IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 23, 2020, accepted January 7, 2021, date of publication January 18, 2021, date of current version January 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052474

l, r-Stitch Unit: Encoder-Decoder-CNN Based Image-Mosaicing Mechanism for Stitching Non-Homogeneous Image Sequences

PREMITH KUMAR CHILUKURI[1], PREETHI PADALA[2], PUSHKAL PADALA[3], VENKATA SUBBAIAH DESANAMUKULA[1], AND PRASAD REDDY PVGD[1]

[1]Department of CS and SE, Andhra University College of Engineering (A), Visakhapatnam 530003, India
[2]Department of Computer Science and Engineering (CSE), National Institute of Technology Surathkal, Mangalore 575025, India
[3]Department of Computer Science and Engineering (CSE), The National Institute of Engineering, Mysore 570008, India

Corresponding author: Venkata Subbaiah Desanamukula (drdvs.2021@gmail.com)

ABSTRACT Image-stitching (or) mosaicing is considered an active research-topic with numerous use-cases in computer-vision, AR/VR, computer-graphics domains, but maintaining homogeneity among the input image sequences during the stitching/mosaicing process is considered as a primary-limitation & major-disadvantage. To tackle these limitations, this article has introduced a robust and reliable image stitching methodology (l,r-Stitch Unit), which considers multiple non-homogeneous image sequences as input to generate a reliable panoramically stitched wide view as the final output. The l,r-Stitch Unit further consists of a pre-processing, post-processing sub-modules & a l,r-PanoED-network, where each sub-module is a robust ensemble of several deep-learning, computer-vision & image-handling techniques. This article has also introduced a novel convolutional-encoder-decoder deep-neural-network (l,r-PanoED-network) with a unique split-encoding-network methodology, to stitch non-coherent input left, right stereo image pairs. The encoder-network of the proposed l,r-PanoED extracts semantically rich deep-feature-maps from the input to stitch/map them into a wide-panoramic domain, the feature-extraction & feature-mapping operations are performed simultaneously in the l,r-PanoED's encoder-network based on the split-encoding-network methodology. The decoder-network of l,r-PanoED adaptively reconstructs the output panoramic-view from the encoder networks' bottle-neck feature-maps. The proposed l,r-Stitch Unit has been rigorously benchmarked with alternative image-stitching methodologies on our custom-built traffic dataset and several other public-datasets. Multiple evaluation metrics (SSIM, PSNR, MSE, $L_{\alpha,\beta,\gamma}$, FM-rate, Average-latency-time) & wild-Conditions (rotational/color/intensity variances, noise, etc) were considered during the benchmarking analysis, and based on the results, our proposed method has outperformed among other image-stitching methodologies and has proved to be effective even in wild non-homogeneous inputs.

INDEX TERMS Deep feature extraction, encoder-decoder cnn, image mosaicing, multi-image registration, non-homogeneous image stitching.

I. INTRODUCTION

Panorama stitching is a process of combining two or more images together to generate high-resolution panoramic images with an extended field of view. The most important application of an image-stitching operation is its ability to summarize and compress the videos taken from a camera into a meaningful image with widened field-of-view. Normally, there is a constraint on the resolution of a digital camera; because of physical and economic reasons, and hence the images generated by the digital camera do not have required field of view even with the use of super-wide angle lenses, which are themselves prone to high lens distortions, hence panoramic stitching operations are necessary to generate high quality panoramic(views) images economically(with no additional equipment) [1], [2]. These image-stitching operations [1], [3]–[21] can also be termed as Image Mosaicing operations. Mosaicing is an image-processing operation that works by accumulating multiple input-images from normal cameras and then stitching them together to form

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaokang Wang.

a panoramic image with a widened field of view (FOV) [21] (refer to Fig. 10 for examples). Recently specialized forms of image mosaicing methods based on deep feature extraction techniques & Deep-Neural-Networks have been introduced [19], [20]. These advanced forms of image-mosaicing methods have played many pivotal roles in the development of multi-spectral satellite image registrations, satellite image stitching, environment modeling/localization [22] techniques.

Image stitching(or)mosaicing operations have various applications in the field of video compression, video stabilization, video matting, video conferencing, video summarization, 3D image reconstruction and applications in the medical field. Moreover, some photographic cameras can stitch image sequences internally with the help of a built-in image stitching function in the computer-aided hardware design of the system's graphic-architecture [7], [23]. The Image stitching methodology was further extended to applications like video indexing, video compression, and genuine generalization of multi-image video stitching [24]. Panoramic videos can be used to create animated video textures, in which different elements of panoramic locations are animated with individually moving video loops, illuminated video-flashlights to create a compound mosaic of a particular scene. Image stitching in the medical domain has many prominent applications to aid the diagonal-process of renal, liver, tissue, abdomen, retinal, cardiac, and other disorders [25]. Localization systems are one of the most important applications of any panoramic image stitching module [3]. Most familiar applications, approaches of these image stitching operations [1], [3], [5]–[19], [21] require indistinguishable exposure differences and specific overlaps in between the input image sequences to produce a seamless output.

The general objective of any image stitching method [23], [24], [26], [27] is to create an intrinsic image-mosaics, which are free of texture artifacts; generally image-stitching artifacts are caused due to inefficiency of the stitching module, optical aberrations, relative camera motion, hardware/external image-noise, ill-placed camera sources, illumination changes, etc. Although image stitching/mosaicing operations have many real-time applications, their implementation during live-scenarios is still considered as a challenging issue in many image-processing, camera-hardware-system use-cases because of the previously stated challenges, limitations & restrictions (refer to Fig. 9 for some sample challenges faced during a panoramic view acquisition process). The observed challenges are mainly caused due to limitations & assumptions made by a particular method during their inference/experimentation phases.

Based on a detailed survey of implementation schematics & working methodologies of many existing image-stitching, panoramic-view-modeling methodologies [10], [24]–[26] we have identified several disadvantages, feature incapacities & some common assumptions (made by a respective author while proposing their methodology).

Following are some of the challenges faced and assumptions made, by the existing methodologies [4], [10]:

- Enough overlapping areas should be maintained between the input image sequences.
- No objects should have large movements in the scene.
- No high distortion in input images.
- No severe exposure, intensity, color differences between the input images.
- The resolution of stereo input image pairs should be restricted to a certain limit(generally less than 2K) so that the image-stitching operation can be efficiently performed.
- The input stereo pairs should be homogeneously synchronized(both spatially and temporally).
- The orientations & resolutions of every image in the input image sequences/register should be synchronized & identical, i.e it's important to maintain geometrical integrity among the input images.
- The input images can only have minimal noise, rotational variances, chromatic aberrations, and artifacts, etc.
- The input images should have semantically rich, dense features/patterns.

We have brainstormed for various implementation features, and based on the analysis we propose a novel Deep Neural Network architecture (Fig. 1) along with several other Deep feature extraction+mapping algorithms (Figs. 1, 2 and 3) to tackle the previously discussed disadvantages and challenges faced by other image-stitching methodologies [1], [3], [5]–[19], [21]. Based on the proposed features, architectures & methods/algorithms, the "l,r-Stitch unit" research work(refer to section 4 for a detailed description) is framed and presented (Fig. 1). The contributions,novelties of the proposed research-methodology are:

- We have proposed a novel l,r-Stitch Unit module for robust & reliable image stitching use-cases.
- We have proposed the l,r-PanoED network, an Encoder-Decoder CNN for deep feature extraction + mapping, intuitive panoramic view re-construction purposes. Split encoding network methodology (for l,r-PanoED network) is introduced in this research work.
- The proposed image stitching methodology l,r-Stitch unit is highly modular and can be adaptively plugged in with other l,r-Stitch units (Fig. 5) to stitch ultra-wide panoramic views(FOV>220^0).
- A deep feature-vector mapping algorithm (F-Mat* (UL-$8_{FT}^{p,q}$, UL-8_{FT})) has been proposed in this research work.
- The proposed method can effectively stitch panoramic views with input field-of-view ranging from $30^0 < \text{FOV} < 320^0$ (Table 6).
- We have proved & illustrated the proposed method's effectiveness and superiority over other methods by performing an extensive benchmarking + performance analysis (Figs. 9 and 10, Tables 1, 2, 3, 4 and 5).

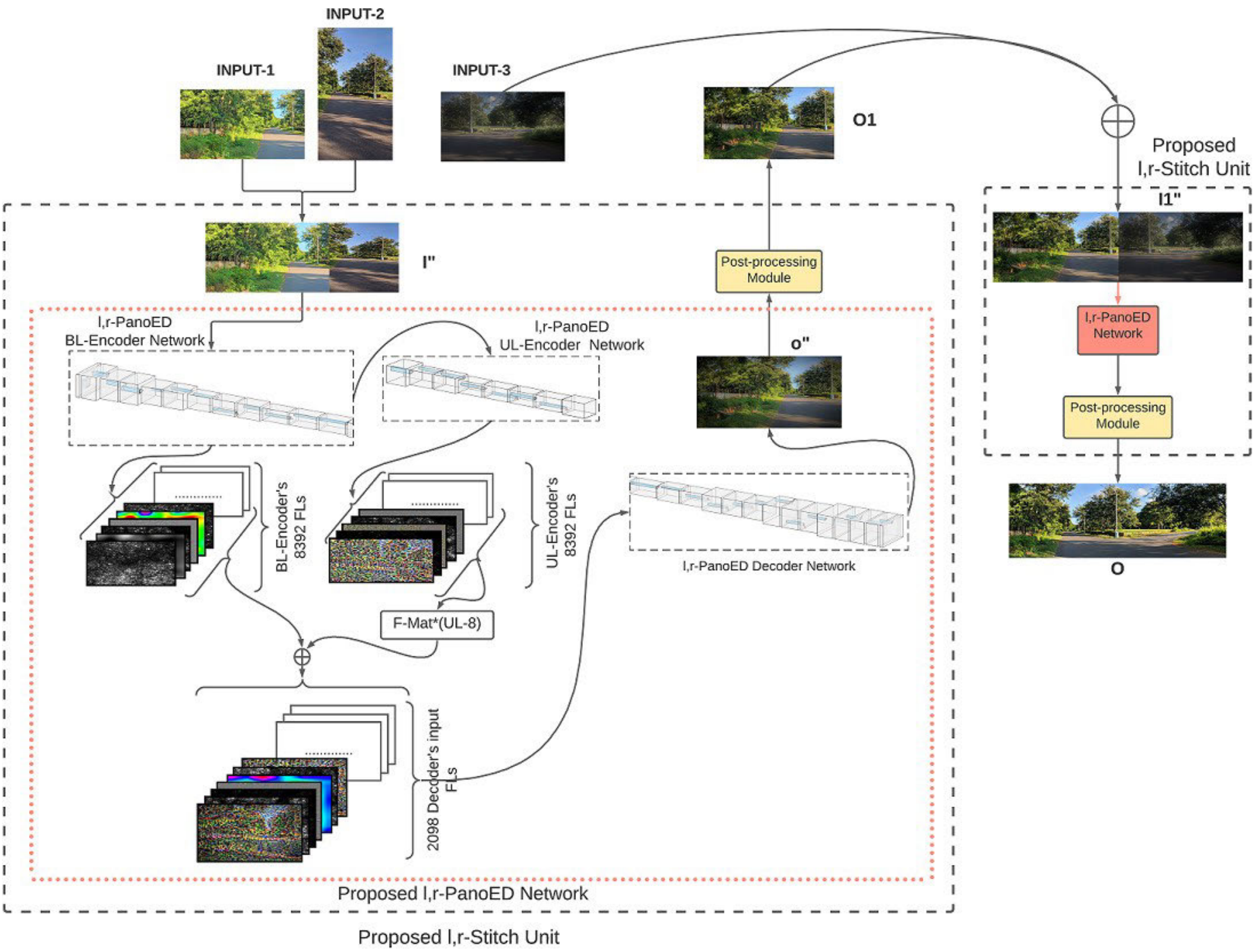

FIGURE 1. **An illustrative overview of the proposed (l,r-Stitch Unit) image-stitching methodology's working mechanism, along with the depiction of schematic work-flow involved in the proposed novel convolutional deep-feature extractor + mapping encoder-decoder network(l,r-PanoED network).**

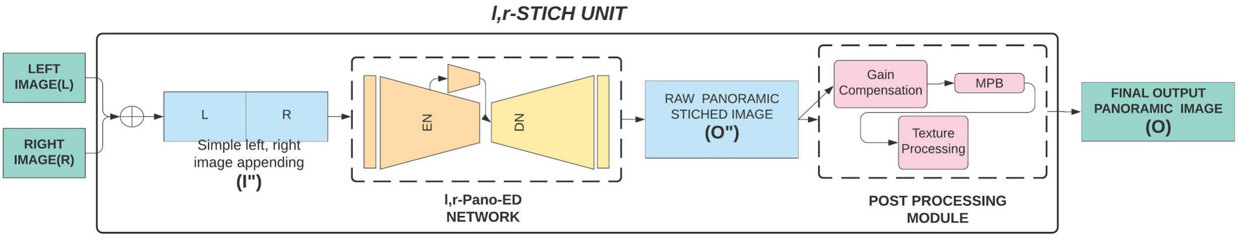

FIGURE 2. **Overall System architecture of the proposed Image stitching mechanism(l,r-Stitch Unit). The l,r-Stitch Unit is a sequential ensemble of the pre-processing sub-module, l,r-PanoED network, and post-processing sub-module. The sub-modules, DNNs present in the l,r-Stitch unit are a robust ensemble of SOTA image-processing/handling, Deep-Learning techniques. Left, right stereo images (L, R) are passed as input to the l,r-Stitch unit to generate a panoramic view(O) with the help of l,r-PanoED network(introduced in this article).**

This analysis consists of subjecting the proposed method & other alternative image-stitching methods to certain extreme wild + non-homogeneous test-case scenarios.[1] Sample results of the proposed method during the performance analysis were illustrated later in section 5.

- We have introduced a new custom-built live traffic dataset [28], which will be open-sourced in the future. Custom loss functions & system pipelines (Figs. 1, 2 and 3) were also introduced in this research work.

Rest of the paper is structured as follows, Section 2 details the abbreviations & acronyms used in this research work, and Section 3 thoroughly surveys, analyzes, and evaluates other existing alternative image-stitching methodologies/research

[1] Refer to Section 5's Table 1, Table 2, Table 3, Table 4, Table 5 for a detailed explanation regarding the wild+non-homogeneous test-case scenarios.

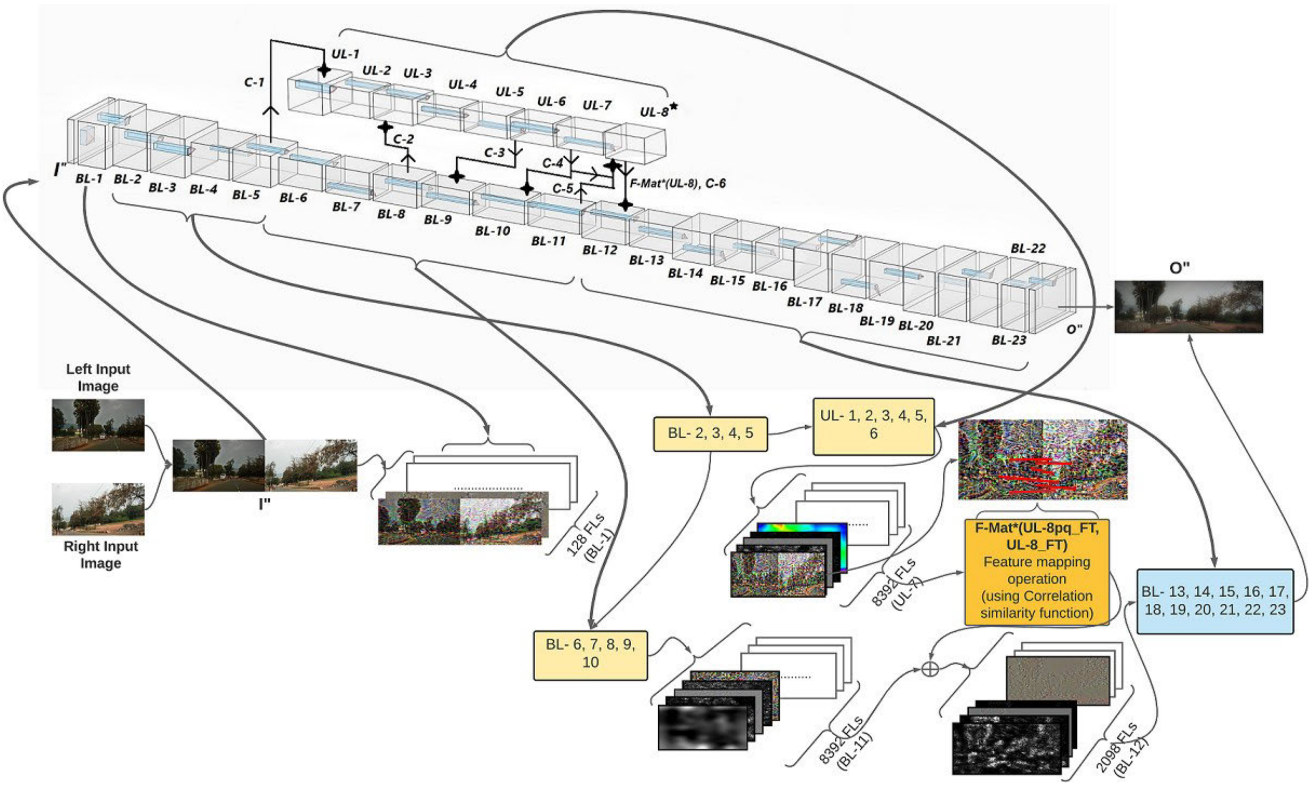

FIGURE 3. Overview of the proposed l,r-PanoED network architecture, with detailed illustrations & description of the feature maps extracted at crucial Encoder-Decoder Network-layers. This figure schematically describes and illustrates the working of the proposed "split encoding network methodology". BL-1 to BL-11 layers belong to the BL-Encoder network of the l,r-PanoED, UL-1 to UL-8 layers belong to the UL-Encoder network of the l,r-PanoED, and BL-12 to BL-23 layers belong to the decoder network of the l,r-PanoED.

works. Section 4 consists of an extensive description and analysis of our proposed image stitching methodology along with other supporting hypotheses. Section 5 illustrates and analyzes our proposed method's results & performance under certain extremely non-homogeneous[2] wild input conditions; Section 5 also consists of training and inference details of the proposed l,r-Stitch unit along with a thorough benchmarking analysis of our proposed method along with other alternative image-stitching methodologies(refer to section 5 & [29], [30]). Section 6 includes the limitations & future-work of our proposed method, and section 7 effectively concludes our proposed research work.

II. ACRONYMS AND ABBREVIATIONS

2D-Conv: "2-Dimensional Convolution", 2D-De-Conv: "2-Dimensional Deconvolution", AED: "Auto Encoder Decoder Network", BL: "Bottom Layered", CNN: "Convolutional Neural Network", DNN: "Deep Neural Network", ED: "Encoder Decoder Neural Network", FL: "Feature Layer", FM: "Feature Map", FOV: "Field of View", FV: "Feature Vector", GT: "Ground Truth", GAN: "Generative Adversarial Network", l: "Left", r: "Right", MPB: "Modified Poisson Blending", MSE: "Mean Square Error", NNDR: "Nearest Neighbour Distance Ratio", Pano: "Panorama", PReLU: "Parametric Rectified linear unit", PSNR: "Peak Signal to Noise Ratio", RANSAC: "Random Sample Consensus", RGB: "Red, Green, Blue", SSIM: "Structural Similarity Index Measure", UL: "Upper Layered".

III. RELATED WORK

In recent times [25]–[27] many static + dynamic conventional feature descriptor methods & CNN/DNN based image stitching methods have been proposed [1], [3]–[21]. Majority of the previously proposed(or existing) image-stitching mechanisms are primarily optimized to perform better in input scenarios, where the input stereo images are homogeneously paired in ideal conditions with slight exceptions in irregularities between them. Some image stitching methodologies [3], [5], [6], [8], [9], [11]–[16] have proposed custom feature extraction & pre + post-processing mechanisms, Some methodologies [17]–[20] have used DNNs to extract deep features from input stereo images to register them together for generating output panoramic-views, while some methodologies [1], [4], [7], [10] have used existing conventional feature extraction techniques and modified the

[2] The term "non-homogeneous" refers to dissimilar(any non-coherent) properties identified between the input left, right stereo images, here dissimilarities between the input images implies to any differences in exposure & contrast, color content, lumination, resolution & orientation, rotations, object-movements & pattern/texture artifacts, geometric misalignments, etc.

matching/mapping methodologies to spike their proposed method's performance in stitching reliable panoramic outputs. For instance, Sampetoding *et al.* [18], Proposed a novel framework for automatic personal photo improvement using photo collections without any 3D regeneration process. Their proposed method [18] consists of two steps: image retrieval and image stitching. They have generated a specified landmark dictionary for every image in the dictionary with the help of NetVLAD descriptor architecture and thresholding operations(to remove any similar image outliers present in the dictionary). After the creation of global feature vectors, their proposed method [18] searches for K-nearest neighbors for an input image based on the similar nature of the global feature vector. Then the K images are subsequently utilized as candidate-images and used for field-of-view expansion. This proposed research [18] work has a narrowed use-case as the input image pair retrieval is constrained to the diversity of their proposed database, internet images available(i.e this image-stitching might fail in any custom use-cases if the input image does not share any corresponding information in their database/internet-gallery); Moreover, the image-stitching operation implemented in their proposed [18] work produced less accurate results compared to other alternative conventional stitching methods, and also their proposed method fails to work in regular wild conditions(with orientational, noise, rotational differences). Kang *et al.* [17], proposed an innovative two-step image alignment technique based on deep-learning and iterative optimizations. A light-weighted end-to-end trainable convolutional neural network (CNN) architecture called ShiftNet was nominated to estimate the introductory shifts amid images, which was more optimized in the subpixel refinement process based on a described camera motion model. Both qualitative & quantitative research results indicate that the cylindrical panorama stitching found on their proposed image alignment method [17] showed significant enhancements over traditional feature-based approaches. The limitations of their proposed work [17] are, their image-stitching module can perform better only on multiple sequential input image-pairs with higher matching percentage (or intersection) in-between the input image-pair sequences. The stitching module implemented in their paper [17] does not perform adequate fine-tuning pre/post-processing operations on inputs & outputs. Tackling Non-Homogeneity among input image-pairs wasn't proposed in [17] research work.

Levin *et al.* [5], Introduced several traditional cost-effective functions for the assessment of quality for image-stitching operations. From these cost functions, the correlation between the seam-appearance and input images were determined in the gradient domain/territory. They [5] have described two approaches for image stitching in the gradient domain. The first explains GIST1, where the mosaic part was inferred directly from the derivatives of the input images. The second describes GIST2, with a two-steps processing pipeline for stitching input images. Their [5] proposed method majorly concentrates on implementing and modulating post-processing techniques to deal with photometric, intensity, coloring inconsistencies along with geometric misalignments to an extent. So, their paper [5] majorly misses the part of effectively proposing a robust image-stitching, feature matching method for projecting the input image pairs to a wide-view panoramic image(although the non-homogeneous inconsistencies among the input stereo pairs were also not handled in [5] paper). Gao *et al.* [6], Proposed Seam driven image stitching methodology where they have Assessed the finest transformation based on the resultant vision quality of the seem-cut, instead of estimating a geometric transform which depends upon the best fit of feature correspondence. Seam-cut was used in masking misalignment artifacts. Their [6] paper has used a conventional non-robust feature detection mechanism which would easily fail in extracting and matching keypoints/feature pairs even in slight wild conditions; moreover, their [6] proposed seam cut requires more latency time compared to other conventional projective stitching mechanisms. Faridul *et al.* [4], Proposed an explicit color correction operation by leveraging the sparse correspondences on input images before performing an image stitching operation. Their [4] approach has two fields. First step consists of all the necessary procedures required for finding an optimal geometric correspondence among input images, then the color information is collected & stored locally to achieve robust performance in geometric correspondences. In the Second step, the proposed method fits a global model that compensates for complex color changes among the collected step-1 colors accordingly. The experimental results stated in their [4] paper showed that their proposed image stitching method was invariant to changes in Exposure status, illumination conditions, and changes in imaging devices, etc. In their [4] proposed paper, the non-homogeneity among input image-pair sequences were handled well, but their proposed image-stitching & feature matching methods were conventional and non-robust. Moreover, noise & rotational variations were not properly handled. Xiong and Pulli [7], Proposed a fast image-stitching methodology with smaller memory footprint consumption for combining sets of source images into panoramic scenic images. In their [7] proposed method, the color correction operation minimizes color differences among source images, while simultaneously maintaining color & luminosity balances throughout the image sequence. Their [7] proposed Dynamic programming discovered optimal seams in imbricate areas between adjacent images to merge them, and finally, an image blending operation was applied to further smooth the color transitions and hide visible seams, stitching artifacts. Their [7] proposed image-stitching methodology was primarily designed for mobile/lite-device use-cases [7], therefore the input image-pairs should always be in ideal conditions for the proposed method to work efficiently, i.e slight non-homogeneity/noise/wild-conditions can highly affect the performance of the proposed stitching methodology. Wang *et al.* [8], Proposed a novel fast image stitching algorithm based on ORB (Oriented FAST and Rotated BRIEF) features. Their [8] proposed algorithm initially selects an ORB algorithm which adds the direction information to the FAST detector for image feature extraction and matching. Later, the RASANC (Random Sample Consensus) algorithm was used in their [8] methodology to eliminate false matching points. Finally, a weighted average method was used to speed up the whole image fusion process. Their method [8] was able to overcome the limitations of speed and accuracy compared to other traditional methods. Wang *et al.* [9], Analyzed the looping path problem which caused error accumulations in traditional image-stitching operations, [9] introduced a multi-image stitching method based on graph models. Their proposed method [9] has used the Weighted Shortest Path Algorithm, in which the input images are stitched automatically. Matching Mean Square Error was introduced as the weight of edges on the graph, which was intuitive and easy to compute. Furthermore, an optimized Dijkstra algorithm was applied to speed up the pathfinding algorithm. Experiments have shown that their [9] proposed algorithm caused less Matching Mean Square Error and has obtained more stable results than other similar methods. [8], [9] research works have proposed robust and efficient feature extraction & mapping mechanisms, but the overall latency time required by these two proposed methods is slightly higher when compared to other alternative conventional stitching methods; moreover, these proposed methods [8], [9] were not able to efficiently handle extreme wild conditions with non-homogeneous integrity among input image sequences. Alomran and Chai [10], proposed a feature-based image stitching Algorithm that consists of image acquisition, image registration, image blending, and compositing operations. Their paper [10] has proposed two main approaches for image-stitching operation, the first approach was based on a direct technique, and the second approach was a feature-based technique. A through experimentation analysis(focused only on lens, focal-length & resolution changes) was carried out on their proposed [10] image-stitching method. Their [10] algorithm can successfully stitch input image sequences only if they are ideally conditioned, i.e the proposed method is restricted to perform optimal only on input image sets that have no exposure differences, enough overlapping areas, minimal lens distortions, no object movements, lens/device invariance restrictions & angular-orientations.

Based on the above extensive survey and further analysis of other alternative image-stitching mechanisms [1], [3]–[21], we can infer that the existing image-stitching methods are only ideal for homogeneously paired input stereo sequences(with little or no wild conditions included); i.e these existing methods tend to generate non-reliable outputs(mostly failure cases) when their respective inputs are subjected to irregularity, wild-conditions or non-homogeneity. So, to tackle the feature incapacities & disadvantages faced by the existing image stitching methodologies we have proposed a novel stereo image-stitching methodology named l,r-Stitch unit (Figs. 1, 2 and 3), which can robustly generate reliable wide + ultra-wide panoramic view images ($30^0 < \text{FOV} < 330^0$) even from non-homogeneously synchronized input image sequences. The proposed l,r-Stitch unit consists of a novel Encoder-Decoder CNN named l,r-PanoED network, which implements a unique split encoding network methodology to extract, detect and map the feature patterns present in input image pairs. The l,r-Stitch unit also consists of several highly efficient & robust post-processing techniques to perfectly fine-tune the output panoramic images generated by the l,r-PanoED network. Our proposed method can also generate ultra-wide panoramic outputs by intuitively ensembling multiple modular l,r-Stitch units. The features and modules(which will be discussed exhaustively in the following section 4) are built based on the proposed research's contributions and novel architectures/methods(mentioned in section 1).

IV. IMPLEMENTATION AND METHODOLOGY

The proposed l,r stitch Unit consists of 3 major subcomponents, 1) Pre-Processing module, 2) l,r-PanoED net- work and 3) Post-processing module (Fig. 2). A single l,r-Stitch unit takes left, right stereo images(either with homogeneous or nonhomogeneous synchronization) as input and generates a robustly mapped panoramic image with low processing latency. The proposed l,r stitch Units are designed as plug-in modules(designed with high modularity), where a single unit can be heuristically plugged-in to other l,r-Stitch units, to form a recursive tree consisting of multiple l,r-Stitch units, now the tree of multiple l,r-Stitch units inputs a series of multiple left, right stereo images(>2) to generate a much wider $\sim 320^0$ panoramic view (Figs. 5 and 10). The steps involved in the proposed pipeline starts with a simple lateral stereo image concatenation operation [24], [31] (pre-processing) to generate I'', the concatenated image(I'') is then sent to l,r-PanoED network to generate a raw panoramically stitched wide view image(O''), and the O'' panoramic image is sent to the post-processing module to minimize illumination, exposure differences [24], [27] across the image, and multi-scale image blending, texture correction operations are further performed on O'' to remove all the occluded edges, blurred, ghosting artifacts [24], [31]. Image generated from the post-processing module is the final output panoramic image(O) of a specific l,r-Stitch unit for respective input l,r stereo images(refer to Figs. 1, 2 and 3 for a working illustration of the proposed method).

A. PRE-PROCESSING (STEREO LATERAL CONCATENATION)

The proposed pipeline pre-process the raw input using a simple stereo image lateral concatenation operation [24], where the individual left, right stereo images (Fig. 4 (a),(b)) are

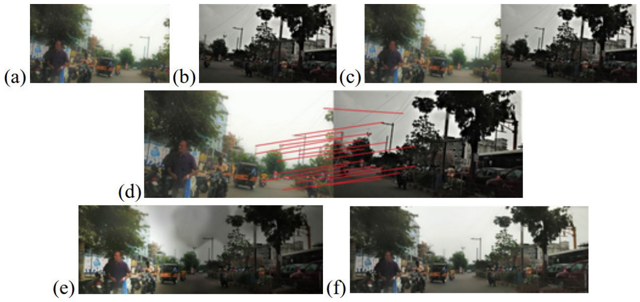

FIGURE 4. (a), (b) are the input non-homogenous stereo images. (c) is the pre-processed input image(I''), i.e output of l,r-Stitch Unit's pre-processing sub-module. (d) is a working illustration of deep-feature stitching/mapping algorithm(F-mat*(UL-8)); red lines indicate the identical feature maps among the input (a), (b) images. (e) is the un-processed raw output(O'') of the l,r-PanoED network, (f) is the final panoramic image(O) generated by the l,r-Stitch Unit by post-processing the output of l,r-PanoED's O''(e).

laterally concatenated into one(I'') (Fig. 4 (c)) by per-forming a lateral matrix concatenation operation (|).

$$I'' = [L_{N1XM1}|R_{N2XM2}]_{(N1+N2)Xm} \quad (1)$$

In, | is matrix lateral concatenation operation [24]; N*, M* are image dimensions. Here L, R are the input stereo images with N1xM1 and N2xM2 dimensions, and after performing the lateral concatenation operation(|) the resulting pre-processed image(I'') has a dimension of (N1+N2) x m(generally m is the minimum of {M1, M2}). The steps involved in the lateral concatenation operation are: 1) making a copy of (N1+N2)*min(M1, M2)*3 dimensional Zero-matrix. 2) Resizing the M1, M2 height of either left or right stereo input images to a value of min(M1, M2), so that the height values of intermediate, output panoramic-views remain homogeneous throughout the stitching process, the pre-processing module performs bi-cubic interpolation [32] of input patterns/features during the resizing operation. After the creation of a zero matrix, and adaptively resizing the dimensions of L, R stereo inputs, 3) we iteratively fill the zero values in zero-matrix with [R, G, B] values present in the resized L, R image,s this iterative filling of [zero matrix]N*xm is performed according to the matrix concate- nation operation (|) [24], [31]. The pre-processed image(I'') is then passed to the l,r-PanoED network to generate a panoramic stitch(O'') based on deep feature mapping.

B. PROPOSED NETWORK'S ARCHITECTURE

l,r-PanoED network is a deep convolutional encoder-decoder architecture [33] with split encoding network methodology with parallel lateral connections between the 2 encoder network architectures. Main operations performed in the l,r-PanoED network are 2D-Convolution (2D-Conv) [34], 2D-Deconvolution (2D-Deconv)/2D-Transposed convolution [35], Batch-Normalization (BN) [36], Parametric Rectified linear unit (PReLU) [37], Max-pooling [38], lateral connections + Feature-Concatenations, Feature similarity [24], [39], [40], loss and optimization operations [41].

$$BN\,(x_i) = \hat{x}^{(k)} = \frac{x_i^{(k)} - (\frac{1}{m}\sum_{i=1}^{m} x_i^{(k)})}{\sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i^{(k)} - (\frac{1}{m}\sum_{i=1}^{m} x_i^{(k)}))^2 + \varepsilon}} \quad (2)$$

BN [36] is used for feature normalization and to minimize covariance-shifts between the extracted feature units. Generally feature normalization is done by recalculating the extracted feature layer units $Y_{(i)}$ according to $BN_{\gamma\beta}$ $(x^{(K)})$ function where γ, β are learnable-parameters, x_i is i^{th} input batch iteration. $y_i^{(k)} = BN_{\gamma,\beta}\,(x^{(K)}) = \gamma^{(k)}x_i^{(k)} + \beta^{(k)}$ where k [1,d], i [1,m] and d is the dimension of feature-space/vector, m = epoch's batch-size. ε is a stability constant. 2D-Conv, 2D-Deconv [34], [35] ($H^1 = W^*_{NXM}H = \Sigma\ [W(i)xH(i)]$) operations are used for the feature extractions(H^1) by 2D convolving W(kernel or filter) on input FMs(H). Generally, during the 2D Conv operation, H layer's input units(x_i^k) present in N (neighborhood of W kernel) are mapped to a single y_i^k output-unit in H' layer (down-sampling); and during a 2D-Deconv operation a single input unit(x_i^k) in H layer is mapped to multiple feature units belonging to N neighborhood of H' layer. 2D-Conv operation extracts concentrated valued semantic feature maps, these 2D-Conv operations are generally implemented in CNNs & AED's encoder networks. Similarly, 2D-Deconv operations extract higher spatial-resolution categorical features from respective input FMs, and 2D-Deconv operations are used in GANs & AED's Decoder networks.

$$PReLU = \begin{cases} y_i^k & (if y_i^k \geq 0 \\ a(y_i^k) & (if y_i^k < 0) \end{cases} \quad (3)$$

In "a" is a learnable parameter [37]. Parametric-ReLU (PReLU) [37] is an activation function used to tackle Vanishing gradient & dying ReLU problems. PReLU introduces non-linearity among the extracted feature maps at respective convolutional layers. Non-linearity in FMs helps to improve robustness and generalization of extracted feature maps. Max-pooling operation [38] is a sample-discretization method, which extracts sharp and distinctive features and also reduces the dimensions of the input FMs for the next layer thereby reducing the chances of overfitting. Max-pooling chooses the max($\{y_i^k\}$) value available in N neighbourhood(i N) [38] for every n-stride non- overlapping regions(n = dim (n*n-max-pooling) operating kernel) covered by a max-pooling kernel. Feature concatenation operation is used for introducing lateral connections between 2 different feature layers. Generally, to maintain dimensional integrity among FLs/FMs while concatenating, we perform cropping & 2D-1*1-Conv operations on input FMs to alter their depths/dimensions to make them suitable for feature-concatenation operation. Feature-similarity functions are used for mapping deep features that are extracted by the CNNs, in this article we use the correlation distance function [40] as a similarity metric for the N.N.D.R algorithm [24], [39] to map the encoder network's extracted features. Loss functions [41] are used to estimate loss/cost incurred during training & validation, simply the loss value calculated by a loss function is used to estimate gradients for optimizing the entire neural network. Optimization functions [41] are used to optimize and adjust the weights of activation/feature units accordingly so that the calculated loss value is minimized during back-propagation. These previously discussed primary DNN operations are used in the architecture of the l,r-PanoED network.

Fig. 3 represents the overall Convolutional encoder-decoder architecture of the proposed l,r-PanoED network, where I^{tt} is the $L_{N1\,XM1}$, $R_{N2\,XM2}$-lateral concatenated image(with (N1 + N2)xm dimension) and O^{tt} is l,r-PanoED output unprocessed raw panoramic stitched image(with W'xH' dimension). l,r-PanoED network consists of a total 31 Feature Layers with lateral connections between some FLs. Out of those 31FLs, 19 feature layers belong to encoder networks and the rest of the 12 feature layers belong to the decoder network. In the encoder network part of l,r-PanoED architecture we implement "split encoding network methodology" (BL-encoder(Bottom_Layered-Encoder) and UL-encoder(Upper_Layered-Encoder)) with lateral connections between these 2 separately split BL & UL encoder networks. BL-encoder network mainly focuses on extracting features from the input images, and the UL-encoder network fine-tunes BL-encoder's Feature Maps by performing additional feature extractions in UL layers, and these fine-tuned Feature maps are mapped at the end of the UL-network encoder using F-Mat*(UL) function, and UL-8*-feature registration is performed based on these matched feature Maps (Fig. 3). Now the decoder network of l,r-PanoED inputs these UL-8*-registered FMs for recreating final robust panoramic view outputs. Lateral connections(C_j; j [1, 5]) between BL and UL encoders helps in the transfer of feature maps during the fine-tuning process. For a detailed explanation of the l,r-PanoED network, and its execution, we split the entire architecture into the Encoder network(Section 4.B.1) & decoder network(Section 4.B.2) during the description.

C. DESCRIPTION OF L,R-PanoED NETWORK's ENCODER ARCHITECTURE

The encoder network of l,r-PanoED consists of 2 subnetworks as discussed above, which are the BL-encoder network and the UL-encoder network. The encoder network consists of a total 19 feature layers (Fig. 3). 11 FLs (BL- 1 to BL-11) belong to the BL-encoder network and 8 FLs (UL-1 to UL-8) belong to the UL-encoder network. Each convolutional feature layer (except UL-8) in the encoder network is generated by applying either one of $EL^1 N,M$ or $EL^2 N,M$ sequences of operations

$$\begin{aligned} L_i &= EL^1_{N,M}\left(L_{i-1}\right) = PReLU(BN(N^*M_{2D} \\ &\quad -Conv(L_{i-1}))) \\ L_i &= EL^2_{N,M}\left(L_{i-1}, C_j\right) = PReLU(BN(1^*1\_2D \\ &\quad -Conv(C_j \oplus N^*M_{2D} - Conv\left(L_{i-1}\right)))) \end{aligned} \tag{4}$$

In, N, M are filter/kernel(W) dimensions, L_i (i^{th}-encoder layer) BL_i| UL_i, C_j is a Lateral connection established between the UL & BL encoders, and is a feature concatenation operation. In the encoder network depths of feature layers(of both BL & UL encoders) increase and the resolution of FLs decreases as we go deep into the encoder network from BL-1 to BL-11 and UL-1 to UL-7(exception for UL-8). In the encoder network, extracted features become semantically concentrated and pattern rich as we go down the network. As discussed previously the encoder network has 19 convolutional FLs (the BL encoder has 11encoding layers, and the UL encoder has 8 encoding layers) where each convolutional FL is generated by applying 2D-Conv(L_i) [34], BN [36], PReLU [37], 2*2-Max Pool-8 layers have 2098 FMs and are generated by applying $EL^1 3X3$ (BL-5), $EL^1 3X3$ (BL-6), $EL^1 3X3$ (BL-7). BL-9, BL-10, BL-11 are deep-bottleneck features maps of the l,r-PanoED network, and BL-9 has 4196 FMs and both BL-10, BL-11 have 8392 feature-map layers, where they are generated by applying $EL^2$3,3(BL-8, C_3), $EL^2$3,3(BL-9, C_4), $EL^1 3X3$ (BL-7) sequence of operation respectively. In $EL^2 XX$ sequence operations, we apply 1*1_2D-Conv() operation to the resulting feature concatenation because, when 2 feature layers are concatenated via C_j lateral connection the number of feature maps at that particular layer doubles in-depth, so to maintain spatial integrity among the peer feature layers we apply 1*1_2D-Conv() operation to reduceing [38] (only to some layers) sequence of operations. Inputs the depth of feature concatenated FMs to its ^{1} (i.e 1*1-to l,r-PanoED can be any N*M*3 dimensional L, R stereo images. BL-1 (of BL-encoder) has a depth of 128 Feature maps and is generated by applying $EL^1 11X11(I^{tt})$ sequence of operations. Both BL-2, BL-3 layers have 384 FMs and are generated by applying $EL^1 7X7$ (BL-1), $EL^1 5X5$ (BL-2) sequence of operations. BL-4, BL-5 encoder layers have 768 FMs(depth) and are generated by applying $EL^1 5X5$(BL- 3), $EL^1 5X5$(BL-4) operations. Similarly, BL-6, BL-7, BL-Conv() operations alter the depth of FMs adaptively so that entropy among them doesn't rise in this case). Generally in the l,r-PanoED network, C_i represents lateral connections between the encoder networks, C_1 connects BL-5 & UL-1, C_2 joins BL-8 & UL-3, C_3 connects UL-6 & BL-9, C_4 joins UL-7 & BL-10, C_5 joins C-4+BL-11 & UL-8 and finally C_6 concatenates UL-8*& BL-12(of decoder network). Now coming to the UL-encoder network, UL-1 has 768 FMs and is generated by applying $EL^2$3,3 (BL-5, C_1). UL-2, UL-3 layers have 2098 FMs respectively and are generated by applying $EL^1$3,3 (UL-1), $EL^2$3,3 (UL-2, C_2). UL-4, UL-5, UL-6, UL-7 layers have 4196, 4196, 4196, 8392 feature-map layers respectively and they are generated by applying $EL^1 3X3$ (UL-3), $EL^1 3X3$ (UL-4), $EL^1 3X3$ (UL-5), $EL^1 3X3$ (UL-6) operations respectively. UL-8 layer of UL-encoder performs several operations besides extracting fine-tuned feature extraction, where UL-8 extracts fine-tuned features by applying UL-8_{FT} (C_4,C_5) = PReLU(BN(1*1_2D-Conv(1*1_2D-Conv(C_5 3*3_2D-Conv

(C_4))))) sequence of operations. We apply 1*1_2D-Conv() operation consecutively 3 times at a single layer(UL-8) because, during the feature concatenation of C_4, C_5 the overall depth of resulting feature maps increase to 16792, now these 16K FMs(depth) should be adapted to the depth of BL-12 in the decoder network, therefore we perform 3x(1*1_2D-Conv) operations to reduce the depth of FMs to $^1/8$ (i.e 2098 FMs) of its current depth. This depth reduction can boost the overall speed and latency time of the feature mapping and registration process of UL-8* function, moreover modulating the depth while concatenation can increase the stability & helps to maintain homogeneity during progressive feature extraction or feature-reconstruction.

$$Correlation\,(p_k, q_k) = \sum\nolimits_{k=1}^{n} \times (\frac{\frac{1}{n}\sum_{k=1}^{n}(\left(p_k - \left(\frac{1}{n}\sum_{k=1}^{n}p_k\right)\right)\cdot\left(q_k - \left(\frac{1}{n}\sum_{k=1}^{n}q_k\right)\right))}{\frac{1}{n}\sum_{k=1}^{n}\left(p_k - \left(\frac{1}{n}\sum_{k=1}^{n}p_k\right)\right)\cdot\frac{1}{n}\sum_{k=1}^{n}(q_k - (\frac{1}{n}\sum_{k=1}^{n}q_k))} \tag{5}$$

After the completion of finely tuned feature extraction, the UL-8 layer performs other operations such as Feature mapping and F-Mat*(UL-8) Feature-registration. Feature mapping at UL-8 is done on UL-8_{FT} (C_4, C_5) feature maps using complete N*xM*-iterative N.N.D.R algorithm [39], [42] with Correlation(p_{2098}, q_{2098}) [40] function as p,q-feature similarity method(), where Correlation(p,q) represents feature correlative similarity score between 2 p_{2098},q_{2098} feature vectors. Now the N*xM*x2098 dimensional UL-8_{FT} (C_4, C_5) feature maps are split into 2 equal l, r-sub-feature maps where each sub-feature map has a dimension of (N*/2 × M*/2 × 2098). Now [1 × 2098] dimensional feature vectors(FVs) belonging to both l, r-sub-features maps are iteratively compared with each other to find the matches using NNDR [39], [42] feature matching method [39]. Generally p^i2098 represents a set of {$FV^i 1x2098$(l-sub-FM)} feature vectors where i [0, N*/2xM*/2], similarly q ^{j}2098 is a set of {FV $^j 1x2098$ (r-sub-FM)} FVs where j [N*/2xM*/2, N*XM*] range. Here in our proposed stitching method, we have implemented a complete N*xM* iterative feature search during l, r-sub-feature mapping in the NNDR-match function, to make our p, q-feature vector matching function in UL-8^*_{FT} () unit more robust and invariant (to input l, r-stereo image orientations and resolutions). During the iterative feature matching, FVs present in the p^i2098 set are Correlatively compared with every feature vector present in the q ^{j}2098 set. During this Correlation(p_k, q_k) similarity analysis, if any 2 vectors have a similarity score less than the Sim-Th(a similarity threshold, Sim-Th = 0.45) then those 2 Vectors (Fig. 4 (d)) are categorized as matched FVs (this process is repeated until vectors present in p,q set are completely analyzed). Finally, the complete set of matched UL-$8^{p,q}_{FT}$ = {p_i, q_j} FV pairs are passed to the F-Mat* (UL-8_{FT} ^{p,q}, UL-8_{FT}) feature registration function for stitching fine-tuned UL-8 FMs to generate a raw initial panoramic mapped FMS. These panoramically mapped UL-8_{FT} FMs along with BL-11 FMs are together passed to the Decoder network for generating O^{tt} panoramic images. In the F-Mat() function, Feature registration [43] of {p_i,q_j} FV paris is performed based on a H-homography matrix calculated using the N-iterative RANSAC algorithm [44].

Algorithm 1 Feature-Map Registration in UL-8∗

```
1.  F-Mat∗(UL-$8^{M}_{FT}$,UL-8ft) ;
    **Input:** UL-8ft fine tuned feature maps and matched {p,q} UL-$8ft^{m}$ FV pairs
    **Output:**
      • H homography matrix parameters calculated using the RANSAC algo.
      • Panoramically stitched UL-Sjq- FMs using H matrix Feature-registration.
2.  **for** i = 0 to N do
3.  randomly select a subset(R*) of 10 {p,q}FVs pairs from UL-Sp/'$^{p,q}$ super set
4.  compute homography H parameters using S.V.D and Direct linear transformation on R*{p,q}
5.  fix K = specific range constant and $FV_{pq}$ are feature vectors belong to respective $\{p_i\}\,\{q_i\}\,R^*\{p, q\}$ set
6.  if $S.S.D(FV_{p)}\ H.FV_q)<K$ then
7.      calculate plausible FV-inliers
8.  else
9.      continue
10. **end if**
11. Structure a key-value map with "i" as key and plausible FV^-inliers as value
12. **end for**
13. Compute least-squared H-params error value for all values present in N-iteration keys using  $FV_P$ ||
14. H = min $(\{\|H^*FV_q - FV_p\|\}_i)$; where ie[0,N] i.e fix the final homography parameters based on i∗ H-parameter's least squared error value calculated in above step-9
15. Perform feature-registration using step-lO's H homography matrix
16. **for** j =0 to (N∗/2xM∗/2) do
17.     UL-STpr are panoramically stitched/ registered Feature maps
18.     $UL\text{-}8^r_{FF}/j] = H.UL\text{-}8_F r[j]$
19. **end for**
20. return (UL-$8^t$FT, H)
21. **end**
```

Based on the above-discussed F-Mat*() Algorithm, UL-8_{FT} feature maps along with the matched pairs of UL-$8^{p,q}_{FT}$ feature vectors are all-together passed to the F-Mat*() function, internally in this function/algo an optimal H-Homography matrix [44] is estimated for performing UL-8_{FT} feature registration. The final output of this F-Mat()

function is a feature-registered initial raw panoramic feature-maps (UL-8^{T}FT). These UL-8^{T}FT FMs are transferred to the decoder network via C_6 lateral connection. The UL-8^{T}FT FMs along with BL-11 FMs(of encoder network) are passed to the decoder network for panoramic reconstruction. Additionally, under some rare scenarios(if the extracted or raw features/FMs between input l, r-stereo images are extremely noisy, highly non-homogeneous/non-continuous) where none of the feature vectors in present in the l, r-sub-feature maps are matched to each other, then in these cases we just simply concatenate the l, r-sub-feature maps laterally and then pass these appended FMs to the decoder network via C6 lateral connection.

1) DESCRIPTION OF L,R-PanoED NETWORK's DECODER ARCHITECTURE

The decoder network of l,r-PanoED architecture (Fig. 3) has 12 decoding layers(BL-12 to BL-23). The decoder network increases the spatial resolution of the feature maps while simultaneously decreasing its no. of FMs(depth), i.e the 2D-Deconv operations adaptively distributes the feature units among ith layer's FMs by increasing its resolution and simultaneously decreasing its depth with minute spatial and temporal pattern occlusions and losses. Our proposed decoder network follows a single stream decoding methodology by applying 2D-Deconv [35], 1*1_2D-Conv [34], BN [36], PReLU [37] operations. In this article, we apply 2D-Deconv operations to FMs similar to U-net [45] and FPN networks [46], but we do not crop any features in the resulting FMs, and instead keep the FM resolutions + spatial features as it is, so that the borders and extreme features present in the input l,r-stereo feed remain intact during the reconstruction process, and the overall feature co-occurrence isn't disturbed. Moreover, we do not store any pooling indices to apply up-sampling(un-pooling) + De- Conv operations (like in De-Convnet [47] & Segnet [48]) because we need to maintain our l,r-PanoED to be storage efficient and responsive during live-inferences(less-latency). The BL-12 of the decoder network has 2098 FMs and are generated by applying sequences of operations.

$$\begin{aligned}&PReLU(BN(1^*1\_2D - Conv(1^*1\_2D\\&-Conv(1^*1\_2D - Conv(3^*3_{2D}\\&-DeConv(C_6|UL - 8^{T}_{FT})))) \oplus (1^*1\_2D\\&-Conv(3^*3_{2D} - DeConv\,(BL - 11))))))\end{aligned} \tag{6}$$

In the BL-12 decoding layer (Fig. 3), feature maps from UL-8^{T}_{FT} via C_6 are Deconvolved to increase their resolution, and then a 1*1_2D-Conv() is applied to modulate the depth of C_6 de-convolved FMs, parallelly final FMs from the BL-encoder(BL-11) are also Deconvolved+1*1_2D-Conv (BL-11); now these 2 FMs from BL(BL-11) and UL(UL-8)-encoders are feature concatenated(), and the resulting concatenated features are reduced to $\frac{1}{2}$ in depth by applying 1*1_2D-Conv operation, now these extracted feature units are batch normalized(BN) [36] and then passed through the PReLU activation function [37] to get final BL-12 FMs. For the decoder network of l,r-PanoED we pass FMs from both UL, BL-encoder networks because, FMs from the UL-encoder contain panoramically stitched raw FMs, and FMs from the BL-encoder contain fine-tuned extracted bottleneck features, these bottleneck FMs contain latent representations of every object present in the input image, So by concatenating these both UL, BL FMs we can encode input pattern's latent co-occurrences while maintaining its Spatio-temporal coherence throughout the O''-reconstruction process. Rest of the convolutional feature layers in the decoder network are generated by applying either one of these DL^{1}*N,M* or DL^{2}*N,M* sequences of operations(refer to).

$$\begin{aligned}L_i &= DL^{1}_{N,M}\,(L_{i-1}) = PReLU(BN(N^*M\_2D\\&\quad -DeConv(L_{i-1})))\\L_i &= DL^{2}_{N,M}\,(L_{i-1}) = PReLU(BN(N^*M\_2D\\&\quad -Conv(L_{i-1})))\end{aligned} \tag{7}$$

The BL-13 convolutional decoder layer has a total of 2098 FMs and they are generated by applying DL^{2}3,3(BL-12) sequence of operations. Each of the BL-14, BL-15, BL-16 Feature layers have 768 FMs individually and are generated by applying DL^{1}3,3(BL-13), DL^{2}3,3(BL-14), DL^{2}3,3(BL-15) operation sets respectively. Similarly, BL-17, BL-18, BL-19 FLs have depths of 384FMs and are generated by applying DL^{1}3,3(BL-16), DL^{2}3,3(BL-17), DL^{2}3,3 (BL-18) sets of operational sequences respectively. BL-20, BL-21, BL-22 are the primary feature reconstruction layers, where the feature maps present in these layers contain features & patterns which are almost relatable to the features & patterns present in stereo input l,r images. BL-23 decoder network layer closely interpolates objects/patterns present in the extracted feature maps correlative to the GT panoramic images present in the training dataset. BL-20, 21, 22, 23 and O^{tt} have identical width and height(W'xH') dimensions, here W'xH' < (N1+N2)xmin(M1,M2); where N1xM1 is the input L-stereo image dimension, N2xM2 is the input R-stereo image dimension and m = min(M1,M2). At the start of BL-20 Feature layer, we interpolate the BL-19 FMs to W'XH' dimension using the bi-cubic interpolation method and later perform a fine-tuned feature extraction process. BL-20, BL- 21, BL-22 layers have 256, 128, 128 FMs respectively, and the decoder network's BL-23 feature layer has 64FMs, each of these BL-20-23 feature layers are generated by applying DL^{1}5,5(BL-19), DL^{2}3,3(BL-20), DL^{2}3,3 (BL-21), DL^{2}3,3(BL-22) respective sequence of operations; and O'' is the final panoramically stitched output generated by the proposed l,r-PanoED network. O '' has 3 [R, G, B] feature layers and are generated by applying DL^{2}3,3 (BL-23) sequence of operations. As discussed above 2-stride 2*2-Max-pooling() operation [38] is used for FM dimensionality reduction to efficiently optimize the l,r-PanoED params(primarily to BL, UL-encoder layers). BL-1, BL-3, BL-5, BL-8, UL-1, UL-3 layers in the l,r-PanoED network are dimensionally reduced to $\frac{1}{2}$ using 2-stride 2*2-Max

Pooling operation. Generally, the Max-pooling operation is only applied after performing fine-tuned feature extraction at respective layers. For UL-5 layer in the UL-encoder, a 2*2-max pooling with stride-1 and padding =2 is applied to refine its previously extracted FMs for optimal generation of semantically dense feature-units in the UL-8_{FT} FVs, for robust Feature matching & F-Mat*() feature registration operations [24], [25], [31], [39], [40], [43], [44]. The l,r-PanoED network's final output (i.e panoramically stitched raw image, Fig. 4 (e)) O" is sent to the post-processing module to yield a final fully processed panoramic image O (Fig. 4 (f)) with minimal reconstruction loss, occlusion & pattern distortions even on non-homogenous stereo input. We have optimized our l,r-PanoED network using a custom-built loss function($L_{\alpha,\beta,\gamma}$), which dynamically considers the reconstruction + panoramic-stitching loss of both l,r-PanoED's raw panoramic image(O") and the post-processed final panoramic image(O) to penalize the entire l,r-Stitch unit & l,r-PanoED network to yield robust and optimal results.

D. POST-PROCESSING MODULE

The Post-processing module involves 3 primary substeps which are 1) applying gain compensation [11] operation on O" image, 2) performing Modified-poisson-blending operation [24], [49]–[51] on the gain-compensated O" image, 3) finally applying texture correction operations for edge refinement & image fine-tuning [24], [31]. The l,r-PanoED network's output O" image contains irregular illumination patches near the stitched areas(generally in the center of the stitched image) (Fig. 4 (e)). These irregular illumination patches are due to the differences in lumination intensities between the l,r-stereo input image feed. To tackle these non-coherent illumination flows and lighting artifacts in O". We apply gain compensation [11], [24], [31] operation on the O" panoramic stitched image, to introduce a homogenous intensify function throughout the O" image(where the center

$$e = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} N_{ij}$$

$$\frac{\left(g_i . \frac{\sum\limits_{u_i \in R(i,j)} I_i(u_i)}{\sum\limits_{u_i \in R(i,j)} 1} - g_j . \frac{\sum\limits_{u_j \in R(j,i)} I_j(u_j)}{\sum\limits_{u_j \in R(j,i)} 1} \right)^2}{10^2}$$

$$+ \frac{(1-g_i)^2}{10^{-2}} \tag{8}$$

Gain compensation operation is necessary to eliminate image artifacts in the stitched mosaic O", these artifacts are generated during the FVs matching phase in F-Mat*(UL-$8_{FT}^{p,q}$, UL-8_{FT}) operation, and also during the recon- struction of panoramic mosaic from BL-12 FMs in the decoder network of l,r-PanoED. In the gain compensation operation, we minimize the normalized total gain error("e") with the help of gain-vectors "g_i" & "g_j" () This "e" error minimization is applied on the overlapping region(R) of the O " panoramic stitched mosaic i.e R(i,j); here n = 1 and i, j represents the stereo image segments(I_1, I_2) which are stitched together to form a mosiac; we restrict the n value to 1 because the proposed l,r-Stitch Unit can only processes pair of l,r-stereo images in a single instance/iteration. $N_{1,2}$ (i.e $N_{i,j}$ is the count of the total number of pixels belonging to an overlapped R region in O"). Now the normalized "e" error function is minimized by equalizing its respective derivative ($\partial(e)$) to 0. After minimizing the "e" function, we get a set of g_1, g_2-gain-vectors for adjusting the brightness/intensities of overall pixels present in O". This intensity adjustment is done by multiplying the current intensity value of a pixel at respective i, jth location with the corresponding g_1, g_2 gain vectors. After performing gain compensation to O" we again apply a robust blending operation [49]–[51] to minimize ghosting artifacts, registration occlusions and edge/pattern distortions(primarily in the l, r-intersection area of O"). The objective of applying a blending operation is to implement a seamless image cloning operation to make the final O panoramic image close to ground truths.

$$E = \int_T^{||divv - \Delta f||_2^2 dt} + \varepsilon \int_T^{||P-f||_2^2 dt} \tag{9}$$

In this proposed article we have chosen Modified-Poisson-Blending(MPB) method [49] in the post-processing module for performing seamless image cloning based on a benchmarking analysis concerning other alternative blend- ing methods. In the Modified Poisson Blending(E) the original poisson energy function [51] is tweaked by adding an extra color-preserving parameter(ε)(Equation (9)). Div v is the divergence operator for v vector field and Δf is the laplacian of constructed image and P, f are the vector-representations of the composed and constructed images respectively; and finally T represents the continuous dt segments ranging from 0 to dim(O) of the whole image. The color-preserving parameter controls the color adaptation level during the blending operation. The backdrop of implementing a normal poisson blending is that the colors in the source image will be completely adapted with respect to the target image. To overcome this disadvantage MPB method is proposed in which the rate of seamless color gradient-field cloning can be controlled by ε parameter.

$$FP\,[u, v] = P_{u,v}^{DcT^{-1}} = \frac{LoI_{u,v}^{DcT} - \varepsilon . I_{u,v}^{DcT}}{\nabla L_{u,v}^{DcT} - \varepsilon} \tag{10}$$

$$F_{u,v}^{DcT} = \sqrt{\frac{2}{N}}\sqrt{\frac{2}{M}}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} \Lambda(i)\,\Lambda(j) \,. \cos\left[\frac{\pi}{2}.\frac{u}{N}(2i+1)\right].\cos\left[\frac{\pi}{2}.\frac{v}{M}(2j+1)\right].f(i,j) \tag{11}$$

Here Λ(x) = 1/1.41, if (x = 0) otherwise x = 1. Moreover, the MPB method [49] is computationally effective and performs robust image cloning in majority of the cases with optimal PSNR,SSIM values [24], [31], [52]. In the proposed paper we perform seamless image cloning/blending in the intensity-frequency domain where the image vector and laplacian kernel are converted into a frequency domain using () discrete-cosine-transformation($F^{DcT}u,v$). The E function in the original MPB operation is converted into $P^{DcT}u,v$ in the frequency domain. In $F^{DcT}u,v$ DCT function, f(i,j) represent the original intensity value present in the $I_{N,M}$ image at i, j^{th} location and i, j values range from 0 to N, M; where N, M are the dimensions of the DCT function's input image I. FP is the final blended panoramic image which is generated by computing the inverse transformation of $P^{DcT}u,v$. LoI^{DcT} is the DcT transformation of the laplacian of O'' image. ε is the color-preserving parameter and we have set this parameter's value to 10^{-9} during our implementation/inference. I^{DcT} is the DcT of raw O'' intensity image, and L^{DcT} is the DcT transformation of a laplacian operator used in P^{DcT} operation. As discussed above, after calculating the inverse of P^{DcT} vector we get the final poisson blended image O with minimal ghosting artifacts, edge/pattern/color distortions, occlusions, and the resulting post-processed image O has high PSNR, SSIM [52] values when compared to its respective ground truth O^{GT} images. The MPB blended O image is further fine-tuned in the post-processing module's texture processing method to remove noise,blurring artifacts, and make the final fine- tuned O panoramic image look more natural and similar to O^{GT}. In the texture processing methods [24], [31] O is convoluted with a median filter to remove higher-level noise, and then we apply wiener filtering to minimize motion blurring artifacts, and finally we apply a image sharpening operation using robustly computed high pass filters. The texture processing method's fine-tuned O panoramic view image (Fig. 4 (f)) is the final robust output generated by the l,r-Stitch Unit by consuming homogeneous or nonhomogeneous synchronized stereo l,r input images. Panoramic images(O) generated by a single l,r-Stitch Unit using l,r-PanoED network has best performing benchmarking scores (SSIM,PSNR values wrt O^{GT}) in both wild and non-wild conditions when compared to other panoramic-stitching methods (Fig. 9, Tables 1, 2, 3, 4 and 5). The Inference time/ latency time consumed by our proposed l,r-Stitch Unit for processing 2 720P-left,right images is ~920ms, inference time required for stitching 2 1080P images is ~1450ms, and to stitch 2 2K images is ~ 2310ms.

E. ULTRA-WIDE PANORAMIC VIEW STITCHING STRATEGY

Based on the above-discussed methodology and techniques a complete l,r-Stitch Unit can be constructed, in which l,r stereo images are passed as input to the l,r-Stitch Unit to generate a robust and reliable panoramic stitched image. The proposed l,r-Stitch Unit runs on-live with low latency time for HD inputs(720p,1080p) videos. Our proposed methodology can also perform Ultra-wide view panoramic stitching($>180^0$) on continuous sequences of input images via camera arrays [12]. Refer to [12], Figs. 2 and 5 for the proposed pipeline followed during the continuous stitching of multiple l,r-stereo input image pairs. Generally, this sequential panoramic stitching is used in mobile phones [7], 3D surroundings recreation, etc where the user has to capture an entire $\sim 360^0$ view of external surroundings. The proposed l,r-Stitch Unit is highly modular and can be instantaneously plugged-in with other units based on-live sequential images requirement(i.e based on amount field of view(Θ^o) to be covered). We can roughly estimate the number (''N'') of independent l,r-Stitch Units required for covering Θ^o using.

$$\begin{aligned} N &= \sum_{i=1}^{K} i;\ where\ K \\ &= \lfloor(\frac{|\{no.of\ \ sequential\ \ camera\, frames\}|\, X63.5}{116})\rfloor \end{aligned} \tag{12}$$

Generally, Θ^o = (|{no. Of sequential camera frames}|-2)*63.5; here 63.5 is the average approximate field of view range covered by a normal single lens smartphone; N value may change when aperture, focal length values of a particular smartphone drastically differ from general smartphone's values(i.e phone-cameras with dual, triple, wide, macro view camera lens), but the end-user can manually append the N l,r-Stitch units to the inference pipeline dynamically based on the (Θ^o) requirement. During the PSNR, SSIM-benchmarking [52], and experimentation hypothesis we have observed that our proposed Ultra-wide view panoramic stitch methodology(with N modular l,r-Stitch Units) yields reliable results with optimal PSNR, SSIM [52] values when the required input Θ^o is in between $[0^0, 275^0]$ range, i.e when the input sequential camera frame count in the memory buffer is of [2], [6] range. Fig. 5 represents an inference pipeline scenario where the proposed Ultra-wide view panoramic stitching module covers $\sim 250^0$ FOV range. So in Fig. 5 (top) case, 6 sequential camera frames (or 3 pairs of l,r-stereo inputs) are passed to the proposed image-stitching module for processing, and to stitch these 6 input camera sequences the proposed method require N l,r-Stitch units to generate an ultra wide view (Here N = 6 based on). These 6 l,r-Stitch units are independently pooled according to the proposed Fig. 5 pipeline. Finally, these are the steps, methods, techniques involved in the proposed module for generating reliable panoramic views from given input image sequences. Next section details about l,r-Stitch unit's proposed loss function.

F. LOSS FUNCTION

$$\begin{aligned} &L_{\alpha,\beta,\gamma} \\ &= \sum_{l=1}^{n} (\alpha(\frac{1}{NM} \sum_{n=1}^{N} \sum_{m=1}^{M} [O_l^{GT}(n,m) \\ &\quad -O_l''(n,m)]^2) + \beta.(\frac{1}{NM} \sum_{n=1}^{N} \sum_{m=1}^{M} [O_l^{GT}(n,m) \\ &\quad -O_l(n,m)]^2) + \gamma.(1 - SSIM(O_l^{GT}, O_l''))) \end{aligned} \tag{13}$$

To optimize the above-discussed l,r-Stitch Unit (specifically l,r-PanoED network, Fig. 3), we have introduced an $L_{\alpha,\beta,\gamma}$

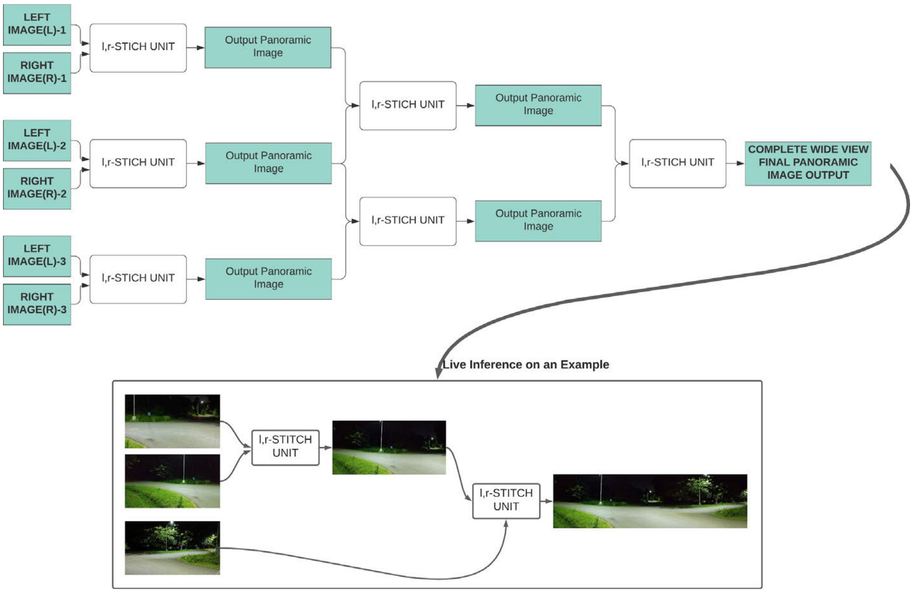

FIGURE 5. Architectural overview of the ensemble-tree created by effectively grouping "N" several modular l,r-Stitch Units together for performing Ultra-Wide panoramic stitching operations, this figure also illustrates a live inference on a sample test-case which consists of 3 input image sequences. Refer to Figs. 1 and 10. For more examples(generated by the proposed method) on ultra-wide panoramic stitching operation.

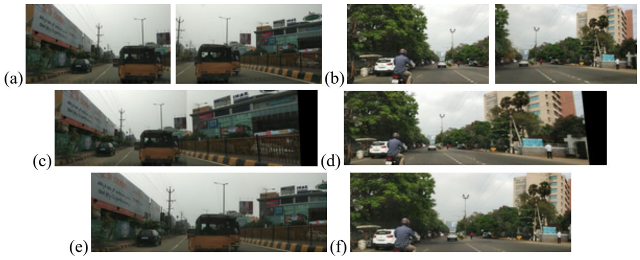

FIGURE 6. A brief performance comparison between the results(respective panoramic stitched) generated by the proposed l,r-Stitch unit & SIFT method. Where (a), (b) are the sample input l,r-stereo image sequences(taken from the test-dataset). (c), (d) are the final panoramic stitches generated by the SIFT method. (e), (f) are the final panoramic stitches generated by the proposed method.

loss function (Equation (13), (14), (15) and (16)) which cumulatively sums MSE, SSIM [24], [31], [52] loss values of both raw(O''), proposed(O) l,r-PanoED outputs compared to the ground truth panoramic images (O^{GT}) belonging to both training, validation datasets. In $L_{\alpha,\beta,\gamma}$ loss function the cumulative summing of MSE(O'', O^{GT}), MSE(O, O^{GT}), SSIM(O'', O^{GT}) loss values is based on adaptively tweakable $\{\alpha, \beta, \gamma\}$ weights, these weight values can be adjusted dynamically to adapt the penalizing rate of $L_{\alpha,\beta,\gamma}$ for better performance during the validation test & on-live inference. Performance(i.e MSE, PSNR, SSIM, $L_{\alpha,\beta,\gamma}$) plots of the proposed method during the training and validation phases are illustrated in Fig. 7.

$$SSIM\left(O_l^{GT}, O_l''\right) = \frac{2.\bar{O}_l^{GT}.\bar{O}_l'' + K_1}{\left(\bar{O}_l^{GT}\right)^2 + \left(\bar{O}_l''\right)^2 + K_1} \times \frac{2.CoVr\left(O_l^{GT}, O_l''\right) + K_2}{{O^{\sigma}}^{2}{}_l^{GT} + {O^{\sigma}}^{2}{}_l'' + K_2}; \; where$$

$$\bar{O}_l = \left(\frac{\sum_{i=1}^{N}\sum_{j=1}^{M} O_l(i,j)}{NM}\right)$$

$$O_l^{\sigma^2} = \left(\frac{\sum_{i=1}^{N}\sum_{j=1}^{M} \left(O_l(i,j) - \bar{O}_l\right)^2}{NM}\right) \qquad (14)$$

We calculate the Mean Squared Error of both raw and post-processed l,r-PanoED network's result, and additionally we also measure the Structural Similarity Index between O^{GT} and O (raw-l,r-PanoED output), and then the error, similarity scores are normalized to make them suitable for including in the $L_{\alpha,\beta,\gamma}$ loss function. "n" in $L_{\alpha,\beta,\gamma}$ loss function represents the total no of training and validation samples considered during the training and evaluation phase of the l,r-PanoED network. N, M are the dimensions of respective O^{GT}, O'', O panoramic images. In SSIM(O^{GT}, O'') function CoVr(O^{GT}, O'') represents [40] the covariance estimate between O^{GT}, O'' panoramic images, $\overline{O}$ represents the mean value of $O_{N,M}$ image and $O^{\sigma 2}$ measures the variance value of $O_{N,M}$ panoramic image. α, β, γ weight values are determined using the Bayesian optimization function [53] (for hyper-parameter optimization) with a set of constraints, i.e where $0 \geq \alpha < 1, 0 \geq \beta < 1, 0 \geq \gamma < 1$ and $\alpha + \beta + \gamma = 1$. The α, β, γ value's Bayesian optimization operation is performed on a sample test-dataset which is a subset of our custom-built dataset (details about our custom-built dataset are detailed in the next section). Next section details about l,r-Stitch unit's training details, benchmarking analysis, experimentation, and results of our proposed wide

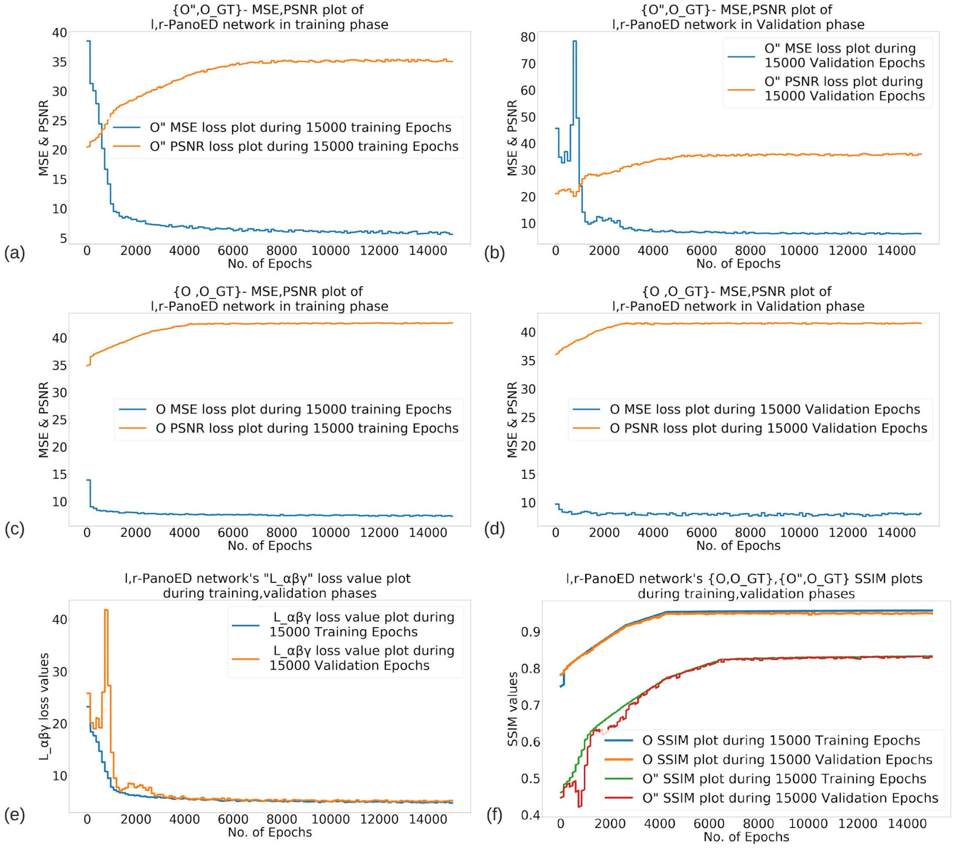

FIGURE 7. (a)-(f) are the MSE, PSNR, Lα, β, γ, SSIM plots of the proposed l,r-PanoED network during training, Validation phases. Where (a) is the MSE, PSNR values plot of l,r-PanoED's {O'', O^{GT}} during the 15,000 epochs training phase. (b) is the MSE, PSNR values plot of l,r-PanoED's {O'',O^{GT}} during the 15,000 epochs validation phase. (c), (d) are the MSE, PSNR values plots of l,r-PanoED's {O,O^{GT}} during the 15,000 epochs training, validation phases respectively. (e) is the Lα, β, γ value plot of l,r-PanoED network during the 15,000 epochs training, validation phases. Similarly, (f) is the SSIM plot of l,r-PanoED during the 15,000 epochs training, validation phases.

view panoramic stitching module.

$$\begin{aligned} &CoVr\left(O_l^{GT}, O_l''\right) \\ &= \frac{1}{NM}.(\sum_{i=1}^{N}\sum_{j=1}^{M}((O_l^{GT}(i,j) \\ &\quad -(\frac{\sum_{i=1}^{N}\sum_{j=1}^{M} O_l^{GT}(i,j)}{NM})).(O_l''(i,j) \\ &\quad -(\frac{\sum_{i=1}^{N}\sum_{j=1}^{M} O_l''(i,j)}{NM})))) \end{aligned} \quad (15)$$

$$\begin{aligned} &L_{\alpha,\beta,\gamma} \\ &= \alpha\left(MSE\left(O^{GT}, O''\right)\right) \\ &\quad +\beta(MSE(O^{GT}, O)) + \gamma\left(1 - SSIM\left(O^{GT}, O''\right)\right) \end{aligned} \quad (16)$$

V. TRAINING & EXPERIMENTATION DETAILS

Based on the previous discussions about l,r-PanoED network's architecture and custom loss function ($L_{\alpha,\beta,\gamma}$) (Equation (16) we train the entire l,r-Stitch Unit accordingly with corre- sponding hyper-parameters to generate realistic and reliable output panoramic images(O) with high PSNR, SSIM[52] values compared to other panoramic/image stitching methods. This proposed image stitching method is trained to generate optimal results even on non-homogeneous, wild l,r-input stereo images. In l,r-Stitch Unit, the stereo lateral concatenation module acts as a fixed pre-processing unit for generating I'' inputs for the l,r-PanoED network, and the Post-processing module performs fixed fine-tuning operations for further refinement(using image processing techniques) of the l,r-PanoED's raw generated output(O''). For training and optimizing the l,r-Pano-ED network present in the l,r-Stitch Unit, we have used an ensemble of multiple public datasets & our custom-built dataset.[3] Adobe Panoramas Dataset,[4] Sun360 [54], Pano-RSOD [55] are the public

[3] The custom-built-dataset "left-right synchronously stereo-paired Indian on-road-traffic" dataset is liscenced, and is subjected to no-objections(under govt. permission and regulatory) during capturing.

[4] Open Sourced at "https://inst.eecs.berkeley.edu/~cs194-26/fa18/upload /files/proj6B/cs194-26-aeh/website/"

datasets used for training; and IIIA Panorama[5] [22], Casual-stereoscopic-panorama stitching [56] are the public-datasets which are used only for testing and validation purposes. In addition to the public datasets used, we also train and validate our proposed l,r-PanoED on our inhouse built custom dataset, which is a collection of raw left, right stereo images, and ground-truth wide view panoramic images which were captured outdoors. Our custom-built dataset consists of outdoor recordings at 10 different geographical locations(total recordings/video-dataset duration is 200^{+}mins). The dataset repository consists of 4 main directories:

- raw-left video frames(extracted from videos captured through left-camera),
- raw-right video frames,
- raw-left, right video recordings,
- ground-truth Panoramic/wide view images(extracted from video captured through a camera with 150^{0} wide lens

The extracted-left, right video frames/images have 1920 × 1080 resolution with a total count of $267{,}300^{+}$ images. $135{,}000^{+}$ GT Panoramic view video frames are present in our custom-built dataset with a resolution of ∼2150 × 900 px. To increase the reliability and diversity of the proposed method. We have sampled over 10,000 raw panoramic images present in our custom-built dataset, and these sampled 10,000 panorama images are now manually stitched using the ORB image stitching [8] mechanism to further create ultra-wide panoramic views with resolutions of ∼3920 × 850px. Now, the public datasets along with our-custom built dataset(along with sampled ultra-wide vide panoramic images) are ensembled together and fed to the proposed l,r-PanoED to train rigorously for generating robust panoramic images(O”).

$$PSNR = 10X log_{10}\left(\frac{255^2}{\frac{\sum\limits_{i=0}^{m-1}\sum\limits_{j=0}^{n-1}[O(i,j)-O^{GT}(i,j)]^2}{m.n}}\right) \tag{17}$$

During training the left, right stereo images inputs are manually paired with the corresponding ground-truth panoramic view image, and in such manner, an over-all 95,000 samples are paired(including custom-built and public datasets) to train the l,r-PanoED network. Similarly, 7,500 samples are paired for testing and validating purposes. Based on the above-curated data we train the l,r-PanoED network for 15,000 epochs with a batch-size of 64. $L_{\alpha,\beta,\gamma}$ is the loss function used to estimate cost/error value during each epoch, and we optimize the estimated loss/error value using the Adam optimizer(with a momentum of 0.9 & weight-decay of 0.0005). We have initialized the a_i learnable parameter in PReLU activation functions [37] to 0.2 value for faster loss function convergence. Learning-Rate is fixed to a value of 0.0002, and all the initial-weights of the l,r-PanoED network are randomly assigned based on a zero-mean-centered gaussian distribution [24] with $\sigma = 0.015$. The above-mentioned training hyper-parameters are determined & optimized using the bayesian hyperparameter optimization technique [53]. The hardware configuration used in this article during training, validation, testing, and benchmarking phases is an Intel i7-9th gen processor coupled with NVIDIA-GTX-1070 GPU. We have employed a single-phase training strategy for the proposed method. The following figures detail about l,r-PanoED network's performance during the training and validation phases.

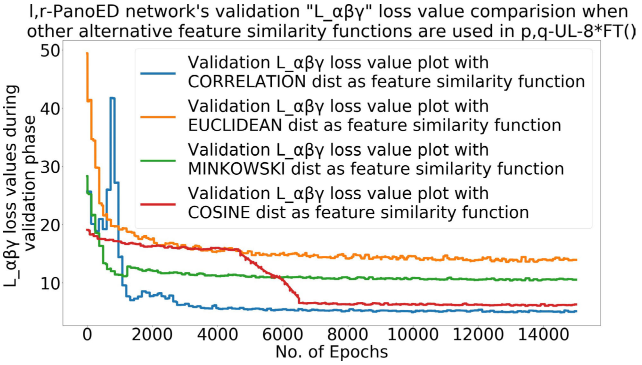

FIGURE 8. Performance comparison of l,r-PanoED network with other alternative similarity functions being used in the F-Mat*(UL-*) algorithm. The alternative similarity functions used in this analysis are CORRELATION distance(preferred in this article), EUCLIDEAN distance, MINKOWSKI distance, COSINE distance. The above performance analysis is based on l,r-PanoED's $L_{\alpha\beta\gamma}$ loss values during the 15,000 epochs validation phase.

Fig. 7 (a) details about {O”, O^{GT}} MSE, PSNR [52] values plot against 15,000 epochs in the training phase, i.e this plot illustrates the error difference between l,r-PanoED's raw output(O”) and the ground-truth O^{GT} images for every training epoch(0-15,000). Similarly, Fig. 7 (a) details about {O”,O^{GT}} MSE, PSNR values plot against 15,000 epochs during the validation phase. Fig. 7 (c), (d) illustrates the mean-square-error, PSNR between l,r-Stitch Unit's post-processed output(O) and ground-truth(O^{GT}) image pairs for every epoch(1-15,000) in both training, validation phases respectively. Fig. 7 (e) details about l,r-Stitch unit's loss value plot for 0-15,000 epochs in both training, validation phases. Similarly, Fig. 7 (f) details about structural similarity index(SSIM) between {O, O^{GT}}& {O”, O^{GT}} pairs for 0-15,000 epochs in both training, validation phases. Finally, Fig. 8 shows the performance comparison of l,r-Stitch units when different similarity functions are used in the feature stitching operation (p,q-UL-8_{FT}*()) in the 8^{th}-FL of l,r-PanoED's UL encoder network. Different similarity functions considered during the benchmarking analysis are Correlation distance(implemented in this article) [40], Euclidean distance [57], Minkowski distance [58], Cosine distance [59] functions.

As our proposed method primarily focuses on performing robust image stitching operation on non-homogeneous

[5] Open Sourced at “https://www.iiia.csic.es/~aramisa/datasets/iiiapanos.html”

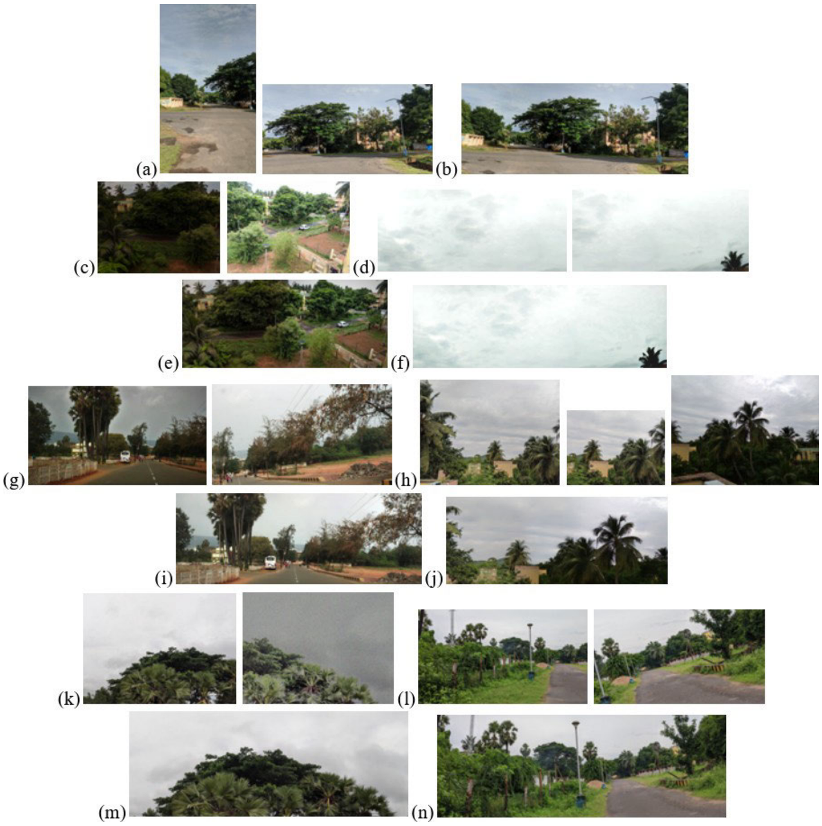

FIGURE 9. (a)-(n) are sample input image sequence sets along with their corresponding panoramic views stitched. These sample images are randomly chosen from the test dataset (the test dataset is a fragment of our custom-built traffic dataset). (a), (c), (d), (g), (h), (k), (l) are the input image-sequence sets, and (b), (e), (f), (i), (j), (m), (n) are their respective corresponding generated panoramic views. Each of these input image sets has at-least one non-homogeneous variance property among them. Refer to Tables 1, 2, 3, 4 and 5 for a detailed benchmarking analysis of our proposed method under these test wild conditions.

wild input stereo images, therefore we have mainly focused on discussing the performance and results of our proposed method on wild non-homogeneous inputs. Tables 1, 2, 3, 4 and 5, Fig. 9 discussion will primarily focus on the performance evaluation of our proposed method in multiple wild conditions and some sample results. Here in this section we compare and benchmark our proposed method along with other image-stitching mechanisms(SIFT [13], ORB [8], KAZE [14], BRISK [15] & SURF [16]) with 4 main evaluation criterias, i.e PSNR, SSIM, Avg. Latency Time [52], Feature-Matching rate.[6] All the above-stated image-stitching modules [8], [13]–[16] along with ours are subjected to multiple wild conditions, and the performance of each method is recorded in these wild conditions and evaluated. Five different wild conditions are considered for this benchmarking analysis, namely 1) Rotational Variation, 2) Resolution & Orientational(portrait & landscape) Variation, 3) Salt & Pepper Noise, 4) Manipulating % of common/matching area between input stereo images and 5) Color & Lumination/Intensity variation. Each table below (Tables 1, 2, 3, 4 and 5) details the performance eval- uation of all image-stitching modules(including l,r-Stitch Unit) under a specific wild-nonhomogeneous condition. All these wild conditions are modeled at multiple scales or levels for better performance analysis, and also to measure the sensitivity of each method under each wild condition at different levels/scales.

We have benchmarked Tables 1, 2, 3, 4 and 5 methods on our custom-built dataset, where we have manually sampled 2,500 pairs(left, right & GT panoramic images) for this benchmarking test dataset. These 2,500 samples from the test dataset are further divided into 5 sub-categories (∼500 images for each), where each sub-category consists of test images modeled according to their respective wild condition(i.e rotation, noise, lumination variance, etc.). Now the 500 samples of each wild conditions are even further split into groups of various scales (according to Tables 1, 2, 3, 4 and 5); for example, scales present in rotational angle variation wild condition (Table 1) are “0^0, 10^0, 20^0, 30^0, 45^0”, and now the 500 categorized samples of this respective wild condition are split into 5 different groups, where each group contains ∼100 test-image pairs

[6] Feature-Matching rate = Average((total no. of matched features/no. of features detected in left input-image), (total no. of matched features/no. of features detected in right input-image))

TABLE 1. Details about Benchmarking analysis-1. In the benchmarking analysis-1, we thoroughly benchmark the proposed method (l,r-Stitch Unit) along with other alternative image-stitching methods (SIFT [13], ORB [8], SURF [16], KAZE [14], BRISK [15] (refer to Column 1)) on a "rotational-variance-wild-condition" dataset. In this benchmarking analysis, we evaluate the performance of each image-stitching method based on PSNR, Mean SSIM, FM-Rate evaluation-metrics (Column 2). Images present in this rotational-variance-wild-condition dataset are skew rotated to either of (0, 10, 20, 30, 45) rotational scales/degrees (Column 3 to Column 7); an even distribution of data is maintained for each rotation-scale during the benchmarking analysis-1.

Methods	Evaluation Metric	0^0	10^0	20^0	30^0	$45^{\ 0}$
		Rotational Variance				
SIFT	PSNR	45.53	37.74	33.15	28.31	24.12
	Mean SSIM	73.62				
	FM Rate(%)	89.68	75.51	66.99	57.92	50.04
ORB	PSNR	48.75	43.58	40.02	35.07	30.48
	Mean SSIM	86.55				
	FM Rate(%)	95.68	86.71	79.98	70.05	61.38
SURF	PSNR	46.75	40.46	35.34	30.6	28.68
	Mean SSIM	78.72				
	FM Rate(%)	91.81	80.07	71.09	62.37	58.12
KAZE	PSNR	41.58	34.96	28.96	24.6	21.58
	Mean SSIM	71.13				
	FM Rate(%)	87.27	74.16	61.99	53.73	47.69
BRISK	PSNR	45.02	39.71	36.5	32.26	28.05
	Mean SSIM	83.85				
	FM Rate(%)	93.9	82.87	77.14	68.79	60.03
Ours	**PSNR**	**46.68**	**44.02**	**40.73**	**37.17**	**32.81**
	Mean SSIM	**88.82**				
	FM Rate(%)	**97.45**	**91.31**	**84.72**	**78.14**	**69.89**
Processing Latency						
SIFT	Avg. Latency Time	1.1402				
ORB		1.1619				
SURF		1.0137				
KAZE		0.7103				
BRISK		1.1742				
Ours		**1.1685**				

TABLE 2. Details about Benchmarking analysis-2. This benchmarking analysis is divided into two parts (1:variations in resolution-scale & 2: variations in orientational differences). Here, we thoroughly benchmark the proposed method (l,r-Stitch Unit) along with other alternative image-stitching methods (SIFT [13], ORB [8], SURF [16], KAZE [14], BRISK [15] (refer to Column 1)) on a "resolution-scale/Orientational variance wild-condition" dataset. In this benchmarking analysis, we evaluate the performance of each image-stitching method based on PSNR, Mean-SSIM [52], FM-Rate evaluation-metrics (Column 2). Images present in the resolution-scale variance wild-condition dataset are either up/down -ampled based on the resize-interpolation scale (0.25x, 0.5x, 0.75x, 1.0x) (Column 3 to Column 6); and every image in the Orientational variance wild-condition dataset is randomly flipped to landscape or portrait orientation; an even distribution of data is maintained for each resolution-scale & orientation during the benchmarking analysis-2.

Methods	Evaluation Metric	0.25x	0.5x	0.75x	1.0x
		Variations in resolution scale			
SIFT	PSNR	24.95	29.95	33.95	41.67
	Mean SSIM	71.68			
	FM Rate(%)	54.84	63.81	72.23	86.67
ORB	PSNR	31.55	36.43	39.58	44.81
	Mean SSIM	85.59			
	FM Rate(%)	67.06	77.01	83.6	92.54
SURF	PSNR	27.58	32.34	36.4	42.51
	Mean SSIM	76.68			
	FM Rate(%)	59.19	68.32	77.09	88.55
KAZE	PSNR	23.13	27.15	33.35	40.18
	Mean SSIM	68.38			
	FM Rate(%)	50.55	58.85	71.08	84.02
BRISK	PSNR	30.56	35.15	37.94	43.72
	Mean SSIM	80.58			
	FM Rate(%)	65.62	74.09	79.93	90.77
Ours	**PSNR**	**35.48**	**38.69**	**42.25**	**45.54**
	Mean SSIM	**90.97**			
	FM Rate(%)	**75.1**	**81.62**	**88.6**	**94.33**
Variations among Orientational Differences (portrait, landscape)					
SIFT	PSNR	39.73			
	Mean SSIM	74.71			
	FM Rate(%)	83.84			
ORB	PSNR	43.86			
	Mean SSIM	86.81			
	FM Rate(%)	92.2604			
SURF	PSNR	40.85			
	Mean SSIM	78.03			
	FM Rate(%)	87.216			
KAZE	PSNR	37.01			
	Mean SSIM	68.28			
	FM Rate(%)	78.162			
BRISK	PSNR	42.06			
	Mean SSIM	83.69			
	FM Rate(%)	89.973			
Ours	**PSNR**	**44.98**			
	Mean SSIM	**92.38**			
	FM Rate(%)	**96.0186**			

which are rotated according to their reactive i^{th} scale's θ^0 rotational angle. Similarly, the 500 samples of each wild condition are grouped and modeled accordingly. Now the 1^{st} test-condition(Table 1) "Rotation angle variation" contains test-samples in which either of the input left, right stereo images are rotated in θ^0 clockwise or anticlockwise direction respectively accord- ing to 0^0, 10^0, 20^0, 30^0, 45^0 scales. The average latency values mentioned in Table 1 are the global average values of processing-latencies taken by a specific method in all of the wild-test conditions(i.e on all 2,500 test samples) during the benchmarking analysis. 2^{nd} test-condition(Table 2) " resolution-scale variation" and "variations in Orientational difference" contains test-samples in which input left, right stereo images are either up-scaled or down-scaled w.r.t i^{th} scale's (0.25x,0.5x,0.75x,1.0x) scaling-factor, and this test-condition also consists of test-samples in which the orientations of input stereo images are altered(either to portrait or landscape and vice-versa). 3^{rd} test-condition(Table 3) "Applying Salt & Pepper Noise" [24], [31] contains test-samples, where salt and pepper noise has been manually added to input stereo images based on i^{th} scale's(5%, 15%, 30%, 45%) noise concentration. 4^{th} test-condition(Table 4) "Manipulating % of common/matching area", contains test-samples which are manually analyzed for categorization into respective groups((0-5)%, (5-10)%, (10-20)%, >20%), in which each specific group contains input left, right stereo images with similar common/matching area between them. The final test condition (Table 5) "color/intensity variation" contains test samples in which hue, saturation co-variations & lumination/ intensity differences are manually induced in between the input stereo images, the variations applied among these input images are modeled according to low, medium, high factors(i.e low factor refers to lower intensity variations among input test-cases, and high factor refers to higher intensity, color variations among the stereo inputs). Based on a detailed analysis of observations and results illustrated in Tables 1, 2, 3, 4 and 5 and Fig. 9, the order of performance in terms of image-stitching quality & feature mapping

TABLE 3. Details about Benchmarking analysis-3. Here we thoroughly benchmark the proposed method (l,r-Stitch Unit) along with other alternative image-stitching methods (SIFT [13], ORB [8], SURF [16], KAZE [14], BRISK [15] (refer to Column 1)) on a "Noise-variance-wild-condition" dataset. In this benchmarking analysis, we evaluate the performance of each image-stitching method based on PSNR, Mean SSIM [52], FM-Rate evaluation-metrics (Column 2). Images present in this noise-variance-wild-condition dataset are subjected to a manual addition of salt & pepper noise with 5%, 10%, 20%, 30% concentrations (Column 3 to Column 6); an even distribution of data is maintained for each noise concentration category during the benchmarking analysis-3.

Methods	Evaluation Metric	5%	10%	20%	30%
		Variations in amount of Salt & Pepper Noise in the input images sequence			
	PSNR	39.58	33.42	25.2	19.59
SIFT	Mean SSIM	69.06			
	FM Rate(%)	82.62	70.66	54.88	43.86
	PSNR	43.05	37.37	29.51	25.28
ORB	Mean SSIM	78.42			
	FM Rate(%)	89.46	78.37	63.48	54.89
	PSNR	40.31	35.09	28.34	22.84
SURF	Mean SSIM	75.16			
	FM Rate(%)	84.34	73.99	60.92	50.52
	PSNR	38.58	31.64	25.23	18.61
KAZE	Mean SSIM	66.92			
	FM Rate(%)	80.39	67.21	55.11	41.78
	PSNR	40.06	36.16	29.22	24.03
BRISK	Mean SSIM	75.41			
	FM Rate(%)	86.97	76.17	62.58	52.77
	PSNR	**43.86**	**40.46**	**33.57**	**29.52**
Ours	**Mean SSIM**	**80.36**			
	FM Rate(%)	**91.32**	**84.76**	**71.48**	**63.08**

TABLE 4. Details about Benchmarking analysis-4. Here, we thoroughly benchmark the proposed method (l,r-Stitch Unit) along with other alternative image-stitching methods (SIFT [13], ORB [8], SURF [16], KAZE [14], BRISK [15] (refer to Column 1)) on a "variable Stereo match-area wild-condition" dataset. In this benchmarking analysis we evaluate the performance of each image-stitching method based on PSNR, Mean SSIM [52], FM-Rate evaluation-metrics(Column 2). Input stereo Images with different matching/common areas are present in this variable Stereo match-area wild-condition dataset, and these images are categorized into either of these ((0-5)%, (5-10)%, (10-20)%, (>20)%) categories based on matching/common areas in between them(Column 3 to Column 6); an even distribution of data is maintained among each category for the benchmarking analysis-4.

Methods	Evaluation Metric	(0-5)%	(5-10)%	(10-20)%	>20%
		Variations in % of matching/common area in-between the input stereo pairs			
	PSNR	17.51	22.88	32.95	41.92
SIFT	Mean SSIM	61.64			
	FM Rate(%)	40.33	50.45	70.08	87.81
	PSNR	23.64	30.36	39.02	46.11
ORB	Mean SSIM	76.82			
	FM Rate(%)	51.81	65.33	82.12	95.16
	PSNR	21.93	25.83	36.23	43.82
SURF	Mean SSIM	70.58			
	FM Rate(%)	48.46	55.66	77.05	91.48
	PSNR	16.93	25.32	33.16	40.62
KAZE	Mean SSIM	62.53			
	FM Rate(%)	39.26	54.72	70.26	84.54
	PSNR	22.83	28.01	37.98	44.53
BRISK	Mean SSIM	73.99			
	FM Rate(%)	49.8	60.46	79.33	92.48
	PSNR	**27.64**	**34.05**	**41.32**	**46.81**
Ours	**Mean SSIM**	**83.07**			
	FM Rate(%)	**59.99**	**71.61**	**86.82**	**96.98**

TABLE 5. Details about Benchmarking analysis-5. Here we thoroughly benchmark the proposed method (l,r-Stitch Unit) along with other alternative image-stitching methods (SIFT [13], ORB [8], SURF [16], KAZE [14], BRISK [15] (refer to Column 1)) on a "Color/intensity variance-wild-condition" dataset. In this benchmarking analysis, we evaluate the performance of each image-stitching method based on PSNR, Mean SSIM [52], FM-Rate evaluation-metrics (Column 2). Images present in this Color/intensity variance-wild-condition dataset are subjected to color/intensity manipulations on different levels (low, medium, high) (Column 3 to Column 5); an even distribution of data is maintained for each manipulation level during the benchmarking analysis-5.

Methods	Evaluation Metric	Low	Medium	High
		Variations in colors & intensities among the input stereo pairs		
	PSNR	38.62	29.12	20.35
SIFT	Mean SSIM		75.87	
	FM Rate(%)	87.17	70.08	52.82
	PSNR	42.49	35.12	27.74
ORB	Mean SSIM		86.29	
	FM Rate(%)	95.11	80.3	66.15
	PSNR	40.81	31.81	23.21
SURF	Mean SSIM		79.63	
	FM Rate(%)	92.071	73.64	58.4
	PSNR	37.91	28.44	18.62
KAZE	Mean SSIM		70.55	
	FM Rate(%)	86.3	68.25	48.74
	PSNR	41.33	33.04	26.05
BRISK	Mean SSIM		82.88	
	FM Rate(%)	92.78	77.03	62.83
	PSNR	**43.89**	**38.26**	**30.05)**
Ours	**Mean SSIM**		**92.16**	
	FM Rate(%)	**98.418**	**87.82**	**71.93**

capability is "**Ours(l,r-PanoED)**> **ORB [8]** > **BRISK [15]** > **SURF [16]** > **SIFT [13]** > **KAZE [14]**" and; "**KAZE [14]** > **SURF [16]** > **SIFT [13]** > **ORB [8]** > **Ours(l,r-PanoED)**> **BRISK [15]**" is the performance order in terms of Avg. Latency speed metric (KAZE [14] requires the least processing latency, and BRISK [15] requires the highest processing latency).

Fig. 9 illustrates the final post-processed results of our proposed method on some random input samples taken from the above discussed benchmarking (Tables 1, 2, 3, 4 and 5) Test-Dataset. Fig. 9 (a) represents the input left, right stereo images set with orientational variations between them(portrait & landscape), and Fig. 9 (b) is the final post-processed output generated by our proposed method for Fig. 9 (a) input. Fig. 9 (c) contains input stereo images with luminous/ intensity variations(on high factor), along with 10%-noise induced among them(left & right images have lower and higher intensities); Fig. 9 (e) is the respective output for Fig. 9 (c) input. Fig. 9 (d) contains input stereo-images with low availability of features/patterns and also 10% of common-area in between the stereo pairs(i.e low availability of features in input images makes it difficult for the feature extractor in a specific method to extract relevant features for mapping & image registration operations); Fig. 9 (f) is the corresponding output for Fig. 9 (d) input. Fig. 9 (g) left input image has lower lumination/intensity level compared to Fig. 9 (g) right image, and Fig. 9 (g) right image is rotated 10^0 wrt ground truth horizontal level. Some test samples with an ensemble of multiple wild-conditions our proposed method has outperformed other image-stitching methods and generates phenomenal outputs(refer to Fig. 9 (i); with lesser error

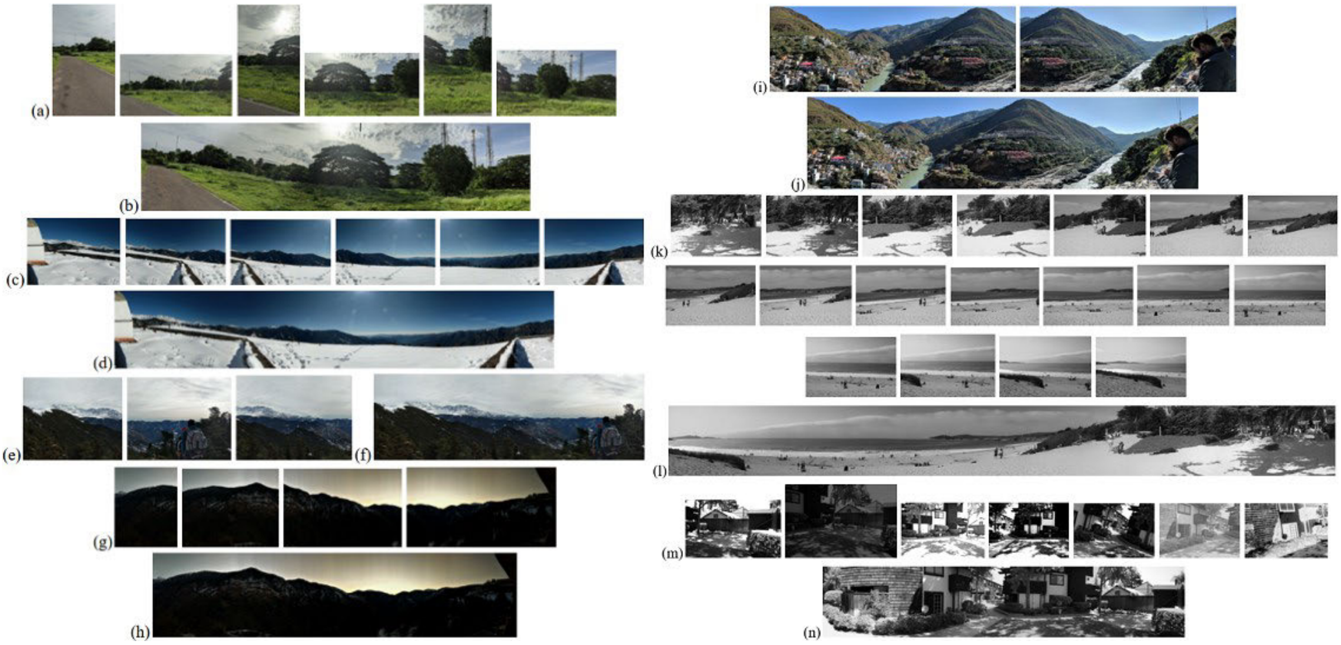

FIGURE 10. (a)-(n) consists of sample multi-input image sequence sets along with their corresponding ultra-wide panoramic views stitched (using an adaptive ensemble of "N" modular l,r-Stitch Units). These sample images are randomly chosen from the test dataset (the test dataset is a fragment of our custom-built traffic dataset & other panoramic public datasets). (a), (c), (e), (g), (i), (k), (m) are the input image-sequence sets, and (b), (d), (f), (h), (j), (l), (n) are their corresponding ultra-wide panoramic views generated {(a), (c), (e), (g), (i) belong to our custom-built dataset and (k), (m) belong to adobe panoramic dataset}. The multi-image sequences present in the test-dataset cover all possible FOV ranges mentioned in Table 6.

TABLE 6. Performance analysis of the proposed method with multiple FOV (O<0<330) ranged input image sequences as evaluation criteria (refer to column 1); The performance analysis is performed on the test dataset, where the test datasets is a fragment of our custom-built traffic datasets and other publicly available panoramic datasets. Column 2 refers to the average resolution of the final stitched panoramic view. Column 3 refers to the average latency time required by our proposed method for stitching a panoramic view for a respective input FOV ranged image sequence. Column 4 lists the number of individual modular l,r-Stitch units required to stitch a particular column 1 FOV ranged panoramic view. Column 5 details the minimum and maximum PSNR, SSIM values scored by our proposed method for a particular column 1 FOV range input image sequence. Column 4 details about the approximate number of "N"-l,r-Stitch Units required for stitching Column 1 input FOV range.

FOV Range	Avg. O' Resolution	Avg. Latency Time	Approx. "N"	{Min, Max} PSNR, SSIM.
$\Theta^0 < 60^0$	~ 1184x829	1.108	1	{45.18, 45.902}, {89.43, 90.90}
$60^0 < \Theta^0 < 120^0$	~ 2251 x 968	1.142	1	{46.87, 47.21}, {92.68, 93.35}
$120^0 < \Theta^0 < 180^0$	~ 3907 x 973	1.726	3	{42.63, 43.22}, {84.27, 85.52}
$180^0 < \Theta^0 < 250^0$	~ 6498 x 982	1.901	6	{39.27, 40.83}, {77.727, 80.74}
$250^0 < \Theta^0 < 300^0$	~ 9386 x 985	2.297	9	{35.15, 37.76}, {69.59, 74.77}
$\Theta^0 > 300^0$	~ 11463 x 979	2.461	11	{30.02, 32.65}, {59.34, 64.57}

deviation & high similarity index w.r.t GT output images). Similarly, Fig. 9 (k) is also an ensemble of multiple test-conditions, Fig. 9 (k) right input image is induced with 30% salt and pepper noise along with lumination/intensity variations involved, Fig. 9 (m) is the final output generated by l,r-Stitch Unit for Fig. 9 (k) input. Fig. 9 (h) test sample consists of multiple input images with an ensemble of "resolution scale variation" & "(0-5)% manipulating matching/common areas" test-conditions, Fig. 9 (j) is the corresponding output for Fig. 9 (h) input. In Fig. 9 (l) test case, the right input image is rotated 30^0 from the ground-truth horizontal level and Fig. 9 (n) is its corresponding generated output.

It's evident from the above discussed benchmarking analysis and sample results, that our proposed image-stitching methodology can generate robust and reliable results even under multiple possible wild conditions (non-homogeneous input). The above-illustrated results in Fig. 9 are primarily raw results generated by a single l,r-Stitch unit. To support some use-cases which require ultra wide-view panoramic stitching(with FOV$>180^0$), we have proposed an adaptive ensemble pipeline which consists of N independent l,r-Stitch Units(detailed in section 4.D). For stitching ultra- wide panoramic views, we have proposed a pipeline that intuitively assembles modular l,r-Stitch units in a tree structure [12] based on "{N, K}" factors. Fig. 10 illustrates some sample results of our ultra-wide view stitching module. Fig. 10 (b), (d), (h), (n) ultra-wide panoramic views cover a field of view greater than 230^0. Fig. 10 (e)'s ultra-wide output Fig. 10 (f) covers a field of view in between 180^0-220^0. Fig. 10 (i) consists of input panoramic stereo wide- view images, and Fig. 10 (j) is its corresponding ultra-wide view panoramic stitch. Fig. 10 (l) ultra-wide panoramic stitch covers $<300^0$ field of view (Fig. 10 (k) as input). Table 6 details the performance analysis of our proposed method under multiple input FOV ranges ($0^0 < \Theta^0 < 330^0$ with normal and ultra-wide views), in Table 6 we have evaluated the proposed pipeline's performance using {MSE, PSNR, Avg latency time [52] } metrics. Refer to Figs. 9 and 10 for sample results, and refer to Tables 1, 2, 3, 4, 5 and 6 for a detailed performance analysis of our proposed module. Based on these benchmarking results, it's evident that our proposed image-stitching method has outperformed other methods [1], [3]–[21] with a minimum margin of 2 in every evaluation metric in all wild/non-homogeneous test-conditions, and input FOV ranges.

VI. LIMITATIONS & FUTURE-SCOPE

Although the proposed method is reliable and robust enough to handle most of the real-life & wild scenarios, there are some limitations and scope for future work. Observed limitations of our proposed method are, higher processing latency compared to other conventional image-stitching methods; higher space/memory footprint is required for an effective inference. Latency time (or) inference time of our proposed method makes it infeasible for integrating to other 3^{rd} party applications that require live processing and analysis. Our proposed method runs efficiently on higher hardware configurations, and is not suitable for inferencing on IoT and general mobile devices. To train the l,r-PanoED network we require a large diversified training data(chances of applying transfer-learning are low in this particular usecase). We follow a supervised learning approach, so the data generation, pairing & labeling of the training data is a hectic and time-consuming task, in the future we would like to introduce an unsupervised learning approach to tackle this problem. To handle a particular(new) real life wild condition while performing the image stitching operation, our proposed method requires at least 50-100 training samples of that particular wild condition to efficiently overcome the situation.

VII. CONCLUSION

This article has introduced a robust & reliable image-stitching/mosaicing methodology named l,r-Stitch Unit (Figs. 1, 2 and 3), which operates efficiently in-between $30^0 <$ field of view$<320^0$ range. The proposed l,r-Stitch unit is a novel system-pipeline that consists of several modules, i.e a pre-processing module, l,r-PanoED network (Fig. 3) (an encoder-decoder CNN proposed in this article), and a post-processing module. We have introduced a unique split encoding network methodology in the l,r-PanoED for simultaneous deep-feature extraction & mapping op- erations. The split encoder network of the proposed l,r-PanoED network was used for extraction + fine-tuning of relevant deep-features, while simultaneously performing mapping & matching(using F-Mat*(UL-8F T p,q, UL-8F T) algorithm) of the corresponding extracted fine-tuned deep-feature maps. The decoder network of the l,r-PanoED was used for intuitive reconstruction of raw-panoramic views (Fig. 4 (e)) from corresponding UL-8 feature-maps belonging to a respective l,r-stereo input image. Custom loss-functions were used during the training phase to optimize the proposed network. The proposed l,r-PanoED network plays a key role in stitching efficient and reliable raw-panoramic views, and based on these raw-panoramic views the proposed l,r-Stitch generates (by performing post-processing) accurate and realistic final panoramic views. An effective ensemble of multiple datasets (which includes our custom-built stereo-traffic dataset) was used to train the l,r-PanoED network. The l,r-Stitch Unit's post-processing module consists of an ensemble of powerful & effective image processing techniques [11], [24], [31], [32], [49]–[51] to minimize exposure differences, distortion artifacts & matching + texture errors present in the outputs of l,r-PanoED. Section 5's extensive benchmarking analysisTables 2, 3, 4, 5 and 6 has proved that our proposed image-stitching mechanism has stitched panoramic views with greater accuracy and reliability compared to other existing image-stitching methodologies [1], [3]–[21]. l,r-Stitch unit has outperformed other imagestitching methods([1], [3]–[21]) by a span of 2^+ in PSNR, SSIM, FM-Rate [29], [52] metrics Tables 1, 2, 3, 5 and 4 within optimal latency time for both homogeneous and nonhomogeneous input sequences. Although significant limitations & challenges were handled in this article, there will be a never-ending quest for improvements in any technical research domain, therefore some of the future enhancements we plan to include are prior mentioned in Section 6.

REFERENCES

[1] Z. Hua, Y. Li, and J. Li, "Image stitch algorithm based on SIFT and MVSC," in *Proc. 7th Int. Conf. Fuzzy Syst. Knowl. Discovery*, vol. 6, 2010, pp. 2628–2632, doi: 10.1109/FSKD.2010.5569813.

[2] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive digital photomontage," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 294–302, Aug. 2004, doi: 10.1145/1015706.1015718.

[3] C. Arth, M. Klopschitz, G. Reitmayr, and D. Schmalstieg, "Real-time self-localization from panoramic images on mobile devices," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, Los Alamitos, CA, USA, Oct. 2011, pp. 37–46, doi: 10.1109/ISMAR.2011.6092368.

[4] H. S. Faridul, J. Stauder, and A. Tremeau, "Illumination and device invariant image stitching," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 56–60, doi: 10.1109/ICIP.2014.7025010.

[5] P. S. W. Y. A. Levin and A. Zomet, "Seamless image stitching in the gradient domain," in *Proc. ECCV*, M. J. Pajdla, Ed., 2004, pp. 377–389, doi: 10.1007/978-3-540-24673-2_31.

[6] J. Gao, L. Yu, T.-J. Chin, and M. Brown, "Seam-driven image stitching," in *Eurographics*, vol. 2013, pp. 45–48, May 2013, doi: 10.2312/conf/EG2013/short/045-048.

[7] Y. Xiong and K. Pulli, "Fast panorama stitching for high-quality panoramic images on mobile phones," *IEEE Trans. Consum. Electron.*, vol. 56, no. 2, pp. 298–306, May 2010, doi: 10.1109/TCE.2010.5505931.

[8] M. Wang, S. Niu, and X. Yang, "A novel panoramic image stitching algorithm based on ORB," in *Proc. Int. Conf. Appl. Syst. Innov. (ICASI)*, May 2017, pp. 818–821, doi: 10.1109/ICASI.2017.7988559.

[9] Z. Wang, Y. Chen, Z. Zhu, and W. Zhao, "An automatic panoramic image mosaic method based on graph model," *Multimedia Tools Appl.*, vol. 75, no. 5, pp. 2725–2740, Mar. 2016, doi: 10.1007/s11042-015-2619-0.

[10] M. Alomran and D. Chai, "Feature-based panoramic image stitching," in *Proc. 14th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2016, pp. 1–6, doi: 10.1109/ICARCV.2016.7838721.

[11] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vis.*, vol. 74, no. 1, pp. 59–73, Apr. 2007, doi: 10.1007/s11263-006-0002-3.

[12] M. Baygin and M. Karakose, "A new image stitching approach for resolution enhancement in camera arrays," in *Proc. 9th Int. Conf. Electr. Electron. Eng. (ELECO)*, Bursa, Turkey, Nov. 2015, pp. 1186–1190, doi: 10.1109/ELECO.2015.7394569.

[13] Y. Li, Y. Wang, W. Huang, and Z. Zhang, "Automatic image stitching using SIFT," in *Proc. Int. Conf. Audio, Lang. Image Process.*, Shanghai, China, 2008, pp. 568–571, doi: 10.1109/ICALIP.2008.4589984.

[14] P. F. Alcantarilla, A. Bartoli, and A. Davison, *KAZE Features*, vol. 7577, C. Fitzgibbon and S. Lazebnik, Eds. Berlin, Germany: Springer, 2012, pp. 214–227, doi: 10.1007/978-3-642-33783-3_16.

[15] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2548–2555, doi: 10.1109/ICCV.2011.6126542.

[16] G. Juan and Oubong, "SURF applied in panorama image stitch- ing," in *Proc. 2nd Int. Conf. Image Process. Theory, Tools Appl.*, Paris, France, 2010, pp. 495–499, doi: 10.1109/IPTA.2010.5586723.

[17] L. Kang, Y. Wei, J. Jiang, and Y. Xie, "Robust cylindrical panorama stitching for low-texture scenes based on image alignment using deep learning and iterative optimization," *Sensors*, vol. 19, no. 23, p. 5310, Dec. 2019, doi: 10.3390/s19235310.

[18] J. L. Ranteallo Sampetoding, B. Satriyawibowo, Williem, R. Wongso, and F. A. Luwinda, "Automatic field-of-view expansion using deep features and image stitching," *Procedia Comput. Sci.*, vol. 135, pp. 657–662, 2018, doi: 10.1016/j.procs.2018.08.230.

[19] H. V. Dung, D.-P. Tran, N. Nhu, T.-A. Pham, and V.-H. Pham, *Deep Feature Extraction for Panoramic Image Stitching*. Cham, Switzerland: Springer, 2020, pp. 141–151, doi: 10.1007/978-3-030-42058-1_12.

[20] M. Rupp, "Stitchnet: Image stitching using autoencoders and deep convolutional neural networks," University of Bern, Bern, Switzerland, Tech. Rep., 2019. [Online]. Available: http://www.cvg.unibe.ch/media/theses/document/maurice-rupp/2019/BA_MauriceRupp.pdf

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[22] A. Ramisa, A. Tapus, R. Lopez de Mantaras, and R. Toledo, "Mobile robot localization using panoramic vision and combinations of feature region detectors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 538–543, doi: 10.1109/ROBOT.2008.4543262.

[23] E. Adel, M. Elmogy, and H. Elbakry, "Image stitching based on feature extraction techniques: A survey," *Int. J. Comput. Appl.*, vol. 99, no. 6, pp. 1–8, Aug. 2014, doi: 10.5120/17374-7818.

[24] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed. New York, NY, USA; Springer-Verlag, 2010.

[25] M. Wyawahare, P. Patil, and H. Abhyankar, "Image registration techniques: An overview," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 2, no. 3, pp. 11–28, 2009.

[26] W. Lyu, Z. Zhou, L. Chen, and Y. Zhou, "A survey on image and video stitching," *Virtual Reality Intell. Hardw.*, vol. 1, no. 1, pp. 55–83, 2019, doi: 10.3724/SP.J.2096-5796.2018.0008.

[27] C.-Y. Chen and R. Klette, "Image stitching—Comparisons and new techniques," in *Lect. Notes Comput. Sci.*, vol. 1689, pp. 615–622, Dec. 19993, doi: 10.1007/3-540-48375-6_73.

[28] V. S. Desanamukula, P. K. Chilukuri, P. Padala, P. Padala, and R. andP Pvgd, "AMMDAS: Multi-modular generative masks processing architecture with adaptive wide field-of-view modeling strategy," *IEEE Access*, vol. 8, p. 198 748-198 778, 2020, doi: 10.1109/ACCESS.2020.3033537.

[29] E. Karami, S. Prasad, and M. Shehata, "Image matching using SIFT, SURF, BRIEF and ORB: Performance comparison for distorted images," 2017, *arXiv:1710.02726*. [Online]. Available: http://arxiv.org/abs/1710.02726

[30] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," in *Proc. Int. Conf. Comput., Math. Eng. Technol. (iCoMET)*, Sukkur, Pakistan, Mar. 2018, pp. 1–10, doi: 10.1109/ICOMET.2018.8346440.

[31] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Englewood, Cliffs, NJ, USA: Prentice-Hall, 2009, doi: 10.1117/1.3115362.

[32] A. Wahab, "Interpolation and extrapolation," in *Proc. Topics Syst. Eng. Winter Term*, vol. 17, 2017, pp. 1–6.

[33] J. Chul Ye and W. Kyoung Sung, "Understanding geometry of encoder-decoder CNNs," 2019, *arXiv:1901.07647*. [Online]. Available: http://arxiv.org/abs/1901.07647

[34] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016, *arXiv:1603.07285*. [Online]. Available: http://arxiv.org/abs/1603.07285

[35] H. Gao, H. Yuan, Z. Wang, and S. Ji, "Pixel deconvolutional networks," 2017, *arXiv:1705.06820*. [Online]. Available: http://arxiv.org/abs/1705.06820

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1026–1034, doi: 10.1109/ICCV.2015.123.

[38] H. Wu and X. Gu, "Max-pooling dropout for regularization of convolutional neural networks," 2015, *arXiv:1512.01400*. [Online]. Available: http://arxiv.org/abs/1512.01400

[39] R. Panigrahy, "An Improved Algorithm Finding Nearest Neighbor Using Kd-trees," in *Proc. Latin Amer. Symp. Theor. Informat. (LATIN)*, vol. 4957. Cham, Switzerland: Springer, Apr. 2008, pp. 387–398, doi: 10.1007/978-3-540-78773-0_34.

[40] Y. Rodriguez, B. D. Baets, M. M. Garcia, C. Morell, and R. Grau, "A Correlation-Based Distance Function for Nearest Neighbor Classification," in *Progress in Pattern Recognition, Image Analysis and Applications*, J. Ruiz-Shulcloper W. G. Kropatsch, Eds. Berlin, Germany: Springer, 2008, pp. 284–291, doi: 10.1007/978-3-540-85920-8_35.

[41] S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A survey of optimization methods from a machine learning perspective," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3668–3681, Aug. 2020, doi: 10.1109/tcyb.2019.2950779.

[42] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005, doi: 10.1109/tpami.2005.188.

[43] L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, Dec. 1992, doi: 10.1145/146370.146374.

[44] U. Baid, "Image registration and homography estimation," M.S. thesis, Shri Guru Gobind Singhji Inst. Eng. Technol., Nanded, India, Aug. 2015, doi: 10.13140/RG.2.2.19709.67043.

[45] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351, N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[46] T. Y. Lin, P. D., and R. Girshick, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[47] N. Noh, S. Hong, and B. Han, "Learning deconvolution networkfor semantic segmentation," in *Proc. IEEE Int. Conf. onComputer Vis. (ICCV)*, Dec. 2015, pp. 1520–1528, doi: 10.1109/ICCV.2015.178.

[48] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/tpami.2016.2644615.

[49] M. Tanaka, R. Kamio, and M. Okutomi, "Seamless image cloning by a closed form solution of a modified Poisson problem," in *Proc. SIGGRAPH Asia Posters*. New York, NY, USA: Association for Computing Machinery, 2012, pp. 1–54, doi: 10.1145/2407156.2407173.

[50] Z. Zhu, J. Lu, M. Wang, S. Zhang, R. R. Martin, H. Liu, and S.-M. Hu, "A comparative study of algorithms for realtime panoramic video blending," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2952–2965, Jun. 2018, doi: 10.1109/TIP.2018.2808766.

[51] J. M. Di Martino, G. Facciolo, and E. Meinhardt-Llopis, "Poisson image editing," *Image Process. Line*, vol. 5, pp. 300–325, Nov. 2016, doi: 10.5201/ipol.2016.163.

[52] U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study," *J. Comput. Commun.*, vol. 7, no. 3, pp. 8–18, 2019, doi: 10.4236/jcc.2019.73002.

[53] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, vol. 12, 2016, pp. 2951–2959. [Online]. Available: http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf

[54] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba, "Recognizing scene viewpoint using panoramic place representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2695–2702, doi: 10.1109/CVPR.2012.6247991.

[55] Y. Li, G. Tong, H. Gao, Y. Wang, L. Zhang, and H. Chen, "Pano-RSOD: A dataset and benchmark for panoramic road scene object detection," *Electronics*, vol. 8, no. 3, p. 329, Mar. 2019, doi: 10.3390/electronics8030329.

[56] F. Zhang and F. Liu, "Casual stereoscopic panorama stitching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2002–2010, doi: 10.1109/CVPR.2015.7298811.

[57] L. Wang, Y. Zhang, and J. Feng, "On the Euclidean distance of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1334–1339, Aug. 2005, doi: 10.1109/tpami.2005.165.

[58] H. Bará Çolakoâlu, "A generalization of the minkowski distance and a new definition of the ellipse," 2019, *arXiv:1903.09657*. [Online]. Available: http://arxiv.org/abs/1903.09657

[59] A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, "Cosine similarity to determine similarity measure: Study case in online essay assessment," in *Proc. 4th Int. Conf. Cyber IT Service Manage.*, Bandung, Indonesia, 2016, pp. 1–6, doi: 10.1109/CITSM.2016.7577578.

PREMITH KUMAR CHILUKURI received the B.Tech. degree in computer science. He is currently pursuing research in deep learning and image-processing domains at Andhra University College of Engineering (A). His research interests include autonomous systems, neural cognitive intelligence, computer vision, deep learning, reinforcement learning, and signal/image processing (with a current expertise of more than three years in these research areas).

PREETHI PADALA received the B.Tech. degree from the National Institute of Technology Surathkal, Karnataka. Her research interests include data sciences, artificial intelligence, image processing, pattern recognition, machine learning, and the IoT, with a current expertise of more than two years in these domains.

PUSHKAL PADALA is currently pursuing the B.Tech. degree in computer science and Engineering with NIE, Mysore. He is actively participating in various workshops at startup cells, Hackathons, in the areas of data security and machine learning. His research interests include data security, image processing, machine learning, and the IoT-based applications.

VENKATA SUBBAIAH DESANAMUKULA received the B.Tech. degree from Nagarjuna University and the M.Tech. degree from BI-HER, Chennai. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, under the Visvesvaraya Ph.D. Scheme, Government of India. His research interests include object detection using deep learning, cyber security, and machine learning.

PRASAD REDDY PVGD is currently a Senior Professor with the Computer Science and Systems Engineering Department, Andhra University College of Engineering (A), Andhra University. His research interests include machine learning, soft computing, software architectures, image processing, number theory, and cryptosystems.