Enhancing Keypoint Detection with Attention Mechanism

Chung-Pang Wang
UCSD ECE
A59025305

Niyas Attasseri
UCSD ECE
A59022844

Abstract

This study explores the enhancement of keypoint detection in images by utilizing attention mechanisms, including attention, multi-head attention, and transformers encoder, to integrate with a VGG19-based encoder-decoder network. The detected keypoints are utilized to generate camera-to-robot transformations from single images. To minimize the simulation-to-reality gap, the model is trained on a domain-randomized simulated dataset. By incorporating attention into the later layers of the network, we observe significant improvements in keypoint detection performance.

1 Introduction

Determining the camera-to-robot transformation is a fundamental challenge in robotic manipulation. This transformation is essential for converting observations from the camera module into the robot's perspective, enabling effective planning and control. Traditional methods for finding these transformations rely on fiducial markers such as ArUco, ARTag, or AprilTag attached to the robot's end effector. By capturing a sequence of images corresponding to various robot configurations, these methods solve a homogeneous linear system to determine the unknown transformation. However, this approach is limited to offline calibration and requires re-calibration if there is any disturbance to the system, such as camera movement relative to the robot. This necessitates restarting the entire calibration process from scratch whenever the camera position changes.

An alternative approach involves using deep learning to establish an implicit relationship between each RGB image and the transformation. This method demands precomputed calibration for each camera location and environment, making it non-generalizable to new tasks and environments without retraining. Therefore, there is a significant need for a tool that enables online, generalizable camera-to-robot calibration.

A promising direction in this field is the Deep Robot-to-camera Extrinsics for Articulated Manipulator (DEAM) framework by Timothy et al. [1], which determines the 2D projections of keypoints of the robotic manipulator in RGB images. By combining this information with the camera intrinsics, the robot joint configuration, and forward kinematics, the transformation is estimated using the Perspective-n-Point (PnP) algorithm [5]. The network is trained on synthetic data generated with domain randomization to bridge the simulation-to-reality gap.

The DREAM framework has demonstrated the feasibility of generating the camera-to-robot transformation using a single image of the robot without fiducial markers. Its results are comparable to classic hand-eye calibration methods that use multiple frames, and its accuracy can be further improved by increasing the number of frames used. The model has shown good results on different robot manipulators (Franka Emika's Panda, Kuka's LBR iiwa 7 R800, and Rethink Robotics' Baxter) and with a variety of cameras.

The DREAM framework utilizes an auto-encoder network to detect keypoints. With the advent of transformers and attention mechanisms, we hypothesize that the model can be significantly improved

by incorporating attention within the network for generating keypoints. By integrating attention mechanisms into a VGG19-based encoder-decoder network, we aim to enhance the accuracy and robustness of keypoint detection, leading to more reliable camera-to-robot transformations in real-time scenarios. Our approach leverages the strengths of attention to improve feature extraction and representation, potentially surpassing the performance of existing methods.

The integration of attention mechanisms and transformer models into the traditional VGG architecture has shown promising improvements in keypoint detection accuracy, particularly in challenging visual recognition tasks. These models significantly reduce false identifications of out-of-frame keypoints, demonstrating robust boundary discernment capabilities. Some model not only minimizes false positives but also excels in identifying in-frame keypoints, achieving the highest in-frame Area Under Curve (AUC), which underscores its superior spatial accuracy and overall efficacy.

2 Related Work

The DREAM framework by Timothy et al. [1] has demonstrated the feasibility of estimating camera-to-robot transformations using deep learning. DREAM utilizes an auto-encoder network to detect 2D projections of keypoints on a robotic manipulator from a single RGB image. This method effectively bridges the simulation-to-reality gap by training on synthetic data generated with domain randomization. DREAM's approach is notable for its ability to operate without fiducial markers and achieve accuracy comparable to traditional hand-eye calibration methods. Our work builds on this foundation by incorporating attention mechanisms to potentially enhance keypoint detection accuracy and robustness.

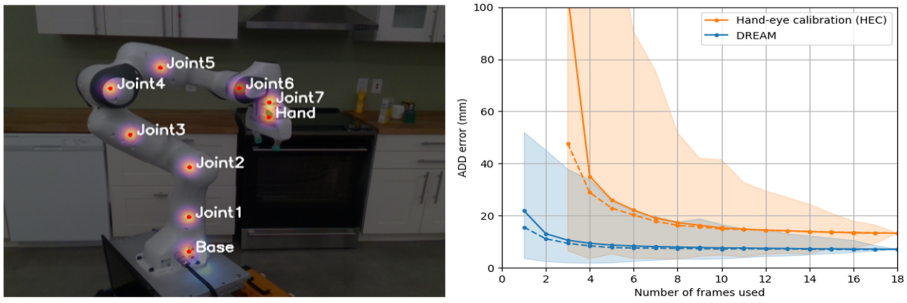

Figure 1: Left: Keypoint belief maps (red dots indicate peaks) detected by DREAM in RGB images on Franka Emika Panda taken by Intel RealSense D415. Right: DREAM vs. HEC, measured by ADD as a function of the number of image frames used for calibration. Shown are the mean (solid line), median (dashed line), and min/max (shaded area), computed over different image combinations. DREAM requires only a single image frame but achieves greater accuracy with more images.

The seminal paper "Attention Is All You Need" by Vaswani et al. [2] introduced the Transformer model, which relies entirely on self-attention mechanisms, dispensing with recurrent and convolutional neural networks entirely. This architecture has revolutionized various tasks in natural language processing and has been adapted for numerous computer vision tasks. The core idea of self-attention allows the model to weigh the importance of different parts of the input data dynamically, improving the model's ability to capture long-range dependencies and complex patterns. In our approach, we leverage the principles of the Transformer model to enhance the detection of keypoints in robotic manipulation tasks. By integrating self-attention mechanisms, we aim to improve the feature extraction and representation capabilities of our neural network, leading to more accurate camera-to-robot transformation estimates.

SegViT [4] proposes an effective structure using plain ViT transformer backbones for the semantic segmentation task. For the first time, SegViT utilizes spatial information in attention maps for semantic segmentation. To implement this idea, the authors introduced an Attention-to-mask (ATM) module that can derive mask predictions during the attention calculation process. SegViT has demonstrated efficiency and state-of-the-art performance across various semantic segmentation benchmarks. This work highlights the effectiveness of leveraging attention mechanisms and spatial

information in vision tasks, which inspires our approach to improve keypoint detection for robotic calibration.

3 Method

An externally mounted camera observes n keypoints $\mathbf{p}_i \in \mathbb{R}^3$ on various robot links. These keypoints project onto the image as $\mathbf{k}_i \in \mathbb{R}^2,\ i = 1, \ldots, n$. Some of these projections may be inside the camera frustum, whereas others may be outside. We consider the latter to be invisible/inaccessible, whereas the former are visible, regardless of occlusion. The network learns to estimate the positions of occluded keypoints from the surrounding context. Technically, since the keypoints are the robot joints (which are inside the robot links), they are always occluded. The intrinsic parameters of the camera are assumed known.

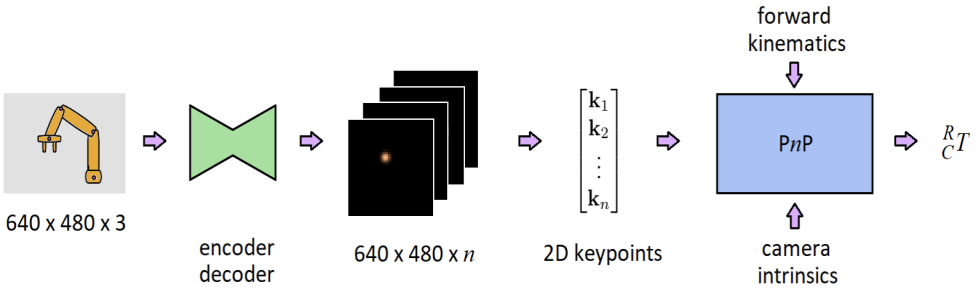

Figure 2: The DREAM framework. A deep encoder-decoder neural network takes as input an RGB image of the robot from an externally-mounted camera, and it outputs n belief maps (one per keypoint). The 2D peak of each belief map is then extracted and used by PnP, along with the forward kinematics and camera intrinsics, to estimate the camera-to-robot pose, ${}^{R}_{C}T$.

Our proposed two-stage process for solving the problem of camera-to-robot pose estimation from a single RGB image frame. First, an encoder-decoder neural network processes the image to produce a set of n belief maps, one per keypoint. Then, Perspective-n-Point (PnP) uses the peaks of these 2D belief maps, along with the forward kinematics of the robot and the camera intrinsics, to compute the camera-to-robot pose, $\mathbf{R}^T_C$. Note that the network training depends only on the images, not the camera; therefore, after training, the system can be applied to any camera with known intrinsics. We restrict $n \geq 4$ for stable PnP results.

3.1 Network Architecture

Inspired by recent work on object pose estimation , we use an auto-encoder network to detect the keypoints. The neural network takes as input an RGB image of size $w \times h \times 3$, and it outputs an $w \times h \times n$ tensor, where $w = 640$, $h = 480$ and n is the same as the number of keypoints. In the previous work [1], they consider a downsample factor $\alpha \in \left\{1, \frac{1}{2}, \frac{1}{4}\right\}$ to crop the image to $\alpha w \times \alpha h \times n$. We found that in most of the cases outputting full images help the later PnP process to be more robust. Therefore, we will use $\alpha = 1$ in our work. The output captures a 2D belief map for each keypoint, where pixel values represent the likelihood that the keypoint is projected onto that pixel.

The encoder consists of the convolutional layers of VGG19 [9] pretrained on ImageNet. The decoder (upsampling) component is composed of four 2D transpose convolutional layers (stride = 2, padding = 1, output padding = 1), and each layer is followed by a normal 3×3 convolutional layer and ReLU activation layer. We also experimented with VGG19-Attention , VGG19-Multi-headAttention and VGG19-Transformer encoder. We add attention, multi-head attention (4 heads) and transformer encoder after the last layer of VGG19 and right before maxpooling. Furthermore, We test with attention layer in several convolutional blocks right before maxpooling to see how adding too much of attention layers affect the performance of the model.

The output head is composed of 3 convolutional layers (3×3, stride = 1, padding = 1) with ReLU activations with 64, 32, and n channels, respectively. There is no activation layer after the final convolutional layer. The network is trained using an L_2 loss function comparing the output belief maps with ground truth belief maps, where the latter are generated using $\sigma = 2$ pixels to smooth the peaks.

The network learns to estimate the positions of occluded keypoints from the surrounding context; technically, since the keypoints are the robot joints (which are inside the robot links), they are always occluded.

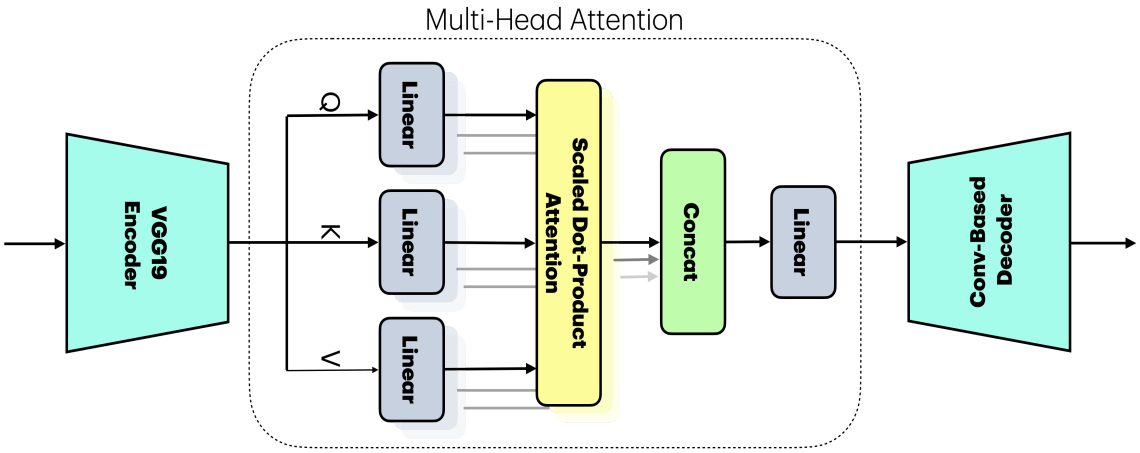

Figure 3: Network Architecture of our proposed VGG-MHA

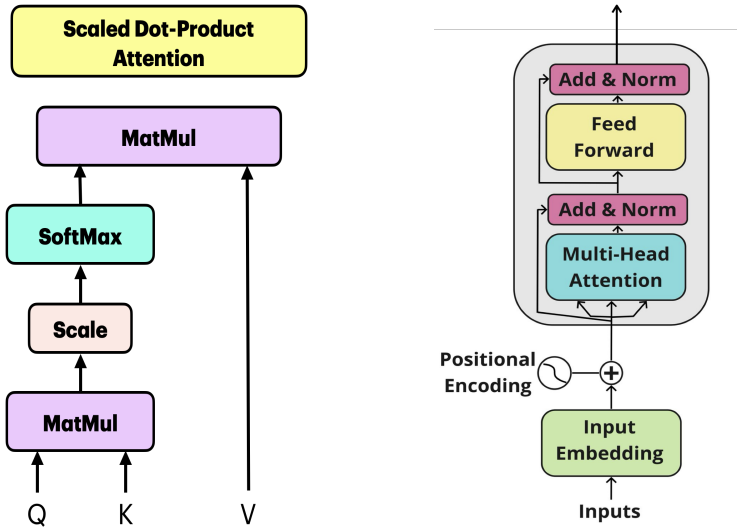

Figure 4: Transformer encoder (Left) and attention mechanism (Right)

4 Experiments

4.1 Dataset

The dataset comprises RGB images of the robotic arm. Each image is annotated with the corresponding keypoints that the model aims to detect. Additionally, for the Perspective-n-Point (PnP) module, camera intrinsics and the robot's forward kinematics are provided.

4.1.1 Training

We used the same synthetic data generated by the DREAM framework on the Panda manipulator for training. This dataset consists of synthetic images generated for a Panda robotic arm in various configurations. The configurations are created by varying the robot joint angles, with the camera positioned freely in a somewhat truncated hemispherical shell around the robot. The data also incorporates domain randomization to bridge the simulation-to-reality gap. This includes randomizing lighting conditions with varying intensity and color, selecting scene backgrounds randomly from the COCO dataset, and placing random objects from the YCB dataset in the environment. These variations create a diverse and robust training set, enabling the model to generalize better to real-world scenarios.

4.1.2 Testing

For testing, we used the real-world images generated by DREAM. These images were captured in their lab using a Microsoft Azure Kinect camera and feature the Panda robotic arm. The robot was moved to five different joint configurations at which the camera collected data, resulting in a dataset consisting of nearly 6k image frames. This setup allows us to evaluate the model's performance in a real-world scenario, verifying the effectiveness of the model trained on synthetic data.

4.2 Metrics

To evaluate the performance of our model, we use the following metrics:

Training Loss
This metric tracks the training error value during the training process, providing insight into how well the model is learning to minimize the error in keypoint detection. The validation error value helps understand if the model is overfitting to the training data.

Percentage of Correct Keypoints
Percentage of correct keypoints (PCK) measures the accuracy of keypoint detection by evaluating the percentage of keypoints that are within a specified threshold distance from the ground truth keypoints. This metric helps assess the precision of keypoint localization.

Average Distance
Average distance (ADD) evaluates the average Euclidean distance between the actual 3D keypoints and their transformed versions using the estimated transform. It combines both rotation and translation errors without having to define an arbitrary weighting between them. This metric is crucial for understanding the accuracy of the model in estimating the camera-to-robot transformation.

4.3 Ablation study

To understand the contribution of different components of our model, we conduct an ablation study with the following variations:

Baseline Model: A VGG19-based encoder-decoder network without any attention mechanisms. This serves as the baseline to compare the impact of adding attention.

Single Attention Layer: A model with a single attention layer added to the late layer of the VGG19-based encoder network. This variation helps evaluate the impact of incorporating a single attention layer on keypoint detection accuracy.

Multihead Attention: A model with multihead attention layers integrated into the network. This setup assesses the benefits of using multiple attention heads to capture different aspects of the input data and improve feature extraction.

Transformer-encoder Block: A model with a transformer-encoder block added at the end of the VGG19-encoder. This variation allows us to compare the effectiveness of the transformer-encoder block in enhancing the keypoint detection capacity of the model.

Attention with Different Configurations: We experiment with different configurations of attention mechanisms, such as varying the number of attention heads and layers, to determine the optimal setup for keypoint detection.

By analyzing the results from these variations, we aim to identify the most effective components and configurations for enhancing keypoint detection accuracy and robustness in our camera-to-robot transformation framework.

4.4 Training and Testing Results

From Figure 5, it can be observed that the training loss curves for all models start relatively high but decrease rapidly, suggesting effective learning early in training across all configurations. As epochs increase, all models continue to improve, with VGG-MHA showing the fastest and most significant decrease in training loss, closely followed by VGG-Transformer and VGG. The VGG-Att model, while improving, shows the slowest rate of decline in training loss. This indicates that the enhancements provided by the MHA and Transformer components lead to better optimization and faster convergence during training compared to the standard VGG architecture.

The validation loss curves reveal a similar pattern to the training loss curves, with the VGG-MHA model achieving the lowest validation loss, implying it generalizes best on unseen data. The validation loss for all models decreases over the first 10-15 epochs before plateauing, suggesting early effective learning that stabilizes as the models begin to fit the data well.

The observations indicate that integrating attention mechanisms and transformer structures into the traditional VGG architecture significantly boosts its performance, manifesting not only in accelerated learning but also in attaining lower loss levels. This enhancement likely leads to higher accuracy and improved generalization to new data. Notably, the training curves suggest that the models have not fully converged, as indicated by the non-zero slopes at the end of 25 epochs. This incomplete training is due to constraints in time and computational resources; each model required approximately 10 hours to train for 25 epochs on an Nvidia RTX A6000 GPU.

In comparison, the DREAM paper reports results from models trained for 50 epochs, demonstrating further potential improvements in performance with extended training periods. However, due to our limitations, we have restricted our comparisons to the original VGG model trained for the same 25 epochs. This approach ensures a fair assessment of enhancements contributed by the attention and transformer mechanisms under equivalent training conditions.

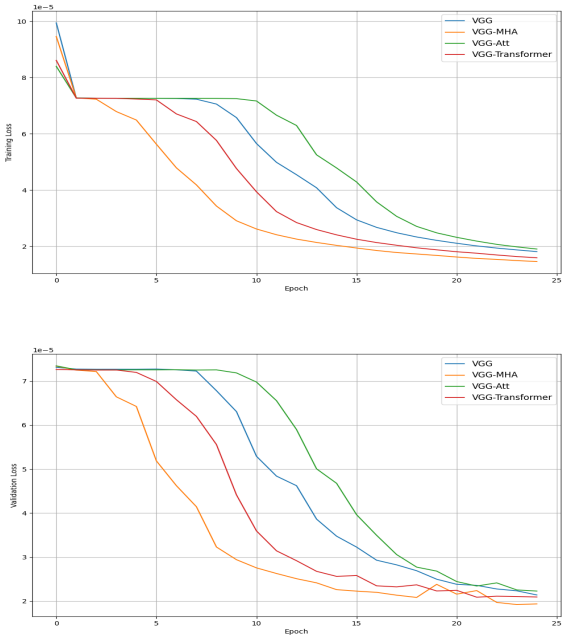

Figure 5: Training and Validation loss

In Figure 6, the first plot representing Average Distance(ADD), showcases the models' accuracy with increasing threshold distances in millimeters. It is evident that all models perform increasingly better as the allowed threshold for errors increases. VGG-MHA shows the highest accuracy across nearly all thresholds, as indicated by the highest area under the curve (AUC) of 0.680. This suggests that integrating multi-head attention with the traditional VGG architecture significantly enhances its keypoint detection accuracy. The VGG-Att model also performs robustly with an AUC of 0.607, followed closely by VGG-Transformer at 0.579, and the standard VGG model trailing with an AUC of 0.500. This hierarchy in performance underscores the benefits of attention mechanisms and Transformer features in improving the precision of pose predictions over the baseline VGG model.

The second plot in figure 6 represents Percentage of Correct Keypoints (PCK) metric, measuring the percentage of keypoints falling within various pixel thresholds. The curve trends demonstrate that the VGG-Att model slightly outperforms other configurations with an AUC of 0.677, closely followed by VGG-MHA at 0.656 and VGG model at AUC of 0.652. The VGG-Transformer scores the lowest at 0.627. Unlike the ADD results, where VGG-MHA led the group, the PCK results highlight the VGG-Att's strengths in keypoint localization within tighter pixel thresholds, suggesting its particular effectiveness in tasks requiring high precision in spatial alignment.

These findings illustrate that while all models enhance upon the VGG baseline in various aspects of keypoint detection, the specific improvements depend on the type of architectural integration and the specific task at hand.

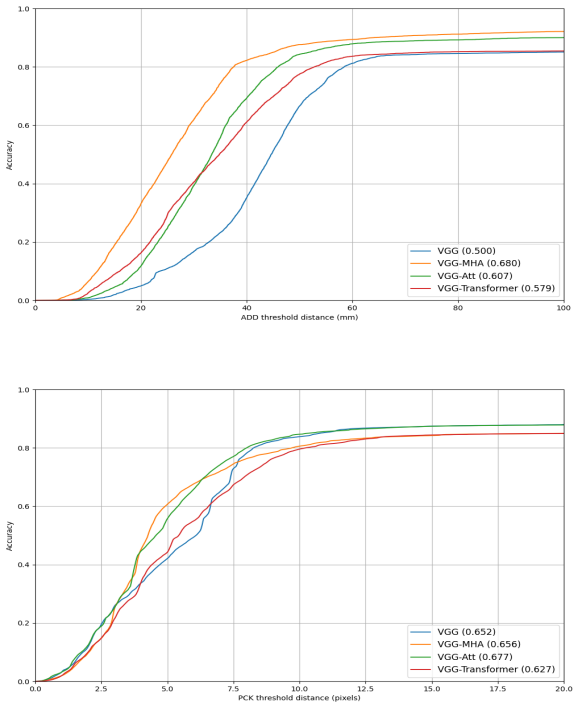

Figure 6: PCK (top) and ADD (bottom) results on the Azure dataset

The performance of our attention-based model outwin the baseline model as shown in a Tab.1. Our models outperforms in the percentage out-of-frame gt keypoints not found of than the baseline model VGG, meaning that the percentage of false detect keypoints is very low in attention-based model. We believed the reason is that attention mechanism can learn the subtle contextual information that the convolutional-only model hard to learn. However, we still see some limitations of the framework where there are some cases that the keypoints are missing. Fig. 7 shows different poses of panda arm that keypoints are well detected, we can see that in 5 different poses, most of the keypoints are well aligned except for the hand. We believed this is because the appearance of the hand is very different from other joints. Therefore, the hand is more likely to be considered as the background and not

detected. Furthermore, the model is not generalized on the poses where the joints are overlapped visually. As shown in Fig. 8, the hand and the first joint are overlapped in the left figure and cause the keypoint of the first join missing. Although the keypoints are the robot joints which are inside the robot links, the model is always tried to detected occluded keypoints, the model only sees the joints on the top (the one only visible) when two joints overlapped together.

Figure 7: Keypoints detection of different poses. The Red points and texts are the prediction while the blue ones are the ground truth

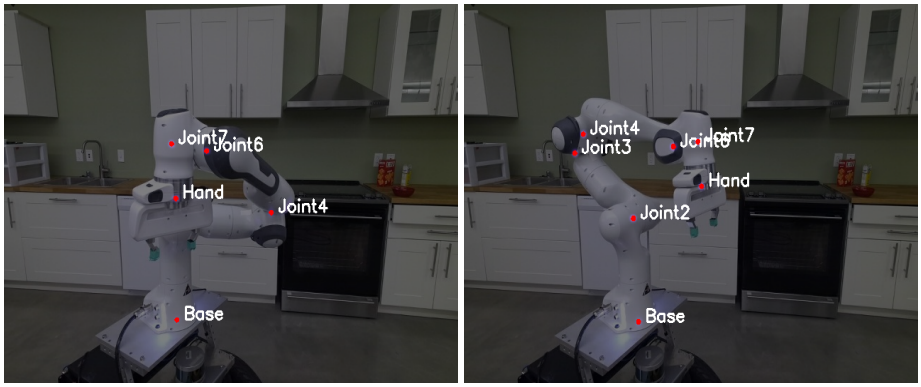

Figure 8: Pose of missing keypoints prediction with occuled/overlapped joints (Left) and the pose pf keypoints detected perfectly detected (Right)

Table 1: Performance Comparison of percentage out-of-frame gt keypoints not found (correct), percentage in-frame gt keypoints found (correct) and L2 error (px) for in-frame keypoints area under the curve (AUC)

Model	Out-of-frame	In-frame	In-frame AUC
VGG	68.76	89.87	65.18
VGG-MHA	**99.52**	85.71	65.65
VGG-ATT	98.74	**89.91**	**67.72**
VGG-Transformer	98.27	86.28	62.70

5 Conclusions

The evaluation of the VGG, VGG-MHA, VGG-ATT, and VGG-Transformer models across various metrics demonstrates the nuanced performance differences brought about by incorporating attention mechanisms and transformer models into the baseline VGG architecture. While all enhanced models significantly reduce false positives for out-of-frame keypoints, VGG-ATT proves superior in accurately detecting in-frame keypoints and achieves the highest In-frame AUC, suggesting its effectiveness in spatial accuracy and keypoint localization. These insights highlight the value of integrating advanced neural network enhancements to improve precision and reliability in tasks requiring detailed spatial awareness, such as keypoint detection within complex visual scenes.

5.1 Limitations

In our observations, the keypoints of the Panda arm were generally well detected across various poses. However, the detection accuracy for the hand was notably lower compared to other joints. This discrepancy is likely due to the distinct appearance of the hand, which causes it to be mistaken for the background and not detected as reliably. Additionally, the model struggled to generalize well to poses where joints visually overlap. Specifically, when joints overlapped, the model often failed to detect the occluded joint and only identified the visible one. This limitation highlights the challenge of accurately detecting keypoints in scenarios with visual occlusions.

6 Acknowledgment

We would like to extend our sincere thanks and appreciation to Prof. Wang and the instructional staff, Nicklas Hansen, Isabella Liu, and Jiteng Mu, for providing an excellent course and continuous support. The lectures covered the latest techniques in visual learning, helping us understand and apply methods to improve existing models.

References

1. Lee, T. E., Tremblay, J., To, T., Cheng, J., Mosier, T., Kroemer, O., Fox, D., & Birchfield, S. (2020). Camera-to-Robot Pose Estimation from a Single Image. In *International Conference on Robotics and Automation (ICRA)*.
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
3. Xu, M., Zhang, C., Jiang, L., Liu, D., & Yuan, J. (2022). ViTPose: Simple Vision Transformer Baselines for HumanPose Estimation.
4. Zhang, B., Tian, Z., Tang, Q., Chu, X., Wei, X., Shen, C., & Liu, Y. (2022). SegViT: Semantic Segmentation with Plain Vision Transformers. In *NeurIPS*.
5. V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," International Journal Computer Vision, vol. 81, no. 2, 2009.
6. S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: Dense 6D pose object detector in RGB images," arXiv:1902.11020, 2019.
7. S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in CVPR, 2019.
8. Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6D object pose estimation," in CVPR, 2019.
9. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in ICLR, 2015.
10. Transformers and Multi-Head Attention Tutorial
11. VGG-Net Architecture Explained