prediction" .The International Conference on Computer Vision (ICCV).2019.