IE643: Deep Learning - Theory and Practice **July-Nov 2023**

End-term Project Report : Self Supervised keypoints discovry model

Team Name: Visionary Paradox *Team Members:M. Hassan Shaikh , Pranjal S.*

Abstract

This report details our project on a self-supervised key-point discovery model designed to extract meaningful key-points in videos featuring a static camera and one or two moving agents. The model attains results comparable to supervised key-point annotations, employing an encoder-decoder architecture with a geometric bottleneck to reconstruct spatio-temporal differences between video frames. Addressing a limitation of the original model, we specifically emphasize improving performance in videos with shadows. Our refined approach demonstrates enhanced results, underscoring the effectiveness of this self-supervised key-point discovery model.

1 Introduction

The recognition of object structure, conveyed through skeletons or keypoints, is fundamental for a model to comprehend the underlying geometry and movements. However, manual keypoint annotation proves both time-consuming and expensive, particularly on extensive datasets. Supervised models for keypoint detection often lack generalization and are tailored for specific, limited datasets.

In addressing these challenges, this model offers a compelling solution. Its significance lies in automating the keypoint discovery process, mitigating the manual annotation burden. Moreover, the model's adaptability to diverse datasets enhances its applicability beyond narrow, task-specific domains. Leveraging deep learning techniques, this model emerges as an efficient and versatile solution to streamline the keypoint recognition process on a larger scale.

2 Literature Survey

This section highlights earlier research done in the field and also outlines some the methodologies used by various studies that have influenced the model employed in this project.

2.1 Unsupervised Learning of Object Landmarks through Conditional Image Generation [1]

[1] is a recent study that proposes a method for learning landmark (key-point) detectors for visual objects (such as the eyes and the nose in a face) without any manual supervision. This model learns object landmarks from various types of data, including synthetic image deformations and videos without any manual supervision.

It casts this as the problem of generating images that combine the appearance of the object as seen with the geometry of the object, where the two differ by a viewpoint change and/or an object deformation. In order to factorize appearance and geometry, it introduces a tight bottleneck in the geometry-extraction process that selects and distils geometry related features. Our project paper took inspiration from this concept and utilises the geometry-extraction method from this paper.

2.2 Automated monitoring and analysis of social behavior in Drosophila [2]

The system in [2] is designed to monitor interacting fly pairs, it computes key features such as location, orientation, and wing posture. These features are instrumental in detecting behaviors associated with both aggression and courtship. Specific to aggression, the system identifies wing threat, lunging, and tussling, while for courtship, circling, wing extension (courtship 'song'), and copulation are recognized. By automatically constructing ethograms from these measurements, labor-intensive nature of behavioral screening is significantly reduced.

3 Methods and Approaches

3.1 Work done before mid-term project review

The initial phase of the project involved acquiring a comprehensive understanding of the self-supervised keypoint discovery model and conducting background research. Challenges encountered during implementation, such as debugging library code and resolving YAML file incompatibility issues, were successfully addressed. The model was trained on Google Colab, and datasets were curated by recording various subjects, including a plastic mouse, a hand in motion, and the rotation of a pen, all captured under a static camera setting. The model was further trained on a subset of the CalMS21 mice dataset, yielding meaningful results in keypoint identification across diverse datasets.

3.2 Work done after mid-term project review

Give details on the work done after the mid-term project review. Explain how it is significant. Explain the network structures used. Explain all other required details.

During the Mid Term reviews, we got two major comments based on our presentation that is (i) We should understand the code of the encoder decoder architecture and and all the other predefined function, Module and python script ,(ii) We have to figure out how to tackle the shadow problem in key point allocation.

Addressing the shadow problem: We tried three approached as discussed in section Experiments section of this report.

3.3 Neural Network Architecture

Model Architecture:

The model employs an encoder-decoder architecture followed by a final reconstruction decoder. The model takes 2 inputs that are adjacent frames of the dataset.

- **Encoder :** The ResNet50 architecture is a variant of the ResNet (Residual Network) family, known for its effectiveness in training very deep neural networks. Developed by Microsoft Research, ResNet50 specifically refers to a ResNet with 50 layers. ResNet uses residual blocks, which include shortcut connections that skip one or more layers, mitigating the vanishing gradient problem and facilitating

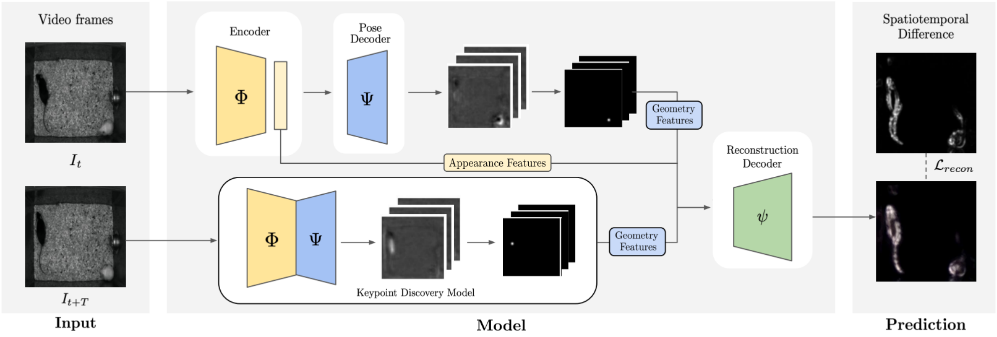
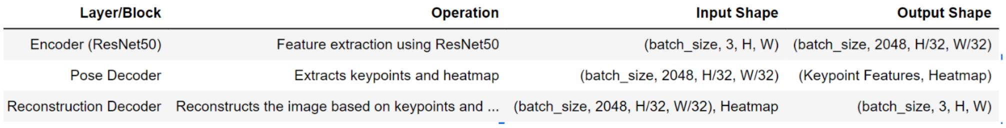


Figure 1: BKinD Model Architecture



Layer/Block	Operation	Input Shape	Output Shape
Encoder (ResNet50)	Feature extraction using ResNet50	(batch_size, 3, H, W)	(batch_size, 2048, H/32, W/32)
Pose Decoder	Extracts keypoints and heatmap	(batch_size, 2048, H/32, W/32)	(Keypoint Features, Heatmap)
Reconstruction Decoder	Reconstructs the image based on keypoints and ...	(batch_size, 2048, H/32, W/32), Heatmap	(batch_size, 3, H, W)

Figure 2: Overview of encoder/decoders used in the model

the training of deep networks. This architecture has proven effective in image classification tasks and serves as the encoder in our model, capturing appearance features of input images.

- **Pose Decoder:** The Pose Decoder in the described model takes the appearance features obtained from the ResNet50-based encoder as input. Its primary goal is to spatially locate and generate meaningful heatmaps of keypoints in the input images. Keypoints are crucial landmarks or distinctive points in an image, and the Pose Decoder aims to identify and highlight these keypoints, contributing to a more detailed understanding of the behavior within the video frames.

- **Reconstruction Decoder:** The Reconstruction Decoder plays a critical role in the model's ability to recreate spatiotemporal differences between adjacent video frames. It takes the appearance features of the initial frame form encoder and geometry features from pose decoder of both the adjacent frames and employs a series of basic block modules (5 times) to decode these features back to the original input dimensions. The reconstructed features are then used to generate an output frame that represents the differences between the input frames.

  The basic block module used in the Reconstruction Decoder typically consists of:

  Upsampling: Increasing the spatial dimensions of the input features.

  Convolution (3x3, Stride 1): A convolutional operation with a 3x3 kernel and a stride of 3, decreasing the number of channels.

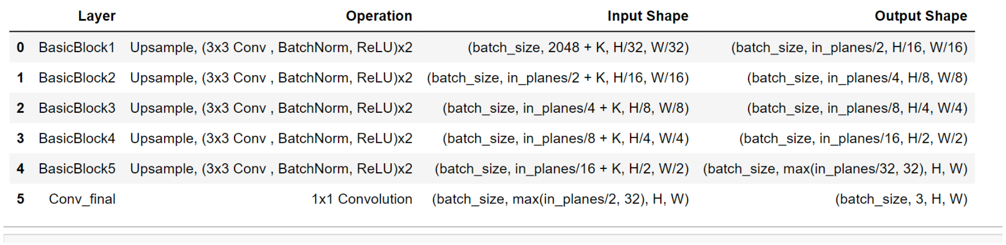
  Batch Normalization: Normalizing the output of the convolutional layer for improved training stability.

  ReLU Activation: Introducing non-linearity through Rectified Linear Unit activation.

  Convolution (3x3, Stride 1): Another convolutional operation with a 3x3 kernel and a stride of 1, preserving the number of channels.

  Batch Normalization: Additional batch normalization.

  ReLU Activation: Another activation function.



	Layer	Operation	Input Shape	Output Shape
0	BasicBlock1	Upsample, (3x3 Conv , BatchNorm, ReLU)x2	(batch_size, 2048 + K, H/32, W/32)	(batch_size, in_planes/2, H/16, W/16)
1	BasicBlock2	Upsample, (3x3 Conv , BatchNorm, ReLU)x2	(batch_size, in_planes/2 + K, H/16, W/16)	(batch_size, in_planes/4, H/8, W/8)
2	BasicBlock3	Upsample, (3x3 Conv , BatchNorm, ReLU)x2	(batch_size, in_planes/4 + K, H/8, W/8)	(batch_size, in_planes/8, H/4, W/4)
3	BasicBlock4	Upsample, (3x3 Conv , BatchNorm, ReLU)x2	(batch_size, in_planes/8 + K, H/4, W/4)	(batch_size, in_planes/16, H/2, W/2)
4	BasicBlock5	Upsample, (3x3 Conv , BatchNorm, ReLU)x2	(batch_size, in_planes/16 + K, H/2, W/2)	(batch_size, max(in_planes/32, 32), H, W)
5	Conv_final	1x1 Convolution	(batch_size, max(in_planes/2, 32), H, W)	(batch_size, 3, H, W)

Figure 3: BKinD Network Architecture for the Reconstruction Decoder

This basic block is repeated 5 times to gradually decode the appearance features to match the original input dimensions. The final layer involves a 1x1 convolution with a stride of 1 to bring the number of channels back to the original value, resulting in the reconstructed frame.

The Reconstruction Decoder aims to capture and represent the spatiotemporal differences between adjacent frames, contributing to the model's understanding of the dynamics and movement patterns within the video sequence.

3.4 Loss Functions

- **Reconstruction Loss :**
  A simple pixel-wise difference often results in a lot of noise, therefore a Structural Similarity Index Measure which takes the product of luminance, contrast and correlation between frames is used in computing this loss.

$$\mathcal{L}_{recon} = \left\| \phi(S(I_t, I_{t+T})) - \phi(\hat{S}(I_t, I_{t+T})) \right\|_2 .$$

Figure 4: Reconstruction Loss

- **Rotational Equivariance Loss :**
  This function is defined to ensure the model learns that agents can rotate too. It finds the difference between original heatmaps but rotated and heatmaps generated from rotated images. This loss must also be the reason that we always get a keypoint in center of the image in our final output images after introducing shadow invariant model, as this loss favours keypoints that are closer to the center due to their radial distance giving less displacement near center.

$$\mathcal{L}_r = \left\| h_g^{R^\circ} - \hat{h}_g(I^{R^\circ}) \right\|_2 .$$

Figure 5: Rotational Equivariance Loss

- **Separation Loss :**
  Keypoints tend to move towards the centre of the image due to rotation equivariance loss. To tackle this we define this loss to ensure the keypoints showcase unique, separated coordinates.

$$\mathcal{L}_s = \sum_{i \neq j} \exp\left(\frac{-(p_i - p_j)^2}{2\sigma_s^2}\right).$$

Figure 6: Separation Loss

- **Final Objective Loss:**
  The Overall loss for our model, where we adopt curriculum learning and apply Lr and Ls once the keypoints are consistently discovered from the semantic parts of the target instance.

$$\mathcal{L} = \mathcal{L}_{recon} + \mathbb{1}_{epoch>n}(w_r\mathcal{L}_r + w_s\mathcal{L}_s).$$

Figure 7: Final Objective Loss

4 Data set Details

The report provides detailed descriptions of the datasets used in the project. The CalMS21 dataset and the custom datasets for computer mice, hands, and pens were utilized. Initial observations revealed issues related to shadow interference in the first two custom datasets, prompting the development of a third dataset that successfully addressed and rectified these issues.

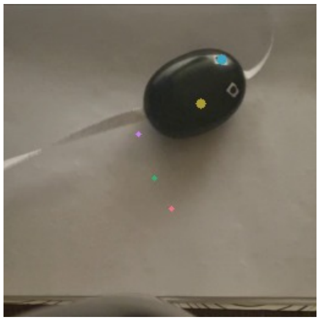


Figure 8: Key points obtained on Computer Mouse(Custom dataset 1)

Custom Dataset 1: Computer Mouse Motion Video footage capturing the motion of a computer mouse, moved by its wire, was recorded. The self-supervised keypoint discovery model was applied to extract motion keypoints, considering spatiotemporal differences. Frames were extracted at a rate of 0.1 frames per second throughout the entire video, revealing insights into the mouse's dynamic behavior.

Custom Dataset 2: Hand Motion In this experiment, the model analyzed hand motion to understand the dynamics of keypoints. The results indicated that shadows were erroneously considered part of the object, leading to incorrect keypoint allocation. This observation prompted the need for further refinement in subsequent approaches to mitigate the impact of shadows.

Custom Dataset 3: Pen Motion and Rotation Video footage capturing the motion and rotation of a pen around a fixed point was recorded. The objective was to identify possible keypoints associated

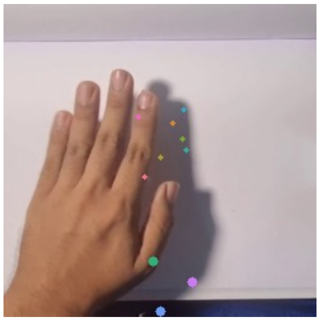


Figure 9: Key points obtained on Hand(Custom dataset 2)

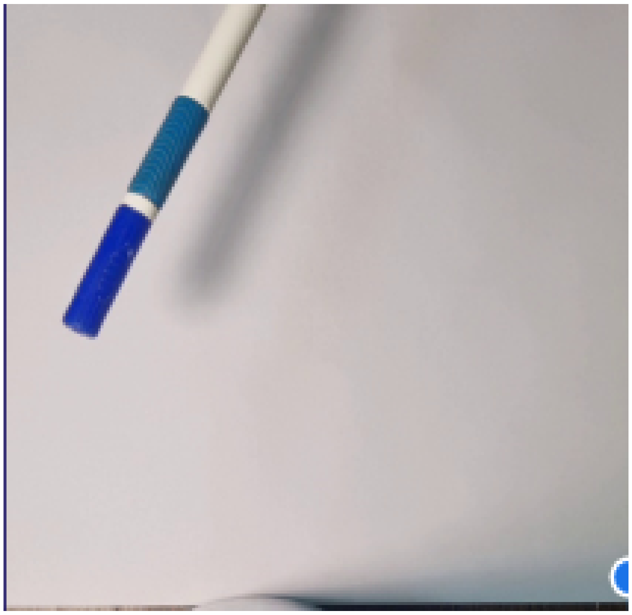


Figure 10: Key points obtained on Pen(Custom dataset 3)

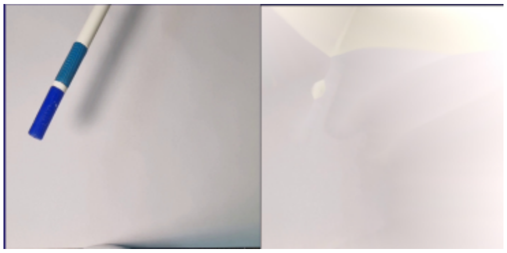


Figure 11: Original image vs Convoluted/ Filtered image

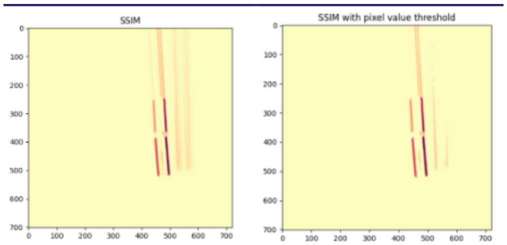


Figure 12: Original image vs Threshold SSIM image

with the pen's movement. This dataset was particularly insightful for assessing the model's performance in scenarios involving rotational motion.

5 Experiments

The primary focus of the experiments was to address challenges related to shadow interference in video analysis as a pre-processing step. Three distinct approaches were employed, each tailored to enhance the accuracy of keypoint identification:

5.1 Convolutional and Filtering

This approach incorporated convolutional operations and filtering techniques to mitigate the impact of shadows on keypoint identification. By applying spatial and temporal filters, the model aimed to enhance the robustness of keypoint detection and minimize the influence of shadow artifacts by equalizing the area of shadows in image but after the process we get a blurry image where even we cant detect the edges.

5.2 Thresholding with SSIM

Utilizing the Structural Similarity Index (SSIM), this approach introduced thresholds to filter out frames with significant shadow distortion. By comparing the SSIM values between consecutive frames, the model identified and excluded frames with undesirable shadow artifacts, contributing to improved keypoint accuracy. Here after the process, the portion of the object is removed instead of shadows and we lost the ability to distinguish between them so we are not going to use this as the prep-processing stage.

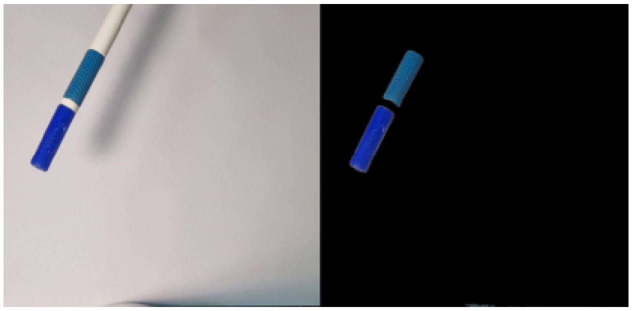


Figure 13: Original image vs bitwise AND operated image

5.3 Bitwise AND Operation

Motivated by insights from the reference book ”Digital Image Processing” [4], this approach employed bitwise AND operations to eliminate shadow-related anomalies in video frames. By selectively preserving pixel information common to consecutive frames, the model successfully removed shadow interference, resulting in enhanced keypoint identification accuracy by segmenting only the part of object.

To utilise approach 5.3 we need to consider a condition on our dataset that is: It has uniform color on the object and relatively higher brightness compare to its background. Custom Dataset 3 adheres to this assumption, capturing the motion and rotation of a pen under conditions of uniform color of pen and higher brightness(Blue pen vs white background)

So we implemented the removeshadowscolor() function[3] defined in another colab notebook which input the folder from the drive as inputfolder where are images extracted from the video. These images are converted to HSV space since we got a better idea of intensity information and color information separately, Then by masking and applying the bit and operation we get the outputs and store them at the outputfolder which will be used for the model. The code for the same is added to the github repository as a new folder named removeshadowcolor function[3][4.

These experiments collectively aimed to refine the self-supervised keypoint discovery model, addressing challenges posed by shadow interference and improving its performance across diverse datasets. The success of the third approach underscored its efficacy in eliminating shadows and enhancing the overall accuracy of keypoint identification.

6 Results

This section will elaborate on the results we obtained from the experiments. we can see that the key points after applying removeshadowcolor() function[3] keypoints lie exactly where they should have been. This is because we extracted or segmented our object and trained on that we are able to only track the object while the rest were blacked out.

Another interesting observation is the green color keypoints which lie approximately at the middle of the images of output of Custom dataset trained on the model with Bit-wise And operation applied(fig 14), we can reason it as the component of total loss i.e. rotation loss tries to fit keypoints in the center and hence giving us green point at center signifying the axis of rotation of pen in the video.

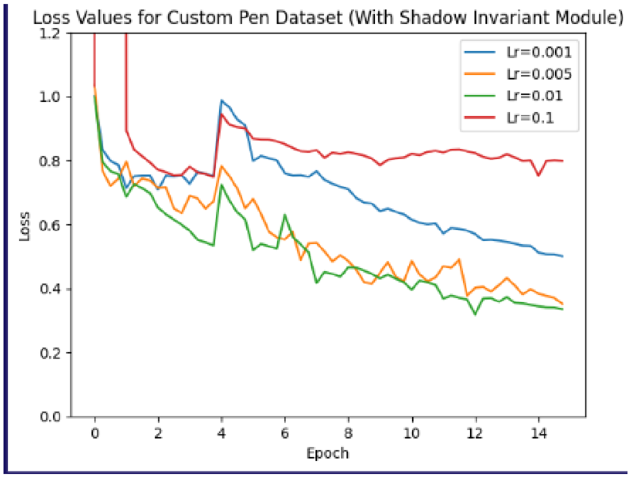


Figure 14: Model Performance

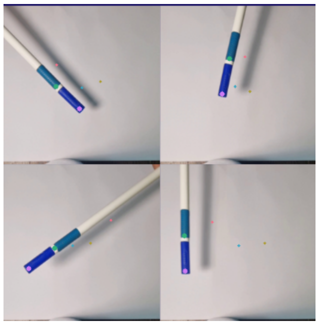


Figure 15: Custom dataset trained on the model without applying Bit-wise And operation

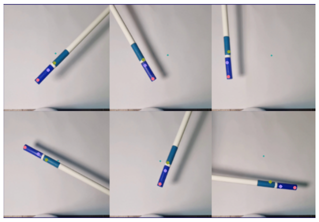


Figure 16: Custom dataset trained on the model with applying Bit-wise And operation

7 Future Work

The report discusses potential enhancements for future work, including the implementation of a dedicated convolutional filtering module to further improve shadow problem resolution. Additionally, exploring applications beyond assumed conditions, such as scenarios involving a dynamic camera, is identified as a promising avenue for future research.

8 Conclusion

The report concludes by highlighting the successful achievement of project implementation with solving shadow invariance in object recognition, underscoring the task-specific nature of the methodology. The segmentation of image portions necessary for keypoint discovery resulted in exceptional results, providing valuable insights into computer vision and the architecture used in the model for addressing the problem statement.

References

[1]Self Supervised Keypoint Discovery in Behavioral Videos (CVPR 2022)

[2]BKinD GitHub Repository:https://github.com/neuroethology/BKinD

[3]Updated repository with Remove shadow function: https://github.com/hassan-byt0/BKinD.git

[4]Rafael C. Gonzalez, Richard E. Woods. ”Digital Image Processing”

[5]Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. ”Image quality assessment: from error visibility to structural similarity,” IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, April 2004. doi: 10.1109/TIP.2003.819861

[6]Bengio, Yoshua et al. “Curriculum learning.” International Conference on Machine Learning (2009)

[7]Justin Johnson, Alexandre Alahi, Li Fei-Fei. ”Perceptual Losses for Real-Time Style Transfer and Super-Resolution”

[8]Yuting Zhang1, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, Honglak Lee. ”Unsupervised Discovery of Object Landmarks as Structural Representations”.University of Michigan, Ann Arbor

[9]Tomas Jakab, Ankush Gupta, Hakan Bilen, Andrea Vedaldi. ”Unsupervised learning of object landmarks through conditional image generation”

[10]Cristina SegalinJalani WilliamsTomomi KarigoMay HuiMoriel ZelikowskyJennifer J SunPietro PeronaDavid J AndersonAnn Kennedy. ”The Mouse Action Recognition System (MARS) software pipeline for automated analysis of social behaviors in mice”

[11]Heiko Dankert, Liming Wang, Eric D Hoopfer, David J Anderson, Pietro Perona. ”Automated monitoring and analysis of social behavior in Drosophila”

[12]Jennifer J. Sun, Ann Kennedy, Eric Zhan, David J. Anderson, Yisong Yue, Pietro Perona. ”Task Programming Learning Data Efficient Behavior Representations” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 2876-2885