JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

ESTIMATION OF COLOUR IMAGES USING DEPTH ENHANCEMENT TECHNIQUES

[1]B.P.Santosh Kumar, [2]Shafiulla Basha Shaik, [3]M. Sai Jahnavi, [4]E.Naveen,[5] K.Prasannalakshmi, [6]M.Akhila.

[1, 2] Associate Professor, Dept. of ECE, YSR Engineering College of YVU.

[3,4,5,6,7] Student of 4th B.Tech, Dept. of ECE, YSR Engineering College of YVU.

Abstract: Depth estimation from a single image is a crucial task in computer vision with applications ranging from autonomous driving to augmented reality. In this project, we employ the UNET++ architecture, a variant of the popular UNET model, for depth estimation. The model is trained on a diverse dataset containing images paired with their corresponding depth maps. During training, we monitor several metrics including training loss, validation loss, structural similarity index (SSIM) scores, and mean squared error (MSE) loss, to ensure both accurate depth predictions and model generalization. Our experiments demonstrate the effectiveness of the UNET++ model in capturing intricate depth details, as indicated by high SSIM scores and low MSE losses on both training and validation sets. This project contributes to advancing depth estimation techniques, offering insights into improving depth perception.

I. INTRODUCTION:

Depth estimation holds significant importance in 3D scene reconstruction and comprehension. Thanks to advancements in deep convolutional neural networks, there have been numerous endeavors to estimate pixel-wise metric depth from RGB images. These efforts encompass both single-view and multi-view methodologies, each leveraging distinct cues with inherent strengths and weaknesses. Single-view techniques rely on monocular cues like texture gradients and known object sizes. Utilizing a deep feature extractor, these cues are encoded into a dense feature map, from which a decoder infers per-pixel depth. Despite their ability to discern depth on weakly textured or reflective surfaces, single-view methods are constrained by the inherent ambiguity of the task. Conversely, multi-view methods capitalize on geometric cues, assuming that correctly estimated depths will project onto visually similar pixels in other images. While this approach minimizes ambiguity and enhances accuracy, it faces challenges such as the need to evaluate numerous depth candidates, issues with occlusion and object motion, and unreliability on texture-less or reflective surfaces. We propose a hybrid approach that combines both monocular and geometric cues to complement each other's limitations. By integrating multi-view matching to alleviate single-view depth ambiguity, enhancing efficiency through targeted depth candidate sampling, and enforcing consistency with single-view depth to address failure cases, we aim to improve overall depth estimation performance. In this project, we implement the UNET++ architecture, a variant of the widely used UNET model, for depth estimation. Training on a diverse dataset comprising image-depth map pairs, we track various metrics including training and validation loss, structural similarity index (SSIM) scores, and mean

squared error (MSE) loss to ensure accurate depth predictions and model generalization. Our experiments highlight the effectiveness of the UNET++ model in capturing intricate depth details, evidenced by high SSIM scores and low MSE losses on both training and validation datasets. This research contributes to advancing depth estimation techniques, providing valuable insights into enhancing depth perception capabilities.

2. Research Methodology:

2.1 System design:

In our project, titled "Extraction of colour images using depth enhancement techniques," we aimed to develop a robust system for depth estimation from single images, a critical task in computer vision with broad applications ranging from autonomous driving to augmented reality. The steps involved in this system were shown in Figure.

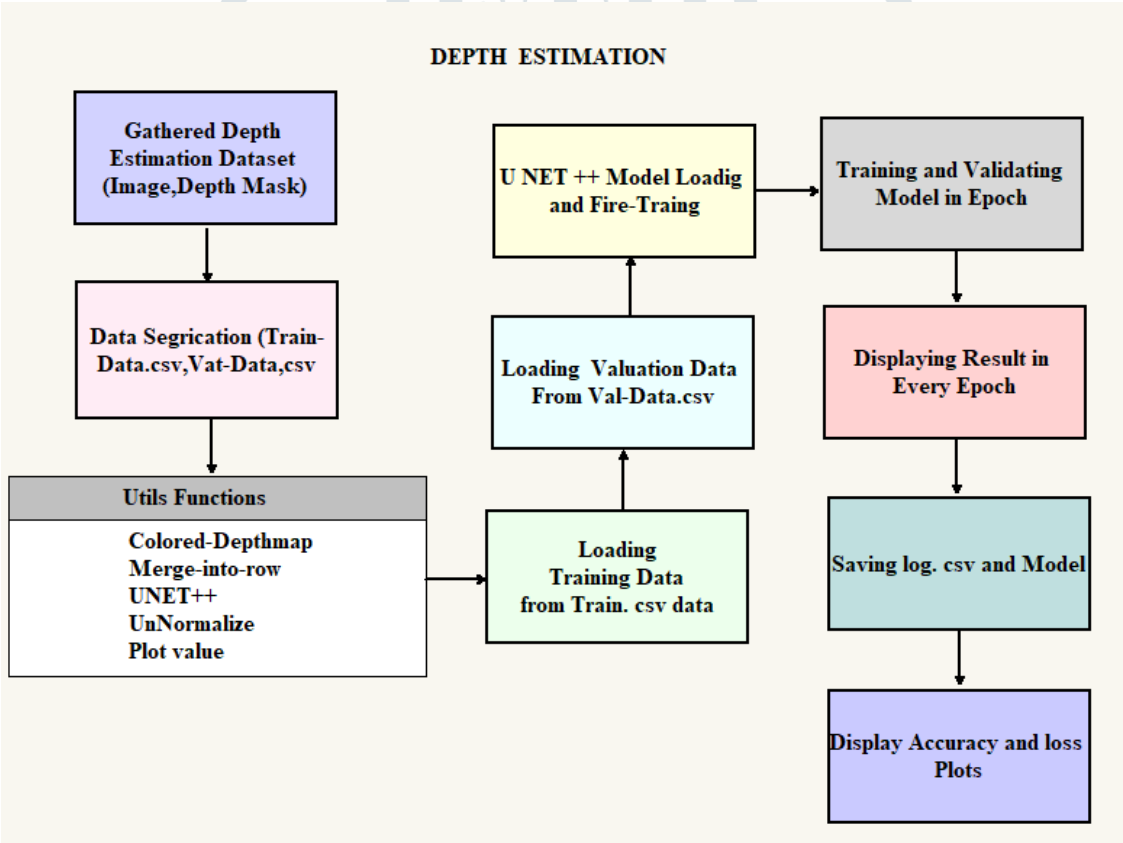

Figure 1: System Architecture

The above Figure 1 depicts a methodology for depth estimation from a single image using a UNET++ model. Here's a breakdown of the process:

1. **Gathered Depth Estimation Dataset:** The system begins by collecting a dataset of images paired with their corresponding depth maps. These depth maps are essentially labels that provide information about the distance of each pixel from the camera.

2. **Data Segregation:** The dataset is then divided into training and validation sets. The training set is used to train the model, while the validation set is used to evaluate the model's performance and prevent overfitting.

3. **U-Net++ Model Loading and Training:** A UNET++ model is loaded. UNET++ is a convolutional neural network (CNN) architecture specifically designed for image segmentation tasks. It's a variant of the popular UNET model known for its ability to capture complex features from images.

During training, the model processes images from the training set along with their corresponding depth maps. The model learns to map the input images to the corresponding depth maps by minimizing a loss function. The loss function measures the difference between the model's predictions and the actual depth maps.

4. **Training and Validation:** Here, the training process iterates through multiple epochs (complete passes through the training data). In each epoch, the model is shown batches of training images and their corresponding depth maps. The model updates its weights based on the calculated loss.

To monitor the training progress, the system also evaluates the model's performance on the validation set after each epoch. This helps ensure the model is generalizing well and not simply memorizing the training data.

5. **Utils Functions:** The system likely utilizes various utility functions to perform tasks such as:
    - Converting the colored depth maps into a format suitable for training the model.
    - Merging images into rows to prepare them for visualization.
    - Normalizing and unnormalizing data (a common pre-processing step in training neural networks).
    - Plotting value distribution for visualization purposes.
6. **Loading Training Data:** During training, the system loads batches of training images and their corresponding depth maps from the training set.
7. **Displaying Results in Every Epoch:** After each epoch, the system likely generates visualizations of the model's depth predictions on a set of images. This helps you monitor how the model's performance is improving over time.
8. **Saving Logs and Model:** The system may also save logs of the training process, including metrics like training loss and validation loss. Additionally, it might save the trained model at regular intervals so you can resume training later or use the model for making predictions on new images.
9. **Displaying Accuracy and Loss Plots:** Finally, the system may generate plots that show the model's training and validation loss over time. This can help you diagnose any training issues and assess how well the model is generalizing.

Overall, the methodology leverages a UNET++ architecture to learn depth estimation from a single image by training on a dataset of images and corresponding depth maps. By monitoring various metrics and visualizing results, you can gain insights into the model's performance and make adjustments as needed.

2.2 proposed architecture:

U-Net++ or Nested U-Net is a deep learning architecture that was introduced in 2019 in the "UNet++. In UNet, the encoder part captures high-level features from the input image through a series of convolutional and pooling layers, while the decoder part upsamples these features to generate a dense segmentation map. However, there can be a semantic gap between the encoder and decoder features, meaning that the decoder may struggle to reconstruct fine-grained details and produce accurate segmentation.

UNet++ introduces the concept of nested skip pathways to bridge this semantic gap. It adds additional skip connections between the encoder and decoder blocks at multiple resolutions. These connections allow the decoder to access and incorporate both low-level and high-level features from the encoder, providing a more detailed and comprehensive understanding of the image. A Nested U-Net Architecture for Medical Image Segmentation" paper. They improved the traditional U-Net architecture by redesigning the skip connections and introducing a deeply supervised nested encoder-decoder network. This article discusses the U-Net++ architecture and also covers its implementation in Python using the TensorFlow library.

U-Net++ architecture is a semantic segmentation architecture based on U-Net. They introduced two main innovations in the traditional U-Net, architecture namely, nested dense skip connections and deep supervision. In their research, they found that using nested dense skip connections bridges the semantic gap between encoder and decoder feature maps and improves the gradient flow. They also found that using deep supervision enhances the model performance by providing a regularization to the network with training.

This illustrates the architecture design of U-Net++. This illustrates the nested encoder and decoder architecture of the U-Net++ architecture. We can notice that instead of a traditional skip connection, the feature map from the lower level is also convoluted with the upper-level feature and then the new combined feature data is then passed further. The basic idea behind UNet++ is to bridge the semantic gap between the feature maps of the encoder and decoder before the fusion. For example, the semantic gap between (X0,0, X2,2) is bridged using a dense convolution block with three convolution layers.

The black dotted skip connection indicated the original skip connection present in U-Net architecture while the blue dotted skip connection indicated the newly introduced nested skip connection. It must be noted that before convoluting the lower level feature map it is upscaled to match the number of channels in that level. The figure also illustrated how deep supervision is also applied to the output of nodes X0,1, X0,2, X0,3, and X0,4 to improve the model learning while training. Deep supervision is an optimization technique where you optimize the model on the final as well as hidden layers (or nodes) in the model. This helps the model to generalize the problem in a better way. In U-Net++ architecture, they optimized the model on the output of X0,1, X0,2, X0,3, and X0,4 nodes by calculating the combined loss on the expected output based on the output of each of these nodes.

The figure provides a detailed analysis of the first skip pathway of UNet++ and below diagram illustrates how the U-Net++ model can be pruned at inference time if it is trained with deep supervision. The design of U-

Net++ L1, U-Net++ L2, U-Net++ L3, and U-Net++ L4 illustrates the U-Net++ design with depth 1, 2, 3, and 4 respectively. These models are used to identify the effectiveness of the model by comparing its performance and inference time over the number of parameters in the model. In their experiments, they found that U-Net++ L3 achieves on average 32.2% reduction in inference time while degrading IoU by only 0.6 points. Results

Analysis:

In the result analysis phase, we scrutinized the performance of our system trained on the UNET++ architecture for depth estimation. Through comprehensive evaluation using SSIM scores and MSE losses on both training and validation datasets, we assessed the model's ability to accurately capture depth details from single images.

RESULTS AND DISCUSSION

The loss plot graph serves as a fundamental tool in assessing the performance and learning dynamics of a deep learning model. By visualizing the changes in loss values over training epochs, it offers valuable insights into the model's convergence, stability, and overall effectiveness.

Training Loss Evolution

The model was trained using the Unet++ architecture for depth estimation. The initial training loss (loss_train) was 0.393687, indicating the average loss per sample on the training set. As the model underwent training iterations, the training loss gradually decreased. After a certain number of epochs, the training loss reached 0.000522, showing the model's improved ability to estimate depth from images in the training set.

Validation Loss Evolution

Similarly, the initial validation loss (loss_val) was 0.190576, representing the average loss per sample on the validation set. Throughout the training process, the validation loss also decreased. Eventually, it reached 0.000725, indicating that the model's performance generalized well to unseen data, as demonstrated by the reduced loss on the validation set.

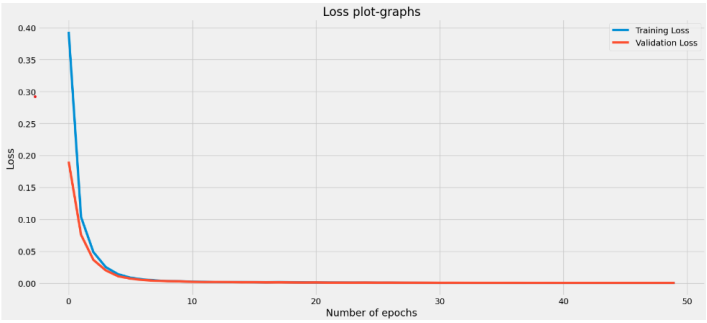

Figure 5: Loss plot-graph

The decreasing trends in both training and validation losses suggest that the model effectively learned to estimate depth from images, with the validation loss showing the model's generalization ability.

Mean Squared Error plot-graphs

In this section, we analyze the results of the depth estimation model using UNet++ through the Mean Squared Error (MSE) plot graph. The MSE metric provides insight into the model's performance by measuring the average squared differences between the predicted depth maps and the ground truth depth maps. A lower MSE value indicates better agreement between the predicted and ground truth depth maps. By examining the MSE plot graph, we can assess how effectively the model learns to estimate depth information and identify any trends or patterns in its performance over the training epochs.

Training MSE Evolution

The model was trained using the Unet++ architecture for depth estimation. The initial Mean Squared Error (MSE) on the training set (mse_train) was 0.394306. This value represents the average squared difference between the predicted depth map and the ground truth depth map for each sample in the training set.

As the model underwent training iterations, the training MSE gradually decreased. After a certain number of epochs, the training MSE reached 0.000522, indicating the model's improved ability to estimate depth from images in the training set.

Validation MSE Evolution

Similarly, the initial MSE on the validation set (mse_val) was 0.190151, representing the average squared difference between the predicted depth map and the ground truth depth map for each sample in the validation set. Throughout the training process, the validation MSE also decreased. Eventually, it reached 0.000703, indicating that the model's performance generalized well to unseen data, as demonstrated by the reduced MSE on the validation set.

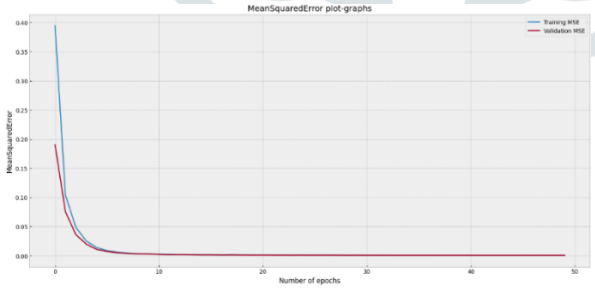

Figure 6:Mean Squared Error plot-graphs

The decreasing trends in both training and validation MSEs suggest that the model effectively learned to estimate depth from images, with the validation MSE showing the model's generalization ability.

Structural Similarity Index Measure plot-graphs

The analysis of results is crucial to evaluate the performance and effectiveness of the 'depth estimation using UNet++' model. In this section, we present a detailed examination of the Structural Similarity Index Measure (SSIM) to assess the quality of the depth maps produced by the model. The SSIM is a widely used metric that quantifies the similarity between two images, with values closer to 1 indicating a higher degree of similarity.

By plotting and analyzing the SSIM values obtained during training and validation, we can gain insights into the model's learning process and its ability to generate accurate depth estimations.

Training SSIM

The training SSIM (Structural Similarity Index) for the depth estimation model using UNet++ increased from 0.01561 to 0.884883. This significant improvement demonstrates the effectiveness of the model in capturing structural similarities between the predicted depth maps and ground truth depth maps during the training process.

The initial SSIM value of 0.01561 indicates a low level of similarity, which suggests that the model's predictions diverged considerably from the ground truth. However, through iterative training and optimization, the SSIM increased to 0.884883, indicating a high level of similarity between the predicted and ground truth depth maps.

This improvement in SSIM highlights the model's ability to learn and refine its predictions over time, ultimately producing more accurate and visually consistent depth maps.

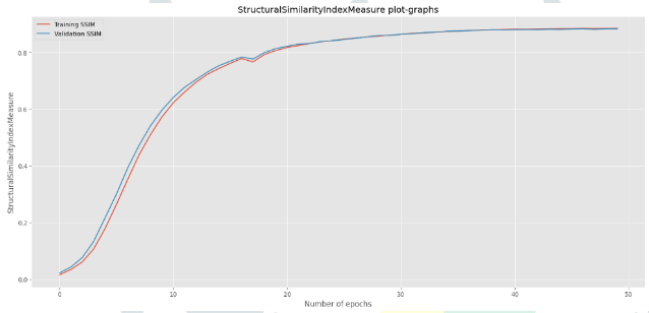

Figure 7: Structural Similarity Index Measure plot-graphs

Name of the losses	Initial loss	Final loss
Training loss	0.393687	0.000522
Validation loss	0.190576	0.000725
MSE loss		
• Training MSE loss	0.3943306	0.000522
• Validation MSE loss	0.190151	0.000703
SSIM loss		
• Training SSIM loss	0.01561	0.884883

Comparison of initial and final losses:

ACKNOWLEDGMENT:

We take this opportunity to express our deepest gratitude appreciation to all those who have helped us directly or indirectly towards the successful completion of project. We wish to express our deep sense of gratitude from the bottom of our heart to our guide Dr. B. P. SANTOSH KUMAR, Head of the Department ECE, Y.S.R Engineering College of Y.V.U, Proddatur for his motivating discussions, overwhelming suggestions, ingenious encouragement, invaluable supervision, and exemplary guidance throughout this project work.

We express our thanks to all our college teaching and non-teaching staff members who encouraged and helped us in some way or other throughout the project work. Finally, we are thankful to all our friends who have in some way or the other helped us getting towards the completion of this project work.

REFERENCES

1.S.C. Chan, H.Y. Shum, K.T. Ng, Image-based rendering and synthesis—technological advances and challenges. IEEE Signal Process. Mag. 24(6), 22–33 (2007). https://doi.org/10.1109/Msp.2007.905702

2. K. Tang, L. Shi, S. Guo, S. Pan, H. Xing, S. Su, P. Guo, Z. Chen and Y. He, "Vision locating method based RGB-D camera for amphibious spherical robots", in IEEE International Conference on Mechatronics and Automation (ICMA), (2017)

3. H.M. Zhu, J.H. Yin, D. Yuan, SVCV: segmentation volume combined with cost volume for stereo matching. IET Com-put. Vision 11(8), 733–743 (2017). https://doi.org/10.1049/iet-cvi.2016.0446

4. A. Roy and S. Todorovic, "Monocular depth estimation using neural regression forest", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2016)

5. K.R. Vijayanagar, M. Loghman and J. Kim, "Refnement of depth maps generated by low-cost depth sensors", in International SoC Design Conference (ISOCC), (2012)

6. N.Y.C. Chang, T.H. Tsai, B.H. Hsu, Y.C. Chen, T.S. Chang, Algorithm and architecture of disparity estimation with mini-census adaptive support weight. IEEE Trans. Circuits Syst. Video Technol. 20(6), 792–805 (2010). https://doi.org/10.1109/Tcsvt.2010.2045814

7. C. Godard, O. Mac Aodha and G.J. Brostow, "Unsupervised monocular depth estimation with left-right consistency", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2017)