5-2021

Matching As Color Imaging: Thermal Feature Detection

Bhavesh Deshpande
bd7491@rit.edu

Matching As Color Imaging: Thermal Feature Detection

by

Bhavesh Deshpande

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering

Supervised by

Dr. Guoyu Lu
Department of Chester F. Carlson Center for Imaging Science
Kate Gleason College of Engineering
Rochester Institute of Technology. Rochester, New York
May 2021

Approved by:

Dr. Guoyu Lu , Professor
Thesis Advisor, Department of Chester F. Carlson Center for Imaging Science

---

Dr. Majid Rabbani , Professor
Committee Member, Department of Electrical and Microelectronic Engineering

---

Dr. Jamison Heard , Professor
Committee Member, Department of Electrical and Microelectronic Engineering

---

Dr. Ferat Sahin, Professor,
Department Head, Department of Electrical and Microelectronic Engineering

Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title:

Matching As Color Imaging: Thermal Feature Detection

I, Bhavesh Deshpande, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

Bhavesh Deshpande

Date

Dedication

I dedicate this master's thesis to my family - Dilip, Dipika, Sunanda and Sheetal who have been a constant source of motivation and support to me.

Acknowledgments

I would like to thank and express my sincere gratitude to Dr. Guoyu Lu for providing me with the opportunity to work with the Intelligent Vision and Sensing (IVS) group at Chester F. Carlson College of Imaging Science. Prof. Lu has guided me ever since I started working under him when I took an Independent study course. It was then I was introduced to the IVS group and then got to know the projects they are working on. He not only backed me up through my master's endeavors, especially when the pandemic of COVID struck, a difficult time for everyone, but also he thought me an important lesson of helping people in need by being the moral support needed. I have also learnt interpersonal skills from working closely with him and have admired his never give up attitude. The courses which I took under him have helped in creating a strong founding platform for my career. I am grateful to have an opportunity to work with him.

I would like to thank Dr. Majid Rabbani and Dr. Jamison heard, my thesis committee members, for their guidance and suggestions through the course of my master's thesis. Dr. Rabbani has been a great mentor since I joined for my master's program. I would also like to thank all the Electrical Engineering professors and staff members for the knowledge I gained by taking the courses.

I take this opportunity to thank my lab mates Sourabh Hanamsheth and Yawen Lu for helping me out not only when I was stuck on any problem but also by providing all the

necessary resources needed. I would like to thank my classmates for a fun and engaging master's journey. Lastly I would like to thank my parents and family members for being a constant source of motivation and helping me achieve my dreams.

Abstract

Matching As Color Imaging: Thermal Feature Detection

Bhavesh Deshpande

Supervising Professor: Dr. Guoyu Lu

Feature detection and extraction is considered to be one of the most important aspects when it comes to any computer vision application, especially the autonomous driving field that is highly dependent on it. Thermal imaging is less explored in the field of autonomous driving mainly due to the high cost of the cameras and inferior techniques available for detection. Due to advances in technology the former is not a major limitation and there lies tremendous scope for improvement in the latter. Autonomous driving relies heavily on multiple and sometimes redundant sensors, for which thermal sensors are a preferred addition. Thermal sensors being completely dependent on the infrared radiation emitted are able to frame and recognize objects even in the complete absence of light. However, detecting features persistently through subsequent frames is a difficult task due to the lack of textures in thermal images. Motivated by this challenge, we propose a Triplet based Siamese Convolutional Neural Network for feature detection and extraction for any given thermal image. Our architecture is able to detect larger number of good feature points on thermal images than other best performed feature detection algorithms with superb matching performance based on our extracted descriptors. To demonstrate our aforementioned claim, we compare the performance of the proposed CNN scheme with traditional as well as state-of-the-art

feature detection and extraction schemes. Future work involves extending the pipeline for motion tracking, SLAM, SFM and many other applications.

List of Contributions

- Sourabh Hanamsheth
- Yawen Lu
- Dr. Guoyu Lu
- **Publication: ICASSP**

  MATCHING AS COLOR IMAGES: THERMAL IMAGE LOCAL FEATURE DETECTION AND DESCRIPTION

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Extracting features on visible images is well established. Features such as SIFT [1], SURF [2], and ORB [3] provide us with good and distinct local image descriptions. There are also several other well-defined edge and corner detection algorithms such as Canny edge detector [13], Harris corner detector [12] etc. These detectors form a core part for SFM and SLAM [8][9] methods. However, these same techniques fail to yield good quality feature detection results when used on thermal images. For the case of autonomous driving or navigation in general, visible images should be sufficient for most of the time, but in conditions of low to no visibility caused by the environment such as rain, fog, snow, night etc., even a good quality high resolution camera is not sufficient. On the other hand, thermal imaging is robust to all these aforementioned conditions and provides us with high quality information of the scene. Nevertheless, there are certain drawbacks when it comes to thermal imaging - the high texture quality observed in visible images is lost in case of thermal images, which also forms the basis for standard feature detection and extraction algorithms. Thermographic cameras detect infrared radiation emitted from the body of an object making it possible to have good visibility even in the absence of illumination. Warm blooded animals or object thus become easy to detect making thermal imaging useful in

military and surveillance applications and has a great potential in the field of autonomous driving as well.

In the deep learning era, obtaining unique, better and more accurate features from images has improved immensely. But almost all the approaches based on images from the visible spectrum which are rich in feature textures failing to completely tackle the aforementioned problems. BRISK [36], SIFT [1], SURF [2], ORB [3], FAST [4] and WADE [11] algorithms are able to detect some features in thermal images, but the quality of detection is rather unsatisfactory, especially for those raw thermal images without further post-processing.

Image-to-image translation using Adversarial networks for thermal-visible domain transfer can be used as a workaround, but the generated images are based on assumptions and also may add additional artifacts which are not desirable.

In this paper, we introduce a feature detection and extraction architecture based of Triplet neural network. For detection network, it takes in three image patches namely Anchor, Positive and Negative. These patches are passed through a series of convolutional and fully connected layers.

The network attempts to project patches with similar embedding together and dissimilar embeddings further away from each other thereby enforcing the distance loss for similar patches to a minimum and a high value for dissimilar patches. The resulting vector is kept being 128-dimensional helping the network retain valuable feature information from the patches. Feature detection and description could be considered as a single step where midpoints for the image patch with a high feature embedding response is saved for feature

matching purpose. However, in order to acquire the high feature response, we introduce an intermediate step by adding fully connected layers to previously trained model weight values. The network is trained on $32 * 32$ patches corresponding to features obtained from KAIST [32] and CSS [10] datasets. The network is able to learn high-quality feature descriptor for given patches and then classify good distinct features which can be identified very accurately through subsequent frames.

To summarize, the main contributions of our work are as follows:

1. I propose a novel Triplet based network to train robust feature detection on thermal scenes.

2. I propose a patch-based feature extraction network to learn 128-dimensional descriptor vectors to overcome the texture and context limitation of the thermal scenes.

3. I integrate both proposed detection and extraction networks into a full pipeline to enable stable and reliable feature matching on thermal images.

4. The proposed method achieves a superior performance in visual and quantitative comparison compared with other widely used classical and most recent deep learning algorithms.

Chapter 2

Related Work

A feature in computer vision and image processing can be defined as a piece of information about the contents in an image determining whether a region has certain properties in them. Typically features could be considered as specific structures in an image such as points, edges, corners, or objects. Features can also be related to motion in image sequence, or shapes defined in terms of curves or boundaries. They can also be a result of neighborhood operation on an image. Typically, the detected features from images are desired to be invariant to changes to be able to detect the same feature over several image frames making it more robust to changes. The changes may include, Translation, Euclidean, Similarity, Affine or Projective transform carried out on an image.

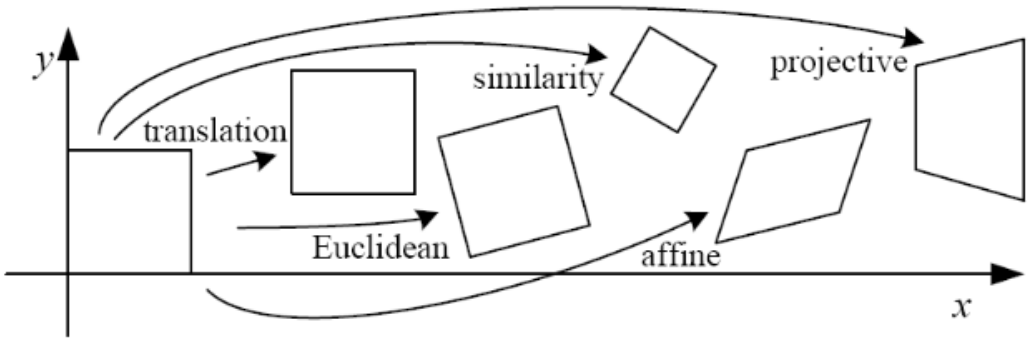

Figure 2.1: Similar feature detected after various transformation performed on an image.

With no exact definition of what represent a feature, we could define it based on the

application being performed. We could say a feature is an interesting region in an image and is a starting point for many image processing and computer vision applications both 2D and 3D.

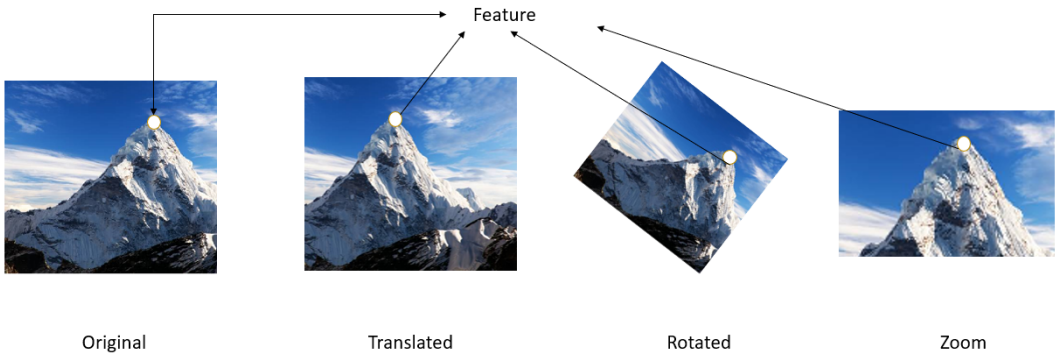

Figure 2.2: Operation performed on images.

2.1 Classic feature detection and extraction methods

Keypoint detection and feature point matching using Canny [13] for edge, Harris [12] for corner detection and Histogram of Oriented Gradients (HOG) [14] had been typically done in the past and were used in applications such as recognition and image matching. Later SIFT was introduced, and due to its high robustness SIFT enjoys the benefits of high accuracy irrespective of the image scale, orientation, or rotation. SIFT localized and learnt good features using Difference of Gaussian (DoG) at multiple scales which made it more popular among the existing feature detection algorithms with the drawback being its higher computing time. In the later years, faster implementations similar to SIFT were introduced namely Features from Accelerated Segment Test (FAST), Binary Robust Independent Elementary Feature (BRIEF), Oriented FAST and Rotated BRIEF (ORB), Speeded up Robust Feature (SURF) which had their share of advantages and disadvantages compared to the

SIFT extractor.

2.1.1 SIFT

Scale Invariant Feature Transform as mentioned earlier is one of the popular feature detection and extraction techniques available which stood the test of time. As the name suggests the detected features are robust to change in scale, rotation, translation, or any other affine transform. For feature detection SIFT implements Difference of Gaussian (DoG) which approximates Laplacian of Gaussian (LoG) mainly because it is a costly process. Difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different Gaussian kernels. The process is carried out for different octaves of the image in the Gaussian Pyramid which is represented in the figure 2.3:

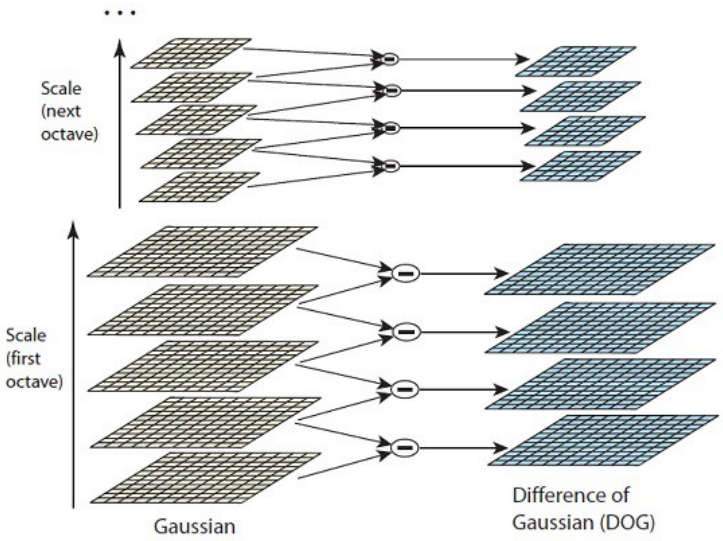

Figure 2.3: Difference of Gaussian operation performed on different octaves of an image.

After Difference of Gaussian, images are searched for local extrema over scale and space i.e., a pixel location is compared with its 8 neighboring pixels as well as with 9 pixels in the previous and next scales. If the pixel is an extremum, it is a potential keypoint. A Taylor

series expansion of scale space is used to obtain more accurate locations of the previously found potential keypoints which is done based on a threshold value (0.03). Further removal of low-contrast and edge keypoints is also carried out to refine the detection. An orientation is assigned to each keypoint by taking a neighborhood around it depending on the scale, where the gradient magnitude and direction are calculated in that region. From that region an orientation histogram is created.

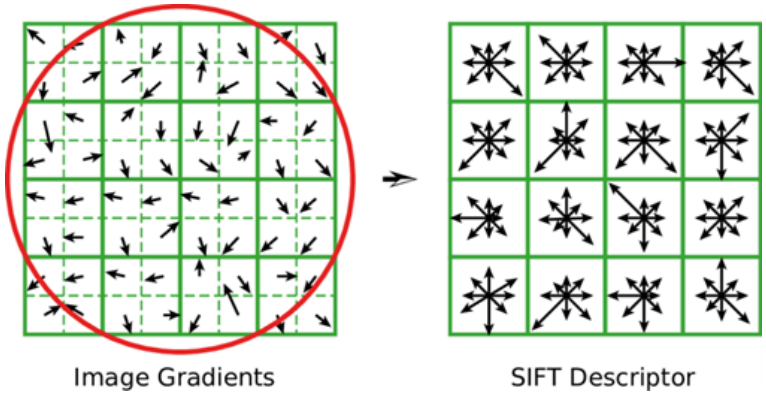

Figure 2.4: Histogram of gradients in $16 * 16$ block around the keypoint

For keypoint description, a $16 * 16$ neighborhood around the keypoint is divided into 16 sub-blocks of $4 * 4$ size and for each sub-block an 8-bin orientation histogram is created resulting in a 128-bin valued vector.

2.1.2 SURF

Speeded Up Robust Features as the name suggests is a fast and robust algorithm for feature detection and extraction. It implements box filtering approach enabling real-time applications like object recognition and tracking. For feature extraction SURF detects keypoints using Hessian approximation. Scale and location are selected using determinant of Hessian matrix. Given a pixel, the Hessian of pixel is:

$$H\left(f\left(x,y\right)\right)=\begin{bmatrix}\frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y}\\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2}\end{bmatrix} \tag{2.1}$$

The image is filtered by a Gaussian kernel to adapt to any scale. The Hessian matrix H(x, σ) is defined as:

$$H\left(x,\sigma\right)=\begin{bmatrix}L_{xx}\left(x,\sigma\right) & L_{xy}\left(x,\sigma\right)\\ L_{xy}\left(x,\sigma\right) & L_{yy}\left(x,\sigma\right)\end{bmatrix} \tag{2.2}$$

To further reduce the computational cost an integral image approach is carried out. Reproducible orientation for the interest points is achieved by calculating Haar-wavelet responses in x, y directions in a circular neighborhood around the keypoint. Sum of vertical and horizontal wavelet responses are calculated till the orientation with largest sum is found to denote the main orientation of the feature. Descriptor values are obtained by taking a rectangular region around the previously found keypoints and a $64 * 64$ length vector is obtained from it.

2.1.3 ORB

Oriented FAST and Rotated BRIEF combines the FAST keypoint detector and the BRIEF descriptor. Keypoints are detected using FAST algorithm, followed using Harris corner detection to find top N points. FAST computes the intensity weighted centroid of the patch

with corner at its center and moments are computed for rotation invariance (Fig. 2.5). A rotation matrix from the orientation of the patch is found and then BRIEF descriptor for the same is computed.

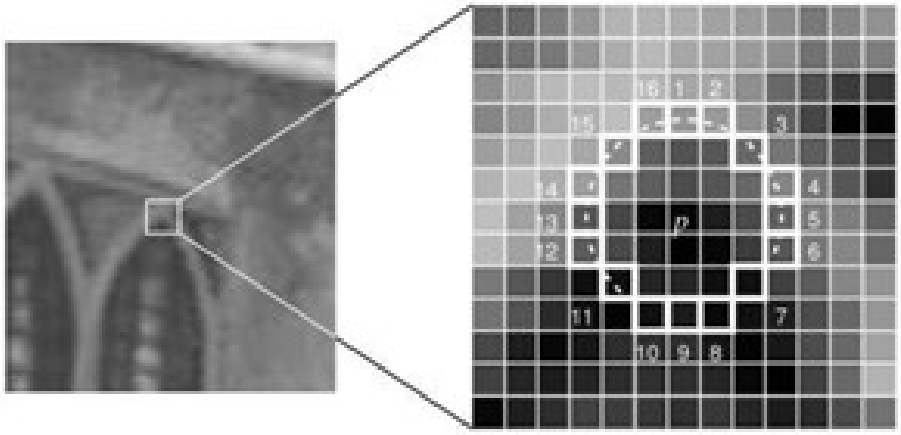

Figure 2.5: Patch corresponding to ORB feature detected.

2.2 Deep learning-based detection and extraction methods

With the increase in popularity of deep learning-based methods, focus was shifted towards learning based feature detection and description. Patch based feature descriptor learning has been also implemented using Siamese network however most of the work has been done in the RGB domain. Faiz et al. [5] depicted a Siamese network trained for detection of change in satellite imagery with the network architecture containing two VGG16 [17] networks. PN-Net [24] took a Triplet based approach to generate descriptors which could be used in traditional matching setup. In contrast to Hinge Embedding loss [25, 28], they introduced SoftPN loss where the pairs of patches represented a soft negative mining. Another Siamese network L2-Net [6] specifically trained for descriptor learning from patches in Euclidean space showed state of the art performance. They had an all-convolutional

structure with a stride of 2 to achieve down sampling, and a loss function having three error terms. Their loss function included three error terms for obtaining descriptor similarity, descriptor compactness and one to obtain intermediate feature maps. SuperPoint used an encoder-decoder based approach having a shared encoder and two different decoders for description and detection of features. Having a good performance, it was limited to RGB images and failed to produce comparable results for thermal images. All the above-mentioned methods can be used for feature descriptor extraction but only on RGB/grayscale images. Our model can be considered as the first patch-based descriptor learning scheme designed for more challenging thermal image data.

2.2.1 PN-NET

PN-Net is a patch-based feature descriptor learning method where the goal is to compute representation vector for image patch. The descriptor vector size matches the feature dimensionality which is obtained from the final layer of a convolutional neural network. Unlike traditional Siamese networks which accept two parallel inputs and share parameters across the network, PN-Net takes a triplet-based approach wherein the input to the network is triplet of patches. In their work they introduce a novel loss function called SoftPN Loss. Any training triplet includes two negative and one positive distance with the goal to converge the positive distance to 0 and the two negative distances to infinity. The SoftPN objective is shown in equation 2.3:

$$l(\tau) = \left[\left(\frac{e^{\Delta(p_1,p_2)}}{e^{min(\Delta(p_1,n),\Delta(p_2,n))} + e^{\Delta(p_1,p_2)}} \right)^2 + \right.$$

$$\left(\left(\frac{e^{min(\Delta(p_1,n),\Delta(p_2,n))}}{e^{min(\Delta(p_1,n),\Delta(p_2,n))} + e^{\Delta(p_1,p_2)}} - 1 \right)^2 . \right. \quad (2.3)$$

The network with the SoftPN loss achieves better performance compared to SoftMax and Hinge loss functions.

2.2.2 SuperPoint

SuperPoint is a self-supervised interest point detection and description framework. A fully connected neural network operating on full sized images producing interest point detection and fixed length description in a single forward pass. It is an encoder decoder architecture where the network consists of a single, shared encoder to reduce the input image dimensionality. Following the encoder there are two separate decoders each learns task specific weights, one for interest point detection and other for description. The SuperPoint uses a VGG style encoding consisting of convolutional layers, spatial downsampling and non-linear activation functions where pixels in lower dimensional output are referred as cells. For every pixel input, the output pixel corresponds to the probability of "point-ness". Descriptor learning is done in a semi-dense grid reducing the training memory. Later a bi-cubic interpolation is performed by the decoder network followed by L2-normalization. The Loss function is the sum of two intermediate losses, respectively for the interest point detector and descriptor. Both the losses are optimized simultaneously by using a pseudo-ground truth interest point location and ground truth corresponding to generated homography between two images. The loss function is given in equation 2.4:

$$L(X, X', D, D'; Y, Y', S) = L_p(X, Y) + L_p(X', Y') + \lambda L_d(D, D', S). \tag{2.4}$$

$$L_p(X, Y) = \frac{1}{H_c W_c} \sum_{h=1,w=1}^{H_c,W_c} l_p(x_{hw}; y_{hw}), \tag{2.5}$$

where,

$$l_p(x_{hw}; y) = -log(\frac{exp(x_{hwy})}{\sum_{k=1}^{65} exp(x_{hwk})}). \tag{2.6}$$

2.2.3 Siamese Neural Network

A Siamese neural network (sometimes called a twin neural network) is a type of convolutional neural network that uses the same weights while working in conjunction with two different input vectors to compute comparable output vectors (Fig. 2.6). Parameters are updated in a mirrored fashion across both sub-networks and the network is used to find the similarity of the inputs by comparing its feature vectors, thereby having a variety of applications.

2.2.4 SuperThermal

In this work [33], authors introduce a siamese based approach for feature detection and description on thermal images. They design a network to learn an efficient matching strategy

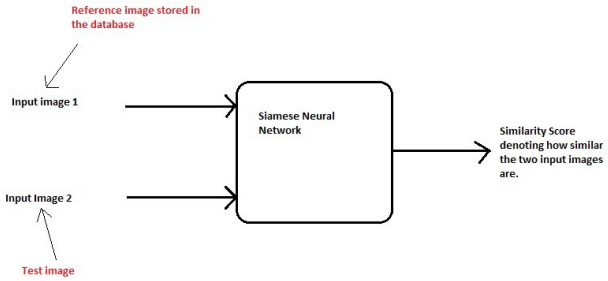

Figure 2.6: Siamese neural network.

with an aim of achieving a robust learning. In their work firstly they introduce an image enhancement module. This module is used to enhance the edge and other detail information in the thermal image since the images are of comparatively low resolution. Once enhanced the patches are fed into the network to effectively extract patches and learn detection and description. The learning is done on multi-scaled patch input information making the architecture and learning more robust. Further image pairs are generated and loss constrained is applied with the image canny detection map to better constrain on thermal images and have more candidates for better matching performance. Data augmentation is also applied to the input data to the network. The architecture pipeline is shown in Fig. 2.7.

2.2.5 Per chapter Synopsis

In Chapter 1, we introduce our work on thermal image feature detection and extraction using neural networks. We briefly describe the available methods for feature extraction and their limitations on the current task. We discuss the motivation for the work and the need for the same in various computer vision applications.

In Chapter 2, we describe in detail classical as well as learning-based feature extraction

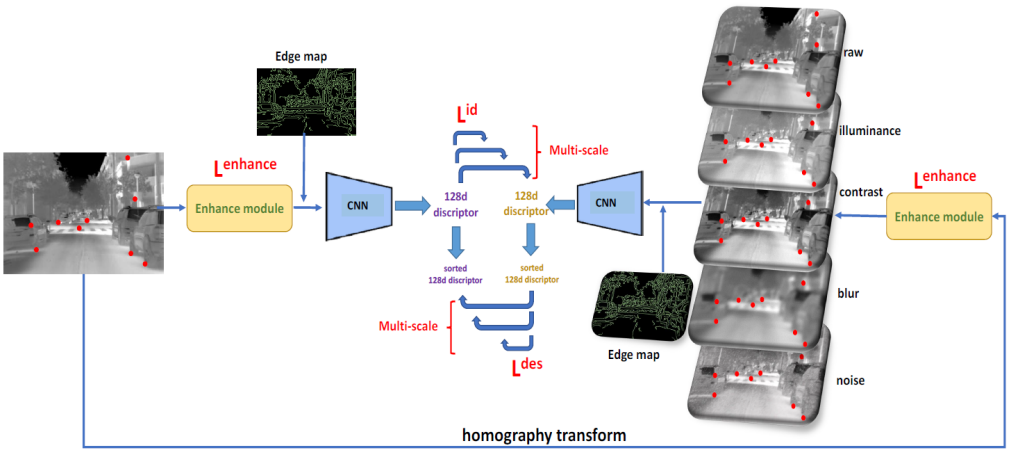

Figure 2.7: Overview of SuperThermal Architecture pipeline.

methods. We discuss the procedure used for obtaining different feature extraction and address the loss functions implemented for the learning-based extraction.

In Chapter 3, we introduce the concepts of 3D multivew geometry and develop a mathematical understanding of the traditional concepts. We discuss the concept of camera projection matrix, homography, Essentianl and Fundamental matrix for pose estimation.

In Chapter 4, we discuss the dataset we used and the specifications for individual camera for each dataset. We further introduce the different types of data augmentations applied to increase the size of the dataset.

In Chapter 5, we introduce our methodology and approach to solve the given task at hand. We discuss the various constraints we implemented from patch selection to the loss function. We also provide an insight on the hyper parameter settings implemented for our learning.

In Chapter 6 and 7, we summarize the results and discuss the experimentations we performed throughout the work. We further discuss the potential work for the future that

can be explored to improve upon the current performance.

Chapter 3

Multiview Geometry Concepts

3.1 Projective Transformation and Homography

Projective transform gives the translation between points in an image or planes in the scene and is one of the fundamental concept in multi view geometry. Homography is a projective transformation where a point in image coordinate x(u,v,1) is mapped onto a point x'(u',v'1).

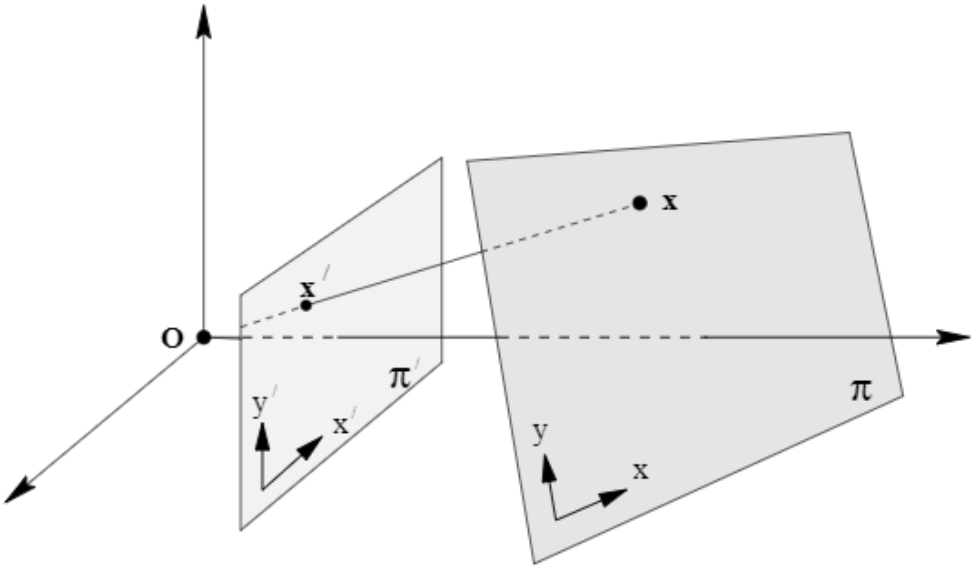

Figure 3.1: Demonstrates the homography relation between two planes.

This can be achieved by a $3 * 3$ matrix know as homographic matrix H given by the form shown in equation 3.1 below:

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \tag{3.1}$$

A point in x can be projected onto the other plane x' using the equation 3.2:

$$x' = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} H_{11}\ H_{12}\ H_{13} \\ H_{21}\ H_{22}\ H_{23} \\ H_{31}\ H_{32}\ H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{3.2}$$

Point x = [x_1,y_1,1] denotes a point at (x_1,y_1) location in its Homogeneous coordinate system. The point (x'$_2$,y'$_2$) in the coordinate system of the second plane are given as:

$$x'_2 = \frac{x'_1}{x'_3} = \frac{H_{11}x_1 + H_{12}x_2 + H_{13}}{H_{31}x_1 + H_{32}x_2 + H_{33}} \tag{3.3}$$

3.2 Essential and Fundamental matrix

3.2.1 Essential matrix

For a pair of stereo cameras viewing the same point X_i in the world space, a 3D point is projected onto two image planes x_i and x'$_i$ as shown in the figure below:

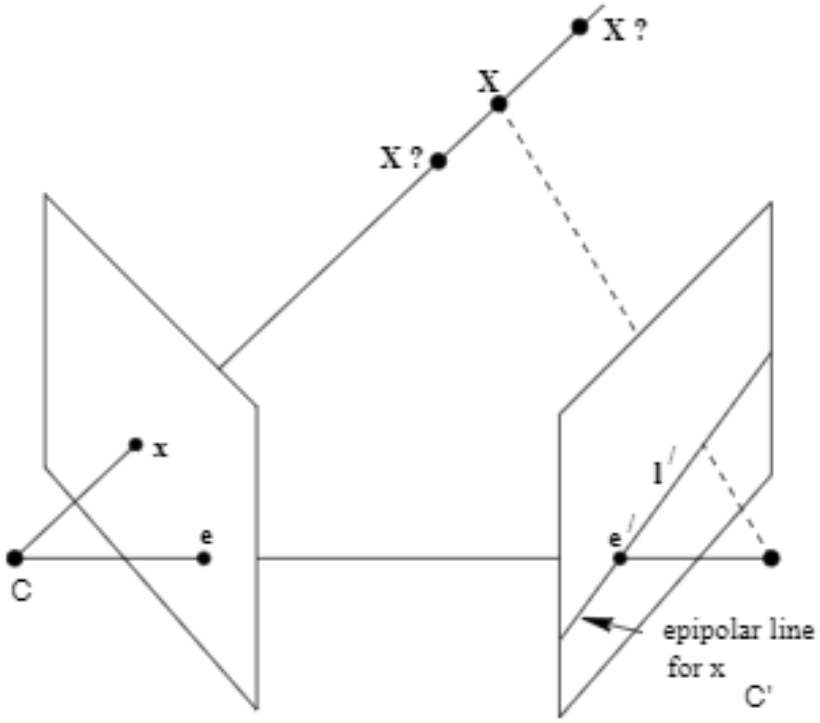

Figure 3.2: Projection of point onto a pair of cameras. Epipoles and epipolar line corresponding to the projection and the two camera centers.

For calibrated cameras, x_i and x'_i are normalised. If the camera matrices (P,P') are known then an estimate of X_i can be obtained by the process of triangulation. The estimated points satisfy the camera projections equations given below:

$$x_i = PX_i \; and \; x'_i = P'X_i \; for \; each \; point \; i \tag{3.4}$$

where P, P' are the estimated camera projections.

The true camera projections are related to the estimated projections by:

$$P = \widehat{P}H_{-1} \; and \; P' = \widehat{P}'H_{-1}. \tag{3.5}$$

where,

$$\widehat{P} = K\,[R|t] \tag{3.6}$$

$$\widehat{P}' = K'\,[R'|t'] \tag{3.7}$$

If the two cameras are calibrated i.e., we know the intrinsic matrix (K) parameters for both the cameras, then x_i and x_i' are given in normalized coordinates i.e, they differ from true points by scale, rotation and translation. Multiplying eq^n 3.4 by K^{-1} we get:

$$\widehat{x_i} = K^{-1}x_i = [R \mid t]\,\widehat{X_i} \tag{3.8}$$

$$\widehat{x_i'} = K'^{-1}x_i' = [R' \mid t']\,\widehat{X_i} \tag{3.9}$$

The transformation matrix corresponding to the points ($\hat{x}_i$, $\hat{x}'_i$) is called the Essential

matrix and is given as:

$$\widehat{x}_i^{'T} E \widehat{x}_i = 0. \tag{3.10}$$

3.2.2 Fundamental matrix

In the case of uncalibrated cameras where the intrinsic parameters of the camera are not determined, we could not multiply K^{-1} to the equation 3.4. This gives us an equation of the form:

$$x_i^{'T} F x_i = 0. \tag{3.11}$$

where F is the fundamental matrix.

From the above equation we can derive a relation between the Essential and Fundamental matrix as follows:

$$F = K^{'-T} E K^{-1} \Rightarrow E = K^{'T} F K. \tag{3.12}$$

3.3 Visual Odometry

In the field of robotics and computer vision, Visual Odometry is a process to estimate the orientation of the robot/vehicle and its position. This is achieved by extracting useful information from a pair of camera images or by using multiple frames of images from the same camera. This can be achieved by a feature based matching method. For stereo cameras, the relation between the two cameras can be found using the finding out point correspondences based on any type of feature detection algorithm. (Eg. section 2)

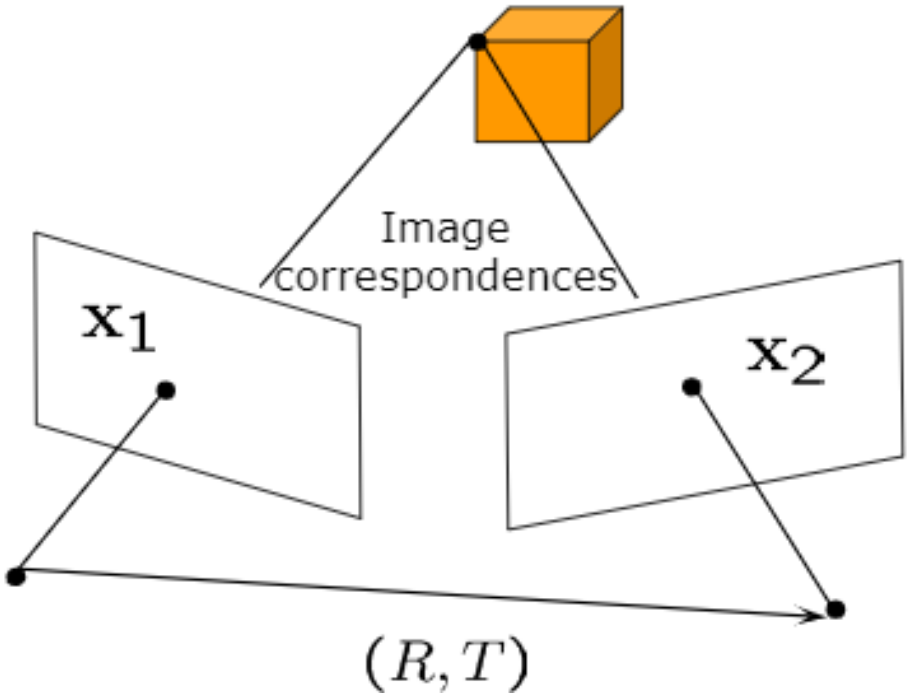

Figure 3.3: Triangulation performed between two cameras extracting information from the same point

The extracted feature points can be used to find the projection matrices for the two cameras and their relation between each other can be computed using the Fundamental matrix for uncalibrated and Essential matrix for calibrated cameras as discussed in the

previous section. Additionally Random Sample and Consensus (RANSAC) algorithm is performed to improve the accuracy of the estimated Fundamental/Essential matrix based on the number of in-liers. Once matrix between the camera pairs has been found, by the process of triangulation we can estimate the relative pose of the camera (in another word, camera motion) which in turn determines the rotation and orientation of the carrier with the camera mounted. There are additional edge cases where the projected points differ from their true locations. This is mitigated by calculating the re-projection error between the estimated and actual points and minimizing the error, which is called bundle adjustment.

Chapter 4

Datasets and Data Pre-Processing

4.1 KAIST Dataset

The KAIST (Fig. 4.1) a multispectral dataset was developed by Soonmin Hwang and his colleagues from Korea Advanced Institute of Science and Technology (KAIST). Multispectral ACF, an extension of aggregated channel feature (ACF) is introduced in the dataset. The hardware for imaging includes a color camera, a thermal camera, a beam splitter, and a three-axis camera. The PointGrey Flea3 global shuttered color camera and FLIR-A35 thermal camera are used. The color camera has $640 * 480$ spatial resolution with a $103.6°$ field of view while the thermal camera has $320 * 256$ spatial resolution with $39°$ field of view. Both the cameras have a frame rate of 20 fps. Both the cameras are mounted on a hardware and a translation is computed between them using stereo calibration. Further, color correction is also applied to the color camera due to uneven reflection ratios of the visible band from the beam splitter.

4.2 CSS Dataset

The dataset (Fig. 4.2) setting consists of a non-verged geometrical setting of pair of cameras: one camera in the visible spectrum and the other in the infrared. The visible camera is an ACE from Basler and has a resolution of $658 * 492$ pixels. The infrared camera is a Long

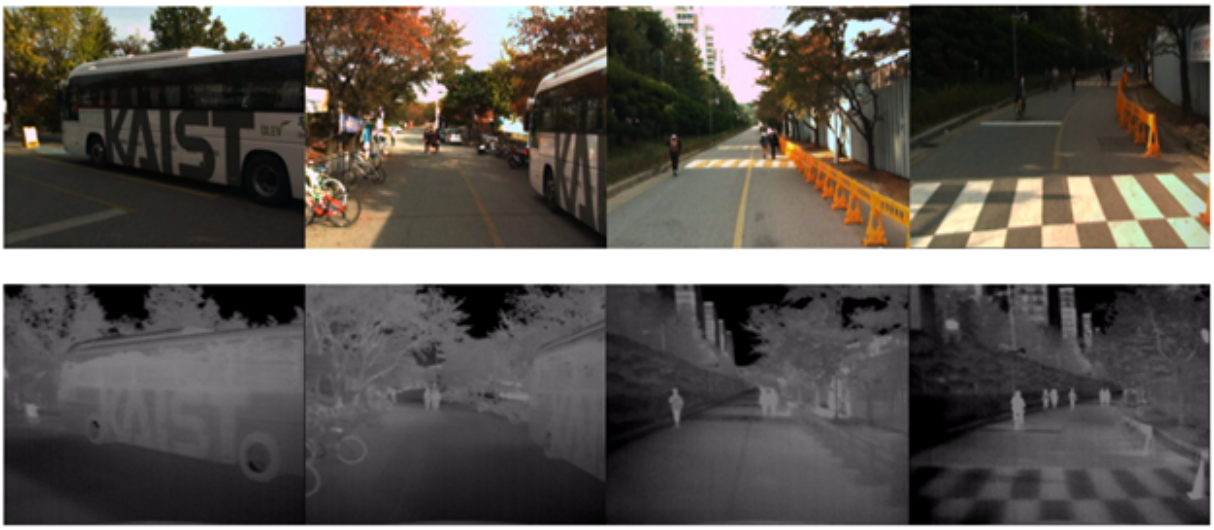

Figure 4.1: KAIST dataset: Top color images and bottom thermal infrared images

Wavelength Infrared (LWIR) device which detects radiation in the range of 8 - 14 µm. Both cameras are synchronized and calibrated. Different video sequence are obtained in urban and semi-urban scenarios.

Figure 4.2: CSS dataset: Top color images and bottom thermal infrared images

4.3 Data Preprocessing and Augmentation

The data from both KAIST and CSS datasets are series of consecutive frames. Learning in our network is performed from these consecutive frames, where there is not much changes between the frames. This results in overfitting of the model to specific patches and false learning. To compensate for this issue the dataset is augmented, which not only increases the data size but also adds regularization to the learning helping the network to generalize better. The data augmentations include:

Figure 4.3: Horizontal flip

Figure 4.4: Vertical flip

Figure 4.5: Zoom

Figure 4.6: Jitter

4.4 Patch Selection

The input to our network is patches cropped from the images. The image patches are selected such that they contain feature information which are robust to scale and illumination changes. The size of the extracted patches is set to $32 * 32$ to encapsulate enough information to be identified as a strong feature. For selection of patches, we make use of SIFT keypoints obtained on visible images and extract a $32 * 32$ patch around it. Another patch with the same spatial coordinates is extracted from the corresponding thermal image, thus allowing to build visible - thermal patch correspondences for feature extraction. This method of extracting patches is not entirely accurate as there are many features that are observed in visible images which are absent in thermal images. To overcome this situation, we further filter the image patch pair by taking into account the variance observed on thermal patches as shown in Fig. 4.7 below, thus only selectively choosing patches for finer training.

We also make use of heavy data augmentations on these patches to increase robustness.

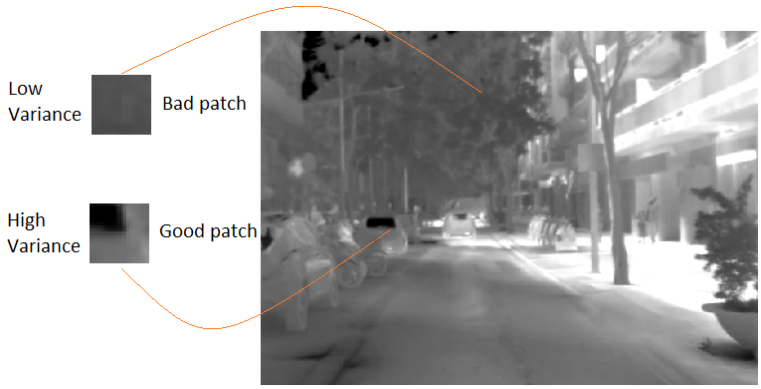

Figure 4.7: Variance based patch selection.

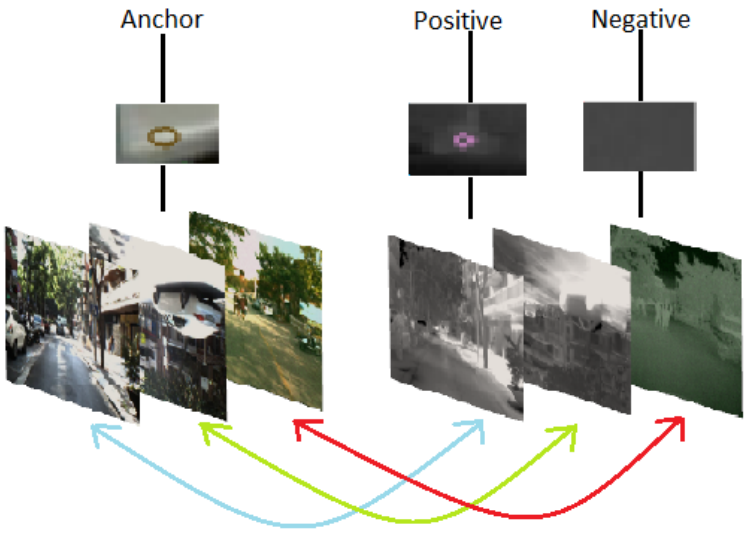

Figure 4.8: Data augmentation for patches given at the time of input.

Chapter 5

Proposed Method

5.1 Feature Detection and Extraction

Feature detection in thermal images is a difficult task mainly due to the high uniformity in those images. Convolutional neural networks have facilitated a deep network-based feature learning in color domain. We implement the same principle in our methodology. To enable the network to learn meaningful features on thermal images, we make use of the concept of Triplet based Siamese Neural Network and design our network in a way, which is trained on image pair patches extracted from visible-thermal image pairs from the KAIST and CSS datasets. For training Triplet Network, we require two images, but 3

Figure 5.1: An illustration of the three types of image patch inputs to the detection network: Anchor from RGB images; Positive and negative patches from thermal scenes.

patches: Anchor, Positive Negative. The Anchor is the patch containing feature keypoint extracted from the visible image. The Positive is the thermal patch corresponding to the Anchor and the Negative is any patch but Positive as seen in Fig. 5.1. The Triplet network learns embeddings of the positive patches but also of the negative patches thus allowing us to accurately localize the keypoint.

5.2 Network Architecture

The network architecture consists of two networks for better keypoint and feature vector learning namely: Feature Description Network and Keypoint Detection Network. The overall network architecture is shown in Fig. 5.2. The individual components of the network are explained in the subsections.

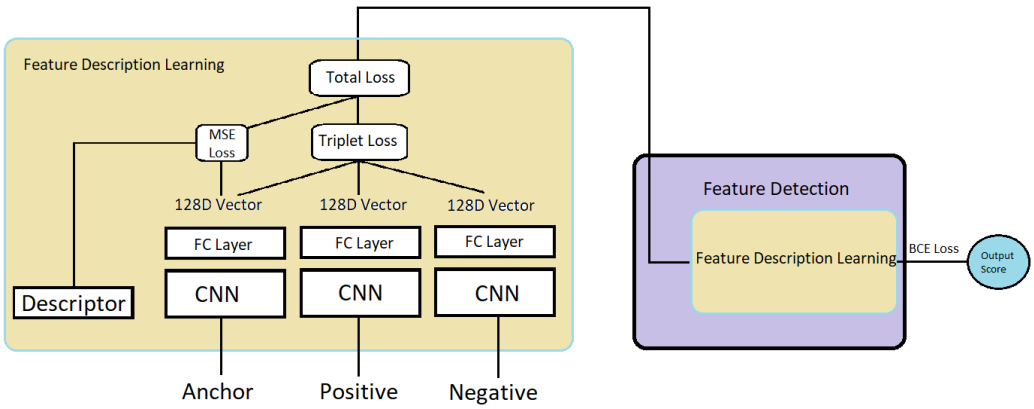

Figure 5.2: Thermal image feature detection and extraction architecture. Inputs given are three image categories: Anchor, Positive and Negative image patches. RGB descriptor values are also provided to regress the anchor feature descriptor.

5.3 Feature Description Network

The Triplet model requires three inputs (in our case image patches) for learning the similarity between images. The Triplet network learns distributed embedding representation of

data points where contextually similar data points are projected in the nearby region and dissimilar data points are projected far away from each other. We use SIFT feature to detect the keypoints and from those detected keypoints we extract $32 * 32$ patches in the RGB and thermal image. This patch corresponds to the detected SIFT feature in the RGB image. On obtaining the anchor patch from RGB image, the keypoint for it is saved. The saved keypoint location from the anchor patch is used where a $32 * 32$ patch with the previously detected keypoint at the center is chosen as the positive patch in the thermal image. For the negative patches we randomly generate keypoint locations and patches with the generated keypoint at the center are selected. Here an additional step is included where we implement patch selection based on the standard deviation of both RGB and thermal patch under consideration. Only those patches are selected which have a deviation value above a threshold indicating a good feature response. This additional step further helps in making the learning better and more robust. The selected patches are then fed into the Triplet network as shown in Fig. 5.3, for feature description learning. The network consists of five convolutional layers followed batch normalization and ReLU activation function. Dropout is used to add regularization in both the models and finally two fully connected layers along with a Sigmoid function are used to output a 128-dimension vector. 128 is selected as many applications take SIFT features as input. We select the same dimensionality as SIFT, but can be changed as necessary.

5.4 Keypoint Detection Network

The 128-dimensional output from the above learned Triplet architecture is connected to the keypoint detection network as shown in Fig. 5.4, which shares the same structure of

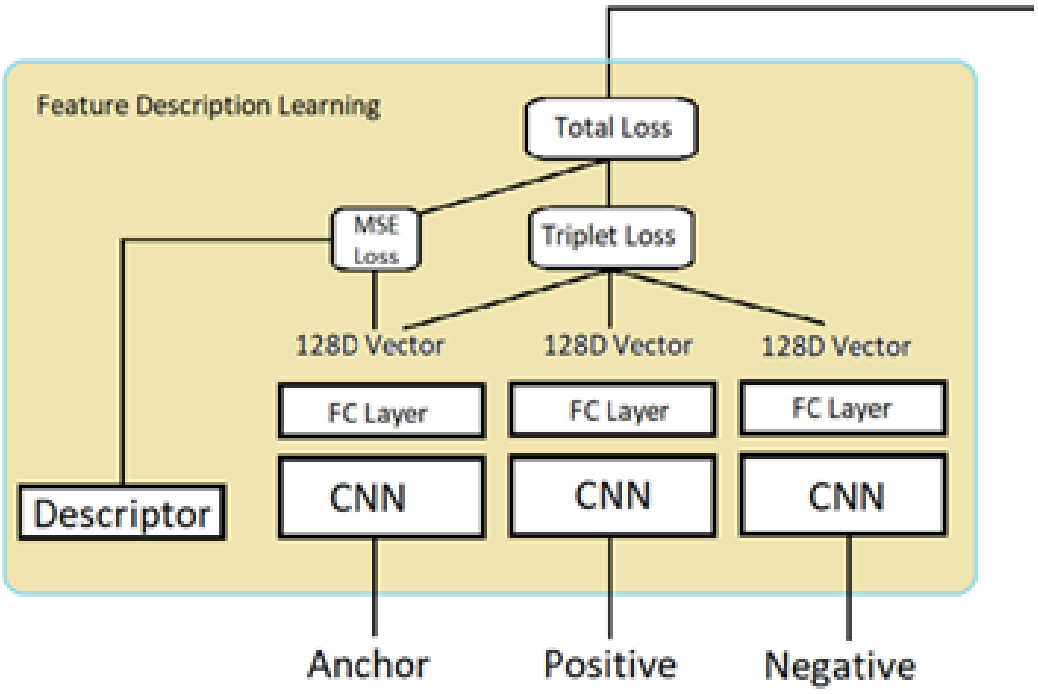

Figure 5.3: Feature description network. Takes in three inputs with two loss constraints for the learning. Output is a 128D vector of feature description.

the feature description network, thus allowing one-step detection and description. This step helps the network classify good and bad feature patches to further improve the feature description and detection. All the intermediate layers are frozen and two fully connected layers followed by ReLU activation function dedicated for detection are added. A Sigmoid activation is used at the end of the fully connected output. The input to this network is $32 * 32$ thermal-thermal patch correspondences. The patches here include augmentation by scaling, flipping etc. Learning is performed on positive and negative patches producing an output score between 0 and 1.

Once a patch with good feature is detected, mid-points of the patch in the image co-ordinate are located and stored as keypoint. After the model is successfully trained and

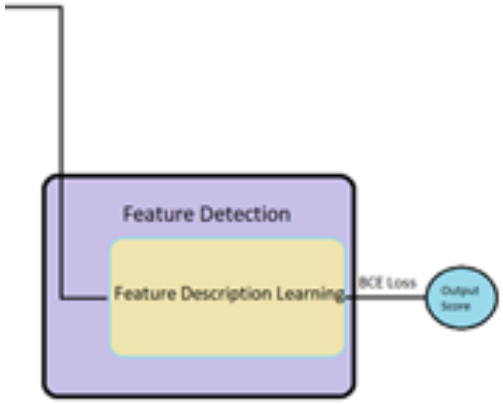

Figure 5.4: Feature detection network. BCE loss criterion used for feature patch classification.

keypoints are obtained we can simply use Euclidean distance measurement amongst the output scores of descriptor values to find the corresponding matches in the two thermal frames. With the shared network and common layers, the feature detection and description can be much accelerated.

5.5 Training and Loss Functions

For the feature description network, we use Margin Ranking loss with a margin of 0.2, with ADAM [18] having a learning rate of 0.001 optimizer. The equation for loss is as:

$$L(A, P, N) = \max\{||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + \alpha, 0\} \tag{5.1}$$

Margin Ranking loss : The Margin Ranking loss is different from other loss functions, like MSE or Cross-Entropy. Unlike the mentioned losses which learn to predict directly

from a given set of inputs, Margin Ranking loss computes a criterion to predict the relative distances between inputs. For the Margin Ranking loss, the loss is calculated using inputs x1, x2 where x1 and x2 are the tensor embedding vectors, as well as a label tensor, y (containing 1 or -1). When y == 1, the first input will be assumed as a larger value i.e., it is ranked higher than the second input. If y == -1, the second input will be ranked higher. Along with the above-mentioned loss we also use a MSE constraint by training the anchor patches with their respective descriptor values as ground truth to improve the learning.

Mean Squared Error Loss : The loss function calculates the average squared difference between the estimated values and the actual value.

We noticed that, by doing so the model learned features not only based on the pixel intensity values but also the information embedded in the patch itself. This further improves the model performance on low resolution dataset such as KAIST. The equation for loss is given as:

$$L_{MSE} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{ij} - y_{ij})^2 \tag{5.2}$$

The overall loss function for the feature description learning architecture is given as:

$$L_{detection} = \lambda_1 L_{(A,P,N)} + \lambda_2 L_{MSE}. \tag{5.3}$$

The keypoint detection network outputs a 0 - 1 valued tensor which is compared to the ground truth of 0 and 1. Classification is performed on the obtained output tensor where Binary Cross Entropy loss (BCE) along with Adam optimization is used for classification.

Binary Cross Entropy loss : The loss function also known as Log loss is used for binary classification problems in the field of machine learning. The loss function tells how good your model is if the model predictions are closer to the actual values making the loss will be minimum and if the predictions are away from the original values then the loss calculated is maximum.

The loss function is given as:

$$L_{det}(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot (p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \tag{5.4}$$

Chapter 6

Experiments and Results

For training we use KAIST and Cross-spectral Stereo (CSS) dataset the details on which can be found in section 4. These datasets are chosen to make our model more robust to low resolution (KAIST) as well as much high resolution (CSS) thermal patches. The image frames are selected such that they consist of unique features for efficient learning. This does cause reduction in the total number of images in the dataset, but it is compensated by augmentation where the frames are flipped (horizontally and vertically), scaled and even jitter is added to the frames before training. The training dataset consists of approximately 10,000 to 15,000 images and $32 * 32$ sized patches are extracted from the images taking our total dataset to increase to more than 150,000 images. For testing we use around 1500 images with patches extracted from them in a similar manner as in training. The testing dataset is completely different from the training dataset.

Initially we used a simple VGG style-based network with Sigmoid activation and BCE loss function. We tried simple classification to check if the network learned directly from a simple solution. The Training loss and accuracy were nearly perfect, but the testing was worse as shown in the Fig 6.1.

From the Fig 6.1 we observed that the model over-fit to the training dataset as seen in

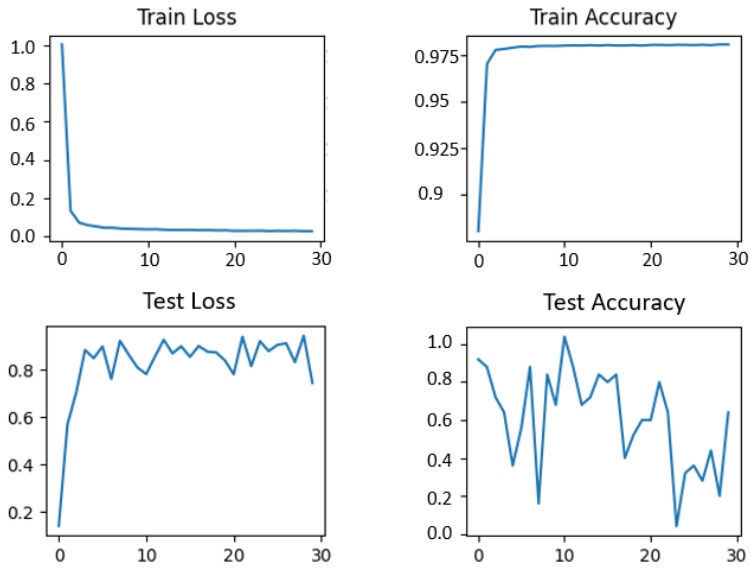

Figure 6.1: Training and testing loss accuracy plot for learning based on only Binary classification. Loss plot: loss value (Y-axis) over the number of epochs (X-axis). Accuracy plot: accuracy (Y-axis) over the number of epochs (X-axis).

the training and testing Loss/Accuracy curves.

Then we proceeded further and tried a Siamese based approach. The results for training and testing are shown in the Fig 6.2. We observe that training plot had a smooth and gradual decrease and the accuracy increased. For the testing initially the network perform well as seen in the testing plot but performance worsened overtime.

Later using PN-Net as an inspiration we trained the model on a simple triplet-based architecture with no modifications to the learning methodology. On completion of the training, we observed that the model learned features based only on the intensity values of the image under consideration and not learning any good features. Slight changes in the intensity of the thermal image caused the performance to drop drastically. This also raised another issue of over-fitting the learning to a particular type of dataset (low resolution/intensity or high resolution/intensity) and resulting in a worse performance for the

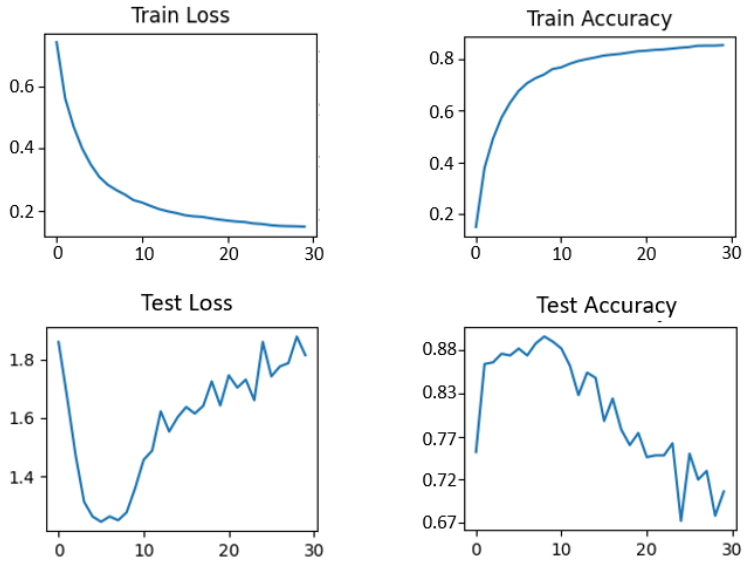

Figure 6.2: Training and testing loss accuracy plot for learning based on Siamese Approach. Loss plot: loss value (Y-axis) over the number of epochs (X-axis). Accuracy plot: accuracy (Y-axis) over the number of epochs (X-axis).

other. The training and testing loss/accuracy plot are shown in the Fig 6.3.

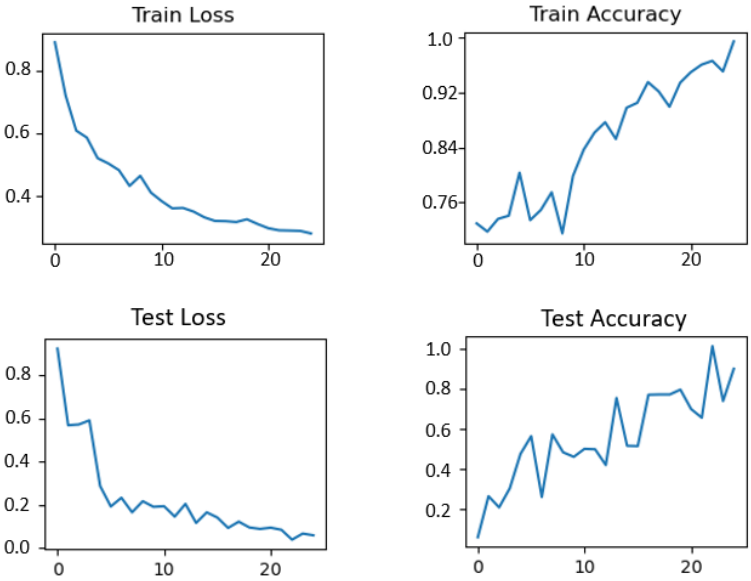

Figure 6.3: Training and testing loss accuracy plot for learning based Triplet based approach without the use of MSE constraint. Loss plot: loss value (Y-axis) over the number of epochs (X-axis). Accuracy plot: accuracy (Y-axis) over the number of epochs (X-axis).

To tackle this issue, we then added an additional constraint of MSE loss to the overall

loss function thereby forcing the anchor description vector to be close to its RGB counterpart.

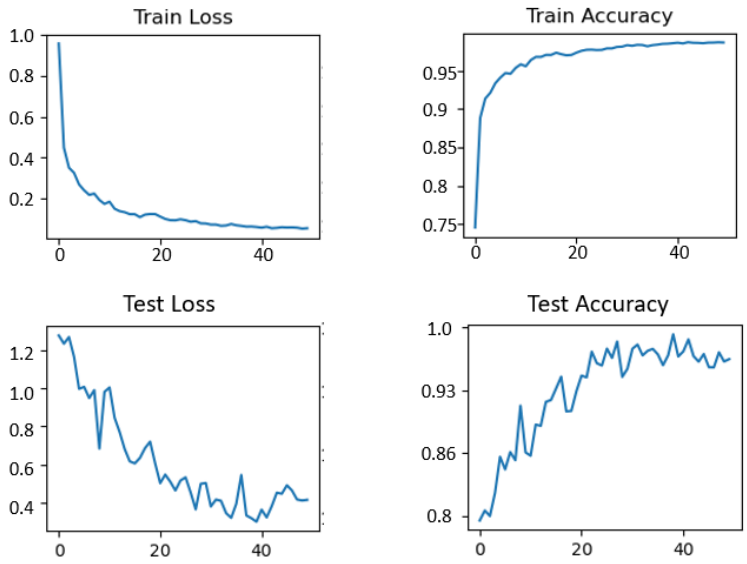

Figure 6.4: Training and testing loss accuracy plot for learning based Triplet based approach with the use of MSE constraint. Loss plot: loss value (Y-axis) over the number of epochs (X-axis). Accuracy plot: accuracy (Y-axis) over the number of epochs (X-axis).

On addition of the above constraint we observe from the Fig 6.4, the model was able to perform well on images with both high and low resolutions.

We also tried using a traditional method of SIFT along with Contrast Limiting Histogram Equalization (CLAHE) applied to the images under consideration. In our experiments we observed that simple modification of the image using a technique like CLAHE drastically improved the feature detection capability with average feature detection of 980 in CSS dataset and 700 in KAIST dataset. Example CLAHE based detection on thermal images is shown in the Fig 6.5:

But when we tried the feature matching algorithm, this approach produced good results for the features matched in the high resolution CSS dataset image with an average of 725

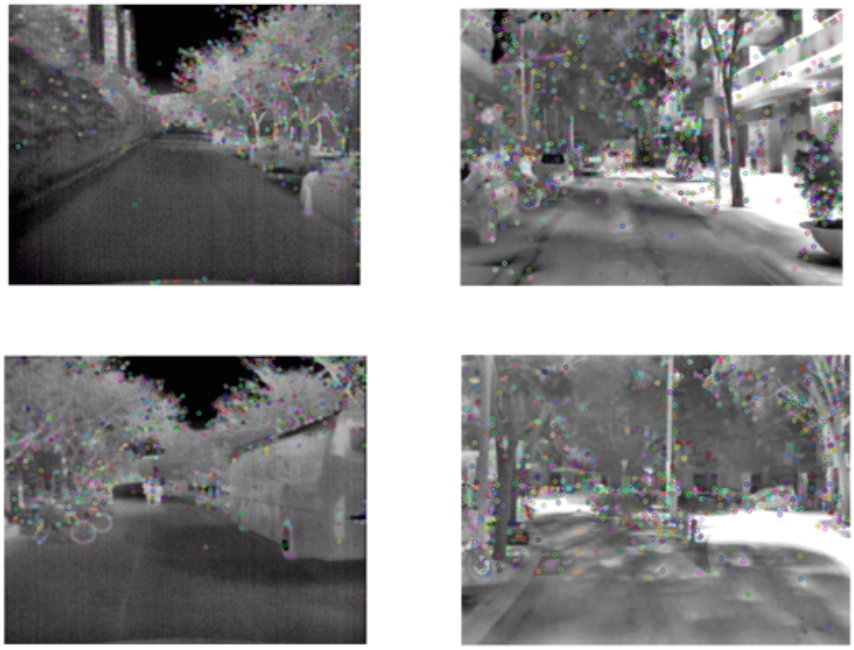

Figure 6.5: Feature detection using CLAHE applied on thermal images.

correct feature matches, but failed to do the same in low resolution KAIST dataset images with an average of only 350 features correctly matched. With these numbers we could determine that CLAHE and other image pre-processing methods may have a better effect on high quality thermal images, which are generally easy to process. With the lower resolution and sensitivity sensors, the advantages of the developed neural network approach is more obvious. As high resolution and temperature sensitivity thermal cameras are quite expensive, the research outcomes could have broader impact on the common market. Also our method directly operates on original thermal images. The performance of our method on images after enhancement will also increase.

	Keypoint Detected	Keypoint Matched
SIFT	KAIST - 121 CSS - 274	KAIST - 77 CSS - 218
Superpoint	KAIST - 97 CSS - 212	KAIST - 63 CSS - 163
Our Model	KAIST - 562 CSS - 778	KAIST - 405 CSS - 582

Table 6.1: Comparisons on detection and matching.

6.1 Quantitative analysis

We compared feature description and matching from SIFT and deep network SuperPoint on CSS datasets. The results of which are shown in Fig. 6.8. It can be observed from Fig 6.8 that our method is able to generate denser and more reliable matchings on the challenging thermal scenes, compared with classic SIFT algorithm and learning based SuperPoint. We also compared feature point detection for different techniques. From the feature detection result in Fig. 6.5, we can see that even though SIFT has a relatively good number of feature detection for the high-resolution CSS dataset, it is unable to produce a comparable result for low resolution KAIST dataset. This is mainly because SIFT is designed for images with high textures and hence gives a poor detection on low resolution and low textured images.

From the Table 6.1 we observe that the number of features detected by our network exceeds traditional SIFT and a neural network model Superpoint retrained on our dataset not only in the high-resolution CSS dataset but also in the low resolution KAIST dataset. Besides the visual comparison, we also report the number of the detected keypoints and the matched keypoints from our trained model on the two datasets, compared with SIFT and SuperPoint in Table 6.1.

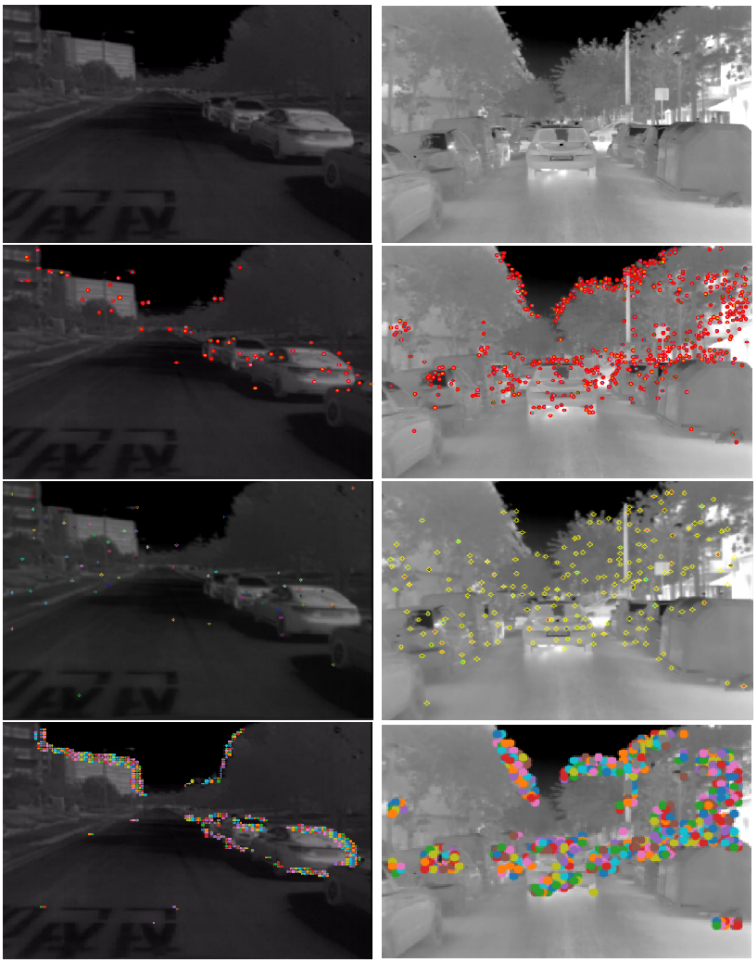

Figure 6.6: Feature detection comparison. Top-bottom: Input Images; SIFT matches; Third row: Superpoint matches; Final row: Our method.

6.1.1 Feature Matching

We use Euclidean metrics between the descriptor values between the frame under consideration and the remaining frames and select the one with the closest match. The matching mechanism is kept based on Euclidean distance so that much faster searching algorithms such as KD-tree search can be implemented making the matching in real time. To further improve the matching capabilities we use Lowe's ratio test (A test where the ratio of the first and the second closest match should be smaller than a certain threshold value) algorithm in

order to be certain no keypoints are mismatched with incorrect correspondences.

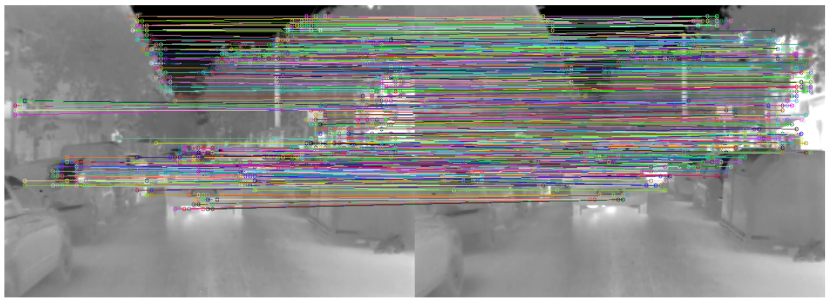

Figure 6.7: Feature detection and matching result on high resolution CSS dataset.

From the detected keypoints and their description values obtained from the network, we performed the task of feature matching to evaluate the model applications of Visual Odometry (VO), Structure from Motion, Simultaneous Localization And Mapping (SLAM) etc.

The next corresponding frame is 5-frame after the first frame. The camera capturing capability is 20 fps, much slower than normal visible video cameras (60-120 fps). As we targets at video tracking, the 5 frames interval sampling can demonstrate the feature matching effectiveness, as normal video tracking is performed between adjacent frames. In our results we clearly see our network outperforms others in the number of keypoints correctly matched.

We implemented our feature matching on a Visual Odometry application and compared the results with ones obtained from an optical flow method on RGB and thermal as shown

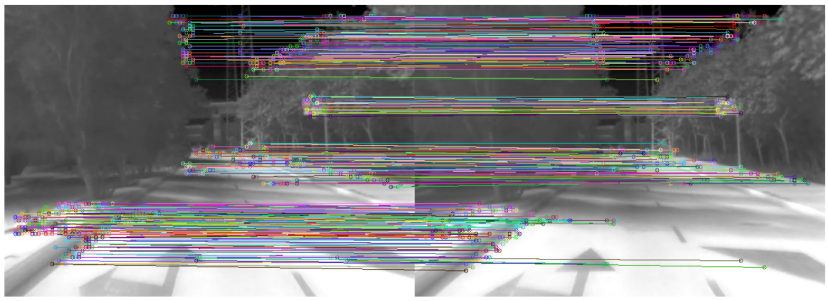

Figure 6.8: Feature detection and matching result on high resolution CSS dataset for a scenario without vehicles.

Fig. 6.6 below:

From the results we can observe that our model performs well in determining the path traversed. Though it is not perfect, the end result can be improved by further optimizing through visual odometry neural network.

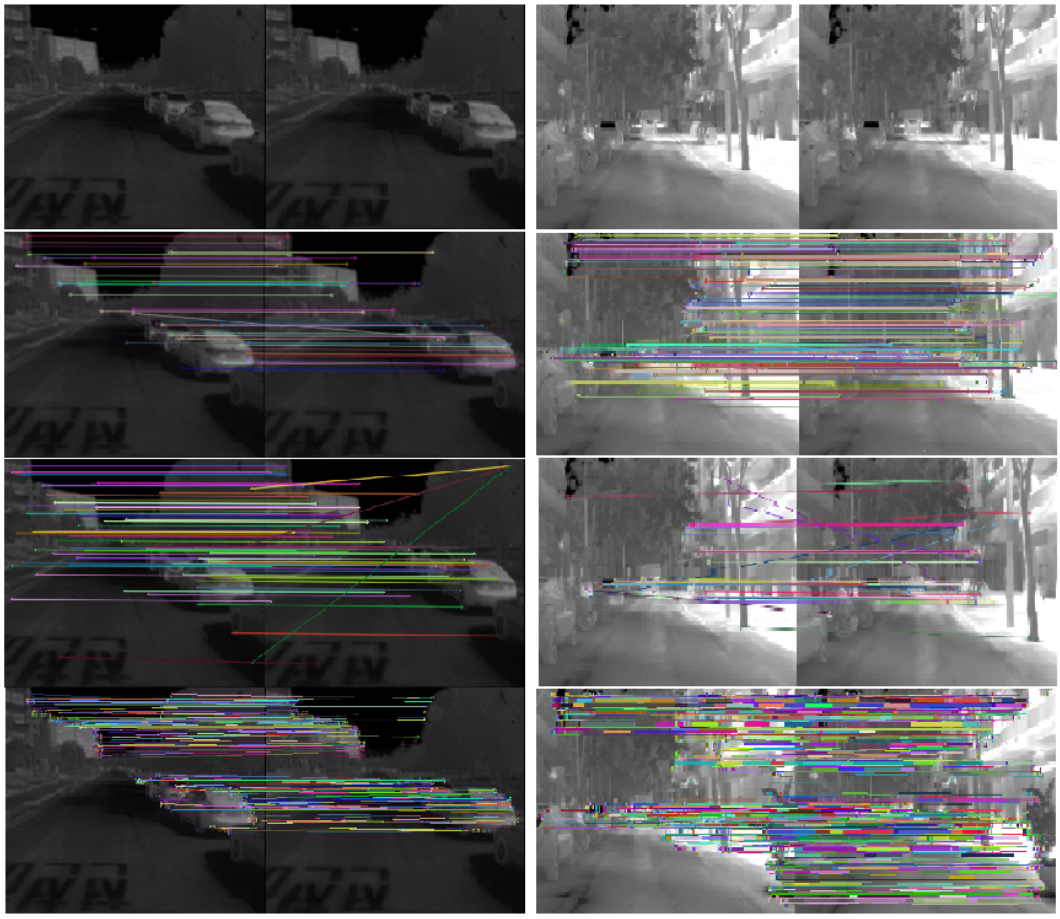

Figure 6.9: Feature matching comparisons: Input images (Top), SIFT feature matching (second row), Superpoint matching (third row) and finally our model (Bottom).

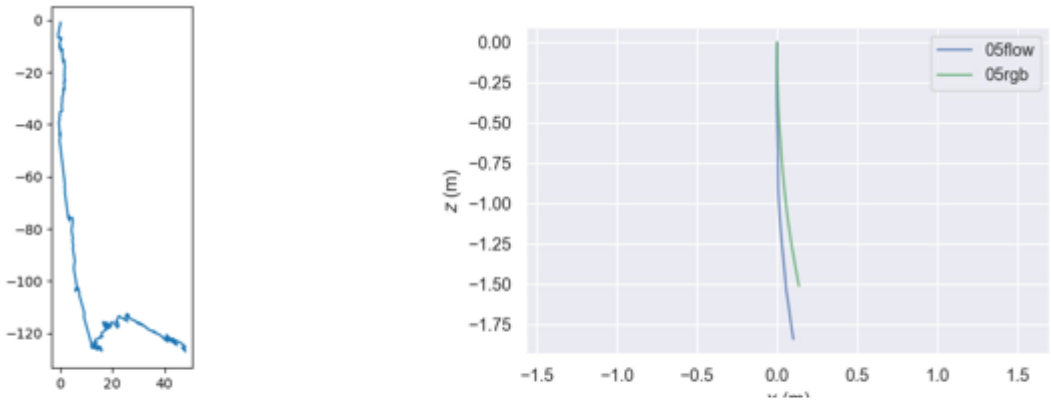

Figure 6.10: Comparison of Visual Odometry outcomes between our model and optical flow based, referenced from the thesis work of Janardhan Choudhary.

Chapter 7

Conclusion and Future work

In the thesis I investigated a novel approach for feature extraction on thermal images using a deep convolutional neural network approach. All the feature extraction methods available are either done traditionally or by CNN that are applied only on visible spectrum images. However, there are methods fusing multi-spectral image data (visible and infrared) and performing feature detection and extraction on the same. These methods do not solely rely on the infrared spectrum, but additional information from the visible spectrum is incorporated in their learning and inference. This approach even though noteworthy but may fail where visible spectrum cameras fail for example in case of low visibility or low illumination regions.

We first examined feature detection and matching from traditional feature extraction methods such as SIFT. From the results we see that SIFT detector is able to perform well comparatively to other available methods but only in the case of high-resolution infrared images and fails to do so in low resolution images.

Later we evaluated state of the art CNN architectures for feature detection, which even though produce good detection results on RGB images, performs much worse than the traditional SIFT when trained on thermal dataset.

We propose a throughout feature detection and description network for thermal descriptor learning based on Triplet Siamese network, which designs an effective method for extracting descriptors to be learned along with the intensity images to obtain much better feature extraction. We initially train on a single loss function without additional augmentation to the data. This displayed an overfitting pattern in the learning. To tackle this issue, we add regularization to our loss function and incorporate data augmentation. Furthermore, we add another MSE loss function to our overall training loss. We observe that the learning improves after the addition of the new loss. Both the learning scheme and loss constraint demonstrate an effective solution compared to other available methods. Our method is easy to implement to be used in practical applications where traditional methods would fail.

Our model even though produces great results, still can be improved. This can be achieved by obtaining much larger and more diverse thermal dataset, which has not been explored in this area. Moreover, by designing new constraints and tuning the hyper parameters the model's performance can be further improved. Also, the use of a different loss functions can help boost the distinctiveness of detected features and their descriptors. Since it is a new field, there is significant space for opportunities to explore and research in the same direction. If achieved, thermal cameras could be a considered as a very well replacement or an add-on for the visual spectrum cameras in a variety of applications.

Bibliography

[1] *David G Lowe, "Distinctive image features from scale-invariant keypoints,"ao International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.*

[2] *Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in European conference on computer vision. Springer, 2006, pp. 404–417.*

[3] *Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski, "Orb: An efficient alternative to sift or surf," in 2011 International conference on computer vision. Ieee, 2011, pp. 2564–2571.*

[4] *Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," in European conference on computer vision. Springer, 2006, pp. 430–443.*

[5] *Faiz Rahman, Bhavan Vasu, Jared Van Cor, John Kerekes, Andreas Savakis, "SIAMESE NETWORK WITH MULTI-LEVEL FEATURES FOR PATCH-BASED CHANGE DETECTION IN SATELLITE IMAGERY"*

[6] *Yurun Tian, Bin Fan, and Fuchao Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 661–669.*

[7] *Vijay Kumar B G, Gustavo Carneiro, Ian Reid, "Learning Local Image Descriptors with Deep Siamese and Triplet Convolutional Networks by Minimizing Global Loss Functions."*

[8] *Josep Aulinas, Marc Carreras, Xavier Llado, Joaquim Salvi, Rafael Garcia, Ricard Prados, and Yvan R Petillot, "Feature extraction for underwater visual slam," in OCEANS 2011 IEEE-Spain. IEEE, 2011, pp. 1–7.*

[9] *J. Hartmann, J. H. Klüssendorff and E. Maehle, "A comparison of feature descriptors for visual SLAM," 2013 European Conference on Mobile Robots, 2013, pp. 56-61, doi: 10.1109/ECMR.2013.6698820.*

[10] *Julien Poujol, Cristhian A Aguilera, Etienne Danos, Boris X Vintimilla, Ricardo Toledo, and Angel D Sappa, "A visible-thermal fusion based monocular visual odometry," in Robot 2015: Second Iberian Robotics Conference. Springer, 2016, pp. 517–528.*

[11] *McConnell, R.K. Method of and Apparatus for Pattern Recognition. U.S. Patent No.*

4,567,610, 28 January 1986

[12] *C. Harris and M. Stephens (1988). "A combined corner and edge detector" (PDF). Proceedings of the 4th Alvey Vision Conference. pp. 147–151.*

[13] *Canny, J., A Computational Approach To Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.*

[14] *N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.*

[15] *Sergey Zagoruyko and Nikos Komodakis, "Learning to compare image patches via convolutional neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4353–4361.*

[16] *Kun He, Yan Lu, and Stan Sclaroff, "Local descriptors optimized for average precision," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 596–605.*

[17] *Karen Simonyan Andrew Zisserma, "VERY DEEP CONVOLUTIONAL NETWORKS*

FOR LARGE-SCALE IMAGE RECOGNITION."

[18] *Adam: A Method for Stochastic Optimization, Diederik P. Kingma and Jimmy Ba, year=2017,1412.6980*

[19] *Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi, "Lf-net: learning local features from images," in Advances in neural information processing systems, 2018, pp. 6234–6244.*

[20] *Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8798–8807.*

[21] *Jianbo Shi et al., "Good features to track," in 1994 Proceedings of IEEE conference on computer vision and pattern recognition. IEEE,1994, pp. 593–600.*

[22] *Engin Tola, Vincent Lepetit, and Pascal Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 5, pp. 815–830, 2009.*

[23] *Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk,*

"Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5173–5182.

[24] *PN-Net: Conjoined Triple Deep Network for Learning Local Image Descriptors Vassileios Balntas, Edward Johns, Lilian Tang, Krystian Mikolajczyk.*

[25] *Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich, "Superpoint: Self-supervised interest point detection and description," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 224–236.*

[26] *https://en.wikipedia.org/wiki/Feature (computer vision)*

[27] *https://media.istockphoto.com/photos/mount-ama-dablam-within-clouds-picture-id938914580?k=6m=938914580s=612x612w=0h=-tQlNL-xuFYSRHAbQ12ms74LG7vjlLfvNwR0AjUPh-o=*

[28] *X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patchbased matching. In CVPR, 2015.*

[29] *E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In ICCV, 2015.*

[30] *S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In CVPR, 2015.*

[31] *Molecular Expressions Microscopy Primer: Digital Image Processing – Difference of Gaussians Edge Enhancement Algorithm", Olympus America Inc., and Florida State University Michael W. Davidson, Mortimer Abramowitz.*

[32] *Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in CVPR, 2015, pp. 1037–1045.*

[33] *Lu, Yawen and Lu, Guoyu,"SuperThermal: Matching Thermal as Visible Through Thermal Feature Exploration",IEEE Robotics and Automation Letters, 2021,2690–2697*

[34] *FA Group et al., "Flir thermal dataset for algorithm training," 2018.*

[35] *Christoph Strecha, Alex Bronstein, Michael Bronstein, and Pascal Fua, "Ldahash: Improved matching with smaller descriptors," IEEE transactions on pattern analysis and machine intelligence, vol. 34, no. 1, pp. 66–78, 2011.*

[36] *Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart, "Brisk: Binary robust invariant scalable keypoints," Ieee, 2011, pp. 2548–2555..*