Generalisation of Monocular Depth Estimation on Aerial Images

1st Shwetang Dubey
(Member IEEE)
Information Technology
Indian Institute of information technology, Allahabad
Prayagraj, India
pro2017002@iiita.ac.in

2nd Yasir Arafat
Information Technology
Indian Institute of information technology, Allahabad
Prayagraj, India
iit2021129@iiita.ac.in

3rd Rehammatullah
Information Technology
Indian Institute of information technology, Allahabad
Prayagraj, India
iit2021187@iiita.ac.in

4th Nitish Bharat
Information Technology
Indian Institute of information technology, Allahabad
Prayagraj, India
iit2021241@iiita.ac.in

5th Anuj Kumar Verma
Information Technology
Indian Institute of information technology, Allahabad
Prayagraj, India
iit2021159@iiita.ac.in

6th Ritik Kumar
Information Technology
Indian Institute of information technology, Allahabad
Prayagraj, India
iit2021205@iiita.ac.in

7th Pavan Chakraborty
(Member IEEE)
Information Technology
Indian Institute of information technology, Allahabad
Prayagraj, India
pavan@iiita.ac.in

***Abstract*—Monocular depth estimate is an important computer vision problem that has applications in autonomous navigation, 3D reconstruction, and scene interpretation. It is the task of calculating depth information from a single input image. The goal of this study is to investigate self-supervised methods for monocular depth estimate from aerial photography that do not require costly ground truth data. We provide a paradigm based on self-supervised learning based on view synthesis between stereo image pairs and variational autoencoders (VAEs). The scene geometry and depth structure are encoded in a probabilistic latent representation that is learned by the VAE architecture. With losses calculated from the image reconstruction error, a self-supervised training procedure uses geometric constraints between stereo images to estimate depth maps and camera posture. To further enhance generalization under varying lighting circumstances, we further explore the incorporation of hybrid transformer encoders and domain separation approaches. We show through extensive tests on both synthetic and real-world aerial datasets that our self-supervised technique can effectively generate accurate depth maps without the need for ground truth supervision. Our approach opens the door to 3D scene knowledge from aerial platforms that is both scalable and economical. Along with dataset descriptions and a thorough discussion of the suggested methodology, including architectural decisions, loss functions, and training protocols, we also offer a thorough literature review.**

***Index Terms*—VAE, Computer Vision, Probabilistic Latent Representation, Self-supervised, Depth map, Ground Truth, Stereo Images.**

979-8-3503-4210-9/23/$31.00 ©2023 IEEE

I. Introduction

Accurate depth estimate from monocular pictures is essential for many computer vision applications, including autonomous navigation, 3D reconstruction, and scene interpretation. However, conventional techniques frequently depend on expensive ground truth data for training, which restricts scalability and financial viability, especially in aerial photo environments. In order to tackle this problem, we investigate self-supervised methods designed for monocular depth estimation, with a focus on the particular requirements of aerial images. By utilizing the capabilities of variational autoencoders (VAEs) and geometric constraints obtained from stereo picture pairs, we present a paradigm that eliminates the need for time-consuming annotations by learning depth information without explicit supervision. Additionally, we investigate the integration of hybrid transformer encoders and domain separation algorithms to enhance adaptability under varying lighting circumstances. We show the effectiveness of our method in producing realistic depth maps through thorough testing on artificial and real-world aerial datasets, highlighting its potential to enable scalable and affordable 3D scene interpretation from aerial platforms. This study advances the field of monocular depth estimate in aerial contexts by offering a thorough presentation of our methodology that includes training procedures, architectural nuances, loss formulations, and a thorough literature survey.

A. *Depth Estimation*

Dense depth estimation from monocular imagery is essential in photo-grammetric computer vision, enabling applications like 3D reconstruction and autonomous vehicle navigation. Establishing a correspondence field between pixels in different images is key, forming the basis for subsequent depth map generation. This field encapsulates pixel relationships across views. Transformation into a depth map relies on camera projection matrices, providing a detailed 3D representation of the scene. Methods for depth estimation include supervised and self-supervised learning approaches, offering avenues for accurate depth map generation without the need for extensive manual annotation.

*1) **Working of depth estimation**:* By the property of similarity of triangles

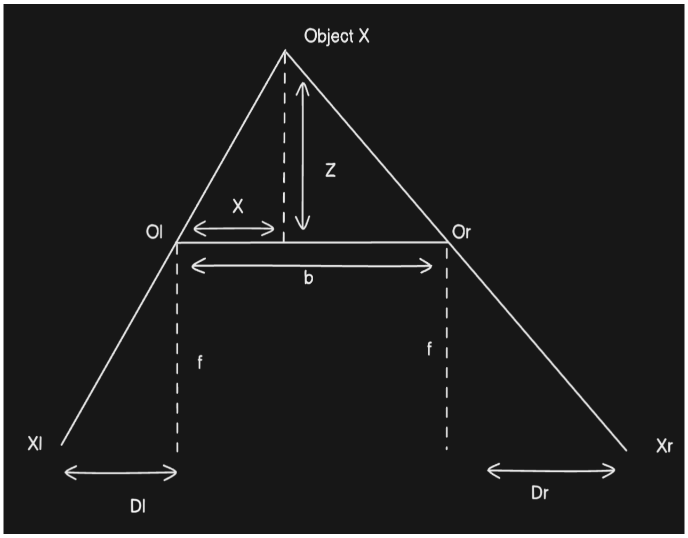

Fig. 1. Similar triangle For Depth Estimation[3]

$$\frac{z}{b} = \frac{z + f}{d_l + d_r + b} \qquad (1)$$

After simplifying, we get

$$z = \frac{f \cdot b}{d_l + d_r} \qquad (2)$$

Here, z is the depth of a point, f is the focal length of the camera, and b is the distance between left and right view points. Instead of relying on heavy mathematics, which may vary with respect to camera intrinsics, we employ deep learning-based models to predict the depth map and disparities of images D_l and D_r.

B. *Supervised Learning*

Learning-based approaches for dense depth estimation often rely on Convolutional Neural Networks (CNNs) trained in a supervised manner, utilizing ground truth depth maps to learn image-depth relationships [5]. However, acquiring extensive training data poses challenges, especially in domains like aerial imagery where datasets are limited. The scarcity of datasets containing both color images and dense depth maps hampers model development. Additionally, transferring trained models across diverse domains presents challenges in ensuring satisfactory performance in varied environments [5]. Generalization capabilities across different domains, including terrestrial and aerial imagery, remain areas for further research and investigation.

C. *Self-Supervised Learning*

Unlike supervised learning methods that rely on ground truth depth maps for training, we use stereo disparity to deduce depth from the geometric connection between corresponding spots in left and right images [3][5]. We use backpropagation to compute errors and adjust weights by creating synthetic picture pairs from predicted depth maps and comparing them with real pairs. Because it trains the model using the intrinsic geometry of stereo pictures rather than costly ground truth data, this self-supervised approach is more scalable and economical.

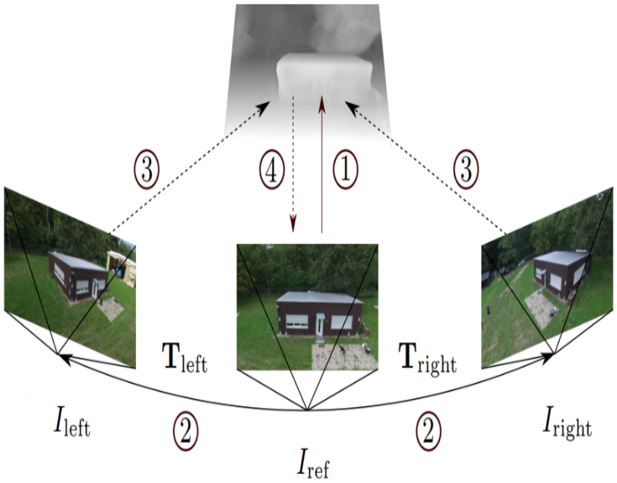

Fig. 2. images of same object taken from different angles and their depth map[3]

The figure above shows image of same object taken from various angle and the depth map generated from it using the geometric relation between the corresponding points on left and right images.

D. Variational Autoencoders

A Variational Autoencoder (VAE) is a type of neural network architecture used to learn a probabilistic latent representation of depth information from single images. Here's how it works [6]:

1) Encoder: The encoder network takes an input monocular image and maps it to a distribution in the latent space. This distribution typically consists of a mean and variance (or alternatively, parameters of a probability distribution) that define a probabilistic representation of the input image in the latent space [6].

2) Latent Space: The latent space represents a lower-dimensional embedding of the input images. It captures the essential features and variations present in the data. In the case of monocular depth estimation, the latent space encodes information about the scene geometry and depth structure [6].

3) Sampling: During training, the VAE samples from the learned distribution in the latent space to generate latent vectors that represent the input images. This stochastic sampling process encourages the model to learn a smooth and continuous representation of the data[6].

4) Decoder: The decoder network takes the sampled latent vectors and reconstructs the input images. It aims to reconstruct the original input with high fidelity while also generating plausible variations of the input data [6].

5) Loss Function: The training objective of the VAE consists of two components: a reconstruction loss, which measures the fidelity of the reconstructed images to the original inputs, and a regularization term, which encourages the learned latent space to follow a predefined prior distribution (typically a Gaussian distribution) [6].

6) Probabilistic Output: Unlike traditional autoencoders, which produce deterministic reconstructions of the input data, VAEs produce probabilistic reconstructions. This probabilistic output allows the model to capture uncertainty in the reconstruction process, which can be useful for tasks like depth estimation where uncertainty estimation is important [6].

E. Benefits of Using VAE for Monocular Depth Estimation

Using a variational autoencoder (VAE) for monocular depth estimation can indeed help improve generalization. Here's how:

1) Latent Space Representation: VAEs are capable of learning a latent space representation of the input data. In the context of monocular depth estimation, this latent space can capture meaningful features and variations in the input depth images. By learning a compact and structured latent space representation, the VAE can generalize better to unseen data.

2) Regularization: The variational aspect of VAEs, which involves learning a probabilistic distribution over the latent space, acts as a form of regularization. This can help prevent overfitting to the training data and encourage the model to learn more robust and generalizable representations [6].

3) Generative Modeling: VAEs are generative models that learn to generate realistic samples from the learned latent space distribution. This generative capability allows the model to hallucinate depth maps for novel input images, which can aid in generalization by capturing variations and patterns present in the training data.

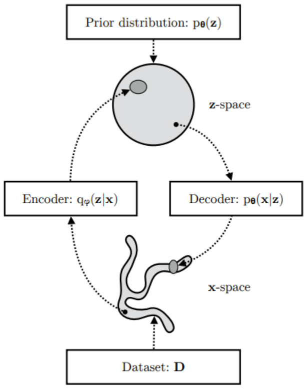
Prior distribution: $\mathbf{p_\theta(z)}$
z-space
Encoder: $q_\varphi(\mathbf{z}|\mathbf{x})$
Decoder: $\mathbf{p_\theta(x|z)}$
x-space
Dataset: **D**

Fig. 3. VAE

A Variational Autoencoder (VAE) learns stochastic mappings between an observed x-space, whose empirical distribution $q_D(x)$ is typically complicated, and a latent z-space, whose distribution can be relatively simple (such as spherical, as in this figure)[4]. The generative model learns a joint distribution $p_\theta(x, z)$ that is often (but not always) factorized as $p_\theta(x, z) = p_\theta(z)p_\theta(x|z)$, with a prior distribution over latent space $p_\theta(z)$, and a stochastic decoder $p_\theta(x|z)$ [4]. The stochastic encoder $q_\phi(z|x)$, also called the inference model, approximates the true but intractable posterior $p_\theta(z|x)$ of the generative model [4].

For variational parameters ϕ, we have:

$$\log p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x)] \tag{3}$$

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log\left[\frac{p_\theta(x,z)}{p_\theta(z|x)}\right]\right] \tag{4}$$

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log\left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\frac{q_\phi(z|x)}{p_\theta(z|x)}\right]\right] \tag{5}$$

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log\left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\right]\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log\left[\frac{q_\phi(z|x)}{p_\theta(z|x)}\right]\right] \tag{6}$$

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log\left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\right]\right] + \mathrm{KL}(q_\phi(z|x)\|p_\theta(z|x)) \tag{7}$$

The second term is the Kullback-Leibler (KL) divergence between $q_\phi(z|x)$ and $p_\theta(z|x)$, which is non-negative:

$$\mathrm{KL}(q_\phi(z|x)\|p_\theta(z|x)) \geq 0 \tag{8}$$

and zero if, and only if, $q_\phi(z|x)$ equals the true posterior distribution.

The first term is the variational lower bound, also called the evidence lower bound (ELBO):

$$L_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x,z) - \log q_\phi(z|x)] \tag{9}$$

Due to the non-negativity of the KL divergence, the ELBO is a lower bound on the log-likelihood of the data:

$$L_{\theta,\phi}(x) = \log p_\theta(x) - \mathrm{KL}(q_\phi(z|x)\|p_\theta(z|x)) \leq \log p_\theta(x) \tag{10}$$

So, the KL divergence $\mathrm{KL}(q_\phi(z|x)\|p_\theta(z|x))$ determines two 'distances':

1) By definition, the KL divergence of the approximate posterior from the true posterior.
2) The gap between the ELBO $L_{\theta,\phi}(x)$ and the marginal likelihood $\log p_\theta(x)$; this is also called the tightness of the bound. The better $q_\phi(z|x)$ approximates the true posterior distribution $p_\theta(z|x)$, in terms of the KL divergence, the smaller the gap.

4) Reparameterization Trick: For continuous latent variables and a differentiable encoder and generative model, the ELBO can be straightforwardly differentiated with respect to both ϕ and θ through a change of variables, also called the reparameterization trick.

5) Change of Variables: First, we express the random variable $z \sim q_\phi(z|x)$ as some differentiable (and invertible) transformation of another random variable ϵ, given z and ϕ:

$$z = g(\epsilon, \phi, x) \tag{11}$$

where the distribution of random variable ϵ is independent of x or ϕ.

After the reparameterization trick, the final loss function is given by:

$$\text{kl\_loss} = -0.5 \cdot \text{mean}\left(1 + \log(\sigma^2) - \mu^2 - \exp(\log(\sigma^2))\right) \tag{12}$$

II. Related Work

A. Self-supervised learning for monocular depth estimation from aerial imagery

The study introduces a method for estimating depth from aerial images taken by UAVs using self-supervised learning. The method trains a Convolutional Neural Network (CNN) to estimate depth and position from image sequences, without the need for annotated data. This approach provides a scalable solution to the difficulties of acquiring ground truth information. The network use sets of three images and a common encoder structure to forecast depth maps and relative rotations, while ensuring training accuracy by minimizing image reconstruction errors. Assessment across various datasets shows encouraging outcomes, reaching a maximum accuracy of 93.5% $\delta_{1.25}$ [3]. Although performance was excellent in rural areas, there were ongoing difficulties in urban surroundings due to problems with image quality and inaccurate ground truth. Comparative examination reveals the benefits of monocular depth estimation over traditional approaches, including its ability to handle camera movement and its superior performance in areas with obstructions. The study highlights the promise of self-supervised learning in producing precise depth estimates without relying on annotated data. It also acknowledges the need for more research to improve accuracy and generalization skills.

B. Self-Supervised Monocular Depth Estimation Using Hybrid Transformer Encoder

The field of depth estimation has made significant advancements, particularly with the introduction of the hybrid transformer-based self-supervised approach suggested in this paper. This method represents a significant development in the field by combining state-of-the-art techniques in image synthesis and cost-volume-based depth estimate. By incorporating self-supervised learning, the depth and pose networks are trained together to minimize photometric errors. This allows for accurate depth estimation even in situations when explicit depth data is not available. In addition, the hybrid encoder-decoder architecture, which combines convolutional and transformer-based methods, improves the representation of features. The attention decoder then further improves depth estimation by utilizing self-attention processes.

The suggested method has been evaluated on benchmark datasets such as KITTI and Cityscapes, and it has been found to outperform existing state-of-the-art models. The hybrid model consistently outperforms both single-frame and multi-frame-based estimate methods, as demonstrated by a detailed quantitative study using metrics such as AbsRel, SqRel, RMSE, and accuracy index. The method's capacity to attain precise accuracy even in the absence of semantic information highlights its resilience and adaptability in various situations. Overall, the hybrid transformer-based depth estimation method provides a reliable and effective solution for a wide range of practical applications, establishing a higher benchmark for accuracy and performance in the field of depth estimation.

III. Methodology

The input image is passed through encoder, residual block and then through decoder respectively.

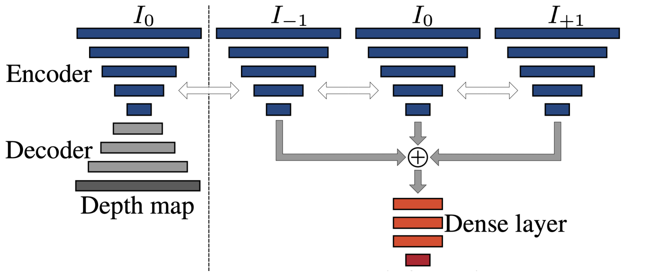

Fig. 4. Network architecture sharing the encoder (blue) with posture estimation (right) and depth estimation (left) networks. The feature maps are concatenated and run through the thick layers with a final global average pooling for pose estimation. Network architecture sharing the encoder (blue) with posture estimation (right) and depth estimation (left) networks. The feature maps are concatenated and run through the thick layers with a final global average pooling for pose estimation [3].

A. Encoder

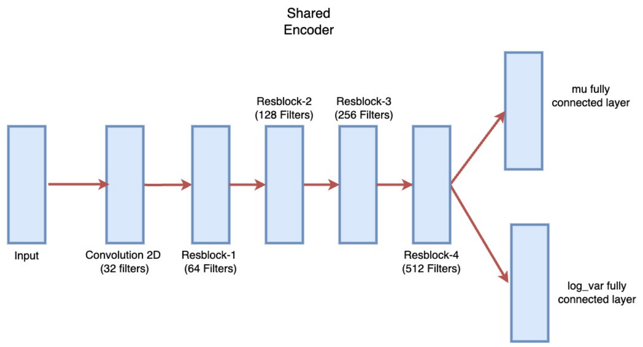

Fig. 5. a shared encoder architecture for representation learning that includes residual and convolutional blocks.

The input image, consisting of 3 channels (RGB) and spatial dimensions 224×384, is passed through 32 filters with a 7×7 filter dimension and a stride of $(2, 2)$. This produces an image with the same spatial dimensions but with 32 channels. The output of this filter is then batch normalized and passed to produce a residual block.

B. Residual Block

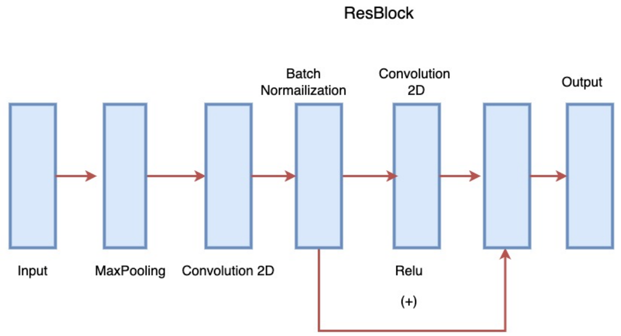

Fig. 6. ResBlock (Residual Block) architecture, consisting of convolutional layers, batch normalization, ReLU activations, and skip connections.

1) **Maxpooling:** The input is pooled using a maxpooling operation with a pool size of $(2, 2)$ and a stride of 2. The output channel size is specified.
2) **Convolution:** The pooled output is convoluted with filters to produce the specified output size.
3) **Batch Normalization:** Batch normalization is applied to the output of the convolution operation.
4) **Activation:** An activation function, such as ReLU, is applied to introduce non-linearity.
5) **Residual Connection:** Another convolution operation with the same output channels and a 3×3 kernel is performed. The output is added to the previous output, creating a residual connection.

First, a 64-channel residual block is produced, which acts as input for subsequent blocks producing 128, 256, and finally 512 channels. The final 512-channel output is passed through two fully connected layers of size 128 to produce μ and $\log(\text{var})$. These are returned along with some skip connections. After features are extracted by the encoder, they serve as input for the decoder.

C. Decoder

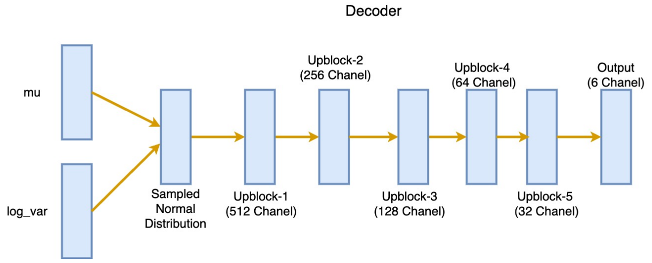

Fig. 7. Decoder architecture employing a series of Up-sampling blocks (Up-blocks) progressively increases channel complexity, transforming a sampled normal distribution into a 6-channel output. This structure is a staple in generative models and auto-encoders for tasks like image generation and data reconstruction

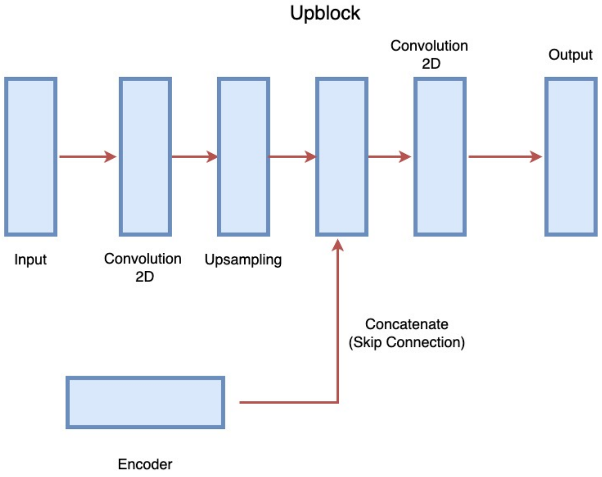

Fig. 8. The image shows the Up-block architecture used in convolutional neural networks, consisting of convolutions, Up-sampling, skip connections, and final output convolution.

1) **Latent Space Sampling:** The decoder first samples the latent space using the reparameterization trick.
2) **Fully Connected Layer:** The sampled latent space is fed to a fully connected layer to match the input size for the decoder to upscale.
3) **Upsampling:**
   a) **Convolution and Reflection Padding:** In upsampling, reflection padding is performed, and the output is then convoluted with a 3×3 kernel and stride $(1, 1)$.
   b) **Upsampling and Concatenation:** The output is then upsampled by doubling the size with stride $(1, 1)$, and it is concatenated with skip connections.
   c) **Activation:** Finally, the result is produced with a sigmoid activation function.
4) **Output Upsampling:** The output is then 4 times upsampled to produce R, G, B, D values, which indicate the depth of a particular point (it contains only one channel and input and output dimension size of the image).

D. Pose Estimation

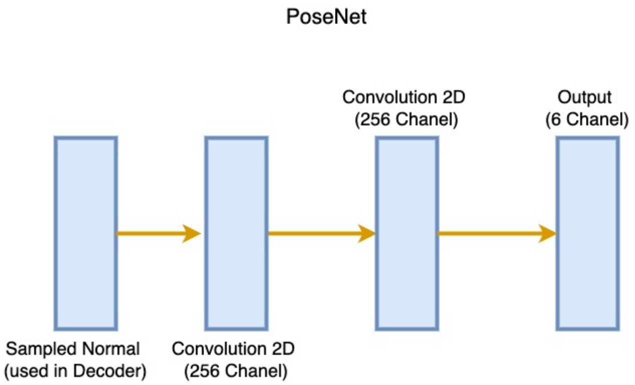
PoseNet
Convolution 2D (256 Chanel)
Output (6 Chanel)
Sampled Normal (used in Decoder)
Convolution 2D (256 Chanel)

Fig. 9. A 6-channel output for pose estimation is obtained by passing an input sampled normal (same as the one used in decoder) through 256-channel 2D convolutional layers.

1) **Latent Space Usage:** The latent space, which was initially sampled during the encoding process, is utilized here. This latent space contains meaningful representations of the input image features extracted by the encoder.
2) **Dense Layer:** The latent space representation is passed through a dense layer to match the size of the image. This dense layer serves to map the latent space representation to a higher-dimensional space that aligns with the spatial dimensions of the input image.
3) **Depth Feature Maps:** Upon passing through the dense layer, the output is reshaped and processed to produce two depth feature maps, one for the left pose estimation and one for the right pose estimation. These depth feature maps encode information about the depth or distance of objects in the scene.
4) **Convolutional Layer:** The depth feature maps are then passed through a convolutional layer with a kernel size of $(1, 1)$ and a stride of $(1, 1)$. This convolutional layer aims to extract spatial information from the depth feature maps and produce six channels representing translation and reflection information. The resulting output provides insights into the spatial positioning and orientation of objects within the image, facilitating pose estimation tasks.

By using depth information and translation and rotation, left and right images are produced, and loss is calculated accordingly for backpropagation.

E. Loss Calculation

1) Reconstruction Loss: : In the realm of image processing, achieving fidelity between reconstructed and reference images hinges on a fundamental metric known as photometric loss. This metric is crafted from two crucial components: Structural Similarity (SSIM) and L1 loss. SSIM evaluates similarity based on luminance, contrast, and structure, while L1 loss measures the pixel-wise differences between images. The combined photometric loss, denoted as:

$$pe(I_a, I_b) = \alpha(1 - \text{SSIM}(I_a, I_b)) + (1 - \alpha) \cdot |I_a - I_b|$$

[3]

orchestrates a balanced interplay between these elements, with α dictating their weighting. SSIM itself is mathematically articulated as:

$$\frac{(2\mu_a\mu_b + k_1)(2\sigma_{ab} + k_2)}{(\mu_a^2 + \mu_b^2 + k_1)(\sigma_a^2 + \sigma_b^2 + k_2)}$$

[3]

It involves statistical evaluations of mean intensity, variance, and covariance of image patches, with its output confined within the bounds of -1 to 1, where 1 signifies an epitome of likeness.

These metrics play a pivotal role in assessing perceptual image quality and are widely employed in various image restoration and enhancement algorithms. By carefully balancing the contributions of SSIM and L1 loss, the combined photometric loss captures both structural nuances and pixel-level disparities, guiding the optimization process towards images that are not only visually pleasing but also faithful representations of their reference counterparts.

Variational Auto encoder is also used which itself generates a loss called as KL divergence. This Loss represents how latent space correctly encodes the distribution of data

2) KL Divergence Loss: : This term represents the Kullback-Leibler (KL) divergence between the approximate posterior distribution of the latent variables (learned by the encoder) and a prior distribution (typically a standard normal distribution). The KL divergence acts as a regularizer, encouraging the encoder to learn a latent space that follows the desired prior distribution.

The Loss function derived for KL-divergence loss is

$$\text{kl\_loss} = -0.5 \cdot \text{mean}\left(1 + \log(\sigma^2) - \mu^2 - \exp(\log(\sigma^2))\right) \tag{13}$$

Total loss is sum of reconstruction loss and the KL-divergence loss. After the total loss is calculated the weights are updated using the back propagation.

IV. Experiments and Results

A. Loss Comparison

In the reference model, the proposed architecture yields an IT loss max of 1.99 and an average IT loss of 1.886, while the smooth loss max and average are 0.0000027231 and 0.000000186 respectively.

B. Our Model's Loss

In contrast, our model (VAE) exhibits slightly higher losses with an IT loss max of 1.998 and an average IT loss of 1.8742. The smooth loss max and average are 0.0000057899 and 0.000000192 respectively.

C. Model Accuracy

Calculating the accuracy of our model(VAE) proves challenging due to the absence of ground truth. To overcome this, we compare our depth maps with a well-defined model's depth maps, yielding an average accuracy of 87.415% and a maximum accuracy of 91.327%.

D. Comparison of Architectures

A comparative analysis of various architectures reveals differences in both loss and accuracy. For instance, MobileNet achieves a remarkably low loss of 0.024 and a high accuracy of 97.6% under supervised conditions.

E. Comparison table of various models

Architecture	**Loss**	**Accuracy**	**Model type**
VGG16	0.073	92.7%	Self-Supervised
ResNet18	0.074	92.6%	Self-Supervised
DenseNet	0.088	91.2%	Self-Supervised
MobileNet	0.024	97.6%	Supervised
Our Model	0.126	87.4%	Self-Supervised

F. Comparison table of our models

Autoencoder	**Max IT Loss**	**Avg IT Loss**	**Max Smooth Loss**	**Avg Smooth Loss**
VAE	1.99	1.8742	5.79×10^{-6}	1.92×10^{-7}
Vanilla Autoencoder	1.998	1.8860	2.72×10^{-6}	1.86×10^{-7}
VAE(1.7xC, 1.7xI)	1.629	1.426	1.80×10^{-4}	1.72×10^{-4}
Vanilla(1.7xC, 1.7xI)	1.645	1.513	1.93×10^{-4}	1.87×10^{-4}

where C is contrast and I is intensity

G. Comparison of Our Models

Within our models, variations in architecture and input data manipulation lead to diverse performance outcomes. For instance, increasing contrast and intensity by a factor of 1.7 in both VAE and Vanilla Autoencoder models result in fluctuations in loss and accuracy.

H. Generalizability Testing

Evaluating the generalizability of our model involves testing it on a modified dataset generated by adjusting contrast and intensity. This process simulates real-world variability and assesses model robustness.

I. Performance on Modified Dataset

Testing our model on the modified dataset yields insights into its adaptability. Our VAE model achieves a peak testing accuracy of 61.303% and an average testing accuracy of 60.654%, demonstrating its ability to generalize to new data distributions.

V. Conclusion and Future Work

Self-supervised monocular depth estimation has emerged as a promising approach for deriving depth information from single images without requiring labelled data. Our model based on variational auto-encoder offer a principled and flexible framework for monocular depth estimation, leveraging probabilistic modeling and latent space representations. With research in this field and addressing the current challenges continues, it will be likely lead to further improvements in the accuracy and generalization of depth estimation.

References

[1] Hwang, Seung-Jun, Sung-Jun Park, Joong-Hwan Baek, and Byungkyu Kim. ”Self-supervised monocular depth estimation using hybrid transformer encoder.” IEEE Sensors Journal 22, no. 19 (2022): 18762-18770.

[2] Liu, Lina, Xibin Song, Mengmeng Wang, Yong Liu, and Liangjun Zhang. ”Self-supervised monocular depth estimation for all day images using domain separation.” In Proceedings of the IEEE/CVF international conference on computer vision, pp. 12737-12746. 2021.

[3] Hermann, Max, Boitumelo Ruf, Martin Weinmann, and Stefan Hinz. ”Self-supervised learning for monocular depth estimation from aerial imagery.” arXiv preprint arXiv:2008.07246 (2020).

[4] Kingma, Diederik P., and Max Welling. ”An introduction to variational autoencoders.” Foundations and Trends® in Machine Learning 12, no. 4 (2019): 307-392.

[5] Masoumian, Armin, Hatem A. Rashwan, Julián Cristiano, M. Salman Asif, and Domenec Puig. ”Monocular depth estimation using deep learning: A review.” Sensors 22, no. 14 (2022): 5353.

[6] https://ar5iv.labs.arxiv.org/html/1906.02691