RBE549: Project 1 - MyAutoPano

Shreyas Devdatta Khobragade
MS in Robotics Engineering
Worcester Polytechnic Institute
skhobragade@wpi.edu

Neel Girish Bahadarpurkar
MS in Robotics Engineering
Worcester Polytechnic Institute
nbahadarpurkar@wpi.edu

I. Phase 1 : Traditional Approach

A. Introduction

This project looks into image stitching methods to create seamless panoramas from overlapping source images. Using images with significant feature overlap (empirically determined to be 30-50% or greater), the methodology applies a strong feature-based approach. This approach includes the following steps:

- **Corner Detection:** Identifying key corner elements within each image.
- **Adaptive Non-Maximal Suppression (ANMS):** Selecting a well-distributed subset of strong corner elements.
- **Feature Extraction:** Creating descriptive feature vectors for each specified corner.
- **Feature Matching:** Finding correspondences between features in overlapping images.
- **Outlier Removal using RANSAC:** Eliminating incorrect feature matches to improve homography estimation.
- **Homography Estimation:** Calculating the geometric transformation between image pairs.
- **Warping and Blending:** Transforming and seamlessly merging images into a single panoramic view.

B. Corner Detection

The initial phase of panoramic image stitching involves the detection of feature points, which is a fundamental task in computer vision. To select prominent picture features in this implementation, Shi-Tomasi corner detection is used, which is accomplished using cv2.goodFeaturesToTrack. This method was chosen because it is effective in locating well-localizedidentifiable features that can then be used for feature matching and estimate. The method's parameters were empirically adjusted to enhance performance across many image datasets. The detected corners for Set 2 are shown in Figure 1.

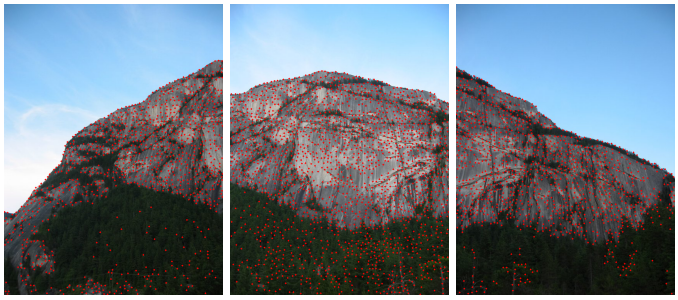

Fig. 1. Output of Corner Detection

C. Adaptive Non-Maximal Suppression (ANMS)

An important step in feature-based image stitching is to ensure that identified corners are distributed spatially uniformly in order to reduce warping artifacts in the final panorama. Raw corner detection algorithms frequently generate clusters of responses around significant features, resulting in redundancy and inefficient feature distribution. Adaptive Non-Maximal Suppression (ANMS) tackles this by picking a subset of the strongest corners that are evenly distributed over the image. The ANMS algorithm emphasizes corners that are real local maxima in a certain neighborhood, thereby suppressing weaker, more closely situated answers. This approach ensures that the selected features correspond to separate image regions, resulting in a more robust homography estimation and a smoother, more visually pleasing stitched panorama. The ANMS corners are displayed in Figure 2.

Fig. 2. Output of ANMS

D. Feature Descriptors

After the corner detection and selection step, feature descriptors are used to encode the properties of each recognized keypoint. These descriptors serve as unique IDs for the local picture portions at the corners. A prevalent approach, is to extract a patch centered around each corner. To reduce computational complexity and obtain some lighting invariance, the patch is subjected to Gaussian blurring and subsampling. This dimensionality reduction reduces the patch to a compact feature vector. Finally, normalization assures that the vector has zero mean and unit variance, which reduces the impact of illumination changes. This approach provides each corner with

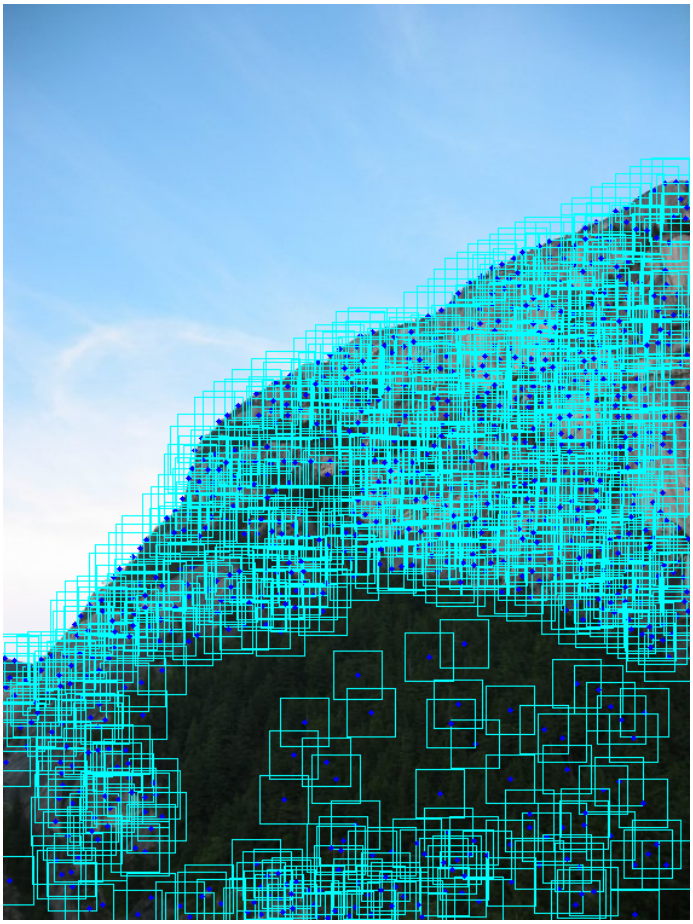

Fig. 3. Output of Feature Descriptor

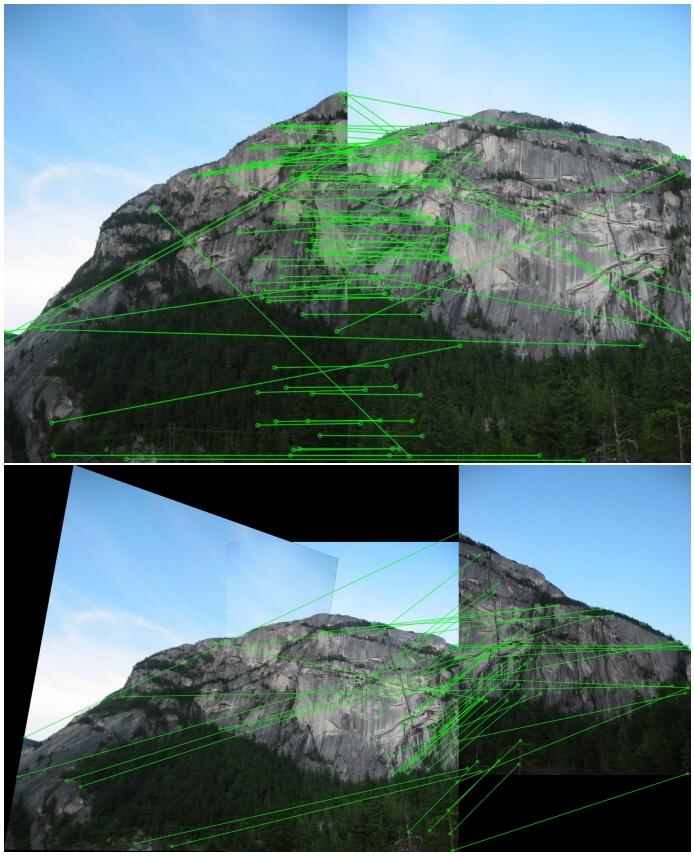

Fig. 4. Output of Feature Matching

a strong descriptor, allowing for rapid feature matching across overlapping images during the panoramic stitching phase. The feature descriptors are displayed in Figure 3.

E. Feature Matching

After identifying feature descriptors, establishing feature correspondences within overlapping pictures is essential for estimating geometric transformations. To discover matching keypoints, feature vectors from different image pairs are compared. To assess the similarity of feature vectors, a common way is to use the Sum of Squared Differences (SSD) distance metric. To improve matching robustness and eliminate spurious matches, a ratio test is used. This test evaluates the distance between the best and second-best matches; matches with a ratio less than a specified threshold are maintained, suggesting a high level of confidence in the connection. This filtered set of feature correspondences serves as the foundation for calculating the homography, which specifies the geometric transformation of the images. The output is diaplayed in Figure 4.

F. RANSAC

After establishing feature correspondences, a vital stage in robust image stitching is the eradication of erroneous matches, often known as outliers. The Random Sample Consensus (RANSAC) algorithm provides a solid foundation for outlier rejection and precise homography estimation. RANSAC works by iteratively picking minimal random groups of matched feature pairs and computing a candidate homography from each subset. The quality of each homography is then assessed by counting the number of inliers, which are defined as matches that adhere to the transformation within a given error margin.

The iterative process continues until the maximum number of iterations is achieved or a sufficient number of inliers are discovered. The homography associated with the greatest number of inliers is chosen as the best transformation. Finally, to improve the homography estimate, all discovered inliers are used in a least-squares fit. This robust approach successfully reduces the effects of outliers, resulting in a more precise and dependable homography prediction, which is required for subsequent image warping and blending. The output for RANSAC is shown in Figure 5.

G. Stitching and Blending

Image stitching is the technique of aligning and integrating two images into a bigger composite via a homography transformation. The stitch_images function first calculates the size of the output canvas by modifying the corner points of the second image using the computed homography matrix. This trans-

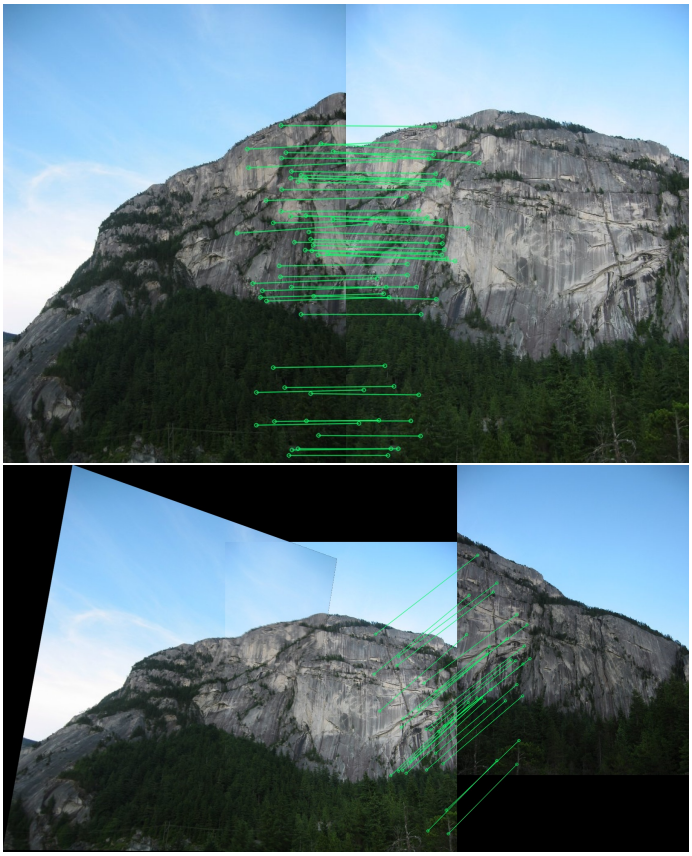

Fig. 5. Output of RANSAC

formation contributes to the establishment of new coordinate boundaries, compensating for any shifts caused by perspective alterations. A translation matrix is used to compensate for negative coordinates and ensure that all pixel positions remain within a valid image space. The second image is then warped into this new canvas with cv2.warpPerspective, which maps its pixels based on the homography transformation, resulting in an aligned image that fits into the stitched output.

Blending is used to provide a smooth transition between the overlapping sections of the photographs. The first image is applied straight to the canvas, retaining its original pixel values. In overlapping regions where both images provide pixel information, a basic averaging method is utilized, in which the intensity value of each pixel is calculated as the average of the corresponding pixels from both images. This blending approach minimizes abrupt borders and visual artifacts that otherwise would result from lighting or color mismatches. The iterative nature of the method allows additional images to be included, gradually enlarging the stitched panorama while retaining seamless transitions between frames.

The method we are following employs an incremental picture stitching strategy in which two images are processed at a time: the current image is aligned with the previously merged result using a homography transformation. The second image is warped onto a shared coordinate system and the overlapping parts are blended to provide a smooth transition. The newly stitched image is then used as the reference for the following iteration, which merges it with the next incoming image. This iterative procedure continues until all photos are included, gradually extending the stitched panorama while retaining alignment and smooth blending. The final outputs for Set2, Set1 and Set3 are shown in Figure 6, Figure 7, Figure 8 respectively. We also created a Custom Set with 6 images and it's output is displayed in Figure 9.

Fig. 6. Output of Image Stitching and Blending for Set 2

H. Observations and Conclusion

1) Train Sets: The stitching algorithm was evaluated on a variety of datasets, with mixed results. For Sets 1 and 2, a large number of RANSAC matches were accomplished, resulting in high-quality panoramas. However, issues developed with Set 3, which had eight images. While the first three images were successfully converted into panoramas, attempts to combine more images (4 and up) were unsuccessful. To solve this, several stitching procedures were used.

One solution included iteratively sewing photos from left to right. We tried to implement a different logic for circum-

Fig. 7. Output of Image Stitching and Blending for Set 1

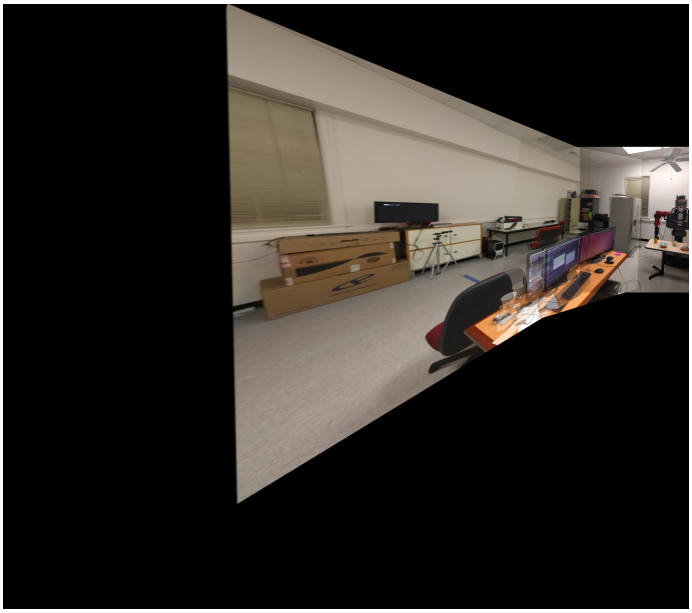

Fig. 8. Output of Image Stitching and Blending for Set 3

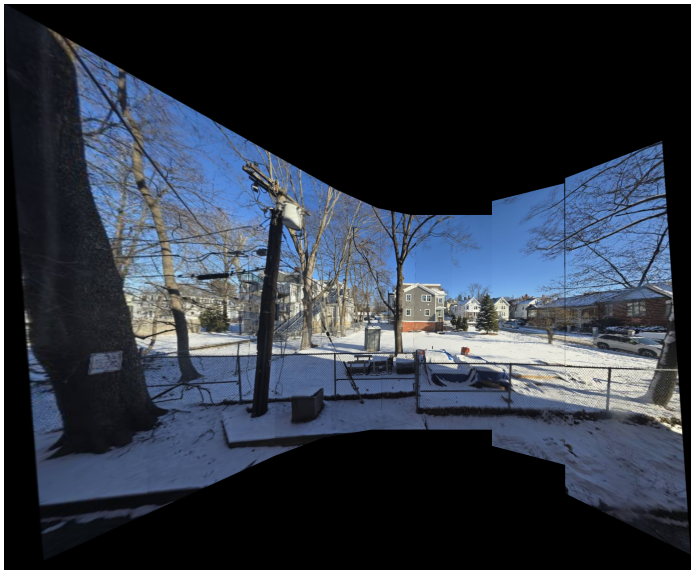

Fig. 9. Output of Image Stitching and Blending for Custom Set

stances where the total number of images exceeded three. In this method, the stitching process distinguished between photographs in the first and second parts of the collection. For the first half of the images, the algorithm used the usual homography matrix to stitch each new image to the existing panorama. For images in the second half, stitching was done in reverse, with the existing panorama matched to the new image using the inverse of the homography matrix. If the total number of images were three or fewer, the stitching method was conventional, which involved joining images sequentially without differentiation. Despite applying this logic and experimenting with cylindrical warping, the stitching procedure for Set 3 remained fruitless after the first three images.

In contrast, the method performed well with a new custom dataset of six images, producing a seamless panorama. These findings emphasize the existing method's limits for big image sets and indicate areas for development in managing complex stitching circumstances.

Another important finding during the RANSAC-based stitching procedure was the presence of several features from the edge of one image mapping to a single location in the next image. This frequently resulted in poor homography estimation and misalignment in the panorama. To remedy this issue, extra logic was built that ignores edge cases. Specifically, points along the margins of images were eliminated from the RANSAC matching procedure. By bypassing these edge locations, the method improved the accuracy of the feature correspondence and provided more trustworthy homography transformations, resulting in the production of better panoramas.

2) Test Sets: For Test Set 1, the algorithm stitched the first three images reasonably well but had difficulty including the fourth image into the panorama as shown in Figure 10. This was also due to the blending approach used, which takes average pixel values from both the images in the output. For Test Set 2, the algorithm successfully matched the first four images, but stitching proved challenging after the fifth image, despite obtaining a significant amount of RANSAC features. For this set, our code repeatedly ended up showing segmentation fault error and killing the task, this was due to the fact that the image size was increasing as more images were merged and our system ran out of memory while stitching any further; we also tried resizing the merged images after every iteration however, which did not work as well. The RANSAC matching result is shown in Figure 11. For Test Set 3, the

Fig. 10. Output of Test Set 1

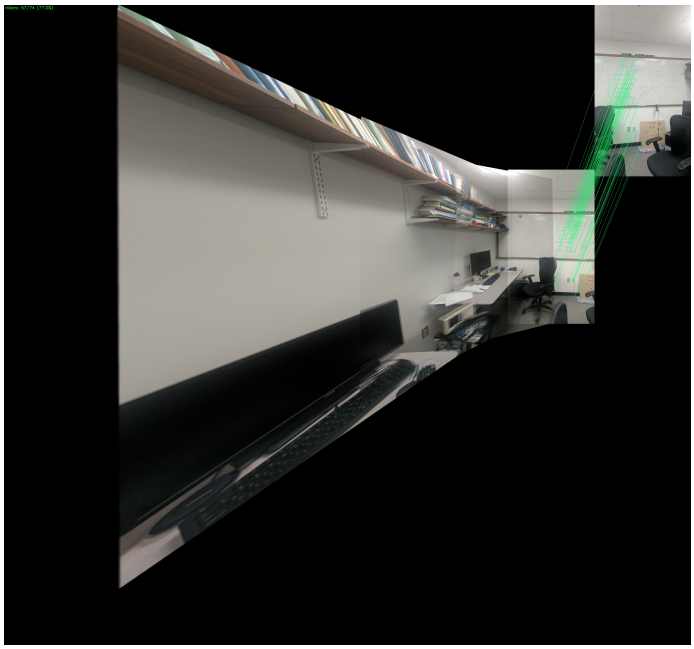

Fig. 11. Output of Test Set 2

Fig. 12. Output of Test Set 3

stitching algorithm performed flawlessly, producing a smooth panorama as shown in Figure 12. However, complications developed in Test Set 4 because images 4 and 5 did not share features with images 1, 2, and 3. To overcome this, we implemented a method that bypassed images during the stitching process if RANSAC discovered fewer inliers than a predetermined threshold. This method worked well, and the program was able to successfully stitch all images up to image 3 (which had overlapping parts) as shown in Figure 13. These findings emphasize the need for enough feature overlap and strong handling of non-overlapping regions for successful panorama production.

II. Phase 2: Deep Learning Approach

A. Introduction

Traditional homography estimation approaches use a sequential pipeline of corner detection, feature extraction, match-

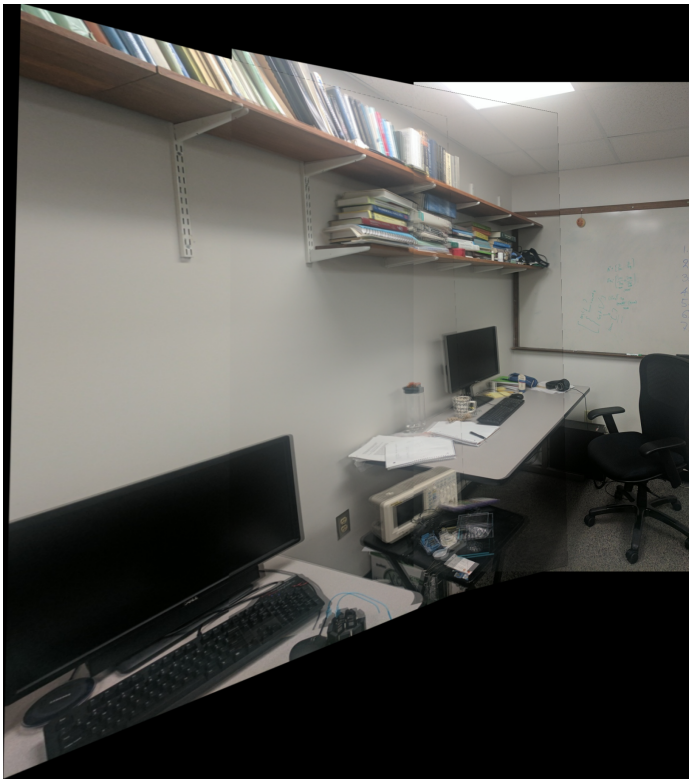

Fig. 13. Output of Test Set 4

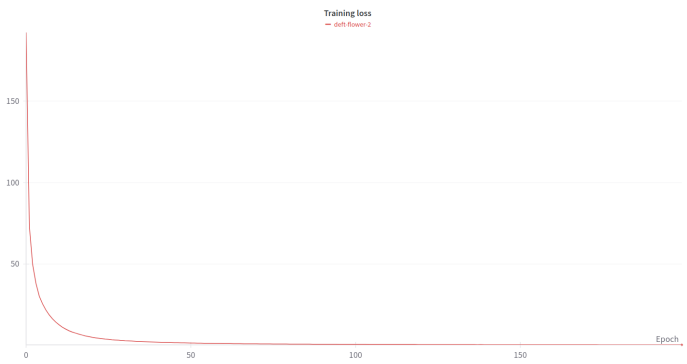

Fig. 14. Training Loss over Epochs for Supervised

ing, and RANSAC-based homography computation, which can be computationally demanding and error-prone at each stage. The introduction of deep learning methodologies has transformed this sector by proposing an end-to-end solution that combines these several phases into a single neural network design. This unified method has the potential to reduce computing cost while also introducing robustness through learned features and transformations, assuming the network has high generalization skills across a wide range of visual contexts.

B. Data Generation

The data generation pipeline for training a deep homography estimation network makes use of the MSCOCO dataset. The procedure starts with converting the incoming photographs to grayscale and scaling them to a standard dimension (320x240 pixels). To achieve correct transformations, multiple 128x128 pixel patches are extracted from each image while taking edge margins into account. The synthetic pair creation process begins with selecting a random patch with corners CA, followed by applying random perturbations to these corners within a range of [-32, 32] pixels to generate CB, resulting in a known geometric change. The Homography matrix H between these corner pairs is calculated using cv2.getPerspectiveTransform, and its inverse is utilized to warp the original image.

The final training sample includes the original patch (PA) and its warped counterpart (PB) stacked channel-by-channel, as well as the ground truth 4-point homography representation (H4pt), which is determined as the difference between CB and CA corners and corners CA. This method generates realistic training data with accurate ground truth labels and avoids artifacts or black patches in the modified images.

C. Supervised Learning

The implemented deep learning architecture is an improved version of the original HomographyNet, with a modified VGG-style network that has greater depth and complexity. The architecture is displayed in Figure 24. In comparison to the initial eight-layer architecture, the network now has twelve convolutional layers divided into six blocks, each of which contains two convolutional layers followed by batch normalization. Each convolutional block gradually increases feature channels (64→64→128→128→256→256) while maintaining spatial dimensions with proper padding, allowing for deeper feature extraction. The network uses Mean Squared Error (MSE) loss for regression of the 4-point homography representation (H4pt), and the dropout layers were deleted from the original architecture since they impeded convergence. This deeper architecture shows higher performance in homography estimation when compared to the original network, yielding more accurate transformation matrices and hence better stitching results. The training loss and validation loss over epochs are displayed in Figure 14, Figure 15 respectively.

Hyperparameters-

Optimizer - Adam
Learning Rate - 0.0005
Number of Epochs - 200
Scheduler Gamma - 0.99
Step size - 1000
Batch Size -32

D. Unsupervised Learning

The unsupervised deep Homography estimation method employs a self-supervised learning paradigm that avoids the requirement for explicit ground truth labels through the use of a photometric loss function. The architecture retains the deep convolutional neural network structure of the supervised version, but adds two critical components: a TensorDLT layer and a Spatial Transformer Network (STN). The training approach

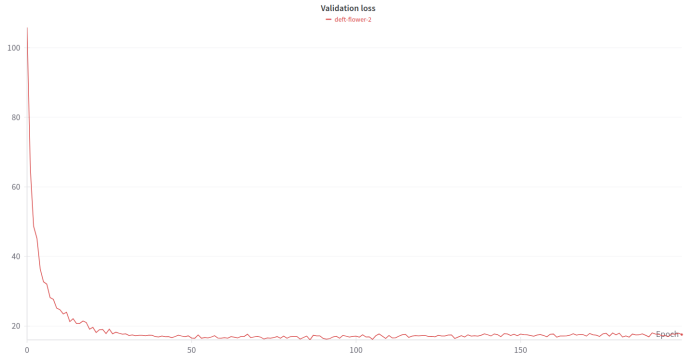

Fig. 15. Validation Loss over Epochs for Supervised

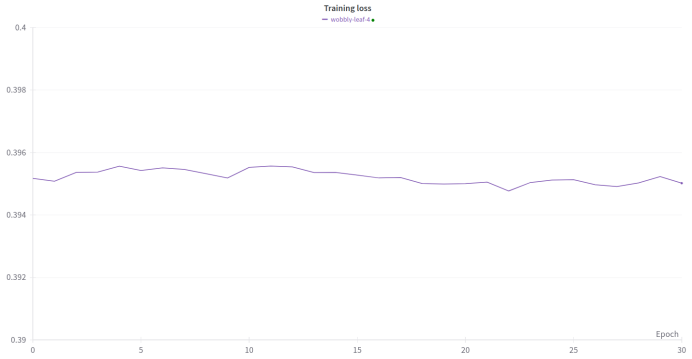

Fig. 16. Training Loss over Epochs for Unsupervised

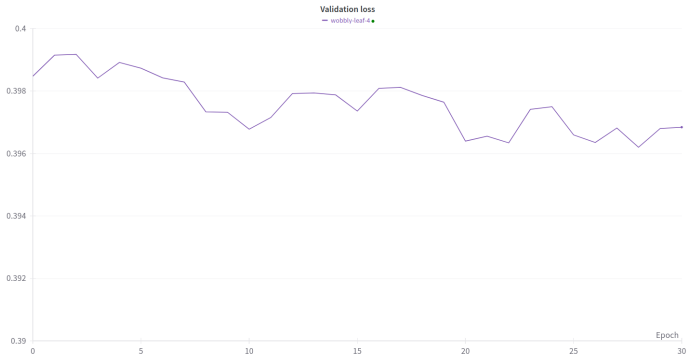

Fig. 17. Validation Loss over Epochs for Unsupervised

starts with predicting the 4-point Homography (H4pt) using the model. This is then translated to a full 3×3 Homography matrix through the TensorDLT layer using patch PA's corner points.

The method uses custom stn function to create a differentiable warping layer that transforms the input patch using the expected Homography. The network learns by minimizing the L1 photometric loss between the warped (predicted) patch PB and the target patch PB, which is calculated after carefully cropping the warped image using the corner coordinates. This approach allows the network to learn geometric transformations directly from image appearances, without the need for explicit supervision, while retaining the supervised architecture's powerful feature extraction capabilities.

The unsupervised implementation had inadequate convergence characteristics, with both training and validation losses showing insufficient decrease over the learning process. In order to improve the model's performance, we used transfer learning techniques, starting the network using pre-trained weights from our successful supervised Homography estimation model. Despite this method, which normally aids in bootstrapping the learning process by utilizing previously acquired feature representations, the unsupervised model demonstrated only minor improvement in performance metrics. This unexpected behavior indicates potential challenges in the unsupervised learning framework, possibly due to the complexity of the photometric loss landscape or the sensitivity of the differentiable warping operations, necessitating further research into alternative loss formulations or architectural modifications for improved unsupervised Homography estimation. The training loss and validation loss over epochs are displayed in Figure 16, Figure 18.

E. Image Stitching

The image stitching pipeline starts with feature detection and matching using the Scale-Invariant Feature Transform (SIFT) technique. The technique starts by transforming input photos to grayscale and then using SIFT to find keypoints and compute their descriptors. A FLANN (Fast Library for Approximate Nearest Neighbors)-based matcher uses k-nearest neighbor matching with a ratio test threshold of 0.6 to filter out high-quality matches. The technique then generates equivalent 128x128 pixel patches around the best-matched keypoints, which serve as inputs to the deep learning model.

In the Homography estimation phase, a deep neural network predicts the four-point Homography (H4pt) between the matched patches. The network transforms the concatenated patches into a normalized tensor. The H4pt cannot be used directly to estimate Homography between two patches; instead, it must first be converted to a Homography matrix. The method uses OpenCV's findHomography function with corner correspondences. Starting with corner points CA and projected H4pt, the target corner points CB are calculated by adding H4pt to $CA(CB = H4pt + CA)$. These associated point pairs are then used with cv2.findHomography() to generate the final Homography matrix. The selection of OpenCV's findHomography over DLT was deliberate, as DLT demonstrated vulnerability to numerical instability and scaling.

The final stitching procedures use perspective transformation and image blending algorithms. The program initially calculates the output canvas size by converting the corners of the first image with the computed Homography and identifying the enclosing rectangle. A translation matrix is used to verify that all converted coordinates are positive. The technique then warps the first image with the final homography matrix and generates a binary mask for blending. The second image is placed in its allocated area, and the final composite is formed

Fig. 18. Comparison of Ground Truth, Classical, Supervised, Unsupervised Approach

by combining both images with the binary mask, ensuring proper alignment and integration of overlapping parts.

F. Results and Conclusion

The experimental results demonstrated significant differences in performance between supervised and unsupervised stitching procedures. The supervised model displayed acceptable stitching skills for smaller datasets, but ran into substantial restrictions with bigger image sets, owing to memory constraints and cumulative mistakes in homography matrix calculations after 6-7 images. In contrast, the unsupervised strategy originally appeared promising, but detailed investigation revealed that its apparent effectiveness was primarily due to correct translation changes, with no suitable accounting for rotational fluctuations. This shortcoming was largely disguised by the test dataset's inherent properties, notably the small rotational variations between consecutive frames due to consistent camera movement.

The Ground Truth (Red), Classical Approach (Yellow), Supervised (Green), Unsupervised approach (Blue) results are displayed in Figure 18. The EPE values for these four images are displayed in Table I.

Test Image	Supervised EPE	Unsupervised EPE	Classical EPE
1	1.93	1.31	26.28
2	3.39	3.97	33.18
3	6.15	6.57	41.73
4	2.92	3.40	30.19

TABLE I
COMPARISON OF END POINT ERROR (EPE) ACROSS DIFFERENT APPROACHES

The Test Set panorama stitching output for Supervised Learning method are displayed in Figure 19, Figure 20, Figure **??**, Figure **??** respectively.

The Test Set panorama stitching output for Unsupervised Learning method are displayed in Figure 21, Figure 22 and Figure 23 respectively. [h]

Fig. 19. Tower Panorama Supervised

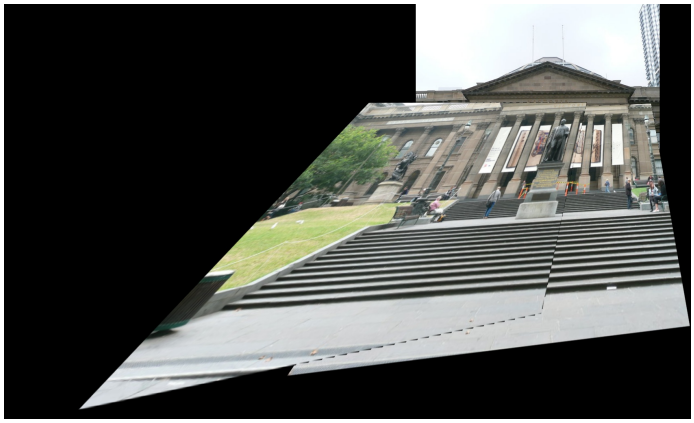

Fig. 20. Our Panorama

III. CHECKPOINTS

You can download the checkpoints folder from this link. Checkpoints Link

REFERENCES

[1] Brown, M., & Lowe, D. G. (2007). *Automatic Panoramic Image Stitching using Invariant Features*. International Journal of Computer Vision, 74(1), 59-73.
[2] Shi, J., & Tomasi, C. (1994). *Good Features to Track*. IEEE Conference on Computer Vision and Pattern Recognition.
[3] Lowe, D. G. (2004). *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision.
[4] OpenCV Documentation: Feature Detection and Description. *cv2.goodFeaturesToTrack()*. OpenCV 4.x Documentation.
[5] OpenCV Documentation: Geometric Image Transformations. *cv2.getPerspectiveTransform() and cv2.warpPerspective()*. OpenCV 4.x Documentation.
[6] OpenCV Documentation: Drawing Functions. *cv2.drawMatches()*. OpenCV 4.x Documentation.
[7] Hartley, R., & Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Second Edition.

Model Type	Testing Loss	Average EPE Loss
Supervised	3.28	4.11
Unsupervised	0.35	4.07

TABLE II
COMPARISON OF TESTING AND AVERAGE EPE LOSS FOR SUPERVISED AND UNSUPERVISED MODELS

Fig. 21. Tower Panorama Unsupervised

Fig. 22. Trees Panorama Unsupervised

Fig. 23. Unity Hall Panorama Unsupervised

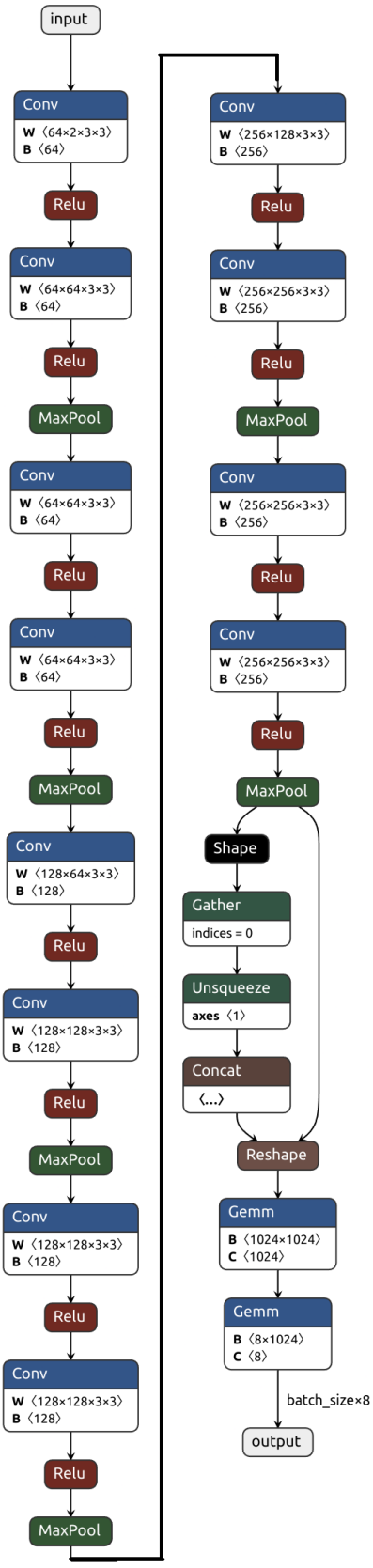

Fig. 24. Modified HomographyNet Architecture.