P1 : MyAutoPano Phase 2 (Using one late day)

1st Venkateshkrishna
Masters in Robotics
Worcester Polytechnic Institute
Worcester, MA 01609
vparsuram@wpi.edu

2nd Mayank, Bansal
Masters in Robotics
Worcester Polytechnic Institute
Worcester, MA 01609
mbansal1@wpi.edu

***Abstract*—This study introduces a neural network trained in both supervised and unsupervised manners to estimate the homography from pairs of grayscale images. The network outputs shifts in corner points that are used to compute the homography matrix, demonstrating the efficacy of deep learning for spatial transformation tasks in computer vision.**

I. Introduction

The estimation of homography is a foundational component in computer vision with relevance to tasks like image alignment and panoramic stitching. Conventional techniques, reliant on feature matching, are often limited by environmental and object variability. This paper introduces a deep learning approach, utilizing a neural network to infer homography from grayscale image pairs. We evaluate the network under supervised and unsupervised training regimes, demonstrating its capability to accurately predict corner point displacements for homography matrix computation. Our findings highlight the effectiveness of deep learning in overcoming the challenges faced by traditional methods, offering a promising direction for advanced homography estimation.

II. Methodology

The following subsections include:

1) Data Generation
2) Supervised Approach
3) Unsupervised Approach
4) Stitching Images

A. Data Generation

To train the supervised model, a dataset of image pairs with precisely known homographies is essential. Acquiring a vast collection of such images poses a considerable challenge, and verifying their exact homography is even more arduous. Therefore, we have created a synthetic dataset derived from the MS COCO dataset through the following four-step procedure:

1) Initiate by randomly cropping a patch P_A with a set of four corners C_A. These corners are then slightly adjusted to obtain a new set C_B. The initial patch size is set to 128×128, and the corner perturbation is applied using a random noise within the range $[-\rho, \rho]$, where $\rho = 32$.
2) Calculate the homography matrix $\mathbf{H}$ linking C_A and C_B via the OpenCV function `cv2.getPerspectiveTransform()`. This establishes a correspondence such that $\mathbf{c_B} = \mathbf{H}\mathbf{c_A}$ for each paired corner position.
3) Apply the inverse homography matrix $\mathbf{H}^{-1}$ to warp the image and subsequently crop it using the set C_A. This results in a new patch P_B.
4) Generate the training data labels $\mathbf{H}_{4points}$ as the difference $C_B - C_A$.

By executing the data generation function with any image from the MS COCO dataset as input, we obtain two patches along with $\mathbf{H}_{4points}$ to serve as training data. We generate 4 patched from each training image. Fig. 1 shows the input set of patches generated for the network to train on.

B. Supervised Approach

The neural network's architecture is identical to the VGG structure referenced in [1] and depicted in Fig. 13. Each convolutional layer is succeeded by a BatchNorm2D layer and then a ReLU layer. We have also put two dropout layer in the network. One after the final convolution layer with a dropout value of 0.4, second one is after the first fully connected layer with a dropout value of 0.4. The output layer consists of eight linear units that yield the estimated $\mathbf{H}_{4points}$. The architecture can be seen in fig. 2. We define the loss function as the L2 norm of the difference between the network's predicted $\hat{\mathbf{H}}_{4points}$ and the actual ground truth $\mathbf{H}_{4points}$:

$$L2\text{-}loss = ||\hat{\mathbf{H}}_{4points} - \mathbf{H}_{4points}||^2 \tag{1}$$

A minibatch size of 64 is utilized along with the SGD optimizer set at a learning rate of 0.005 and momentum of 0.9. The learning rate is scheduled to decrease by a factor of 10 after every 30000 iterations, throughout the course of 100 training epochs. Each epoch consists of 20000 images synthesized from the MS COCO dataset, by varying the perturbations acting as a data augmentation technique.

Fig. 3 shows the training loss per batch for every epoch and fig. 4 shows the validation loss per batch for every epoch.

Fig. 5 shows the result of supervised approach on one set of training images.

C. Unsupervised Approach

For unsupervised learning, the network's structure is identical to that utilized in the supervised methodology, eliminating the need for actual $\mathbf{H}_{4points}$. The loss is instead evaluated

Fig. 1: Input set of patches generated as dataset

through the photometric error between the warped input patch P_B and the neural network's output patch $f(P_A)$, which is processed using the predicted $\hat{\mathbf{H}}_{4points}$:

$$L1\text{-}loss = |f(P_A) - P_B| \tag{2}$$

The unsupervised technique includes two novel stages. Initially, the tensor DLT process converts $\hat{\mathbf{H}}_{4points}$ into the homography matrix $\mathbf{H}$, as described in [2]. Unlike the $cv2.getPerspectiveTransform()$ function, this tensor-based method allows for gradient tracking essential for learning. Subsequently, the input patch P_A is warped to generate $f(P_A)$, utilizing `kornia.geometry.transform.HomographyWarper()` rather than a spatial transformer network.

Training for the unsupervised model follows parameters akin to the supervised approach and it was trained for 100 Epochs as well.

D. Stitching Images

In the supervised approach, once we obtain the network-generated $\hat{\mathbf{H}}_{4points}$, we stitch them using the approach fol-

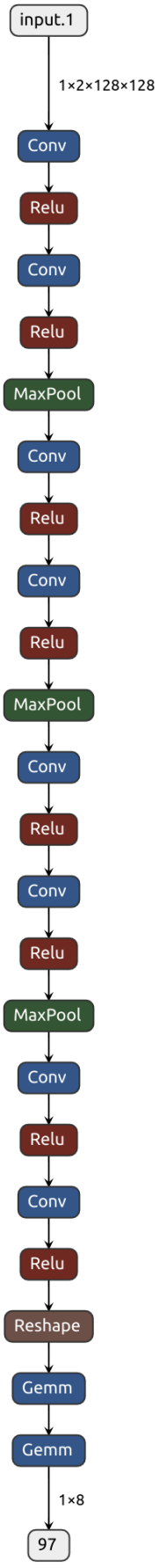

Fig. 2: Model Architecture

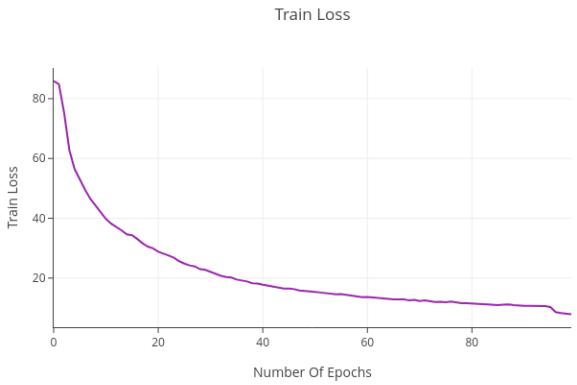

Fig. 3: Train Loss

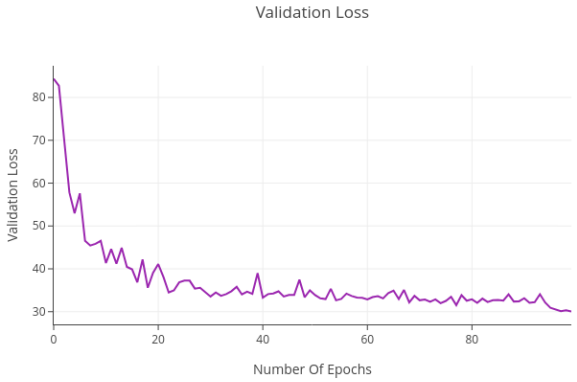

Fig. 4: Validation Loss

lowed in phase 1 of this project.

1) Determine the homography between the first and second images.
2) Apply the homography to the four corner points of the first image to ascertain the necessary translations.
3) Calculate the minimum and maximum x and y translations for the image.
4) Formulate a homography matrix specific to the translation and integrate it with the initial homography.
5) Warp the first image using this refined homography and then stitch it with the second image to create the panorama.

Fig. 5: Result of supervised network after stitching

Fig. 6: Final stitched image for test image 1

Fig. 7: Final stitched image for test image 2

In the unsupervised approach, since the model is the same, we use the same approach to get the homography and stitch the images together.

III. Results

Fig. 6 and Fig. 7 shows the result of stitching using the supervised network for the Test set. We can see that the network correctly estimates the homography and stitches both the patches. We can see that the bottom part of knife is added in the stitched image(fig. 6) and the top part of the seat and the side wall is added in the stitched image(fig. 7). Fig. 8 shows the output bounding boxes for the supervised and unsupervised networks respectively. So does fig. 9 for another test image. The white box is the output of network and black box is the ground truth. We can see that the predicted corners of patch B resembles very closely to the ground truth, showing that the network learned well. Table 1 shows the EPE error and runtime for the two networks over the different datasets. We can see that the unsupervised network did not perform as well as the supervised network. The performance can be improved by training the network for a longer time and by tuning the model parameters.

Fig. 8: Output bounding box for the two networks for test image

Fig. 9: Output bounding box for the two networks for test image

	EPE (pixel)	Run time (ms)
Supervised - Train	8.4	0.4
Supervised - Val	11.7	0.4
Supervised - Test	12.1	0.4
Unsupervised - Train	34.1	0.4
Unsupervised - Val	38.9	0.4
Unsupervised - Test	38.3	0.4

TABLE I: Evaluation metrics for Supervised and Unsupervised models for the different datasets

REFERENCES

[1] My AutoPano: link