Deep Learning in Feature Detection and Matching in Computer Vision: From SIFT to SuperPoint+SuperGlue

Yubo Mao
mao00071@morris.umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

Abstract

This paper introduces the significant shift techniques of feature detection and matching from classical methods like SIFT and NN(Nearest Neighbor) to advanced deep learning-based methods like SuperPoint and SuperGlue. Feature detection, a critical process in computer vision, involves identifying and describing salient points or regions within images, laying the foundation for numerous applications such as object recognition and augmented reality. SuperPoint and SuperGlue, used in conjunction, represent a paradigm shift in feature detection and matching, replacing the traditional role of SIFT. This paper explains the functioning of SIFT, SuperPoint, and SuperGlue, and compares their performance, illustrating how deep learning has reshaped feature detection and matching in the field of computer vision.

Keywords: features, neural networks

1 Introduction

Computer vision is the field of computer science which aims to enable machines to interpret and make decisions based on visual data or similar. One of the foundational challenges in this field has been the task of reliably detecting, describing and matching features within images. These features, identifiable as distinct points or regions in an image, lay the groundwork for numerous applications, from image recognition to augmented reality.

A feature in computer vision refers to distinct elements within an image, such as edges, corners, or objects, that are significant for analyzing and understanding the image's content. A feature usually consists of two parts: the keypoint and the descriptor. A keypoint is a point in image to point out where a feature is, so the keypoint stores 2D spatial information of a feature. A descriptor is the appearance of a feature, and its purpose is to distinguish a feature from other features. Homography, a critical concept in image processing, involves a transformation that maps points from one plane to another, typically used in tasks like image stitching and 3D reconstruction. It is essential to understand these concepts for effective feature detection and matching, as illustrated in Figure 1.

In the early days of computer vision, handcrafted feature detectors and descriptors began to emerge, and Scale-Invariant Feature Transform (SIFT) became the most significant example of this era. SIFT is completely handcrafted, meaning it is designed based on predefined algorithms and mathematical models that specify how to identify features in images. The core technique SIFT utilizes is Gaussian pyramid. A Gaussian function blurs an image, and this process can be repeated, creating a series of increasingly blurred images. This sequence forms a Gaussian pyramid. By examining the differences between these blurred images at various scales, key information is extracted to determine keypoints related to important features in the image. The robustness of SIFT to changes in image scale, rotation, and illumination made it a preferred choice for a variety of applications and set a benchmark for future algorithms. However, while SIFT and its contemporaries were groundbreaking, they had limitations, particularly when it came to adaptability and handling a broader range of visual distortions.

With the advent and success of neural networks in various fields, computer vision too experienced a great change. Instead of handcrafting features and algorithms, researchers could now "teach" systems to learn these features directly from vast amounts of data. This paradigm shift led to significant advancements in accuracy, adaptability, and capability. The emergence of methods like SuperPoint and SuperGlue, which utilize deep learning for feature detection and matching, showcased the potential of this new era. In the indoor/outdoor localization challenges of CVPR2020 /ECCV2020, the solution using SuperPoint and SuperGlue tops the list, fully demonstrating the advantages of these two methods in feature extraction and matching.

2 Background

Both SuperPoint and SuperGlue use neural networks and SuperPoint uses Fully Convolutional Networks while SuperGlue uses Graph Neural Network. Therefore, before diving into SuperPoint and SuperGlue, it is necessary to understand the neural networks that they use.

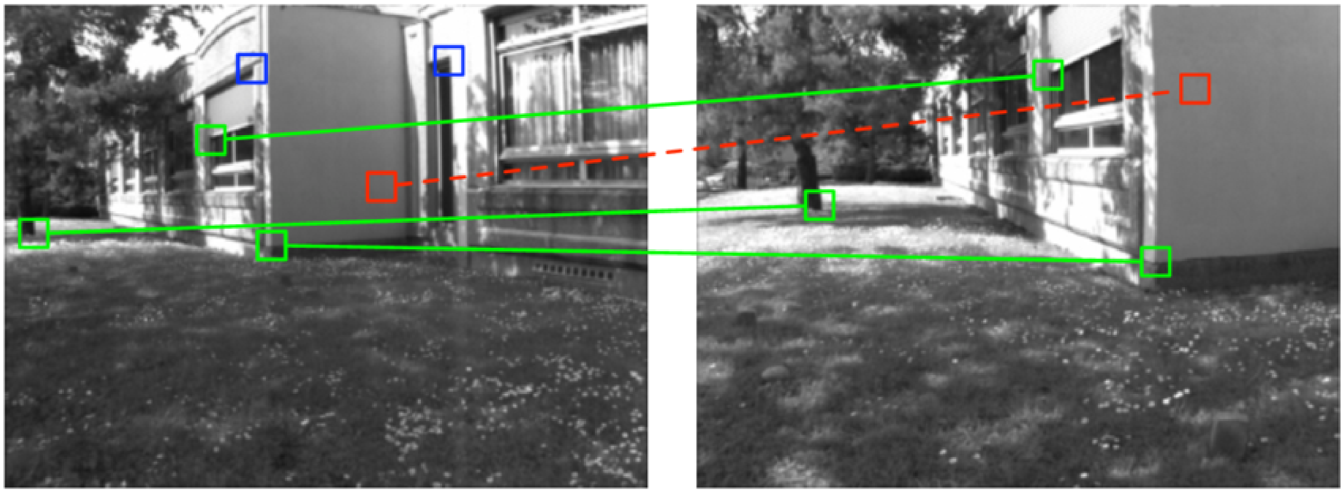


Figure 1. Feature detection and matching. The small boxes in the image represent the local features that were extracted. The connecting line between two images represents the matching relationship between features. In this example, green represents a correct match. Red represents an incorrect match. Blue means there is no corresponding feature to match. [6]

2.1 Neural Networks

In the realm of machine learning, neural networks play a pivotal role. These networks, conceptualized as a series of interconnected nodes or neurons, perform calculations involving intermediate results. The structure of a neural network is layered, with each layer processing a collection of nodes in a stepwise pattern, and these calculations are iterative and occur in batches. The layers are interconnected, with each layer's output forming the input for the subsequent layer. This design allows the network to extract and process complex patterns in the input data. The operations within a neural network are governed by parameters, which are 'weights'. These weights are crucial in controlling the network's responses and learning process. A more detailed and technical explanation of neural networks can be found in specialized textbooks on the subject, which delve into the intricacies of their design and function.

2.2 Fully Convolutional Networks (FCN)

As the neural network technology used by SuperPoint, FCN represents a class of deep neural architectures tailored for image semantic segmentation, where the goal is to assign a class label to each pixel in an input image. FCN is a variant of traditional convolutional neural network (CNN). Unlike CNN that often includes fully connected layers for classification, an FCN continuously uses convolutional layers, allowing it to handle inputs of any size and generate spatially dense outputs. By converting the fully connected layers into convolutional layers, FCNs can produce pixel-wise segmentation maps, offering detailed spatial information about the image's content. FCN has played an important role in driving recent results in the task of image semantic segmentation. The following illustrates the differences between FCN and CNN by explaining the main structure of CNN.

Convolutional Layers: The purpose of convolutional layers is to detect local features, such as edges, corners, and textures. This is achieved using the convolution operation, where a small matrix known as a filter (or kernel) slides or convolves across the input data, often an image. At each position, a dot product is computed between the filter and the input, resulting in a pixel in the output feature map. The network trains multiple filters, allowing it to detect a variety of features, from vertical and horizontal edges to more complex patterns in deeper layers.

Pooling Layers: The purpose of pooling layers is to reduce the computational demand and the number of parameters by downsampling the spatial dimensions of an input. This not only speeds up the computation but also enhances the ability to recognize features. The pooling operation involves sliding a filter(often square-shaped, e.g., 2x2 or 3x3) over the input feature map and aggregating the values within that filter into a single value using a specific aggregation function. Pooling is divided into average pooling and maximum pooling, with maximum pooling being the most commonly used. In the process of aggregating values within the filter, max pooling takes only the maximum value as its output while average pooling computes the average of the values.

Fully Connected Layers: In CNN, the fully connected (FC) layers serve as the decision-making units, typically situated at the end of the network. After the convolutional and pooling layers extract features from the input data, the FC layers process these features to produce a final outcome, such as class probabilities in image classification tasks. They are termed "fully connected" because every neuron in these

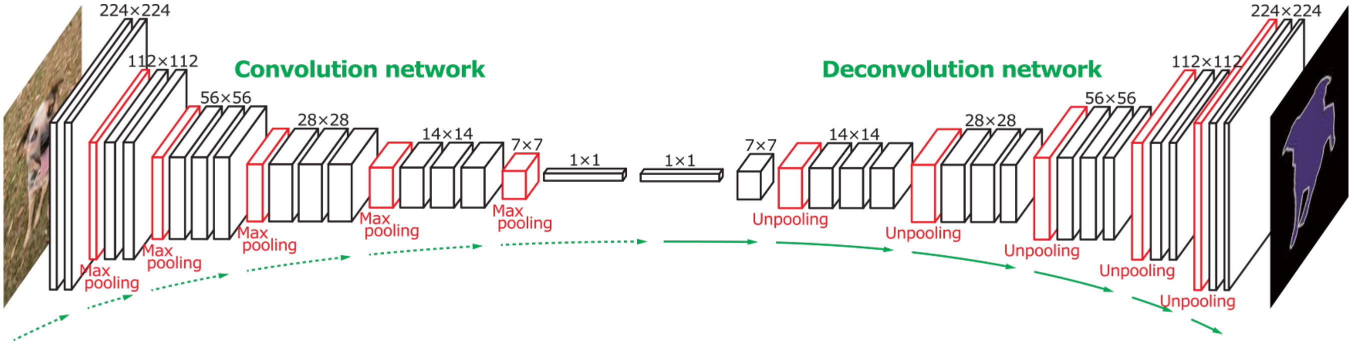


Figure 2. This is one example of FCN doing Semantic Segmentation. In the Convention networks, The size of the image becomes smaller and smaller after the conventional and max pooling layers until its size is 1. Then in the Deconvolutional networks, the size of the image is gradually restored until it is the same size as the input image. [4]

layers is interconnected with all neurons from the previous and subsequent layers, ensuring comprehensive integration of feature information for decision-making. Each neuron uses ReLU(Rectified Linear Unit), which is a widely used activation function in neural networks and directly outputs the positive input and ouputs zero for else input, to speed up the computation. Because of the strict limitations of the fully connected layer on the size of the input, CNN can only accept images of a specific size.

The main difference between CNN and FCN is FC layers. FCN replaces the FC layer with a convolutional layer, and then uses deconvolutional layers to upsampling the feature map of the last convolutional layer (which can also be called the heatmap in FCN), restoring it to the same dimensions as the input image, so that it can generate pixel wise prediction. Since there is no fully connected layer, FCN can accept images of any size. Figure 2 clearly shows the architecture of a FCN.

2.3 Graph Neural Network (GNN)

As the neural network technology used by SuperGlue, GNN is a type of neural network designed to operate directly on graphs, a data structure that captures entities (nodes) and their interactions (edges). Traditional neural networks, such as CNN and RNN, are often unsuited for graph-structured data due to the irregular and dynamic nature of graphs. GNN, on the other hand, is specifically engineered to handle this kind of data, effectively passing information through graph structures.

The basic and core job of GNNs is to perform pairwise message passing. All nodes in a graph have their own information. At each layer of GNN, each node aggregates information from its neighbors and then possibly updates its own information based on the aggregated information. This local aggregation and update mechanism allows GNN to capture complex patterns and dependencies in graph-structured data.

Attention mechanisms in GNNs enhance the model's ability to focus on specific parts of the graph structure. In a GNN, attention operates by calculating and assigning different weights to the nodes or edges in the graph. These weights represent the importance or relevance of each node or edge in the context of a given task. Weighting works by influencing the message passing process. In general, each node aggregates more information from neighboring nodes that have higher weights on it. For instance, in node classification tasks, the attention mechanism can help the model to focus more on neighboring nodes that have a higher influence on the target node's class. This selective focus allows GNNs to be more accurate and efficient in processing graph-structured data. Attention in GNNs is particularly useful in tasks where the importance of nodes or their connections varies significantly, such as in social network analysis, molecule structure prediction, or recommendation systems. The complexity of attention is beyond the scope of this paper. For an in-depth understanding, the interested reader is referred to 'Attention is all you need'. [7]

3 Scale Invariant Feature Transform(SIFT)

SIFT is an algorithm in computer vision developed by David Lowe in 1999 to detect and describe local features in images. It takes an image as input and outputs extracted features in the image with keypoints and descriptors. The SIFT algorithm has four main steps to generate a set of image features:

Scale-Space Extrema Detection: To identify potential interest points that are invariant to scale changes in the input image, SIFT first employs a Gaussian pyramid to produce blurred images at different scales. It then computes the Difference of Gaussians (DoG) between successive Gaussian-blurred images and identifies potential keypoints as local maxima and minima in the DoG images across scales. This step ensures that the features detected are consistent across varying scales, making the algorithm robust to scale changes in the input image.

Keypoint Localization: In this step, each candidate keypoint is examined for its stability. SIFT performs a detailed fit to the nearby data for location, scale, and ratio of principal curvatures. This helps in discarding low-contrast points and edge responses, reducing the likelihood of instability and noise. This step enhances the reliability of the keypoints detected, ensuring they are distinctive and stable, which is important for the matching process.

Orientation Assignment: The purpose of this step is to ensure the keypoints are rotation invariant. In this step, the gradients of image intensities at surrounding pixels of each keypoint are computed. A histogram of gradient orientations is created, and the peaks in this histogram correspond to dominant directions of local gradients. The highest peak and other peaks within 80% of the highest are assigned as the keypoint's orientations. [3]

Keypoint Descriptor: In this step, a descriptor is generated for each keypoint based on the intensity gradients in its surrounding region. This involves dividing the region around the keypoint into sub-regions and creating a histogram of gradient directions for each sub-region. These histograms are then combined to form the final descriptor, which is robust to variations in illumination, 3D viewpoint, and slight changes in geometry.

4 SuperPoint

SuperPoint is a deep learning based method of feature extraction using FCN. It can be said to be the deep learning version of SIFT in terms of classic degree. Superpoint is fundamentally different from SIFT because it can learn to detect keypoints and generate descriptors from vast amounts of data, rather than relying on handcrafted rules. The SuperPoint network is trained in a self-supervised manner using both synthetic and real-world images. It is also designed to output both keypoint locations and their corresponding descriptors in one forward pass. The learning-based nature of SuperPoint allows it to potentially adapt and generalize better to various image conditions and scenarios, as it derives its behavior from patterns observed in training data. This approach contrasts with the fixed operations of SIFT, offering flexibility and adaptability at the cost of requiring training data and computational resources for training. The SuperPoint is mainly composed of two parts: MagicPoint and Homographic Adaptation. [1]

MagicPoint: The MagicPoint is the predecessor of the SuperPoint algorithm that specializes in detecting salient points in images. Magicpoint is a FCN trained on a large amount of virtual data, which is consisted of some basic shapes such as line segments, triangles, rectangles and cubes, and these basic shapes have uncontested feature point locations. [1] After being trained on a large amount of data, MagicPoint's performance in detecting feature points in virtual images is significantly better than traditional approaches, but it still does not perform well in real, complex images. Therefore, the authors had to improve Magicpoint.

Homographic Adaptation: This is an improvement proposed and applied by the authors to Magicpoint. The homographic adaption enhances the robustness of keypoint detection by applying various geometric transformations, known as homography, to images. These transformations include scaling, rotating, and translating the image to simulate different viewing conditions. Magicpoint is called SuperPoint after the homographic adaption training. Training the algorithm on features invariant to these transformations ensures that SuperPoint can reliably identify keypoints across diverse images and conditions. This adaptability distinguishes SuperPoint from traditional feature detection methods, showcasing its effectiveness in handling complex visual tasks.

5 SuperGlue

SuperGlue is the feature matching algorithm based on graph neural networks. It introduces an attention mechanism to strengthen the network's ability to represent features, thus making it possible to still find a good match between feature points between two images with large parallax. Unlike traditional methods like Nearest Neighbor algorithm which only use descriptors to match features, SuperGlue takes the keypoints coordinate of the two images and their corresponding descriptors as input and outputs an assignment matrix representing the matching relationship between two sets of features. This ensures the resulting matches can take into account both the spatial information and appearance of features. The structure of superglue has three parts: the feature encoding, the Attentional Graph Neural Network, and the optimal matching layer. They are showed in Figure 3 [5]

Feature Encoding: In contrast to traditional feature matching methods, which only focus on descriptors while often neglecting the spatial aspects of keypoints, SuperGlue introduces an innovative approach. It integrates a feature encoding process that combines both spatial and appearance attributes of features. This integration results in a comprehensive feature representation, encapsulating both spatial location and visual characteristics. The equation that combines space and appearance is as follows:

$$^{(0)}\mathbf{x}_i = \mathbf{d}_i + \mathrm{MLP}_{\mathrm{enc}}(\mathbf{p}_i). \tag{1}$$

where d_i is the descriptor of the number i feature, and p_i is the 2D position(keypoint) of feature i. The MLP is a multilayer perceptron to increase the dimension of p_i to make it able to merge with descriptor. x_i represents the newly created node after merging the descriptor and spatial information of a feature. [5]

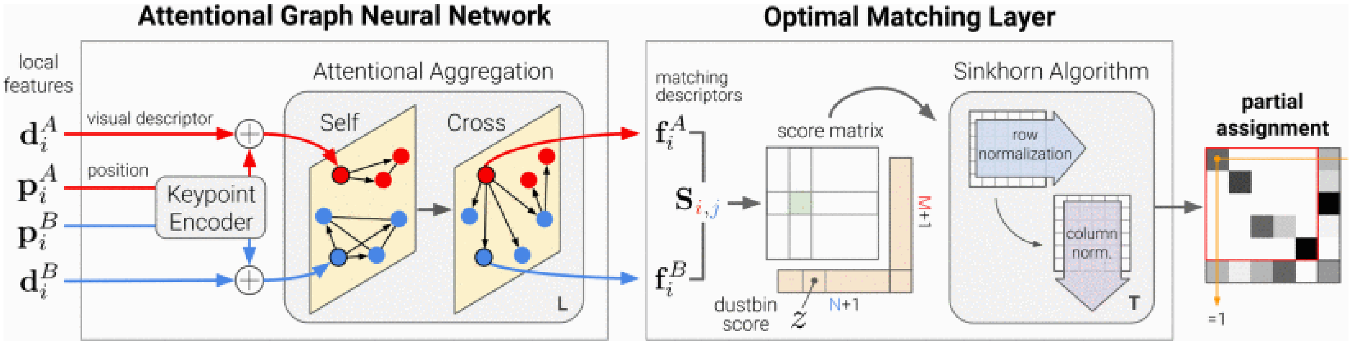


Figure 3. The SuperGlue architecture.

Attentional Graph Networks: The attentional graph neural network in SuperGlue is a key component of the system, and it's designed to process sets of features by considering the relationships between them within the same image (self-attention) and across two images (cross-attention).

Graph Construction: SuperGlue treats the set of local features extracted from images as a graph. Each feature x_i is considered a node in the graph, and the goal is to determine the edges that represent the relationships between these nodes.

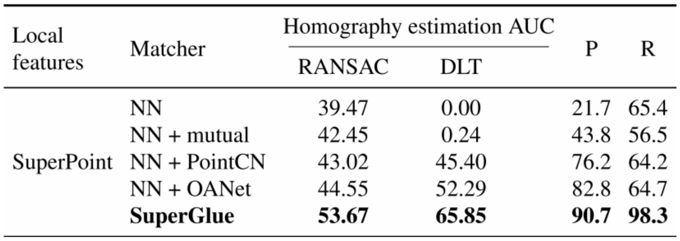
Self-Attention: The self-attention mechanism allows the network to learn contextual information for each feature within the same image. It does this by considering all the features (nodes) and learning which features are relevant to each other. This part of the network helps to refine the features before matching, by allowing each feature to be influenced by the presence of other features in the same image.

Cross-Attention: Cross-attention extends the self-attention mechanism across two different images. The network learns to attend to features in one image based on the information from the other image. This process is crucial for feature matching because it allows the network to compare features from one image with features from another image, considering not just individual feature similarities but the global context as well.

Optimal Matching Layer: After the Attentional graph part, each feature point x_i will be transformed into f_i called feature descriptor, which is a powerful representation for feature matching. [5] When combining the feature descriptor f_i in image A and the feature descriptor f_j in image B, a matching score $S_{i,j}$ will be generated. A higher matching score represents a higher match potential for the two features. Therefore, if we arrange all feature descriptors in image A vertically and all feature descriptors in image B horizontally, a score matrix, which contains scores of all possible matches, can be constructed. The author also adds a dustbin for each set of feature descriptors so that the feature without a corresponding matching feature can be thrown away. Then, the optimal matching layer uses the Sinkhorn algorithm to translate all scores in the score matrix into percentage, that means the sum of any column or row should be one. The principle of the Sinkhorn algorithm is to divide each element of a row/column by the sum of that row/column so that the sum of that row/column becomes 1. The process of applying the Sinkhorn algorithm to all rows/columns in the score matrix is called row/column normalization. The optimal matching layer will iterate row and column normalization T times until the sum of every row and column is 1, and the result is the final partial assignment.

6 Comparsion

One of the applications of feature detection and matching is Homography estimation. Homography estimation lies at the heart of many computer vision applications, serving as a foundational bridge between images of the same scene captured from varying perspectives. At its core, a homography is a 3x3 transformation matrix that describes how points from one image plane correspond to points on another. Homography estimation is also a projective transformation, meaning it preserves straight lines but not necessarily angles or distances, and it is able to capture the complex variation of rotation, translation, scaling, and even perspective shifts between two images. This technique becomes invaluable in scenarios where understanding the spatial relationship between images is crucial, such as in image registration, panorama stitching, and 3D scene reconstruction. In order to estimate the homography, The first step is to extract local features from images and match these features to establish a global correspondence. Then, based on these matched features, we will use an estimator such as RANSAC(Random Sample Consensus) and DLT(Direct Linear Transform) to estimate homography. The difference between RANSAC and DLT is that RANSAC is an estimator with robustness while DLT has no robustness. Usually, incorrect matches of features known as outliers can lead to an inaccurate estimation result. When there are more outliers, RANSAC with robustness works better than DLT because RANSAC is able to discard the outliers that it detects while DLT directly receives all



Local features	Matcher	Homography estimation AUC		P	R
		RANSAC	DLT		
SuperPoint	NN	39.47	0.00	21.7	65.4
	NN + mutual	42.45	0.24	43.8	56.5
	NN + PointCN	43.02	45.40	76.2	64.2
	NN + OANet	44.55	52.29	82.8	64.7
	SuperGlue	**53.67**	**65.85**	**90.7**	**98.3**

Figure 4

matches. But when there are only a few outliers, DLT will produce better results than RANSAC.

Figure 4 is a comparative study on homography estimation, which uses SuperPoint as the local feature extractor and uses SuperGlue, NN and NN's combination with outlier rejector mutual constraint, PointCN, Order-Aware Network (OANet) as the feature matchers. [5] The Homography estimation AUC(Area Under Curve) in the middle is a common evaluation index of estimation accuracy, and higher AUC represents higher accuracy. For more details on the AUC, please read 'An introduction to ROC analysis'. [2] The P and R on the right represents matching precision(P) and recall(R). They are calculated using the following formula:

$$Prec = TP/(TP + FP) \tag{1}$$

$$Recall = TP/(TP + FN) \tag{2}$$

Where TP(True Positive) is the number of positive/correct matches found by a matcher, FP(False Positive) is the number of negative/incorrect matches, and FN(False Negative) is the number of positive/correct matches that were ignored by a matcher. [8] The higher precision means a matcher has more correct matches in all matches found by it. The higher recall means a matcher overlooks fewer correct matches. It is evident from the data that SuperGlue outperforms other matchers in terms of precision and recall, achieving an impressive 90.7% precision and 98.3% recall. This shows that SuperGlue is adept at recognizing most correct matches while simultaneously filtering out incorrect ones. Meanwhile, due to the low precision of NN, non-robust estimator DLT almost got 0 on estimation accuracy, while SuperGlue's match precision is so high that even DLT performs better than RANSAC. This suggests that the matches provided by SuperGlue are predominantly correct, rendering the outlier rejection capability of RANSAC redundant.

7 Conclusion

The evolution of feature detection and matching in computer vision has witnessed a transformative shift from handcrafted techniques like SIFT to the contemporary deep learning-based methods like SuperPoint and SuperGlue. While the traditional method SIFT laid the foundational groundwork, it's the integration of neural networks and deep learning that has significantly elevated the capabilities of feature detection and matching. SuperPoint's proficient feature point extraction combined with SuperGlue's robust matching algorithm underscores a synergistic approach, setting new standards in accuracy and adaptability across various computer vision tasks. This combined efficacy, as observed in homography estimation and pose estimation, affirms the unmatched superiority of deep learning-based methods. As the field of computer vision continues to evolve, deep learning-based methods like SuperPoint and SuperGlue represent the future of development.

Acknowledgments

I would like to thank Peter Dolan for all the support with the research and development process in writing this paper.

References

[1] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2018. SuperPoint: Self-Supervised Interest Point Detection and Description. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 337–33712. https://doi.org/10.1109/CVPRW.2018.00060

[2] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.

[3] David G Lowe. 1999. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, Vol. 2. Ieee, 1150–1157.

[4] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. 2015. Learning Deconvolution Network for Semantic Segmentation. *CoRR* abs/1505.04366 (2015). arXiv:1505.04366 http://arxiv.org/abs/1505.04366

[5] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. SuperGlue: Learning Feature Matching With Graph Neural Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 4937–4946. https://doi.org/10.1109/CVPR42600.2020.00499

[6] Joan Solà. 2007. Towards Visual Localization, Mapping and Moving Objects Tracking by a Mobile Robot: a Geometric and Probabilistic Approach. (02 2007).

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[8] Cui Ni Guangyuan Zhang Wenjun Huangfu Weilong Hao, Peng Wang. 2023. SuperGlue-based accurate feature matching via outlier filtering. (July 2023). https://doi.org/10.1007/s00371-023-03015-5