THE UNIVERSITY OF ELECTRO-COMMUNICATIONS

DOCTORAL DISSERTATION

Deep Learning Techniques for 2D Hand Pose Estimation Integrating Spatial Attention-based Contextual Features for Enhanced Efficiency

---

Author:

Sartaj Ahmed Salman

Supervisor:

Hiroki Takahashi

A thesis submitted in fulfillment of the requirements

for the degree of Doctor of Philosophy

in the

School of Informatics and Engineering

Department of Informatics

The University of Electro-Communications

July 31, 2024

Declaration of Authorship

I, Sartaj Ahmed Salman, declare that this thesis titled, " Deep Learning Techniques for 2D Hand Pose Estimation Integrating Spatial Attention-based Contextual Features for Enhanced Efficiency " and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. Except for such quotations, this thesis is entirely my work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work I did jointly with others, I have clarified exactly what others did and what I have contributed myself.

Signed: Sartaj Ahmed Salman

Date: July 24, 2024

Abstract of the Dissertation

Hand Pose Estimation has appeared essential in advanced computer vision tasks, particularly for real-world AI applications, such as human-computer interaction, augmented reality, and virtual reality. However, this task is challenging, mainly due to the complex nature of hand joints, dexterity, and self/object occlusion. This dissertation presents several innovative algorithms to overcome these issues in 2D hand pose estimation from RGB images. Firstly, we propose a novel network that combines traditional graph-based probabilistic models with deep convolutional neural networks to integrate the hand's structural constraints, improving the accuracy of hand pose estimation. Despite the accuracy improvements, it comes with a significant computational cost. To tackle this, we streamline the model, making it more shallow while maintaining accuracy by incorporating attention mechanisms and efficient feature extractors to learn spatial features effectively. However, achieving state-of-the-art performance remains a challenge due to the delicate nature of human hands. To address this, we integrate a global contextual module and deformable convolutions to aid the models in learning contextual information and geometrical constraints. These enhancements efficiently increase the precision while maintaining low computational cost. We conducted extensive experiments on the publicly available CMU dataset, and our approaches demonstrate state-of-the-art performance. By integrating these novel approaches, our research aims to push the boundaries of hand pose estimation and contribute to developing more robust and accurate AI systems for real-world applications.

Acknowledgements

I extend my heartfelt gratitude to my respected advisor, **Professor Hiroki Takahashi**, whose unwavering support and encouragement have been instrumental throughout my academic journey. His invaluable feedback, profound knowledge, and guidance have significantly enriched my understanding of the research topics. I am grateful for the opportunity to be part of his laboratory and for his unwavering belief in my abilities. His positive energy, patience, and unwavering support during challenging times in my Ph.D. course have uplifted me.

I would also like to express my sincere appreciation to my co-supervisors, **Professor Keji Yanai** and **Professor Hayaru Shouno**, for their insightful guidance, expertise, and support throughout my research. Their contributions have been invaluable in shaping my academic work and broadening my perspective. I would also like to sincerely thank my dear brother, friend, and lab mate, Ali Zakir, whose unwavering support and assistance have been invaluable since my master's degree and throughout my Ph.D. journey. His companionship and support have been a source of strength and inspiration.

I am also thankful to all my lab members, especially Yuki San, for their support and friendship over the years. Their friendship and collaborative spirit have enriched my research journey and provided me with the emotional support I needed during challenging times. Their presence has made a significant difference in my academic journey. To my beloved family and friends in Pakistan, I owe a debt of gratitude that words cannot fully express. Your unwavering support, encouragement, and love have been the bedrock of my journey. Your belief in my abilities and unwavering support have been my pillars of strength and filled my heart with love and gratitude. I am deeply appreciative of your contributions to my success.

I am very grateful to every individual who has played a role, big or small, in shaping my academic and personal journey. Your support and encouragement have been invaluable, and I am deeply appreciative of your contributions to my success.

Contents

List of Figures

List of Tables

List of Abbreviations

HPE	**H**and **P**ose **E**stimation
CV	**C**omputer **V**ision
AR	**A**ugmented **R**eality
HCI	**H**uman **C**omputer **I**nteraction
DL	**D**eep **L**earning
CNN	**C**onvolutional **N**eural **N**etworks
DCNN	**D**eep **C**onvolutional **N**eural **N**etworks
SA	**S**patial **A**ttention
GCB	**G**lobal **C**ontextual **B**lock
SE	**S**queeze-and-**E**xcitaion
DC	**D**eformable **C**onvolution
CPM	**C**onvolutional **P**ose **M**achine
FIM	**F**irst **I**nference **M**odule
SIM	**S**econd **I**nference **M**odule
GIM	**G**raphical **I**nference **M**odule
PGM	**P**ose **G**raph **M**odel
PCK	**P**ercentage of **C**orrect **K**eypoints
MSE	**M**ean **S**quared **E**rror

Chapter 1

Introduction

Nowadays, the role of computers and robots in society is rapidly growing and advancing [1]. As we exceed previous technological difficulties, our everyday lives progressively depend on varied interactions between humans and technology. Yet, with the advancement of these technologies comes increased complexity, sometimes leading to complex interaction methods that can complicate their use [2, 3]. Simplifying these interactions to reflect human-like communication involves employing natural dialogue with virtual assistants and connecting Computer Vision (CV) for applications like emotion recognition, Augmented Reality (AR), and precise human and hand pose estimation, enhancing our engagement with technology [4, 5, 6].

Hand Pose Estimation (HPE) methods are essential for identifying key hand points in images or videos, fundamental for Virtual Reality (VR), Human-Computer Interaction (HCI), and other areas of CV [7, 8, 9]. Substantial progress has been made in estimating hand poses, yet practical applications face numerous hurdles [10]. One major challenge is acquiring datasets; the shortage of labeled training data is difficult in HPE [11, 12]. Neural networks need wide-ranging, labeled datasets for training, and obtaining detailed real-hand data is particularly challenging, often due to complex backgrounds in natural acts that complicate image processing tasks like lighting, viewing angles, weather conditions, and more [13, 14, 12]. Such background complexity

can restrict with accurately estimating hand poses, where issues such as self-occlusion and object interference further complicate the process [11]. Previously, the research community struggled to address these challenges using machine learning algorithms. However, these methodologies often relied on manually designed features, limiting the model's learning capacity and generalizability [15, 16]. While deep learning advancements have progressed in overcoming some of these problems, searching for comprehensive solutions that can adeptly handle the intricate aspects of visual recognition and hand motion is ongoing. Thus, despite improvements in precision, speed, and efficiency, developing more advanced and adaptable HPE technologies remains an active area of research [15, 17, 18].

Generally, two methods are utilized for HPE, including detection-based and regression-based [15, 19, 20]. Detection-based methods are popular in HPE because of their reliability and precision. These models work by creating a heatmap for each necessary point on the hand, showing how likely it is for that point to be at each pixel in the image. Often, these heatmaps are made using a two-dimensional Gaussian distribution focused on the keypoints, which helps pinpoint their locations more accurately by giving higher importance to pixels near the keypoint [21, 22]. In relation to detection-based methods and their application in creating heatmaps for HPE, the process focuses on locating specific hand positions, referred to as keypoints, that illustrate the hand's exact posture, labeled as P. This procedure evaluates RGB images or video sequences, indicated as I. Each keypoint, denoted by k_i, is linked to a particular area on the hand, like joints or fingertips, and is deduced from separate heatmaps H. The aim is to forecast the heatmaps for every keypoint $H_1, ., ., H_i$. As a result, the pose P is depicted as a collection of coordinates $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$, with each coordinate pinpointing the most probable location on its respective heatmap. The total number of keypoints K varies with the dataset but often includes 21 points. The essence

of this task is to calculate the pose P by pinpointing these keypoints, a process meticulously detailed in Algorithm 1, which maps out the steps for estimating the hand pose P from an RGB image or video frame I by detecting keypoints and generating corresponding heatmaps to identify the most probable coordinates for each point.

Data: RGB image or video frame I
Result: Estimated hand pose P represented as keypoints
Initialize $P = \emptyset$ (Set to store keypoints);
Detect keypoints K representing distinct hand regions in I;
for $i = 1$ *to* K **do**
  Generate heatmap H_i for k_i in I;
  Extract coordinates (x_i, y_i) with highest probability from H_i;
  Add (x_i, y_i) to P as a keypoint;
end
Return P as the estimated hand pose;

Algorithm 1: 2D Hand Pose Estimation using Heatmaps

On the other hand, Regression-based methods estimate the coordinates of keypoints of the hand directly from the input image [23, 24]. Despite lower computational costs than detection-based approaches, regression-based methods often face the problem of occlusion and complex hand configuration problems. Despite limitations, regression-based methods remain relevant for real-time applications due to their speed and simplicity. Regression-based methods face problems of robustness to spatial generalization and occlusion. Incorporating prior knowledge into the regression framework improves performance but can lag behind detection-based approaches in complex scenarios [23, 24, 22, 21]. In this dissertation, we have utilized detection-based methods, and all proposed architectures are based on this approach. Detection-based methods provide the most effective means of achieving our objectives, and we have worked hard to ensure that our proposed architectures are optimized for this approach.

1.1 Problem Statement

Despite considerable improvements in Deep Learning (DL) network-based methods for 2D HPE, there remains a critical challenge in achieving a balance between computational cost and accuracy because there is always a trade-off between accuracy and computational cost. Current models, while effective, often require massive computational resources, making them less practical for real-time applications or edge devices with limited processing capabilities. This limits their feasibility in fields such as VR, AR, and HCI, where precision and efficiency are required. Furthermore, existing techniques may not fully capture the complex geometrical constraints and spatial features characteristic of hand movements, leading to a negotiation in accuracy or performance. This research proposes to address these gaps by developing a novel 2D HPE model that minimizes computational resources without losing accuracy and includes novel strategies like attention mechanisms, global contextual modules, and deformable convolutions to enhance feature learning and geometrical understanding. Throughout this research, we seek to determine a new benchmark in HPE that is efficient and highly accurate, outfitting the needs of advanced applications in interactive technologies.

1.2 Objective

Enhance Model Efficiency Without Losing Accuracy—A primary objective is to enhance the 2D HPE with advanced computational approaches that reduce its complexity. This includes exploring efficient neural network architectures like COnvNext and the EfficientNets family to lower computational demands. The objective is to develop a lightweight and highly accurate model suitable for real-time applications, utilizing knowledge from structural optimization and resource-efficient methodologies.

Enhance Feature Extraction and Geometrical Understanding - Our second objective focuses on enhancing the model's ability to identify complex spatial details and navigate the complex geometry of hand movements. To achieve this, we incorporate unconventional techniques like attention mechanisms that direct the model's focus to essential features while suppressing the less important features and global contextual modules that offer a comprehensive view of the entire scene. We will also investigate using deformable convolutions, which adjust convolutional filters for better orientation with the unpredictable shapes and positions of hands. We aim to increase the model's effectiveness and precision in estimating 2D hand poses by providing advanced feature detection and geometry recognition capabilities.

1.3 Contribution

To advance the 2D HPE field, this dissertation presents a comprehensive approach containing innovative techniques to enhance model performance and robustness. The key contributions of my dissertation are as follows:

Utilization of Efficient Backbone Models

Efficiency in computational resources is not just a theoretical concern but a practical necessity for deploying HPE models across various platforms. By leveraging lightweight and efficient backbone models, such as ConvNext and EfficientNet B0, we aim to strike a delicate balance between model complexity and computational efficiency. The adoption of these efficient architectures in the proposed model helps the model to make a balance between accuracy and computational complexity.

Incorporation Attention Mechanisms

Attention mechanisms are crucial in guiding the model's focus toward salient spatial features, enhancing discriminative power and robustness in HPE. This dissertation integrates spatial attention and Squeeze-and-Excitation mechanisms into the model architecture. These attention mechanisms enable the model to dynamically allocate attention to the most informative regions within the input data, effectively suppressing noise and irrelevant information. The model better captures complex hand configurations through adaptive attention mechanisms, advancing state-of-the-art HPE research.

Further feature Enhancement with Global

Contextual Block Integrating a global contextual module is essential for enriching feature representation and enhancing pose estimation accuracy. To this end, a Global Contextual block is incorporated into the model architecture. This block facilitates the integration of holistic spatial context, enabling the model to consider the relationships and dependencies between different hand regions within the scene. The model gains a deeper understanding of hand configurations by capturing global contextual information, improving inference accuracy and robustness in diverse real-world scenarios.

Deformable Convolution

Hand poses exhibit complex geometric deformations that require flexible and adaptive modeling techniques. Deformable convolution layers are employed within the model architecture to effectively capture these geometric variations. Unlike traditional convolutional layers, deformable convolution layers enable the model to learn spatial transformations flexibly, enhancing feature extraction and localization accuracy. By accommodating spatially variant filters and dynamic, receptive fields, the model becomes adept at capturing

fine-grained spatial details and geometric intricacies inherent in hand poses, resulting in superior performance and generalization ability.

Incorporating efficient backbone models, attention mechanisms, global contextual information, and deformable convolution represents an effort to advance the state-of-the-art 2D HPE. Collectively, these contributions contribute to developing a robust and accurate HPE model with significant implications for applications in human-computer interaction, robotics, and augmented reality.

1.4 Dissertation Structure

This dissertation is structured as follows: Chapter 1 provides the primary context by providing background information, an overview of the problem statement, a summary of the objectives, details of the contributions, and an overview of the dissertation's structure. Following this introduction, the dissertation moves on to chapter 2 related work and Chapters 4 to 7, each dedicated to exploring the complexities of 2D HPE. These chapters utilize various methods, including probabilistic graphical models and deep convolutional and graph neural networks. Concluding this dissertation, Chapter 7 summarizes the findings and provides insights and recommendations from the research. It also outlines potential directions for further exploration in this dynamic field and lays the groundwork for future endeavors.

Chapter 2

This chapter presents the literature review related to the HPE.

Chapter 3

To address the constraints inherent in current techniques for estimating 2D hand pose from RGB images, we introduce a novel framework called Spatial Attention-Based Deep Pose Graph Network (SDPoseGraphNet) in this chapter. This framework builds upon our prior work by incorporating the Spatial-Attention (SA) module of the VGG-19 model into the backbone. The SA module empowers the network to dynamically emphasize significant spatial areas within the input image, a critical aspect for precise pose estimation. By integrating this module, we aim to augment the model's ability to represent features, enhancing its performance across various hand pose estimation tasks.

This chapter was previously published in a multidisciplinary journal, Sensors.

Chapter 4

In this chapter We introduced a compact CPM approach tailored to 2D HPE to make the previous approach more computationally efficient. Our strategy uses a customized ConvNext architecture as the backbone, keeping the convolution layer while discarding the fully connected layer. This adjustment aims to improve feature extraction, especially for HPE tasks. Including the Global Context Block (GCB) is a notable improvement in our methodology. This addition allows us to leverage the backbone model's feature extraction capabilities and detect global context information from features extracted by the convolution layer.

This chapter was previously published in an International Workshop on Image Technology (IWAIT 2024).

Chapter 5

However, the computational cost of the previous method is relatively high. A new architecture named Attention-Driven Contextual Features Based Convolution Network (ACENet), which involves a shallower network design, is introduced to overcome this challenge. EfficientNet is used with a squeeze-and-excite attention mechanism to improve feature extraction and enable efficient learning of spatial features.

This work was previously published in IVCNZ 2023 Image and Vision Computing.

Chapter 6

Recognizing that previous methods are still insufficient in computational efficiency, in this chapter we have developed a new solution to further reduce computational cost without compromising accuracy: A multi-stage deformable convolution network specifically designed for 2D hand HPE. We introduce the Deformable Pose Network (DPN), a multi-stage deformable convolution network. Our approach overcomes the challenges mentioned earlier by using deformable convolutions that prioritize the incorporation of geometric constraints into the convolutional operations. Meanwhile, the network backbone overcomes additional computational hurdles by addressing the handling of hidden information. This integrated strategy strives to increase efficiency while maintaining high accuracy in HPE.

This chapter was previously published in The 19th International Joint Conference on Computer Vision, Imaging, and Computer Graphics Theory and Applications (VISAPP 2024).

Chapter 7

In a continuous effort to improve the efficiency of our model, in chapter 7 we have made adjustments to the DPN and made it shallower. Furthermore, in the quest for higher accuracy, we have introduced an innovative architecture: Attention-Driven Contextual Feature-Enhanced Deformable Convolutional Based Network for 2D Hand Pose Estimation (ACDCNet). This innovative model integrates the Squeeze-and-Excitation (SE) attention mechanism and the GCB into the EfficientNet backbone. By incorporating these components, the system gains the ability to learn both spatial and contextual information efficiently, reducing computational costs while maintaining accuracy.

This research chapter will be submitted to an interdisciplinary journal to evaluate further and disseminate the results.

Chapter 8

This chapter concludes the dissertation and gives the future direction for HPE.

Chapter 2

Related Work

2D hand pose estimation is crucial in computer vision, with applications in human-computer interaction, sign language recognition, and gesture-based control. HPE is a challenging task due to variations in hand poses, limited depth information, and issues related to appearance and occlusion. We will discuss Multi-View RGB-based, Depth-based, and RGB-based 2D hand pose estimation methods.

2.1 Multi-View RGB Approaches

Multi-view RGB models offer a promising approach to address these challenges by leveraging information from multiple camera viewpoints. The work by Simon et al. in [25] introduced a multi-view RGB model that combines features from multiple views to estimate hand poses. Their approach utilizes a convolutional neural network (CNN) to extract view-invariant features from each RGB image, which are then fused using a view-pooling layer. This multi-view fusion strategy effectively mitigates self-occlusion and improves the overall accuracy of HPE. Similarly, In [26], Sun et al. proposed a multi-view RGB model incorporating a hierarchical multi-scale feature aggregation module. This module combines features from different scales and views, enabling the model to capture local and global information for accurate HPE. Their approach demonstrated superior performance compared

to single-view and other multi-view methods, particularly in scenarios with severe occlusion. While multi-view RGB models have shown promising results, specific camera setups often limit their practical implementation. For instance, the models proposed by Chen et al. in [27] and Ge et al. in [28] require a fixed number of cameras positioned at predetermined locations around the target area. Such rigid camera configurations can limit the performance of these models in real-world scenarios where camera placements may vary or be suboptimal.

Researchers have explored more flexible multi-view RGB models that can adapt to different camera configurations to address this limitation. For example, in [29] Yuan et al. introduced a multi-view RGB model that can dynamically adjust its feature fusion strategy based on the available camera views. Their approach employs an attention mechanism to adaptively weight the contributions of different views adaptively, enabling the model to handle varying camera setups effectively. Despite these advancements, multi-view RGB models still face challenges regarding computational complexity and data requirements [27, 28]. As the number of camera views increases, the computational cost of feature extraction and fusion can become prohibitive, especially for real-time applications. Additionally, training these models often requires large-scale multi-view datasets, which can be time-consuming and expensive to acquire [30, 7].

2.2 Depth-Based Approaches

Depth-based models for HPE leverage depth information captured by specialized sensors, such as time-of-flight cameras or structured light sensors. These models offer several advantages over Multi-view RGB approaches, including accurate hand localization and robustness to variations in lighting

conditions and appearance [31, 32]. One of the pioneering works in depth-based HPE is the method proposed by Ge et al. in [28]. Their approach utilizes a CNN to directly estimate hand poses from depth maps, eliminating the need for intermediate representations or post-processing steps. By leveraging the inherent 3D information in depth maps, their model can effectively capture the complex spatial relationships between hand joints, improving accuracy.

While depth-based models have demonstrated impressive performance in controlled environments, they can be sensitive to noise and environmental factors that affect depth data quality. For instance, ambient lighting conditions, sensor limitations, and occlusions can introduce artifacts or missing depth values, which can degrade the performance of these models [32, 31]. One notable depth-based method is the work by Tompson et al. in [33], which uses a CNN to directly regress the 2D hand joint locations from depth images. The authors demonstrate that their approach can accurately estimate hand poses in real time, making it suitable for interactive applications. Oberweger et al. present another depth-based approach [23]. They propose a feedback loop-based architecture to refine the hand pose estimation iteratively. Their method leverages depth information to guide the refinement process, improving accuracy and robustness.

However, depth-based hand pose estimation methods also have limitations. They may struggle with hand occlusions, where parts of the hand are obscured from view, leading to inaccuracies in pose estimation. Additionally, depth sensors may encounter challenges in environments with complex backgrounds or varying lighting conditions, affecting depth data quality and impacting hand pose estimation accuracy. Furthermore, depth-based methods may require careful calibration and alignment of depth sensors to ensure accurate depth measurements, adding complexity to the setup process [32, 31].

2.3 Monocular RGB-Based Approaches

With the widespread availability of RGB cameras and the advancements in deep learning, RGB-based methods have gained significant attention in HPE. These methods aim to directly estimate hand poses from RGB images, eliminating the need for specialized depth sensors or multi-view camera setups [34]. One of the key challenges in RGB-based HPE is the accurate localization and estimation of 2D hand keypoints, which serve as an essential intermediate representation for 3D pose estimation. Panteleris et al. [35] and Zimmermann et al. [36] emphasized the importance of accurate 2D HPE for overall performance in 3D HPE techniques. Their work demonstrated that even minor errors in 2D keypoint localization can propagate and amplify in the subsequent 3D pose estimation stage, leading to significant inaccuracies.

One approach to RGB-based HPE is holistic regression, where a CNN is trained to directly regress the hand pose from the input RGB image. In [37], Tekin et al. proposed holistic regression models that capture global constraints and correlations between keypoints, eliminating the need for intermediate representations for human pose estimation, which can apply to hand pose estimation. However, these methods can suffer from generalization issues and sensitivity to translational variance, limiting their performance in real-world scenarios. Heatmap-based methods, such as Convolutional Pose Machines (CPM) [38] and Optimized Convolutional Pose Machine (OCPM) [39], have emerged as a popular and effective approach for RGB-based HPE. These methods leverage CNNs to predict heatmaps, representing the likelihood of each hand joint being present at different spatial locations. Heatmap-based methods can achieve precise hand keypoint localization by combining these heatmaps with additional refinement stages, enabling accurate 2D pose estimation [40, 41, 42, 43].

Recent advancements in RGB-based HPE have focused on improving these

models' robustness and generalization capabilities. For instance, in [44], Wan et al. introduced a self-supervised learning approach that leverages synthetic data and domain adaptation techniques to enhance the generalization of RGB-based HPE models across different domains and scenarios. Despite the significant progress in RGB-based HPE, challenges remain in handling occlusions, complex backgrounds, and varying lighting conditions [45, 46].

Chapter 3

Spatial Attention Based Deep Pose Graph Network

Introduction

The Convolutional Pose Machine (CPM) [38] is proficient at generating robust feature maps, but often struggles with capturing geometric correlations between joints. Consequently, this can lead to inconsistencies in the final predictions of joint positions, presenting a significant hurdle in tasks related to human pose estimation. This challenge is magnified in 2D Hand Pose Estimation (HPE) due to increased articulation and self-occlusion, exacerbating the issue. To address these limitations, we introduce the Spatial Attention Based Deep Pose Graph Network (SDPoseGraphNet) in this chapter. This novel framework enhances VGG-19 [47] capabilities with Spatial Attention (SA) [48], as depicted in Figure 3.1.

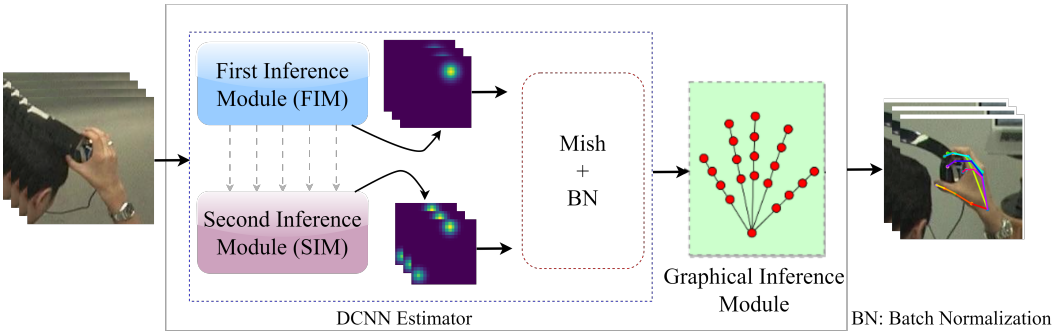


Figure 3.1: Illustration of the SDPoseGraphNet architectural design.

3.1 SDPoseGraphNet Architecture

SDPoseGraphNet consists of the First Inference Module (FIM) and the Second Inference Module (SIM). A final Graphical Inference Module (GIM) integrating deep convolutional neural networks (DCNNs) with the Pose Graph Model (PGM) connects FIM and SIM sequentially. FIM provides an initial feature score for hand keypoints during the preliminary stage, easily integrated with the reverted feature score from the VGG-19 block in SDPoseGraphNet. The final module utilizes parameters generated by SIM to represent spatial constraints among critical hand keypoints. This framework sets itself apart by leveraging SIM, enabling the association of the PGM with Deep Convolutional Neural Networks (DCNN)[49, 50].

In our approach, parameters are not treated as independent; instead, they are tightly coupled with the input image through VGG-19, ensuring adaptability and responsiveness to varying input images. This integration effectively captures and utilizes relevant information from the input image, resulting in improved performance and adaptability across different scenarios as shown in Figure 3.6.

The process of predicting hand poses is formally described through a graph represented by $G = (V, \mathcal{E})$, where Here, the vertices $\mathbf{V}$ are directly associated with the salient keypoints of the hand, denoted as K, and can be expressed as $V = \{v_1, v_2, \ldots, v_k\}$. Each vertex v_i corresponds to a specific two-dimensional keypoint, represented as $x_i \in \mathbb{R}^2$, which provides the position of that keypoint relative to v_i. Equation (3.1) expresses the joint probability of hand poses, modeling the interrelationships between keypoints and their positions within a graphical structure.

$$p\left(\mathbf{X} \setminus I, \Theta\right) = \frac{1}{Z} \prod_{i=1}^{|V|} \phi_i(x_i \setminus I; \Theta_f) \prod_{(i,j)\in\mathcal{E}} \varphi_{i,j}\left(x_i, x_j \setminus I; \Theta_s\right) \tag{3.1}$$

In the equation, $X = \{x_1, x_2, \ldots, x_K\}$ denotes the set of hand keypoints, where i and j represent their positions. The term $|\boldsymbol{V}|$ indicates the cardinality, or number of elements, in set $\boldsymbol{V}$, $\boldsymbol{Z}$ is the partition function, and $\mathbf{I}$ correspond to the input image. The parameter $\boldsymbol{\Theta}$ encompasses the combination of the FIM and SIM; $\boldsymbol{\Theta} = \{\boldsymbol{\phi}_i(x_i \setminus \mathbf{I}; \boldsymbol{\Theta}_f); \boldsymbol{\varphi}_{i,j}(x_i, x_j \setminus \mathbf{I}; \boldsymbol{\Theta}_s)\}$. In this context, the equation models the joint probability of hand poses by considering the interrelationships between keypoints and their positions within a graphical structure. It provides a formal representation of the predictive process used to estimate hand poses, where various components, such as hand keypoints and the input image, are considered in calculating this probability. The parameter $\boldsymbol{\Theta}$ encapsulates the combination of specific modules contributing to the overall predictive model.

Further details and comprehensive explanations regarding each component of SDPoseGraphNet are provided in the subsequent subsections.

3.2 Intergration of VGG19 with Spatial Attention for Enhanced Feature Extraction

Attention mechanisms in neural networks enhance the focus on essential parts of input data while reducing the significance of less relevant components, a technique known as visual attention. This approach has seen significant advancements in deep learning research and has proven effective for text and image data. Various methods incorporating visual attention have been developed to improve convolutions' efficiency. This study proposes a novel HPE task approach by integrating a SA [48] module with the VGG-19 [47] model. This fusion capitalizes on the feature extraction capabilities of the VGG-19 architecture and the ability of attention mechanisms to highlight salient spatial regions, creating a powerful combination [51].

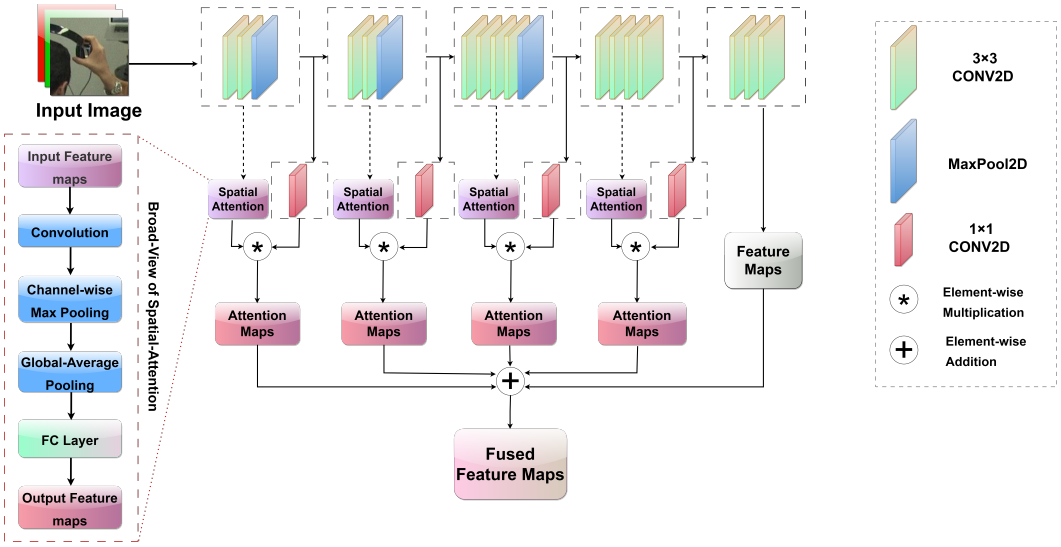


Figure 3.2: Architecture of VGG-19 with SA module for enhanced 2D HPE.

The SA [48] modules enable each feature map to employ a distinct attention mechanism. These attention maps are aggregated along the channel dimension and then passed through a convolutional layer with a kernel size k to produce the final attention map. The final attention map undergoes normalization through a sigmoid activation function to ensure the values are within the range of 0 and 1.

VGG-19 generates five distinct four attention maps and one feature map, as depicted in Figure 3.2. SA and $\mathbf{1} \times \mathbf{1}$ convolutions for channel reduction are applied to the first four feature maps to obtain the attention maps.

$$\mathbb{F}_i = \hat{\mathbf{F}}_i \otimes \mathbf{F}_i \tag{3.2}$$

$\hat{\mathbf{F}}_i$ is the feature map after channel reduction. This convolutional layer series reduces the feature maps' channel dimension to 128. $\otimes$ denotes element-wise multiplication, and $\mathbb{F}_i$ represents the attention feature map.

Bilinear interpolation from torch.nn.functional aligns the spatial dimensions, as shown in Equation (3.3).

$$\hat{\mathbf{S}}_i = \boldsymbol{F.interpolate}(\mathbb{F}_i, t) \tag{3.3}$$

$\hat{\mathbf{S}}_i$ represents attention maps after interpolation, and t denotes the target size of the last feature map without spatial attention. After interpolation, the attention maps are fused with the last feature map from the backbone by element-wise addition, as shown in Equation (3.4).

$$\mathcal{S} = \hat{\mathbf{S}}_1 + \ldots + \hat{\mathbf{S}}_4 + f_5 \tag{3.4}$$

Here, $\mathcal{S}$ represents the fused feature map.

3.3 Operational insights of FIM, SIM, and GIM

In the initial module, the VGG-19 [47] architecture was utilized up to Conv $\mathbf{3} \times \mathbf{3}$ as the primary feature extraction network, followed by three additional convolutional layers to generate the initial heatmap. A SA module was incorporated to enhance the VGG-19 architecture, generating 128 feature maps. These feature maps then undergo information processing through a six-stage module comprising continuous convolution layers with a specific kernel size of $\mathbf{3} \times \mathbf{3}$, incorporating a heatmap label as a supervisory mechanism.

The heatmap labels were generated by applying a Gaussian function to the corresponding ground truth, as expressed by Equation (3.5):

$$Heatmap = exp\left(\frac{-[(x - x_k)^2 + (y - y_k)^2]}{2\delta^2}\right) \tag{3.5}$$

where δ represents the extent of the heatmap, and x_k and y_k denote the coordinates of the keypoints. The final stage produces 21 unique feature maps, each representing a keypoint. These feature maps serve as static weights during the training of the graphical module. The output of the initial module is denoted as $F(I;\Theta_f) \in \mathbb{R}^{|V| \times h_{F-heatmap} \times w_{F-heatmap}}$, where the dimensions of the output heatmaps are determined by the corresponding values of height

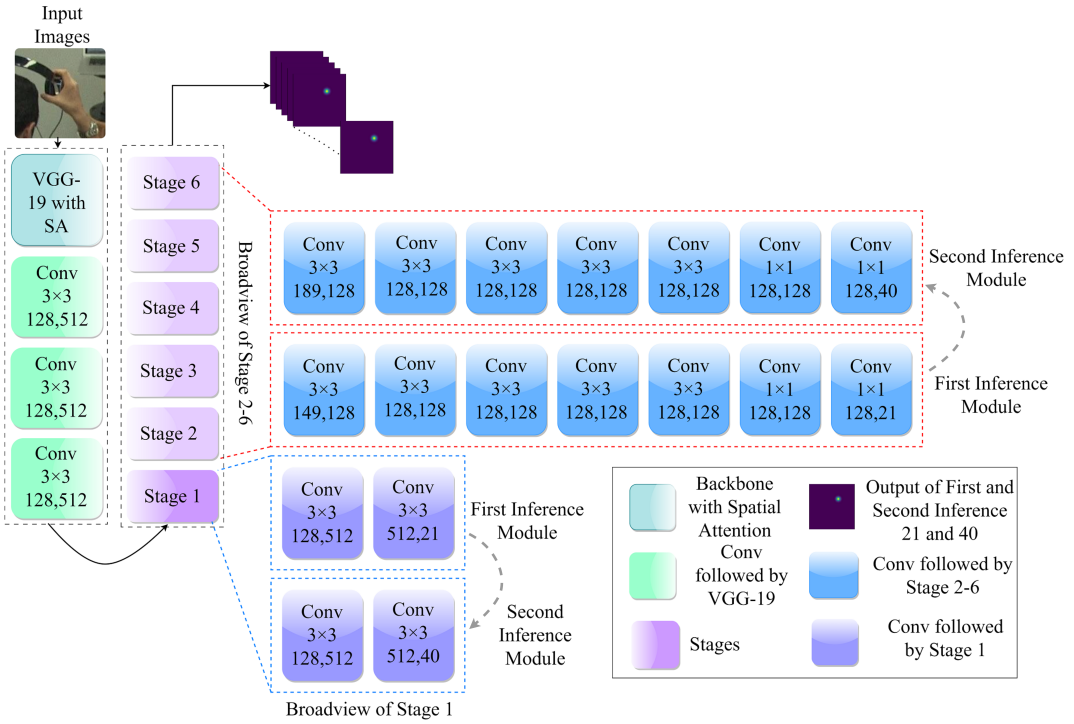


Figure 3.3: Comprehensive overview of First and Second Inference Modules (FIM and SIM).

$h_{F-heatmap}$ and width $w_{F-heatmap}$. Figure 3.3 illustrates the sequential convolutional detailing of the FIM.

The SIM follows a methodology similar to the FIM, maintaining the framework for 128 feature maps but generating 40 instead of 21. These 40 feature maps represent information about the relationships between hand keypoints, capturing details about relative positions, distances, and interactions between different pairs of keypoints on the hand. The output produced by the SIM is denoted as $S(I;\Theta_S) \in \mathbb{R}^{|E|\times h_s \times w_s}$, ($h_s$ and w_s height and width of the input image) indicating the SIM channel kernels. The primary objective of the SIM is to learn the relative positions between hand keypoints.

During the SIM training, we keep the weights of the FIM fixed, effectively 'freezing' them. As depicted by the gray arrows in Figure 3.1 and 3.3, there is a directional flow of information from the FIM to the SIM at the end of each stage. Throughout this information flow process, the feature sets generated at each FIM stage merge with those from the corresponding SIM stages.

For example, the features from the first stage of the FIM combine with those from the first stage of the SIM, and this composite feature set feeds into the second stage of the SIM. This consistent information exchange approach is maintained across all SIM's training phase stages, as illustrated in Figure 3.3.

The message-passing algorithm is widely used in GIM, enabling the effective calculation of marginal probabilities through the sum-product operation within a graphical module. The equation for marginal probability is expressed in Equation (3.6):

$$p_i(x_i \setminus I; \Theta) = \sum_{\mathbf{V} \setminus x_i} p(X \setminus I; \Theta) \tag{3.6}$$

In this context, the argmax probability function optimizes the marginal probability for predicting the location of the hand keypoint labeled as i, as shown in Equation (3.7):

$$x_i = argmax\, p_i(x_i \setminus I; \Theta), \tag{3.7}$$

Here, $\Theta = \{\Theta_f, \Theta_s\}$ represents the collection of all parameters, amalgamating the parameters of the initial two modules.

In the graphical model, each vertex $\mathbf{V}$ can send and receive messages $\mathbf{M}$ to and from its corresponding neighboring nodes Nbn. The sum-product algorithm updates messages sent from hand keypoints from i to j. The complete message exchange is denoted by $\mathbf{M}ij$, with $\mathbf{M}ij \in \mathbb{R}^{h_w \times w_u}$ representing the message passing formulation, as shown in Equation (3.8):

$$\mathbf{M}_{ij}(x_j) = \sum_{x_i} \varphi_{i,j}(x_i, x_j) \phi_i(x_i) \prod_{k \in Nbn(i) \setminus j} m_{ki}(x_i) \tag{3.8}$$

After multiple iterations and convergence, marginal probabilities are approximated as shown in Equation (3.9):

$$p(x_j) \approx \frac{1}{\mathcal{Z}} \phi_i(x_i) \prod_{k \in Nbn(i)} m_{ki}(x_i), \tag{3.9}$$

Here, $m_{ki}(x_i)$ represents a message from node k to node i, and $\mathcal{Z}$ is the normalization constant.

This research adopted a tree-structured graphical model, which accurately derives marginal probabilities using belief propagation. Figure 3.4 illustrates the hand model arranged in a tree-like structure, facilitating precise marginals by transmitting messages from the bottom-most nodes to the topmost node and then back down to the lowest nodes. The schedule of message updates is denoted by the number 3, with 40 message transmissions sufficient for obtaining accurate marginals.

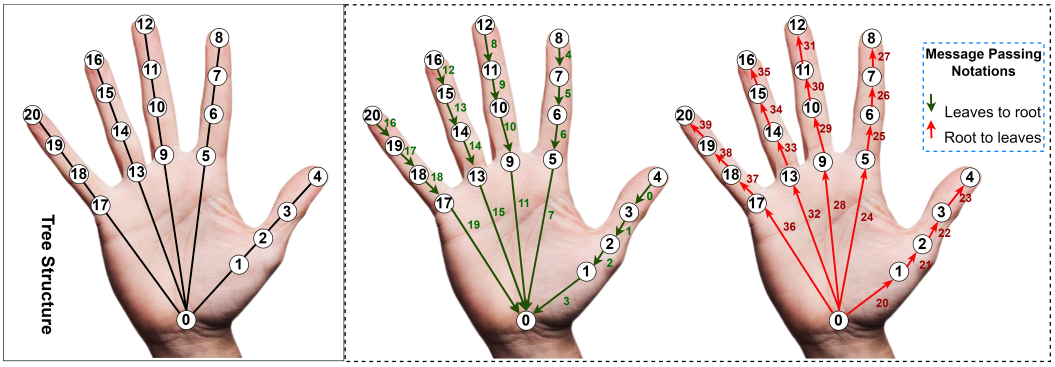


Figure 3.4: Illustrative representation of message passing within a hand tree structure.

3.4 Experimental Setups

The proposed model is implemented using the PyTorch framework version $\mathbf{1.11.0 + cu102}$. The present model underwent a three stages training process, where each stage was trained with a consistent learning rate of $\mathbf{1e^{-04}}$, a batch size of 32, and 4 num workers. The first two stages of the model were

Table 3.1: Distribution of data

Dataset	Training	Validation	Testing
CMU Panoptic	11,853	1482	1482

trained for 100 epochs, and an early stop technique was implemented to mitigate overfitting. In contrast, the last stage was trained for a notably shorter duration of 10 epochs with a weight decay of 0.01.

3.4.1 Dataset

The Carnegie Mellon University (CMU) Panoptic Hand Dataset [25] was utilized during my dissertation to assess the proposed model. The dataset consists of a total of 14,817 annotations that correspond to the right hand of individuals captured in images from the Panoptic Studio. The current research examines the process of HPE as opposed to hand detection. To achieve this objective, annotated hand image patches were extracted from the initial images using a square bounding box with dimensions 2.2 times larger than the hand size. The dataset was partitioned into three subgroups using a random sampling technique. Specifically, these subgroups were designated as the training set, comprising **80%** of the data; the validation set, comprising **10%**; and the test set, comprising **10%** as shown in Table 3.1.

3.4.2 Loss Function

The Mean Squared Error (MSE) is utilized as the loss function in the model. The loss function is scaled by a coefficient of 35 to prevent the loss from diminishing to nominal values.

Formulating the loss calculation for a model involves a weighted sum of the loss function of each inference.

$$L = \alpha_1 L^{First} + \alpha_2 L^{Second} + \alpha_3 L^{Final} \tag{3.10}$$

where L^{First} is the MSE loss of FIM, L^{Second} represents the MSE loss of SIM, and L^{Final} denotes the PGM MSE loss. These loss terms collectively drive the training process for enhanced model performance. While α_1, α_2, and α_3 are the coefficients for fine-tuning the model, the values are set to 1, 0.1, and 0.1, respectively.

3.4.3 Model Optimization

An optimizer aims to decrease the loss function and steer the network toward improved performance by identifying optimal parameter values. Utilizing a newly derived variation of the Adam optimizer [52] called AdamW can bolster the refinement of model optimization techniques. In contrast to its predecessor, the Adam optimizer, the AdamW algorithm effectively disentangles the weight decay component from the learning rate, allowing for individualized optimization of each component. This feature effectively addresses the issue of excessive overfitting. The outcomes reveal that the models optimized through AdamW exhibit superior generalization performance compared to those trained using other optimizers, particularly Adam. The AdamW optimizer was employed to train our final graphical module.

3.4.4 Activation Functions

Several activation functions, namely ReLU [53], SoftMax [54], and Mish [55], introduce nonlinear components to the neural network, allowing it to comprehend complex patterns and correlations in the data. The Mish activation function has demonstrated superior performance to alternative activation functions, primarily due to its nonlinear nature. The definition of the term can be expressed using the following formula:

$$f(x) = x \cdot \tanh(\ln(1 + e^x)) \tag{3.11}$$

The experimental findings demonstrate that Mish's efficacy surpasses widely utilized activation functions, including ReLU and SoftMax, among others, in diverse deep network architectures operating on complex datasets.

3.4.5 Evaluation Metric

We normalized the Percentage of Correct Keypoints (PCK) [56] for this dissertaion. The PCK metric is a commonly employed evaluation measure for HPE. Specifically, it quantifies the likelihood that a predicted keypoint is located within a designated distance threshold, denoted as σ, from its corresponding ground truth coordinate. The application of σ, restricted to the scale of the hand-bounding box, is utilized within this study. The threshold was uniformly distributed within the range of 0 to 0.10, and the PCK formula is

$$PCK_{\sigma}^{k} = \frac{\mathbf{1}}{||D||} \sum_{D} \mathbf{1} \left(\frac{||p_k^{pt} - p_k^{gd}||_2}{\mathbf{max}(w,h)} \leq \delta \right) \tag{3.12}$$

Where $\mathbf{p}_k^{gd}$ is the ground truth of the keypoint, 1 is the indicator function, and $\mathbf{p}_k^{pt}$ is the predicted keypoint. k represents the number of keypoints, D represents the number of test or validation samples, and h and w represent the height and width of the sample images, respectively.

3.5 Experimental Results

A comparative analysis is conducted between my proposed network and traditional networks used for HPE. Finally, the predicted outcomes are visually understandable to underline my results.

3.5.1 Quantitative Results

Table 3.2 presents the PCK performance of our proposed model on the CMU Panoptic Hand Dataset, showcasing its superiority over contemporary state-of-the-art models. The empirical results reveal that SDPoseGraphNet, on average, improves accuracy by nearly 3.14% compared to AGMN [40], and achieves a 1.24% increase compared to CDGCN [57]. Additionally, Figure 3.5 illustrate our proposed model's PCK compared with other models on CMU dataset.

Table 3.2: SDPoseGraphNet performance in comparison with previous state-of-the-art models.

Threshold σ	**0.04**	**0.06**	**0.08**	**0.10**	**0.12**	**Average**
CPM [38]	56.76	74.66	82.50	86.67	89.45	78.01
AGMN [40]	83.70	90.27	93.23	95.20	96.45	91.77
CDGCN [57]	85.52	91.53	94.33	96.02	97.18	92.91
SDPoseGraphNet	**87.34**	**92.73**	**95.21**	**96.79**	**98.64**	**94.14**

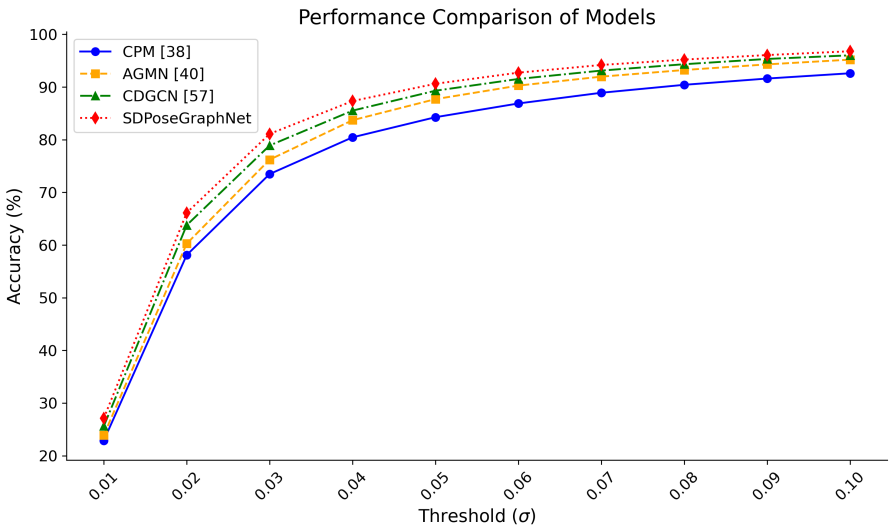


Figure 3.5: PCK evaluation for performance comparison: pro- posed model against existing models.

3.5.2 Qualitative Results

To illustrate qualitative results, we selected a diverse range of images showcasing different angles, challenging scenarios, instances of occlusion, and complex backgrounds. Figure 3.6 demonstrates the robustness and consistency of SDPoseGraphNet across various test scenarios and conditions, highlighting its resilience against interference even in complex backgrounds. In situations where image clarity was compromised, acquiring a higher resolution or more detailed depiction proved beneficial for better interpretation and analysis, emphasizing the significance of our proposed model.

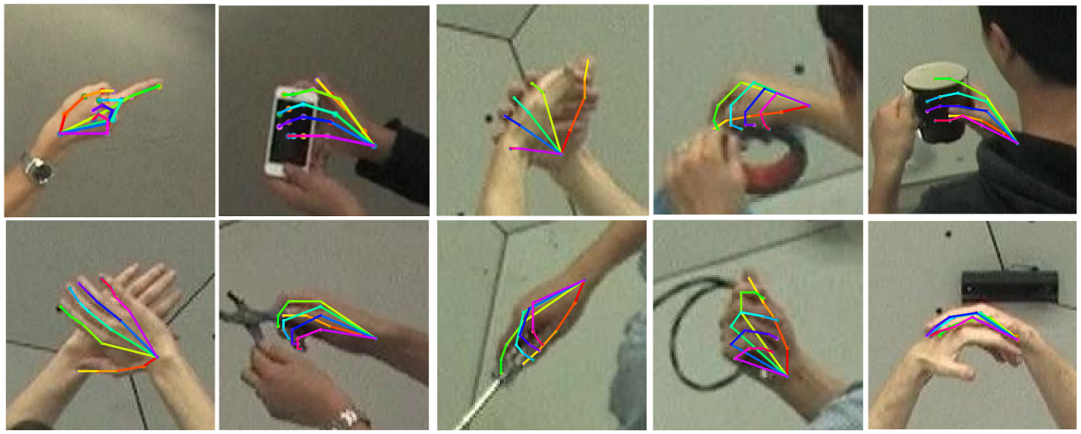


Figure 3.6: Visualizing the performance of SDPoseGraphNet: random image analysis the complexity increases from left to right.

Figure 3.7 visually presents the performance of our model on a random selection of images, featuring (a) ground truth, (b) our proposed SDPoseGraphNet model, (c) CDGCNN, and (d) AGMN. It shows the effectiveness of SDPoseGraphnNet even with the occluded images.

3.6 Ablation Study

An ablation study conducted using the Panoptic dataset aimed to validate the effectiveness of our optimization strategy. The SA module was integrated into the FIM to assess the impact of the SA module while keeping all other

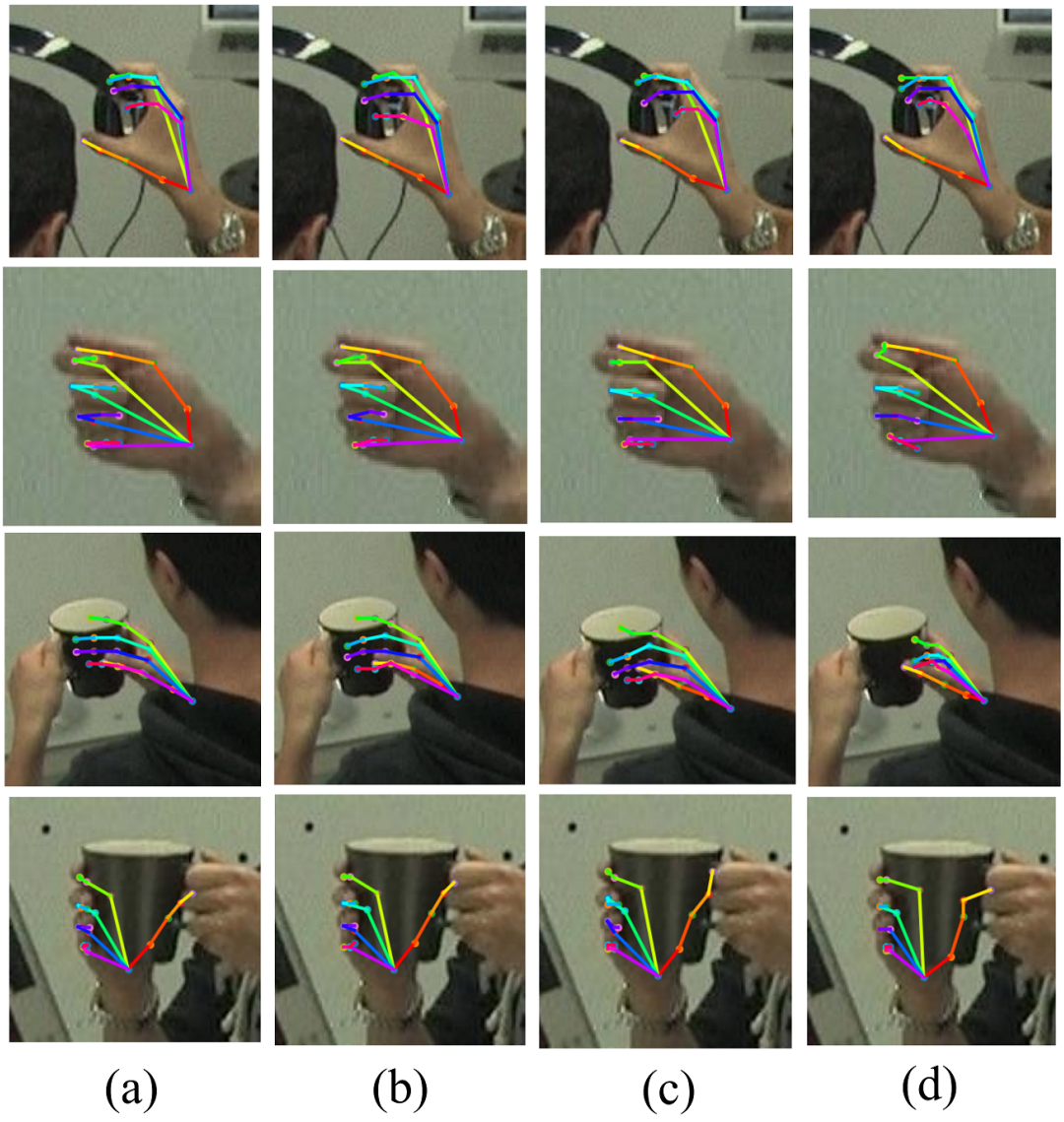
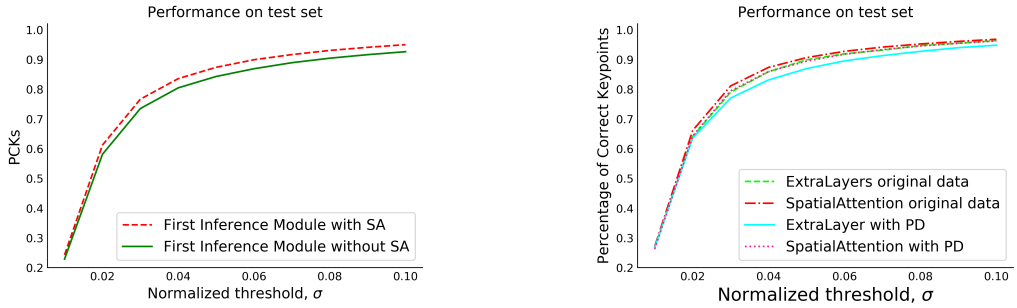


Figure 3.7: Illustrative comparison of 2D HPE: (**a**) Ground truth; (**b**) Ours; (**c**) CDGCN [57]; and (**d**) AGMN [40].

aspects unchanged. As shown in Table 3.3, the experimental results demonstrated an average performance improvement of 3.52%. Figure 3.8a illustrates the significant improvement in network output by integrating VGG-19 with SA, and Figure 3.8b shows the results of our model with pre and post-processed data.

Table 3.3: Comparative performance evaluation of FIM with and without SA integration.

Threshold σ	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**	**0.10**	**Average**
FIM	22.88	58.10	73.48	80.45	84.27	86.88	88.91	90.42	91.61	92.61	76.96
FIM with SA	24.28	61.21	76.63	83.55	87.36	89.90	91.64	93.03	94.08	94.97	79.66



(a) (b)

Figure 3.8: PCK comparison: (a) FIM with and without the integration of the SA module, (b) FIM with preprocessed data (PD) and original data.

Regarding the SIM, it generates 40 feature maps, as explained earlier. However, the SIM consistently predicts the exact 2D coordinates for pairs of neighboring hand keypoints that share a common edge in the tree structure. This consistency arises because the relative positions between these keypoints are fixed and learned during training. Consequently, during testing, the SIM consistently predicts the same relative positions, resulting in consistent 2D coordinate predictions for these 21 keypoints.

Additionally, we employed a VGG-19 backbone model with several additional layers and batch normalization for feature enhancement. Our analysis

of each module in Table 3.4 indicates an average improvement of 2.48% and 0.39% over AGMN [40] and CDGCN [57], respectively. While performance in terms of accuracy increases, computational speed decreases due to adding layers.

Table 3.4: Module performance comparison with integrated extra feature extraction layers.

Threshold σ	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**	**0.10**	**Average**
FIM	24.53	60.82	75.84	82.72	86.47	89.07	90.98	92.42	93.52	94.44	79.08
SIM	23.85	60.11	76.21	83.68	87.87	90.52	92.44	93.84	94.85	95.63	79.90
SDPoseGraphNet	26.25	64.22	79.44	85.93	89.41	91.74	93.30	94.51	95.42	96.22	81.64

(a) (b)

Figure 3.9: Preprocessing stages: (a) Original image (B) Preprocessed image.

In line with the acknowledgment of noise's detrimental impact on model performance [58], we undertook preprocessing measures on the CMU Panoptic dataset, employing a median filter. Subsequently, the model underwent training using the processed data, yielding the following insights. It was observed that while denoising filters introduced a level of smoothness to the dataset, this smoothness could compromise edge clarity and discernibility, thus presenting suboptimal conditions for hand pose estimation. As depicted in Figure 3.9, it is evident that the preprocessed image exhibits a

significant degree of smoothness compared to the original, resulting in information loss. However, in regions with noise, the model showcased improved performance post-denoising. Conversely, areas lacking noise may experience a detrimental impact on the model's performance. Table 3.5, 3.6 shows the numerical results of our model with preprocess dataset.

Table 3.5: Comparative performance of the enhanced model with preprocessed data and additional feature layers.

Threshold σ	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**	**0.10**
FIM	25.43	61.30	75.30	81.48	85.02	87.56	89.45	90.80	91.96	92.97
SIM	23.97	60.26	76.03	83.29	87.21	89.86	91.84	93.14	94.21	94.99
SDPose GraphNet	26.75	63.57	77.01	83.13	86.91	89.52	91.28	92.72	93.96	94.79

Table 3.6: Comparative analysis of model performance with preprocessed data and SA integration.

Threshold σ	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**	**0.10**
FIM	25.90	62.87	76.64	82.77	86.33	88.54	90.37	91.63	92.79	93.74
SIM	24.38	61.67	77.71	84.69	88.59	90.98	92.69	93.92	94.87	95.71
SDPose GraphNet	26.25	64.12	79.01	85.89	89.89	91.88	93.14	94.67	95.44	96.33

3.7 Discussion and Analysis

The proposed SDPoseGraphNet framework is an essential advancement in HPE. By combining the strengths of the VGG-19 architecture with spatial attention mechanisms, the model efficiently captures the spatial relationships between the hand's joints, leading to more accurate pose estimation. Integrating the SA module with VGG-19 improves the model's ability to draw

attention to spatial regions, thereby improving feature extraction. This is also evident from the results of ablation studies, which show significant performance gains when the SA module is integrated into the FIM. In addition, the ability of SIM to recognize the relative position of keypoints also contributes to improved accuracy.

Furthermore, the messaging algorithm used in the proposed graphical model facilitates the refinement of joint prediction by considering the interdependence between keypoints. This allows a more accurate hand pose estimation in complex scenarios with occlusion and different backgrounds. Experimental results show that SDPoseGraphNet outperforms existing state-of-the-art models such as CPM [38], AGMN [40], and CDGCN [57]. In particular, the model achieves significant performance gains in accuracy and correctness on CMU dataset, including raw and preprocessed data. Despite the problems associated with noise in the dataset, SDPoseGraphNet demonstrates robustness and stability, maintaining reliable performance despite the loss of information due to noise. This emphasizes the adaptability and effectiveness of the model in real-world scenarios.

However, it is important to note that the SDPoseGraphNet model does have a limitation in terms of computational complexity. Due to its large size and intricate architecture, the model requires significant computational resources for training and inference. This aspect should be considered when deploying the model in resource-constrained environments or applications requiring real-time processing.

In conclusion, SDPoseGraphNet is a versatile and robust framework for hand pose estimation that can be applied to HPE and other computer vision tasks, such as 3D pose estimation and human pose estimation. SDPoseGraphNet is amenable to end-to-end training and performs well, making it a valuable asset in computer vision research and application development, albeit with considerations for its computational requirements."

Chapter 4

Compact Convolutional Pose Machine

Introduction

In this chapter, we present an evolution of our previous work, proposed in chapter 3 for 2D Hand Pose Estimation (HPE), aimed at reducing computational complexity while enhancing feature integration. Our previous model, SDPoseGraphNet, while effective, had limitations in terms of computational complexity due to its large size and intricate architecture. This aspect required significant computational resources for training and inference, which could be challenging in resource-constrained environments or applications requiring real-time processing. Addressing these limitations, we introduce the adoption of ConvNeXt architecture and contextual representation to achieve our objectives effectively. We employ a customized ConvNext [59] as a backbone, preserving the convolutional layers while removing fully connected layers. This optimization aims to improve HPE feature extraction while reducing the model's overall complexity. A key improvement in our approach is including a Global Context Block (GCB) [60]. This addition allows us to leverage the backbone model's feature extraction capabilities, enabling the model to learn global contextual information of the features

obtained from the convolutional layers. The refined features then undergo a six-processing block process within the CPM framework, resulting in accurate 2D HPE outcomes while maintaining a more efficient computational profile

4.1 CCPM Architecture Components

2D HPE using heatmaps typically involves detecting keypoints to derive the pose **P** of human hands from RGB images or video frames $\boldsymbol{I}$. Each keypoint $\boldsymbol{k_i}$ corresponds to a specific area of the hand, such as joints and fingertips, and is represented by a heatmap $\boldsymbol{H}$. The task then becomes predicting a set of heatmaps $\{\boldsymbol{H_1, ., H_i}\}$, where the pose $\boldsymbol{P}$ consists of coordinates with the highest probability in each heatmap. While the number of key points $\boldsymbol{K}$ varies across datasets, most contain 21 key points, making the objective to estimate $\boldsymbol{P}$ as the set of key points. Our proposed 2D HPE model, depicted in Figure 4.1, is based on a simplified version of the CPM [38] baseline model, carefully balancing complexity and accuracy. The approach begins with a customized ConvNeXt [59] backbone integrated with GCB [60] for feature extraction, followed by a convolution layer to produce the initial heatmap. Subsequently, the initial heatmaps undergo processing through a module comprising six blocks of 3 × 3 convolutions, each supervised using heatmap labels calculated from ground truth via the Gaussian function. The mathematical formula is given as :

$$\boldsymbol{Heatmap} = \boldsymbol{exp}\left(\frac{-[(x - x_k)^2 + (y - y_k)^2]}{\boldsymbol{2\delta^2}}\right) \tag{4.1}$$

where δ denotes the extent of the heatmap, while x_k and y_k represent the underlying coordinates on the ground.

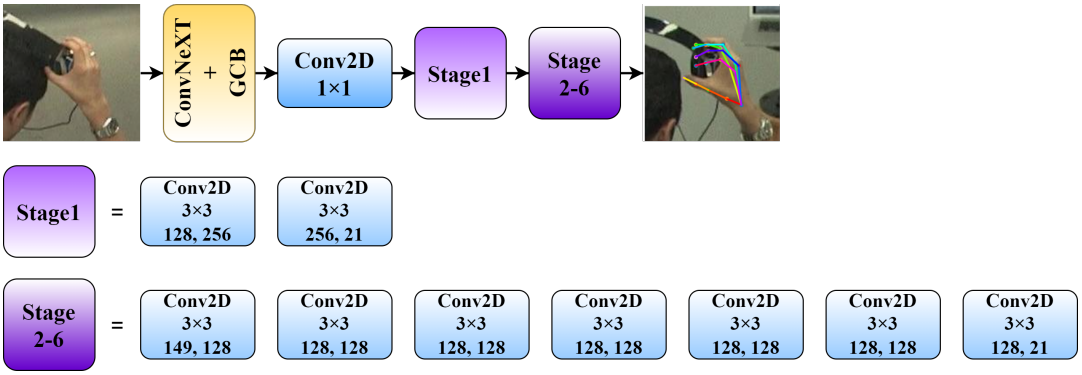


Figure 4.1: General overview of our lightweight, compact CPM information processing module.

4.2 Feature Extraction with ConvNeXt

In our proposed methodology, we integrated a customized ConvNeXT [59] architecture to serve as the backbone for feature extraction. As depicted in Figure 4.2, this architecture consists of convolutional layers meticulously designed to capture and encode intricate patterns and features in the input image $\boldsymbol{I}$. The initial convolutional layer, equipped with a $\mathbf{3} \times \mathbf{3}$ kernel size,

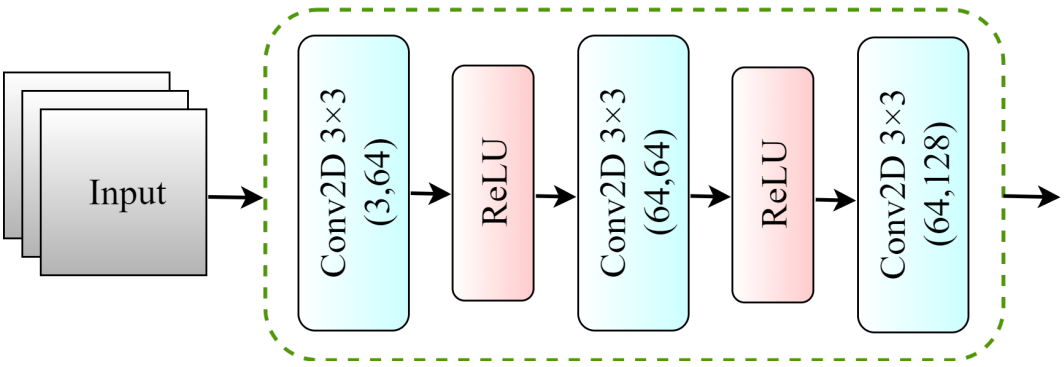


Figure 4.2: Overall architecture of our customized ConvNeXT.

processes the input $\boldsymbol{I}$ and produces 64 feature maps. Subsequent convolutional layers refine these feature maps while maintaining the same kernel size, generating 128 feature maps. The ReLU [53] activation function is applied throughout the architecture to introduce non-linearity, enhancing the model's capacity to capture complex relationships within the data.

The output feature maps serve as the foundational representation in our proposed model, playing a pivotal role in accurately estimating 2D hand

pose estimation. Additionally, these features are subjected to a GCB to facilitate learning of contextual representations, further enhancing the network's capability to discern intricate hand poses.

4.3 Extracting Contextual Information using GCB

To enrich the model's comprehension of global contextual information within feature maps, we incorporated the GCB [60] as depicted in Figure 4.3, where w, h, and c represent the features' width, height, and channels, respectively. The module initiates with a global average pooling layer, which aggregates spatial details across each feature map by computing their average values, thus condensing the dimensions of each feature map into a single channel. Subsequently, the module integrates a fully connected network comprising two linear layers. ReLU activation introduces non-linearity to the network, empowering it to discern intricate patterns within the data. The output of this network undergoes processing through a Sigmoid activation function to yield the final weights.

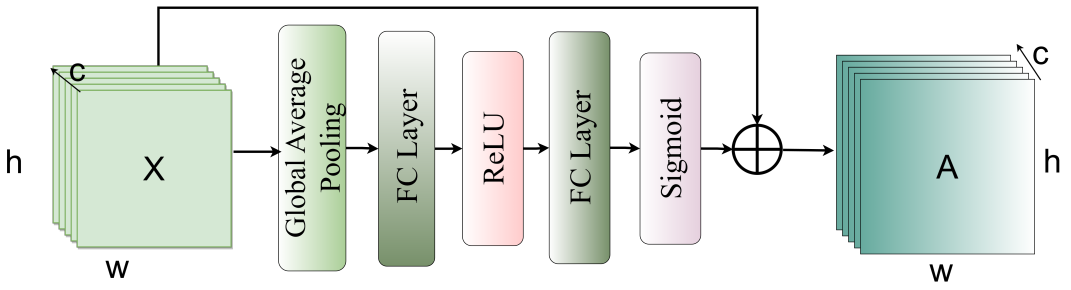


Figure 4.3: Detailed overview of global context block utilized in our framework.

This module utilizes an adaptive average pooling layer to condense spatial information into a single channel per feature map. Subsequently, a fully connected network comprising two linear layers introduces non-linearity through ReLU activation, followed by the production of weights via a Sigmoid activation function. The calculation for the weight vector $\hat{\mathbf{w}}$ is articulated in

Equation (4.2),

$$\hat{\mathbf{w}} = Sigmoid(W_2(ReLU(W_1(\mathbf{F_{avg}})))) \tag{4.2}$$

where $\mathbf{F_{avg}}$ signifies the global average-pooled feature map, and W_1 and W_2 denote the weights of the linear layers. These computed weights are then applied to the input feature map X, emphasizing the salient regions and capturing global context within local features, as delineated in Equation (4.3),

$$\mathbf{A} = X \odot w \tag{4.3}$$

where $\mathbf{A}$ represents the feature map with the applied weights, and $\odot$ signifies element-wise multiplication.

Incorporating this module enriches the model's ability to make accurate predictions by leveraging global contextual cues.

4.4 Information Processing in CCPM

The CPM [38] architecture, widely used in pose estimation tasks, has been modified to reduce complexity and create a new, more compact design, as shown in Figure 4.4. The architecture consists of two main modules, the first of which extracts features from the backbone and then passes through six processing blocks. These blocks consist of successive convolutional layers, with two convolutional layers with a core size of $\mathbf{3} \times \mathbf{3}$ included at the initial stage. This reduces the number of channels from 512 to 256. The subsequent block (2-6) contains seven $\mathbf{3} \times \mathbf{3}$ convolutional layers, reducing the number of channels from 256 to 128. This adjustment made our model more compact and less complex than the original CPM architecture.

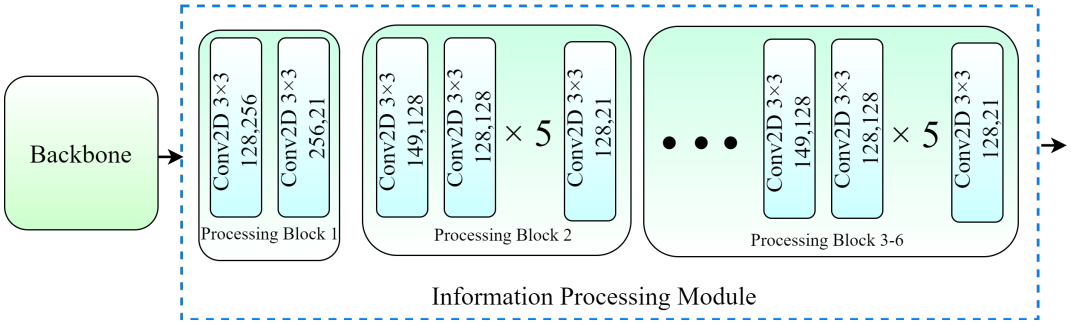


Figure 4.4: General overview of our lightweight, compact CPM information processing module.

4.5 Experimental Setup

4.5.1 Dataset and Evaluation matric

We used a publicly available dataset, Carnegie Mellon University (CMU) Panoptic Hand Dataset [25], to assess our lightweight 2D HPE model. The dataset includes 14,817 annotations corresponding to the right hands of individuals captured in Panoptic Studio images. To achieve our goal, annotated patches of hand images were extracted from the original image using a square bounding box with a size **2.2** larger than the hand size. The dataset is divided into training, validation, and testing, covering **70%**, **15%**, and **15%**, respectively, using a random sampling technique.

We utilized a commonly used evaluation metric, the Percentage of Correct Keypoints (PCK) [56], to test the performance of our proposed model. In this experiment, the threshold σ of PCK was set to $\{\mathbf{0.04, 0.08, 0.12}\}$, and **0.1** was selected to determine the best model weight while testing the model prediction accuracy on the test set.

4.5.2 Loss function and Implementation Details

The loss function used in the model is the mean squared error (MSE). To avoid the loss from becoming too small, it is scaled by a coefficient of 30, and

the formulation is as follows.

$$\text{Scaled MSE} = \mathbf{30} \cdot \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{i=1}^{n} (\mathbf{P}_i - \hat{\mathbf{P}}_i^{\,2} \tag{4.4}$$

where $\boldsymbol{n}$ is the number of keypoints, $\mathbf{P}_i$ and, $\hat{\mathbf{P}}_i$ are predicted and target pose.

We implemented our model using the PyTorch framework with a batch size of **64** and a $\mathbf{10^{-4}}$ learning rate. The model was trained for **100** epochs.

4.6 Experimental Results

The detailed quantitative analysis in Table 4.1 shows that our lightweight model achieves higher precision accuracy than the other state-of-the-art models. Our model accuracy σ at **0.04** is **68.43** which is improved by **4.76%** and σ at **0.12** is **94.05** improved by **1.05%** and an average of **2.62%** against the OCPM [39].

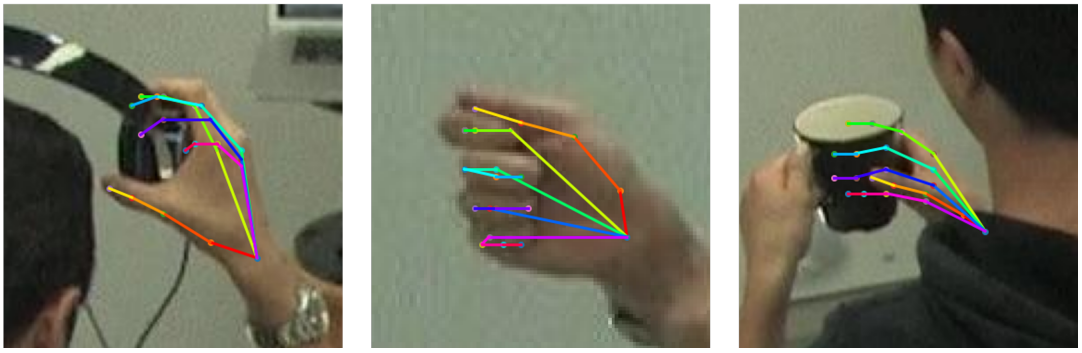


Figure 4.5: Visualization on occluded random test images.

Along with the model accuracy, we also reduced the model complexity to demonstrate that we performed a parameters and GFlops comparison; our proposed model has fewer parameters and GFlops than the state-of-the-art models. Figure 4.6 illustrates the PCK comparison on the test set of our model in contrast with the previous models, and Figure 4.5 shows the Visualization on occluded images. We also performed an ablation study to demonstrate the importance of the integration of GCB in ConvNeXt; Table

Table 4.1: Experimental results on CMU panoptic hand dataset.

Model	PCK (%)					Ave	Par (M)	GFLOPs
	σ 0.04	σ 0.06	σ 0.08	σ 0.10	σ 0.12			
CPM [38]	56.76	74.66	82.50	86.67	89.45	78.01	36.80	103.23
LDM-6 [42]	59.51	76.19	83.77	87.83	90.27	79.51	38.19	95.18
LPM-6 [42]	60.71	77.60	84.93	88.76	91.10	80.62	38.38	92.18
OCPM [39]	63.67	80.26	87.10	90.65	93.01	82.94	29.28	80.53
Our	**65.43**	**81.25**	**89.17**	**91.45**	**94.05**	**84.27**	**8.15**	**18.53**

Table 4.2: Experimental results of our model with and without GCB on CMU panoptic hand dataset.

Model	PCK (%)					Ave	Par (M)	GFLOPs
	σ 0.04	σ 0.06	σ 0.08	σ 0.10	σ 0.12			
Our without GCB	63.43	80.15	87.57	90.78	93.25	83.03	7.55	17.01
Our with GCB	65.43	81.25	89.17	91.45	94.05	84.27	8.15	18.53

4.2 clearly shows the importance of learning the global contextual information from the features to leverage the model's accuracy.

4.7 Discussion and Analysis

The proposed lightweight CMU for 2D hand pose estimation represents a promising advancement in this area. It utilizes the capabilities of ConvNeXt [59] and GCB [60] for feature extraction and learning global contextual information. Evaluations conducted on the CMU dataset show that our model performs better than existing lightweight systems, improving accuracy and computational efficiency. This suggests that our model can significantly impact various hand-related computational tasks, including gesture recognition and human-computer interaction, by providing more accurate and efficient

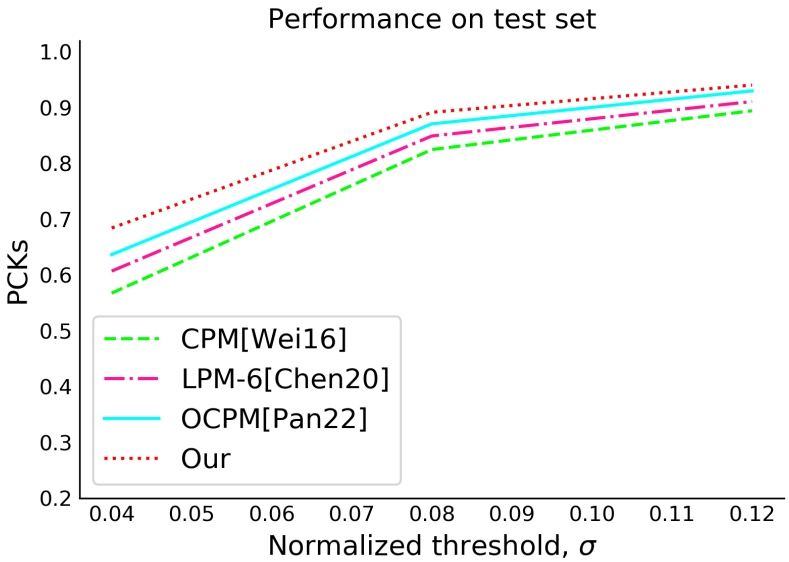


Figure 4.6: PCK comparison on the test set of our model with CPM [38], LPM-6 [42], and OCPM [39].

solutions. One of our approach's strengths is our ability to balance model complexity and performance: using ConvNeXt and GCB, we have developed a rational architecture that minimizes computational cost while maintaining competitive accuracy. This makes our model suitable for resource-constrained environments and real-time applications where computational efficiency is crucial.

However, despite the promising results, areas still require improvement and further study. For example, although our model outperforms existing lightweight frameworks, there may still be room for optimization to achieve even greater accuracy or reduce computational cost. Furthermore, extending this work to 3D hand pose estimation with minimal computational cost is an exciting direction for future research. We use this study's lightweight and efficient design principles to develop models that accurately estimate 3D hand pose in real-time or resource-constrained environments.

Our proposed CCPM offers an attractive combination of accuracy, efficiency, and versatility, significantly contributing to hand pose estimation.

Through ongoing research and development, we will strive to refine and improve our model to address emerging challenges and advance the field of hand-related computational tasks.

Chapter 5

Attention Driven Contextual Features Based Convolution Network

Introduction

It is essential to note that the network has some limitations, such as being computationally expensive and having an imperfect structure. These limitations suggest that there is room for improvement in the design of the CPM [38] and our previous approaches in chapter 3 and 4. In this chapter, we proposed an innovative approach Attention Driven Contextual Features Based Convolution Network (ACENET) for 2D HPE utilizing a CPM architecture combining the power of EfficientNet (EN) [61] with a Squeeze and Excitation Attention (SE) block [62] and Globel Contextual (GC) [63] block named ACENET. Our proposed model aims to accurately predict the intricate poses of human hands from monocular RGB images. We leverage the state-of-the-art EN [61] architecture to extract rich and hierarchical features from the input images, enhancing the model's ability to capture intricate hand structures and pose variations. The SE block can be viewed as an attention mechanism

that learns to focus on the most relevant features. It uses a squeeze operation to reduce the spatial dimensions of the feature maps and a set of fully connected layers to capture channel-wise dependencies. The excitation operation then scales the feature maps based on the learned importance weights, resulting in enhanced feature representation.

5.1 ACENet Architecture

We present an innovative approach ACENet for 2D HPE by utilizing a CPM [38] architecture, combining the power of EN [61] with SE [62] and GC [63]. The overall architecture of the proposed model is shown in Figure 5.1. Our

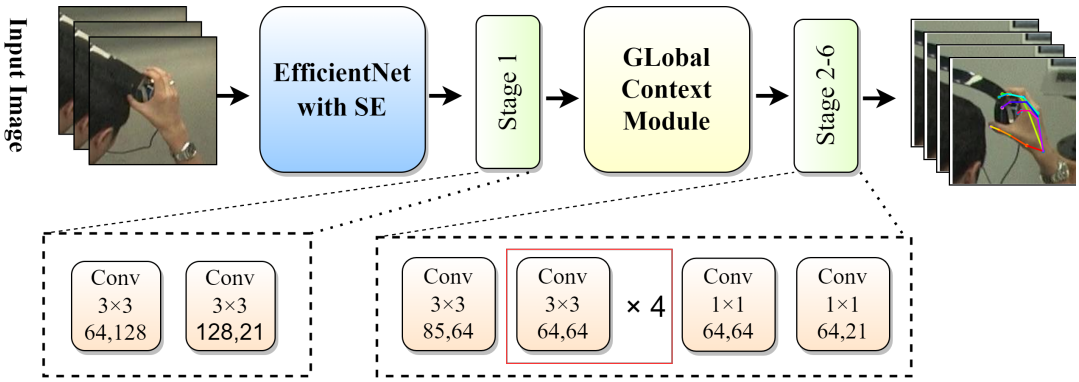


Figure 5.1: Detail architecture of ACENet.

approach is unique in that it utilizes a CPM [38] architecture designed to predict the location of hand joints by generating a confidence map that assigns a probability to each pixel in the image. To make the previous approaches more efficient, This confidence map is then used to estimate the location of the hand joints.

5.2 Feature Extraction with EN

The study incorporated EN [61], an advanced convolutional neural network architecture renowned for balancing model accuracy and computational efficiency. EN is particularly well-suited for scenarios with limited computational resources due to its efficient feature extraction capabilities. Leveraging an EN model renowned for its vast collection of annotated images allowed us to harness high-level features crucial to our task. The architecture of the EN model used for feature extraction is elaborated in Figure 5.2.

Our investigation employed the EN-B0 baseline, comprising seven blocks containing varying numbers of Mobile Inverted Bottleneck Convolution (MBConv) layers. Our adaptations to the EN architecture involved removing the final fully connected layers, thus reducing the model's size and repurposing it as an efficient feature extractor. The data flow within the network entails sequential processing, starting with a $\mathbf{3} \times \mathbf{3}$ convolutional operation, followed by MBConv operations. These operations extract distinctive features, subsequently fed into the SE [62] block.

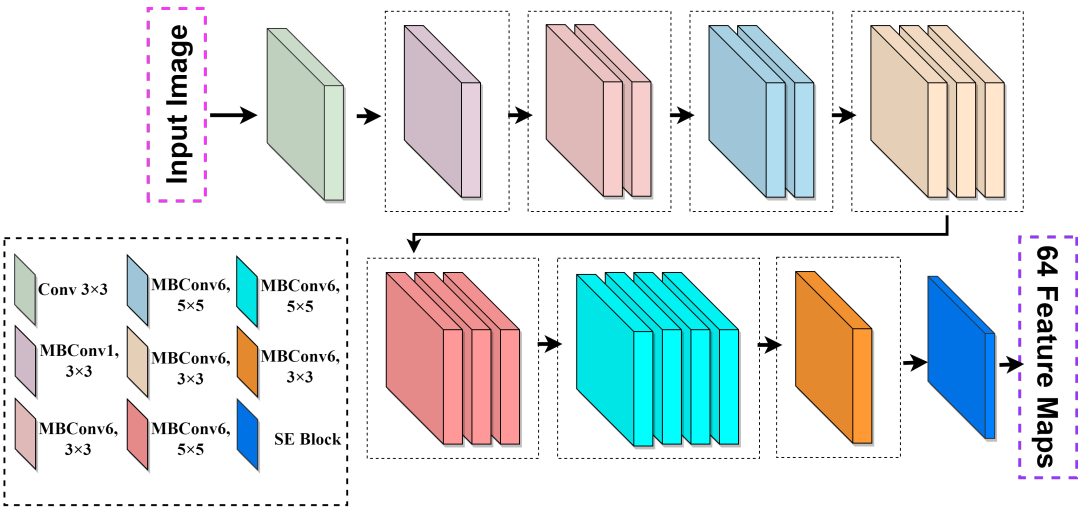


Figure 5.2: Detailed architecture of EfficientNet

5.3 Spatial Feature Extraction with SE

The SE [62] Block employs a series of transformations to enhance the representation of input feature maps. First, a channel-wise transformation is applied to the input feature map $\mathbf{X}$, as expressed by Equation (5.1.

$$\mathbf{U} = f_c(\mathbf{X}) = \sigma(\mathbf{W_2}\delta(\mathbf{W_1}\mathbf{X}) \tag{5.1}$$

This transformation involves the application of learnable weight matrices $\mathbf{W_1}$ and $\mathbf{W_2}$, along with ReLU [53] and sigmoid activation functions δ and σ, respectively, resulting in the output feature map $\mathbf{U}$.

Next, recalibration factors are computed to adjust the dimensions of $\mathbf{U}$ to match those of the original feature map $\mathbf{X}$, as described in Equation 5.2. Here, $\mathbf{W_3}$ represents a learnable weight matrix, and $\mathbf{s}$ denotes the recalibration factor.

$$\mathbf{s} = f_s(\mathbf{U} = \mathbf{W_3}\mathbf{U} \tag{5.2}$$

Finally, feature recalibration is achieved through element-wise multiplication, as depicted in Equation (5.3).

$$\mathbf{Y} = f_r(\mathbf{X}, \mathbf{s}) = \mathbf{X} \odot \mathbf{s} \tag{5.3}$$

This operation involves the element-wise product of the original feature map $\mathbf{X}$ and the recalibration factor $\mathbf{s}$, yielding the amplified feature map $\mathbf{Y}$.

By dynamically adjusting input features based on channel importance, the SE block enables the neural network to highlight critical information, thereby enhancing its ability to identify complex patterns in the data. This capability improves performance across various computational tasks, including HPE. For a more detailed illustration, refer to Figure 5.3.

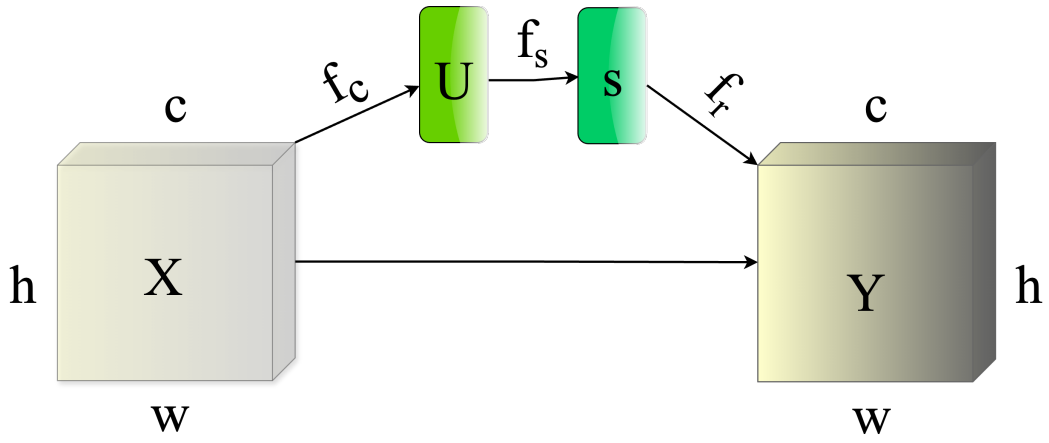


Figure 5.3: Squeeze and Excitation Block

5.4 Lightweight CPM

Our proposed framework retains the core CPM [38] structure through modifications while effectively addressing the intricacies of hand pose estimation. The architecture comprises six stages, with the initial stage featuring two $\mathbf{1} \times \mathbf{1}$ convolutional blocks, followed by seven $\mathbf{1} \times \mathbf{1}$ convolutional blocks in subsequent stages. After the first stage, we integrate the GC [63] block, augmenting the model's perceptual capability by incorporating multi-scale contextual information.

Moreover, we downsized the number of channels in each convolutional block across all stages to streamline our framework's complexity. Specifically, we reduced the channel count from 512 to 128 in the first stage and 256 to 64 in subsequent stages. This reduction in channel count significantly mitigated the computational complexity of the architecture, yielding a more lightweight yet efficient model.

Furthermore, we employed the Mish [55] activation function instead of the ReLU [53] activation function in the first stage to enhance the model's capacity. This switch amplifies the model's non-linearity, empowering it to learn more intricate patterns. Collectively, these adjustments culminated in an optimized and lightweight CPM architecture, striking a balance between

accuracy and computational efficiency. For a comprehensive overview, refer to Fig. 5.2.

5.5 Extracting Hierarchical Contextual Feature with GC Block

Our HPE architecture incorporates the GC [63] block, drawing on hierarchical context aggregation to enhance the model's ability to discern intricate spatial relationships. To integrate global context, we employ adaptive average pooling, as depicted by Equation (5.4):

$$\mathbf{z} = \text{AdapAvgPool}(\mathbf{X}) \tag{5.4}$$

Here, $\mathbf{X}$ represents the input feature maps, and $\mathbf{z}$ denotes the global context representation.

To refine this context, attention weights are computed through convolutional transformation, prioritizing essential features by assessing relative importance, as shown in Equation (5.5):

$$\mathbf{w} = \text{Conv2d}(\mathbf{z}) \tag{5.5}$$

Here, $\mathbf{w}$ denotes the attention weights post-convolution.

Subsequently, the module proceeds with hierarchical context aggregation, a crucial process that examines multiple scales. By iteratively rescaling input feature maps $\boldsymbol{X}$ with scale factors (s) like **0.5**, **1.0**, and **2.0**, distinct scaled input representations $\boldsymbol{X_s}$ are generated. These scaled representations simplify the computation of contextual maps $\boldsymbol{M_s}$ embedded within their respective scales. To maintain alignment with the original feature map dimensions, contextual

maps are adjusted using interpolation techniques, as demonstrated in Equation (5.6):

$$\mathbf{M'}_s = \text{Interpolate}(\mathbf{M}_s, \mathbf{S(X)}) \tag{5.6}$$

Here, $\mathbf{M'}_s$ represents the reshaped contextual map.

The essence of the GC block becomes apparent when diverse contextual aspects are merged, achieving aggregation denoted as $\mathbf{A}$ through the summation of elements from different scales, as expressed in Equation (5.7):

$$\mathbf{A} = \sum_s \mathbf{M'}_s \tag{5.7}$$

The illustration of the GC block emphasizes leveraging varied viewpoints to optimize the accuracy and effectiveness of HPE. For a detailed insight, refer to Figure. 5.4, where X, w, h, and c denote input feature, width, height, and channels, respectively.

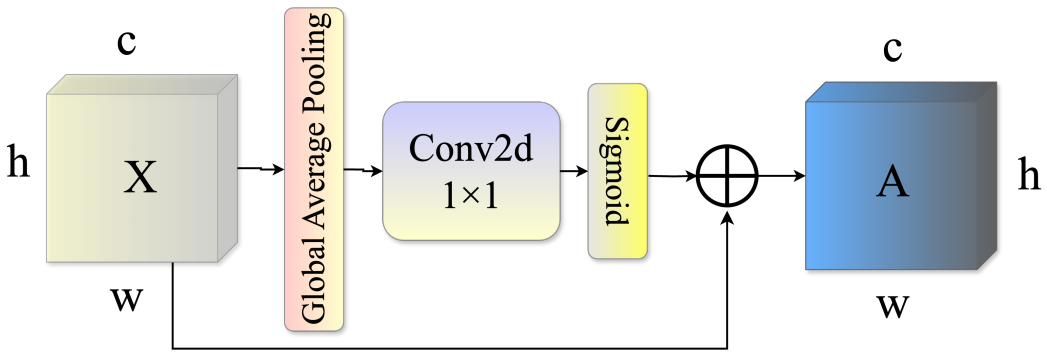


Figure 5.4: Detailed overview of Global Context Block

5.6 Experimental Setups

5.6.1 Dataset

The study evaluated the proposed model's effectiveness using the CMU Panoptic hand dataset [25]. In the dataset, 14,817 detailed annotations exclusively

Table 5.1: Distribution of data.

Dataset	Training	Validation	Testing
CMU Panoptic	11,853	1482	1482

represent the right hand of individuals captured in images obtained from the Panoptic Studio. The annotated hand images were appropriately cropped from the original images by employing a square bounding box that was 2.2 times greater in size than that of the hand. The dataset is split into two parts using a random sampling approach. In particular, the mentioned subgroups were designated for training and accounted for 70 % of the data. The remaining 30 % was set aside for validation. The 30 % of the data used for validation were further employed for testing, effectively utilizing the entire dataset. The data distribution of the dataset is shown in Table 5.1.

5.6.2 Evaluation Matrics

We utilized the normalized Percentage of Correct Keypoint [56] metric as an evaluation metric. The likelihood of a predicted keypoint being close to its actual location is measured by a distance threshold σ. To ensure consistency across different hand sizes, a normalized threshold σ is used, which varies between 0.04 and 0.12 based on the size of the hand-bounding box.

5.6.3 Implementation Details

We implemented our model using the PyTorch framework with a batch size of 64 and a learning rate of $10^{-}\mathbf{4}$. The model was trained for 100 epochs to prevent overfitting; we used early stopping with a patience of 3. Before being fed into the model, we resized the images to 368×368, resulting in a final score map of size 46×46 for each keypoint. Additionally, all images

were scaled to $[\mathbf{0,1}]$ and normalized using $\boldsymbol{mean} = (\mathbf{0.485, 0.456, 0.406})$ and $\boldsymbol{std} = (\mathbf{0.229, 0.224, 0.225})$.

The optimizer utilized in this study was AdamW, which can effectively separate the weight decay component from the learning rate, enabling independent optimization of each component. Additionally, we employed Mean Squared Error (MSE) as the loss function for our model. To prevent the loss from reaching nominal values, it was scaled by a factor of 30.

5.7 Experimental Results

This section presents a performance analysis of our proposed framework, including numerical and visual results.

5.7.1 Quantitative Results

Fig. 5.5 shows the Percentage of Correct Keypoints (PCK) performance comparison of ACENet with other state-of-the-art models on the CMU Panoptic hand dataset. The graphical representation indicates that Our-B0 outperforms the accuracy of the other method, with a more minor keypoint offset of regression.

Table 5.2 shows a numerical comparison with previous methods OCPM [39], CPM [38], LDM-6 [42], and LPM-6 [42], our proposed framework on CMU Panoptic hand dataset with a consistent, normalized threshold with the previous methods to maintain the same standard shows a significant improvement. Comparing the results, ACENet achieves a 2.11% increment in average accuracy to OCPM and 7.04% from CPM.

We compared the parameters, excluding LPM-6 from the analysis due to the absence of additional parameters. The researchers input a 368×368 color image into the network to determine the number of parameters required for the operation. The results, presented in Table 5.3, indicate that the ACENet

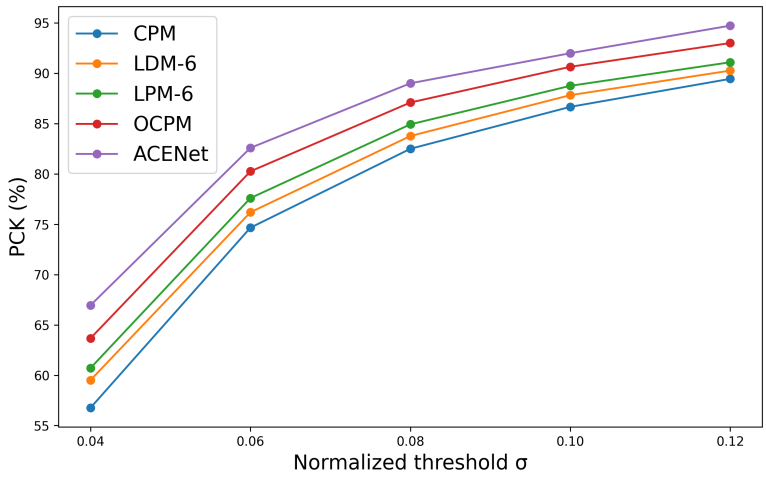


Figure 5.5: PCK comparison of ACENet with other models

Table 5.2: Experimental Results Comparison of ACENet with Other Models on CMU Panoptic Hand Dataset

Threshold σ	***0.04***	***0.06***	***0.08***	***0.10***	***0.12***	***Average***	***Improvement***
CPM [38]	56.76	74.66	82.50	86.67	89.45	78.01	–
LDM-6 [42]	59.51	76.19	83.77	87.83	90.27	79.51	1.50
LPM-6 [42]	60.71	77.60	84.93	88.76	91.10	80.62	2.61
OCPM [39]	63.67	80.26	87.10	90.65	93.01	82.94	4.93
ACENet	**66.95**	**82.59**	**89.01**	**92.00**	**94.74**	**85.06**	**7.04**

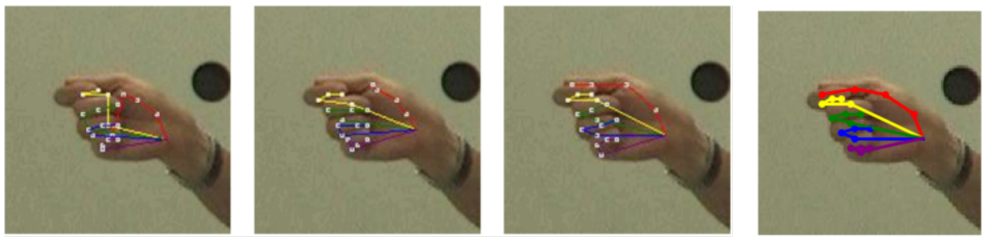
parameters were reduced in each aspect, signifying a significant reduction in the computational cost.

Table 5.3: Parameters Comparison With Previous Models

Model	*Parameters (M)*	*FLOPs (G)*
CPM [38]	36.80	103.23
OCPM [39]	29.28	80.53
ACENet	**12.45**	**20.23**

5.7.2 Qualitavie Results

To evaluate the effectiveness of the proposed framework, we randomly selected images as inputs to the network for visualization. The results, as shown in Fig. 5.6 and Fig. 5.7, demonstrate that our ACENet exhibits robustness, anti-interference capability, and severely self-fingers-occluding hand. The network's ability to operate efficiently in low light or blurred images, as observed in some CMU Panoptic Hand dataset samples, is noteworthy. These results indicate that ACENet can accurately detect hand key points, even in challenging conditions. ACENet's performance in such scenarios is a significant advantage, as it can be used in real-world applications where the lighting and background conditions may not be optimal. The study's findings suggest that ACENet is a promising approach for accurate HPE.



(a) (b) (c) (d)

Figure 5.6: Visualization results, each subfigure shows the result of (a) CPM [38], (b) LPM [42], (c) OCPM [39] and (d) ACENet

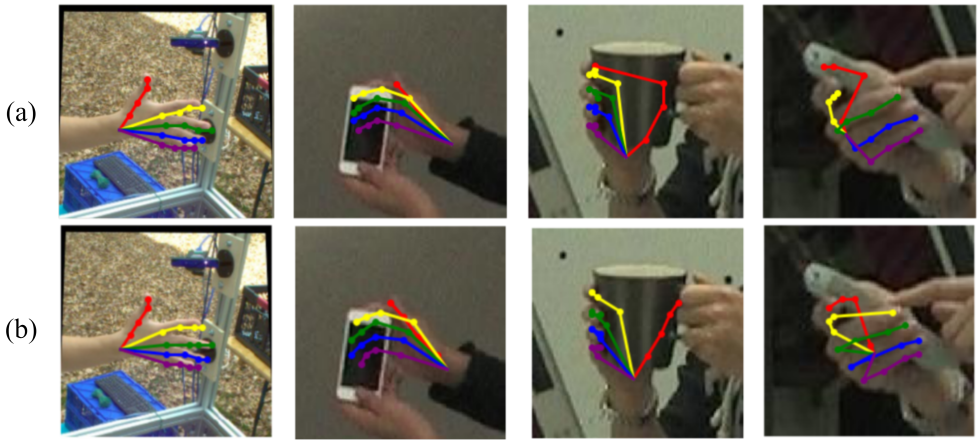


Figure 5.7: Visualization results, (a) LPM [42] and (b) ACENet

5.8 Ablation Study

To assess the efficacy of our optimized approach, we conducted an ablation experiment using EfficientNet as the backbone.

Furthermore, we conducted experiments to evaluate the impact of the SE block, comparing frameworks with and without it. The results, outlined in Table 5.4, unequivocally demonstrate the SE block's effectiveness in enhancing the framework's performance. Notably, integrating the SE block leads to a marked improvement in accuracy, indicating its ability to improve performance without substantially increasing computational overhead.

In addition to the SE block, we introduced a GC block featuring hierarchical context aggregation after the first stage of the information processing block. Designed to capture global context information from the input image, the GC block aims to bolster the framework's accuracy. Evaluation results, presented in Table 5.5, confirm the efficacy of the GC block, with its inclusion resulting in a significant enhancement in accuracy. This underscores the effectiveness of the GC block as a means to augment the framework's performance.

Overall, these findings underscore the effectiveness of both the Squeeze-and-Excitation modules and the GC block in improving the framework's performance.

Table 5.4: Experimental results of Our Model with and without SE

Threshold σ	***0.04***	***0.06***	***0.08***	***0.10***	***0.12***	***Average***
ACENet*	62.09	78.82	85.64	90.27	92.42	81.85
ACENet$_{SE}$	64.05	80.75	87.85	91.16	93.10	83.32

[a]ACENet* is Without SE and ACENetr$_{SE}$ is With SE

Table 5.5: Experimental results of ACENet with and without GC block

Threshold σ	***0.04***	***0.06***	***0.08***	***0.10***	***0.12***	***Average***
ACENet$_{SE*}$	64.05	80.75	87.85	91.16	93.10	83.32
ACENet	66.95	82.59	89.01	92.00	94.74	85.05

[a]ACENet$_{SE*}$ is With only SE, and ACENet with both SE and GC blocks

5.9 Discussion and Analysis

This study presents ACENet, a novel 2D HPE framework. ACENet is based on integrating EfficientNet as the underlying architecture combined with a SE block for attention-based feature extraction. In addition, integrating the GC block after the first stage enhanced the perceptual capabilities of the model. It allowed it to utilize global context information to estimate hand position accurately. When evaluating our model on the CMU Panoptic Hand dataset, promising results were obtained; ACENet achieved better performance on several metrics while significantly reducing the number of parameters. This suggests that our approach balances model complexity and performance, achieving accuracy and computational efficiency. The significance

of this research is not limited to hand pose estimation but has potential applications in human-computer interaction, virtual reality, and robotics. By providing accurate and efficient hand pose estimation, ACENet paves the way for improved user interfaces, immersive experiences, and intuitive human-machine interaction. However, it is important to acknowledge certain limitations of ACENet. Despite efforts to reduce computational complexity, the model still requires significant computational resources, which may limit its applicability in some real-time or resource-constrained scenarios. Additionally, ACENet showed limitations in accurately estimating hand poses in heavily occluded situations, indicating room for improvement in handling complex occlusions. These challenges highlight areas for future research and refinement of the model

Chapter 6

Deformable Convolution Pose Network

Introduction

In this chapter, we introduce the Deformable Pose Network (DPN), a novel approach to 2D Hand Pose Estimation (HPE) that builds upon the advancements made by ACENet. ACENet, as presented in the previous chapter 5, represents a significant milestone in HPE research, leveraging EfficientNet architecture and attention mechanisms to accurately estimate hand positions. While ACENet demonstrated considerable progress, it faced certain limitations. Despite efforts to reduce computational complexity, the model still required significant computational resources, potentially limiting its applicability in some real-time or resource-constrained scenarios. Additionally, ACENet showed limitations in accurately estimating hand poses in heavily occluded situations, indicating room for improvement in handling complex occlusions. Addressing these challenges, DPN aims to further enhance the accuracy and efficiency of HPE. Inspired by the need to incorporate geometrical constraints into convolutional operations, DPN introduces the concept

of Deformable Convolution. This approach focuses on addressing the challenges posed by complex hand poses and occlusion scenarios, ensuring robust performance in diverse environments. Similar to ACENet, DPN adopts a multi-stage architecture consisting of a backbone and a Deformable Convolution Block (DCB) [64, 65]. The choice of EfficientNet [61] B0 as the backbone ensures a balance between computational cost and model efficiency, aligning with the principles of ACENet. Additionally, the integration of a four-stage DCB, inspired by Convolutional Pose Machine (CPM) [38], allows DPN to effectively capture geometrical constraints and hidden information critical for accurate HPE.

6.1 DPN Architecture

We proposed a new Deformable Pose Network (DPN) approach for efficient and accurate 2D HPE. A multi-stage deformable convolution is utilized in our work inspired by the workflow of CPM [38] stages, combining the power of EN [61] as a backbone for feature extraction. Figure 6.1 shows the detailed architecture of our proposed method.

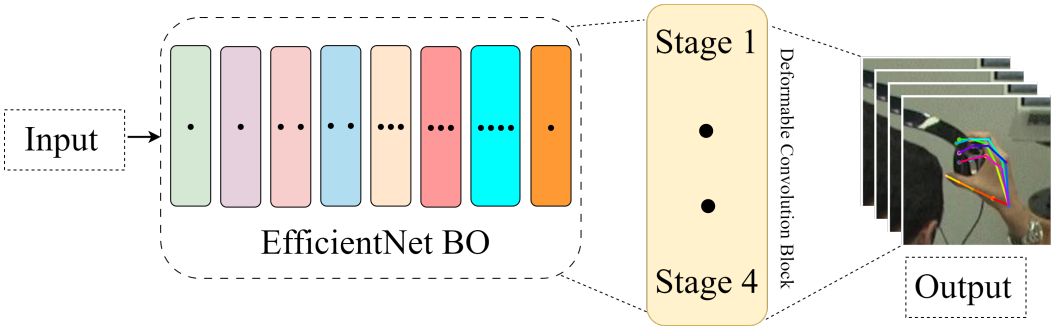


Figure 6.1: Detailed overview of Deformable Pose Network.

6.2 Feature Extraction with EfficientNet

We use EN [61], a state-of-the-art network architecture, as the basis of our proposed feature extraction method. EN is renowned for its ability to balance model accuracy and computational efficiency. There are different EN (B0-B7) versions, each varying in depth and complexity. Our approach chose the B0 version because of its lightweight nature. The architecture of the modified B0 EN, which serves as the backbone of our network, is shown in Figure 5.2.

To simplify our approach compared to other variants and feature extraction networks such as ResNet [66] and VGG [47], we specifically chose the B0 EN version. The B0 EN version consists of seven blocks, follows the structure of MobileNetv2, and includes a variable number of MBConvs. It has a variable number of MBConvs. The model parameters have been effectively reduced by removing the last fully connected convolutional layer, making it suitable for feature extraction. Several layers sequentially process the input data. First, the $\mathbf{3} \times \mathbf{3}$ convolution operation is applied, followed by the MBConv operation; the last layer EN generates 64 feature maps, passed to the deformable convolution block for further processing.

6.3 Four-stage Information Processing Block

The CPM is among the baseline CNN-based pose estimation models addressing HPE complexities. However, it faces limitations due to unknown geometrical constraints and other challenges. To tackle these issues, we integrated the DC into CNN-based models, focusing on managing geometrical constraints and enhancing adaptability in learning unknown features. Our proposed approach features a four-stage network. The initial stage includes two $\mathbf{3} \times \mathbf{3}$ DCBs with 256 channels, followed by subsequent stages with seven

3 $\times$ **3** DCBs, each containing 128 channels. Figure 6.2 provides a detailed overview of this information processing DCB.

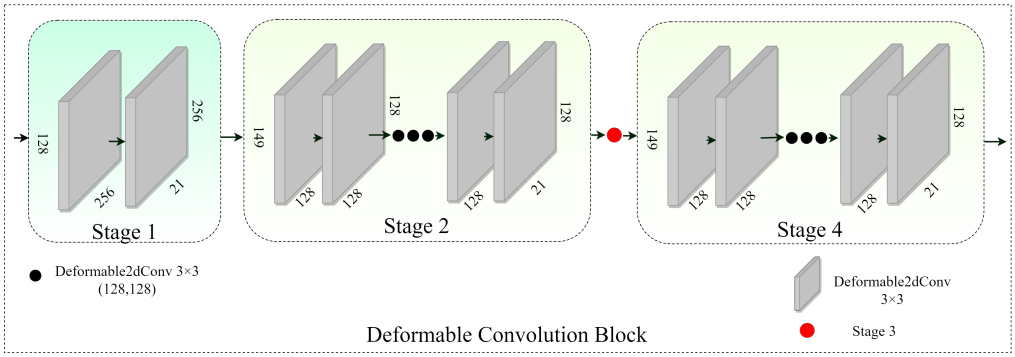


Figure 6.2: Detailed overview of stages of deformable convolution block.

The output feature maps from the backbone are directed to the initial stage of our network for subsequent information processing. Within each stage, the DC comprises two Convolutional Layers (CL), offset CL, and modulator CL, along with a DC operation discussed in detail below:

6.3.1 Offset Convolutional Layer (CL)

This layer computes spatial offsets through learnable parameterization from the input feature map $\boldsymbol{x}$, which represents the output feature map of the backbone and is denoted as $\boldsymbol{OF}$. Mathematically, it can be represented as shown in Equation 6.1:

$$\mathrm{OF} = \mathrm{OFC}(\boldsymbol{x}) \tag{6.1}$$

Here, $\boldsymbol{OFC}$ denotes the convolutional operation on the input $\boldsymbol{x}$ for computing the offsets. This process helps determine the sampling location in $\boldsymbol{x}$, providing flexibility to receptive fields.

6.3.2 Modulator Convolutional Layer (CL)

This layer governs the significance or modulation of sampled regions by generating modulation weights. It utilizes the output of a sigmoid function, as shown in Equation 6.2:

$$\boldsymbol{M} = \mathbf{2} \times \sigma(\boldsymbol{MC}(\boldsymbol{x})) \tag{6.2}$$

Here, $\boldsymbol{M}$ represents the modulator, σ represents the sigmoid function and $\boldsymbol{MC}$ denotes the convolutional operation. This factor facilitates adaptive feature adjustments according to their importance.

6.3.3 Deformable Convolution (DC) Operation

Following the first two Convolutional Layers (CLs), the DC operation serves as the core component of the DCB. It integrates the offset and modulator with the regular CL. Mathematically, this operation can be expressed as shown in Equation 6.3:

$$\boldsymbol{x} = \boldsymbol{deform\_2d}(\boldsymbol{x}, \boldsymbol{OF}, \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{M}) \tag{6.3}$$

Here, $\boldsymbol{x}$ represents the input feature map, $\boldsymbol{OF}$ signifies the spatial offsets, and $\boldsymbol{w}$, $\boldsymbol{b}$, and $\boldsymbol{M}$ represent the convolutional weights, bias, and the modulating factor, respectively. This incorporation dynamically adjusts the receptive fields, enabling the model to learn adaptive features and geometrical constraints. The final output from the initial stage progresses to the second stage.

The sequence iterates across all four stages, and in the final stage, we obtain the 21 final keypoints. Along with the DC, we reduced the number of stages and channels compared to CPM, enhancing our model's overall adaptability and computational efficiency.

Table 6.1: CMU panoptic hand dataset distribution.

Dataset	Training	Validation	Testing
CMU Panoptic	11,853	1482	1482

6.4 EXPERIMENTAL SETUP

6.4.1 Dataset

Our research used a publicly available data set, The Carnegie Mellon University Panoptic Hand Dataset (CMU) [25] from Panoptic Studio to evaluate our proposed model. The dataset includes 14,817 annotations of the right hand of individuals captured at the studio; the distribution is shown in Table 6.1 as our research is HPE, the annotated image patches were extracted from the entire image using a box size of 2.2 times larger than the hand to achieve this objective. The dataset is randomly divided into three subgroups by a random sampling technique for training, validation, and testing comprised of **80 %**, **10 %**, and **10 %** of the dataset, respectively.

6.4.2 Implementation details

We implemented our model using the PyTorch framework, with a batch size of 64 and a learning rate of **0.0001**. The model is trained up to 100 epochs. The input images were scaled to $[\mathbf{0, 1}]$ and normalized using a mean and standard deviation of $(\mathbf{0.485, 0.456, 0.406})$ and $(\mathbf{0.229, 0.224, 0.225})$ respectively. The Mean Squared Error (MSE) is utilized as a loss function. The loss function is adjusted using a scaling factor of 35 to prevent the loss from decreasing to a meager value.

6.4.3 Activation Function and Model Optimizer

To incorporate nonlinear aspects into the network, various activation functions were proposed, such as ReLU [53], Softmax [54], and Mish [55]. However Mish outperforms others notably due to its nonlinear nature; its mathematical representation is as follows:

$$f(x) = x \tanh(\ln(1 + e^x)) \tag{6.4}$$

Experimental results highlight Mish's superficiency over other activation functions.

The model optimizers aim to decrease the loss function and enhance network performance by finding the best parameter values. We adopted a newly derived version of the Adam optimizer; AdamW can significantly bolster model optimization techniques. In contrast to the Adam optimizer, the AdamW algorithm separates the weight decay component from the learning rate, enabling individualized optimization of each component. This feature effectively addresses the issue of excessive overfitting. The results indicate that the model optimized with AdamW demonstrates better generalization performance. The AdamW optimizer was employed in the training of our proposed approach.

6.4.4 Evaluation Metric

As an evaluation metric commonly used for pose estimation, the Percentage of Correct Keypoints (PCK) [56] was utilized in this study. It measures the probability that the predicted keypoints fall in a specified threshold distance, represented as σ from the ground truth. σ was uniformly distributed in a

range of 0.04 to 0.12; it is formulated as:

$$PCK_{\sigma}^{k} = \frac{1}{||D||}\sum_{D} 1\left(\frac{||p_k^{pt} - p_k^{gd}||_2}{max(w,h)} \leq \sigma\right) \tag{6.5}$$

Here p_k^{gd} represents the keypoints ground truth, 1 is the indicator function, p_k^{pt} denotes the predicted keypoints, k for the number of keypoints, D refers to the number of test or validation sample, and w and h indicates the height and width of the input image respectively.

6.5 Experimental Results

This section analyzes the performance of the proposed network and compares it with different HPE methodologies.

6.5.1 Quantitative Results

The quantitative analysis of the proposed model is presented in Table 6.2 and Figure 6.3, both numerically and graphically. The results show noticeable improvements. Our model achieves an improvement of 5.13 % at σ 0.12 and an average improvement of 7.29 % over CPM. It also outperforms OCPM [39] by 1.57 % at $sigma$ 0.12 and an average improvement of 2.36 %.

Figure 6.3 shows the PCK comparison between DPN and CPM [38], LDM-6 [42], LPM-6 [42], and OCPM [39], demonstrating DPN's superiority over existing methods. To compare the computational complexity, the parameters were compared except for LDM-6 and LPM-6 which have no parameters. As shown in Table 6.3, it can be seen that the proposed architecture has fewer parameters and lower computational complexity compared to CPM and OCPM.

Table 6.2: Numerical comparison of DPN with other models on CMU panoptic hand dataset.

Threshold σ	***0.04***	***0.06***	***0.08***	***0.10***	***0.12***	***Average***	***Improvement***
CPM [38]	56.76	74.66	82.50	86.67	89.45	78.01	–
LDM-6 [42]	59.51	76.19	83.77	87.83	90.27	79.51	1.50
LPM-6 [42]	60.71	77.60	84.93	88.76	91.10	80.62	2.61
OCPM [39]	63.67	80.26	87.10	90.65	93.01	82.94	4.93
DPN	**67.19**	**82.81**	**89.27**	**92.63**	**94.88**	**85.36**	**7.29**

Table 6.3: Parameters comparison.

Model	***Parameters(M)***	***Flops(G)***
CPM [38]	36.80	103.23
OCPM [39]	29.28	80.53
DPN	**8.55**	**16.38**

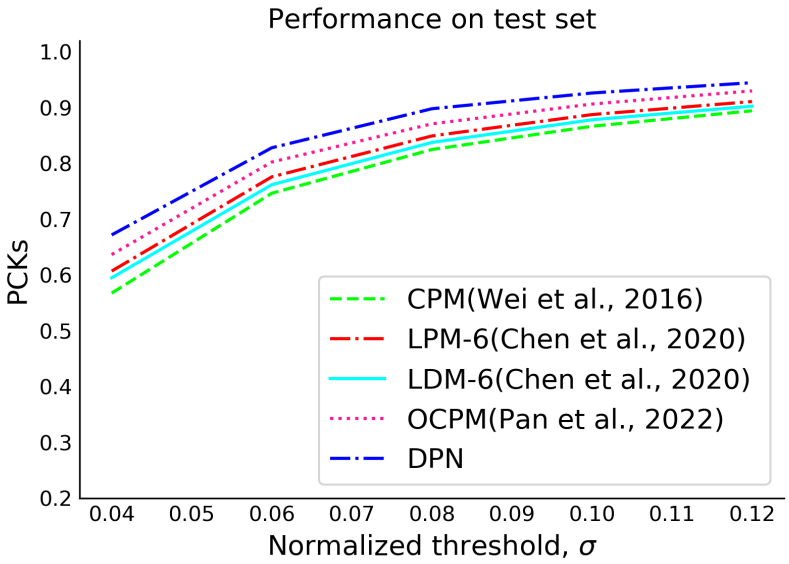
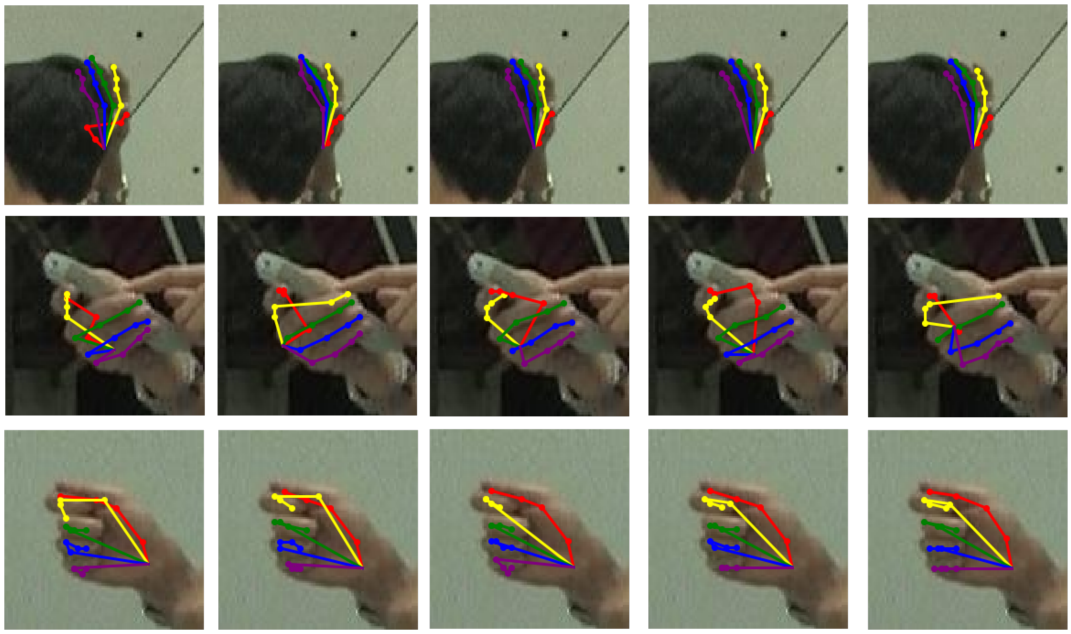


Figure 6.3: PCK comparison with other lightweight 2D HPE models.

6.5.2 Qualitative Results

To visually evaluate the performance of DPN, we selected random images from the test set and visualized them. Figure 6.4 shows that our proposed network performs well, especially when processing low-light and blurred images. These results show that our DPN outperforms other lightweight state-of-the-art models.



(a) CPM (b) LDM (c) LPM (d) OCPM (d) DPN

Figure 6.4: Visual illustration of predicted hand keypoints.

6.6 Ablation Study

To demonstrate the effectiveness of DC in the stages, an ablation study was conducted by training a network without DC. The results showed that, surprisingly, DC outperforms convolution even when compared to a 6-stage network with many parameters. The numerical results shown in Table 6.4 confirm that the inclusion of DC significantly improves the accuracy of the network.

Table 6.4: Comparison of six-stages without DC and four-stage with DC.

Threshold σ	***0.04***	***0.06***	***0.08***	***0.10***	***0.12***	***Average***
6 stage without DC	62.09	78.82	85.64	90.27	92.42	81.85
4 stage with DC	67.19	82.81	89.27	92.63	94.48	85.30

6.7 Discussion and Analysis

The lightweight multi-stage deformable convolutional network for 2D hand pose estimation proposed in this paper utilizes EfficientNet as a framework to improve feature extraction. By integrating deformable convolutions at each step to solve geometric constraints, our approach significantly improves over traditional convolutional methods. Through evaluation of a publicly available CMU hand dataset, our proposed method outperforms state-of-the-art networks in both accuracy and computational complexity; using EfficientNet as a basis facilitates the exploration of hidden information, contributing to improved performance. Furthermore, including deformable convolution improved the adaptation to geometric changes in hand pose, leading to more accurate predictions. However, it is important to note that despite these advancements, the model still faces certain limitations. While efforts have been made to reduce computational complexity, the model remains computationally expensive, which may pose challenges in some real-time applications or resource-constrained environments. Additionally, the model's performance in highly occluded scenarios, though improved, still shows room for enhancement. These limitations highlight areas for future research and refinement.

This study highlights the effectiveness of combining EfficientNet and deformable convolution for 2D HPE. The results emphasize the importance of innovative approaches to address the challenges inherent in the HPE task

and advance the state-of-the-art in this field. Future work could focus on further optimizing computational efficiency and improving the model's robustness in heavily occluded situations.

Chapter 7

Attention-Driven Contextual Feature-Enhanced Deformable Convolutional Based Network

Introduction

In this chapter, we introduce a novel approach to tackling the complexities of 2D Hand Pose Estimation (HPE) was created by creating a multi-stage network called ACDCNET that addresses HPE's challenges that still exist in the previous approaches discussed in Chapter 3, 4, 5, and 6. Our model consists of two primary components: the EfficientNet (EN) [61] backbone and the Deformable Convolution [65, 64] (DC) block. EfficientNet is used for its ability to balance computational efficiency with model effectiveness. It comes in several versions, from B0 to B7, each offering a different balance between processing speed and accuracy. The B0 model, being the smallest, is particularly useful for situations where quick computations are as necessary as precision.

However, making the network shallower decreases performance due to the trade-off between accuracy and computational cost. We incorporate a Squeeze and Excitation (SE) [62] block as an attention mechanism to enhance

our model. This mechanism focuses on identifying and emphasizing the most significant features. It compresses the spatial dimensions and then uses a fully connected layer to capture channel-specific dependencies. Following this, an excitation process adjusts the feature map according to learned importance weights, improving feature representation. Furthermore, we integrate a Global Context (GC) [63] block. This addition is crucial for efficiently generating a confidence map that assigns probabilities to each pixel in the image, enabling the model to aggregate contextual information from different levels better. The GC block helps understand the image's global context, making it easier for the model to accurately estimate hand poses by considering the overall scene alongside local features. This comprehensive approach ensures that our model captures detailed local features with high precision, leading to more accurate and robust hand pose estimations. Moreover, our model's DC block draws inspiration from the well-known Convolutional Pose Machine (CPM) framework, which is structured around a six-stage Convolutional Block (CB) architecture. We have designed this by adopting more streamlined four-stage DC blocks, mainly constructed to address geometric constraints more efficiently. This modification raises the computational performance of our approach and improves its capacity to uncover and understand fine, complex details, such as geometric constraints. Consequently, these enhancements lead to more precise outcomes in 2D Hand Pose Estimation (HPE).

7.1 ACDCNet Architecture Components

We propose an innovative approach, ACDCNet, for 2D HPE that utilizes a multistage deformable convolutional network; as a backbone for the network, we utilize the modified EN [61] B0 with SE [62] attention block and

GC [63] block to get the enhanced features. The architectural components of the proposed model are shown in Figure 7.1.

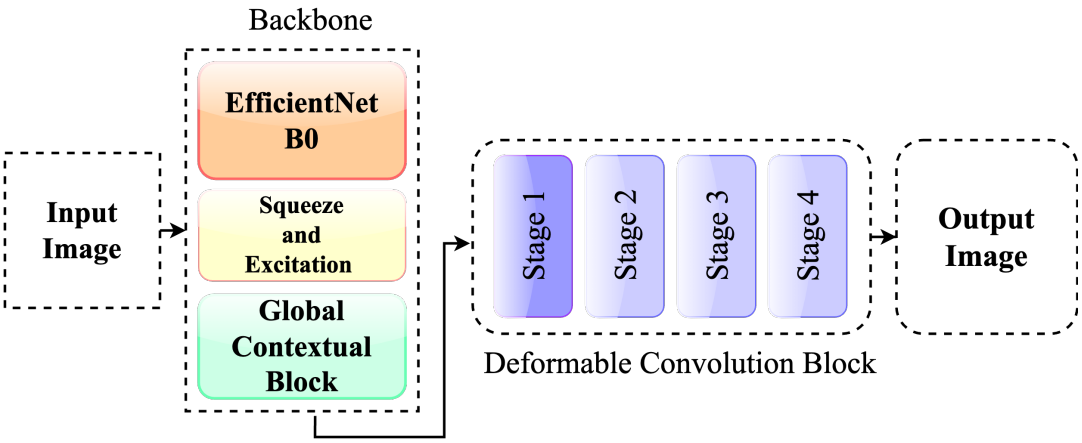


Figure 7.1: Architectural components of ACDCNet.

7.1.1 Enhanced feature extraction using EN B0

EN is known for its excellent balance between model accuracy and computational efficiency. It forms the basis of our proposed approach for feature extraction, using EN's state-of-the-art architecture, specifically the EN-B0 version, for optimal resource utilization. The enhanced EN-B0 architecture, shown in Figure 7.2, consists of seven blocks, a structure inspired by MobileNetv2, with a different number of MBConvs in each block. Notably, the last fully connected convolutional layer has been eliminated to streamline the model's parameters and increase its capacity as a feature extractor. Sequential processing starts with a $\mathbf{3} \times \mathbf{3}$ convolution operation on the input data, followed by an MBConv operation, and finally, 64 feature maps are generated. These features are further refined in the SE and GC blocks to facilitate robust and efficient feature extraction.

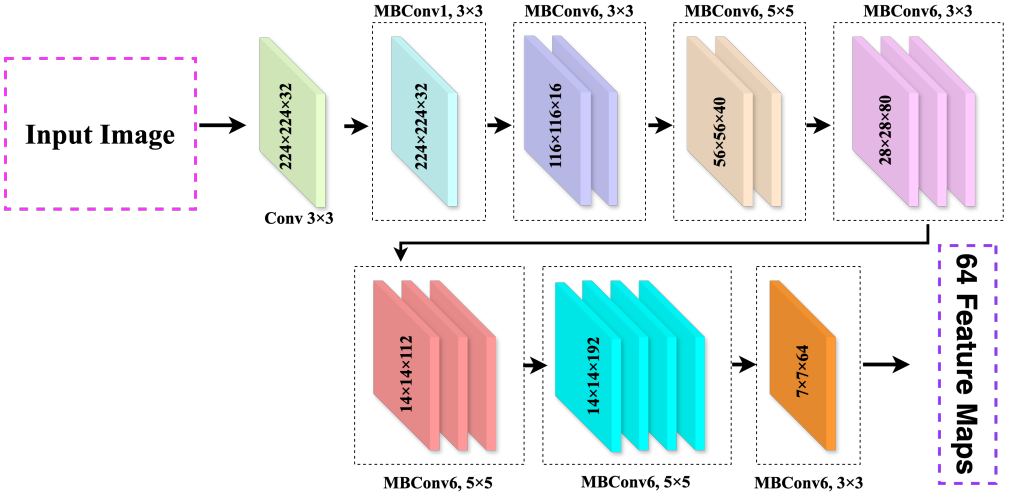


Figure 7.2: Detailed architecture of modified EfficientNet B0.

7.1.2 Improving feature representation with SE Block

The SE block first performs a channel-wise transformation on the feature maps produced by EN, denoted as:

$$U = f_c(X = \sigma(W_2\delta(W_1)X) \tag{7.1}$$

Where X is the feature map, W_1 and W_2 are the adaptive weight matrices, δ is the ReLU activation function, σ is the sigmoid activation function and U is the result of the channel-wise transformation.

The recalibration coefficients are then determined by aligning the U dimension with the X dimension of the original feature map:

$$s = f_s(U) = W_3U \tag{7.2}$$

Here, W_3 represents the trainable weight matrix and s the recalibration coefficients.

Finally, feature recalibration is performed by element-wise multiplication:

$$Y = f_r(X, s) = X \odot s \tag{7.3}$$

where $\odot$ represents element-wise multiplication and $\boldsymbol{Y}$ represents the augmented feature map. By dynamically adjusting the input features according to the importance of the channel, the SE block empowers the neural network to emphasize important information, thus improving its ability to identify complex patterns in data and increasing its effectiveness in various computational tasks, including HPE. Figure 7.3 shows an overview of the SE block.

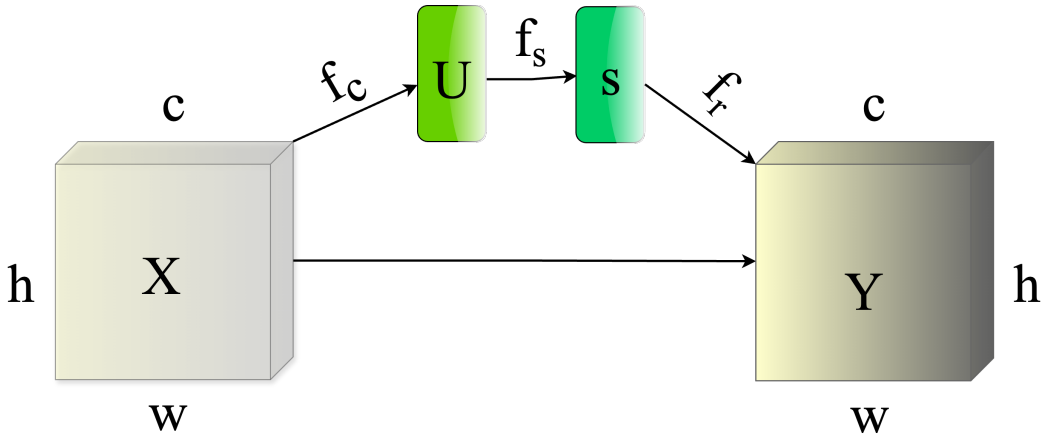


Figure 7.3: Detail overview of SE block.

7.1.3 Enhancing features through GC block

Our HPE architecture combines GC block with a backbone and incorporates hierarchical context aggregation to enhance the model's ability to describe complex spatial relationships.

The global context is included through adaptive average pooling, defined as follows:

$$\boldsymbol{Z} = \text{AdapAvgPool}(\boldsymbol{X}) \tag{7.4}$$

Where $\boldsymbol{X}$ is the feature map after passing through the SE block, and $\boldsymbol{Z}$ represents the global context representation.

To better specify this context, convolutional transformations are used to calculate attention weights and estimate the relative importance of different

contextual aspects:

$$W = \text{Conv2d}(Z) \tag{7.5}$$

Here, W denotes the attention weights after the convolutional operation.

This process generates an input representation (X_s) at different scales (s) such as **0.5**, **1.0**, and **2.0** and simplifies the computation of the context map. The context map embedded in the corresponding scale M_s is carefully aligned to the dimensions of the original feature map using interpolation methods:

$$M'_s = \text{Interpolate}(M_s, S(X)) \tag{7.6}$$

Here, M'_s represents the reshaped contextual map.

Here GC block achieves contextual feature map, and the aggregation by summing elements from different scales, denoted as A:

$$A = \sum_s M'_s \tag{7.7}$$

Using different perspectives in the GC block optimizes the accuracy and efficiency of HPE. Figure 7.4 shows the detailed characteristics of the GC block, where X, W, H, and C represent the input features, width, height, and channels, respectively.

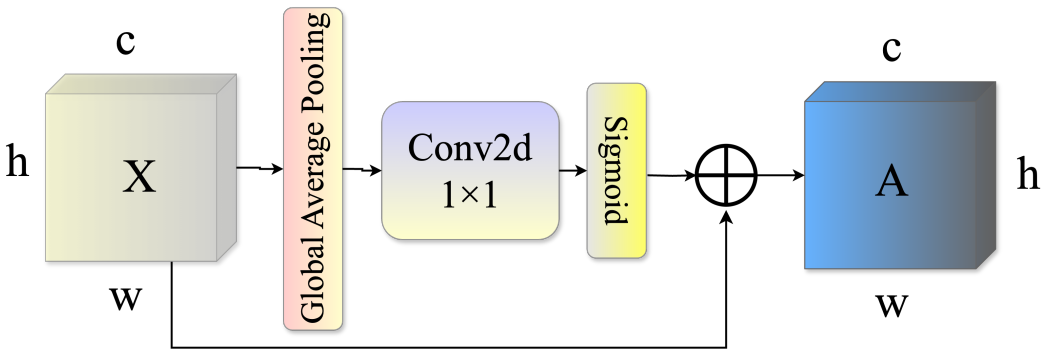


Figure 7.4: Visualization of GC block.

7.1.4 Multi-stage DC Block

CPM serves as a basic CNN-based model for pose estimation and addresses various complexities inherent in HPE. However, CPM has limitations, mainly unknown geometric constraints and other problems mentioned in the literature; we utilize a DC to mitigate these shortcomings of CNN-based models. This convolution is designed to manage geometric constraints and improve the model's adaptability when learning unknown features during information processing.

Our proposed approach includes a four-stage network architecture. The initial stage comprises two $\mathbf{3} \times \mathbf{3}$ Deformable Convolutional Blocks (DCBs) with 64 channels. The next stages consist of seven $\mathbf{3} \times \mathbf{3}$ DCBs, each DCB has 32 channels. A detailed overview of information processing by these DCBs is shown in Figure 7.5.

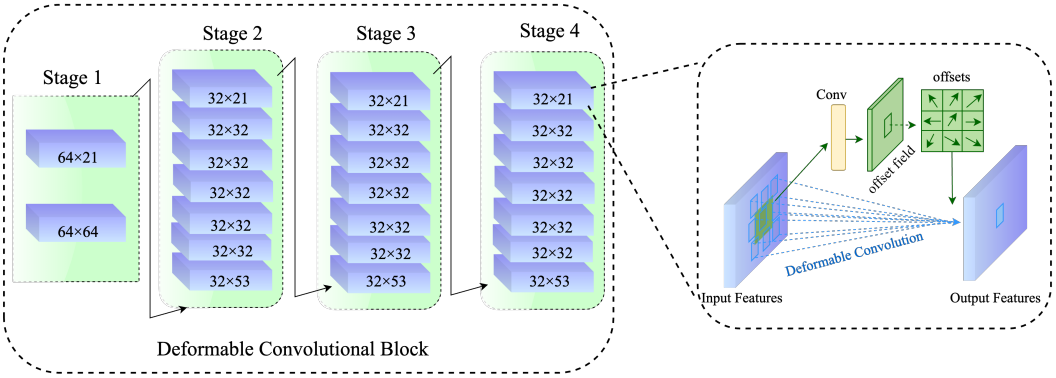


Figure 7.5: Detailed overview of information processing multi-stage DCB.

The output feature maps generated by the backbone network are fed to the initial DCB stages in this network for further information processing. Each stage contains a DC mechanism consisting of two convolutional layers (CL), an offset CL, and a modulator CL. The DC operations detailed below provide further refinement of the feature representation for accurate pose estimation.

7.1.4.1 Spatial offset calculation

The offset CL computes the spatial offset from the contextual feature maps $\boldsymbol{A}$ obtained from the GC block, which is the reference output feature map, using a trainable parameterization. This process is called OF and is mathematically expressed as follows:

$$\boldsymbol{OF} = \boldsymbol{W_1} * \boldsymbol{A} \tag{7.8}$$

Here, $\boldsymbol{W_1}$ signifies the convolutional operation applied to the input $\boldsymbol{A}$ to compute the offsets. These offsets aid in determining the sampling location within $\boldsymbol{A}$, thereby enhancing the flexibility of receptive fields. The difference of sampling location of standard convolution and deformable convolution is shown in Figure 7.6.

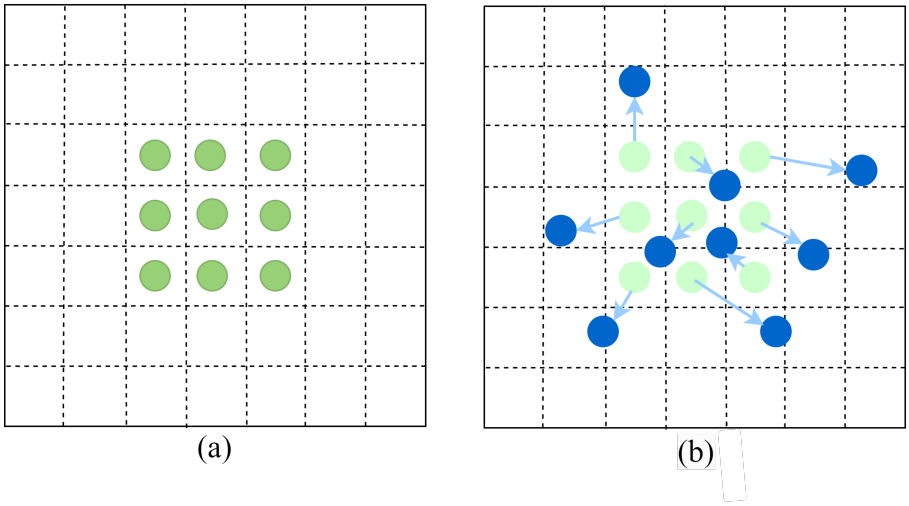


Figure 7.6: (a) Sampling of standard convolution (b) Sampling of deformable convolution

7.1.4.2 Modulation of sampled regions

Modulation weights are generated using a sigmoidal function applied to the input $\boldsymbol{A}$ that facilitates and dominates the modulation of the sampling region using the modulator CL. Mathematically, this process can be expressed as follows:

$$\boldsymbol{M} = \boldsymbol{2} \times \sigma(\boldsymbol{MC}(\boldsymbol{A})) \tag{7.9}$$

Where $\boldsymbol{M}$ is the modulator, σ is the sigmoid function and $\boldsymbol{MC}$ is the convolution operation applied to the contextual feature map $\boldsymbol{A}$. These modulation weights allow features to be adaptively tuned based on their importance.

7.1.4.3 Dynamic adaptation in DC

After applying the first two convolutional layers (CL), the essence of deformable convolution (DC) is revealed through the DC operation. This operation easily combines spatial offsets and modulation coefficients with conventional convolutional layers, contributing to a dynamic learning environment. Mathematically, this operation is expressed by the following equation:

$$\boldsymbol{x = deform2d(A, OF, w, b, M)} \tag{7.10}$$

$\boldsymbol{A}$ is the contextual feature map, $\boldsymbol{OF}$ is the spatial offset, and $\boldsymbol{w}$, $\boldsymbol{b}$, and $\boldsymbol{M}$ are the convolution weights, offset, and modulation factor, respectively. This integration facilitates adaptive tuning of the receptive field, allowing the model to perceive subtle features and navigate complex geometric constraints. The results of the initial stage are seamlessly transferred to subsequent stages to move the computational pipeline forward. This process is repeated in all four stages, resulting in the extraction of 21 key points. Notably, the operation is simplified by reducing the number of stages and channels compared to the CPM and our DPN, improving overall adaptability and computational efficiency.

7.2 Experimental Setups

We utilize the PyTorch framework to implement our proposed architecture and train for 100 epochs with a batch size of 34. The images were scaled from 0 to 1, then normalized using mean and standard deviation values of

$(\mathbf{0.485, 0.456, 0.406})$ and $(\mathbf{0.229, 0.224, 0.225})$, respectively. Mean Squared Error is used as a loss function. To prevent the diminishing of loss, we applied a scaling factor of 35 to adjust the loss function accordingly.

7.2.1 Model Optimizer and Activation Function

Optimization methods guide networks to achieve optimal results to enhance model performance. AdamW, an advanced version of the Adam optimizer, stands out by separating the weight regularization parameter from the learning rate. This allows for precise tuning of optimization settings. Unlike Adam, AdamW effectively addresses overfitting issues, resulting in better generalization capabilities. Our research demonstrates that models optimized with AdamW surpass those trained using other optimizers. This reinforces the significance of selecting the right optimizer in model optimization.

The activation function adds non-linearity, enabling the network to learn intricate patterns in data. The Mish [55] function has emerged as a promising activation function due to its unique non-linearity defined mathematically as:

$$f(x) = \boldsymbol{x tanh(ln(1 + e^x))}. \tag{7.11}$$

Compared to popular alternatives like ReLU [53] and SoftMax [54], Mish [55] has demonstrated exceptional performance across various deep network architectures and complex datasets. This emphasizes the activation function's influence in optimizing network performance and enhancing its ability to generalize to unseen data.

7.2.2 Evaluation Metric

This study utilized the Percentage of Correct Keypoints (PCK) [56] metric widely used in HPE. The PCK metric measures the accuracy of predicted keypoints within a certain distance threshold, denoted as σ and corresponding. It evaluates the accuracy of predicted keypoints by measuring their proximity to true coordinates. In our experiments, σ was limited by the scale of the bounding box of the hand. The σ threshold was uniformly distributed between 0.04 and 0.12. The PCK equation is expressed as follows:

$$PCK_{\sigma}^{k} = \frac{1}{||D||}\sum_{D} 1\left(\frac{||p_k^{pt} - p_k^{gd}||_2}{\max(w,h)} \leq \delta\right) \tag{7.12}$$

Where p_k^{gd} is the true keypoint, p_k^{gd} is the predicted keypoint, k is the number of keypoints, D is the number of tests or validation samples, h and w are the height and width of the sample image, respectively.

7.3 Experimental Results and Analysis

7.3.1 Quantitative Results

The average performance of the different models is summarized in Table 7.1, which shows the relative performance of each approach at different σ thresholds. Our proposed model achieves an average PCK of 86.16 %, showing robustness and consistency in accurately predicting key points at different levels of complexity. In contrast, CPM [38] and OCPM [39] achieve average PCK scores of 78.01 % and 82.94 %, respectively. Figure 7.7 shows the PKC comparison of Our with CPM [38], LDM [42], LPM [42], HRNet [45], Hourglass [46] and OCPM [39], which shows better performance compared to the existing lightweight methods. These results indicate that our model outperforms, in terms of average performance, performs better on a wide variety of

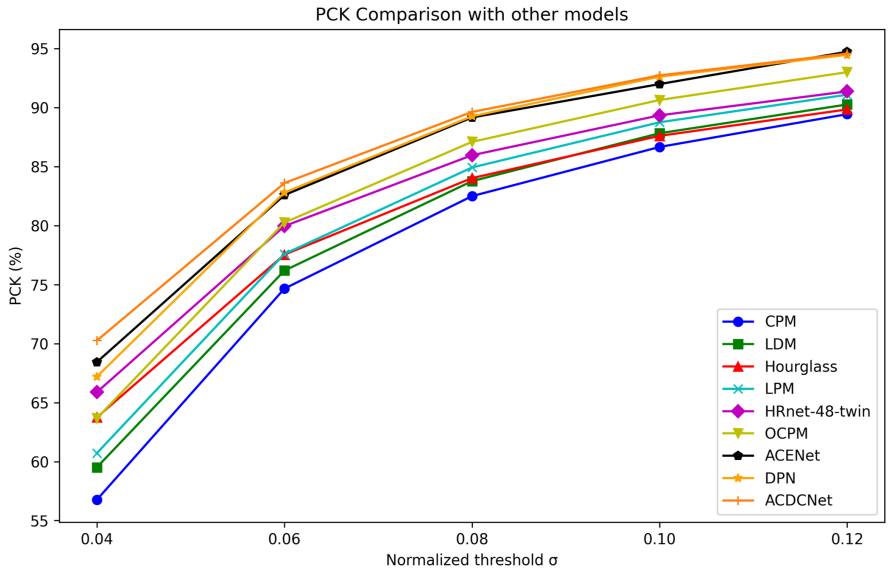


Figure 7.7: PCK visualization of the proposed approach and SOTA.

poses and hand configurations. Comparing the different models' parameters yields impressive results regarding computational complexity. As shown in Table 7.1 and Figure 7.8, our proposed architecture has 7.88 million parameters, which is significantly less than the 36.8 million parameters of CPM [38] and 29.28 million parameters of OCPM [39]. Notably, our method's reduced number of parameters represents a more rational and efficient model architecture, contributing to computational efficiency without sacrificing performance.

7.3.2 Qualitative Results

Various images with different perspectives, complex situations, self/object occlusion cases, and complex backgrounds were selected to demonstrate qualitative results. Figure 7.9 shows our approach's high robustness and consistency, which is reflected in its reliable performance in numerous test scenarios and conditions. Even in complex backgrounds, the interference avoidance feature of ACDCNet worked effectively. In situations where image

Table 7.1: Performance Comparison of ACDCNet with the state-of-the-art models

Model	PCK (%)					Ave	Par (M)	GFLOPs
	σ 0.04	σ 0.06	σ 0.08	σ 0.10	σ 0.12			
CPM [38]	56.76	74.66	82.50	86.67	89.45	78.01	36.80	103.23
LDM [42]	59.51	76.19	83.77	87.83	90.27	79.51	38.19	95.18
Hourglass [46]	63.75	77.54	84.03	87.61	89.85	80.56	-	-
LPM [42]	60.71	77.60	84.93	88.76	91.10	80.62	38.38	92.18
HRnet-48-twin [45]	65.88	79.96	85.97	89.35	91.38	82.51	-	-
OCPM [39]	63.67	80.26	87.10	90.65	93.01	82.94	29.28	80.53
DPN [67]	67.19	82.81	89.27	92.63	94.88	85.36	8.55	16.38
ACDCNet	**70.25**	**83.61**	**89.64**	**92.74**	**94.98**	**86.24**	**7.88**	**14.89**

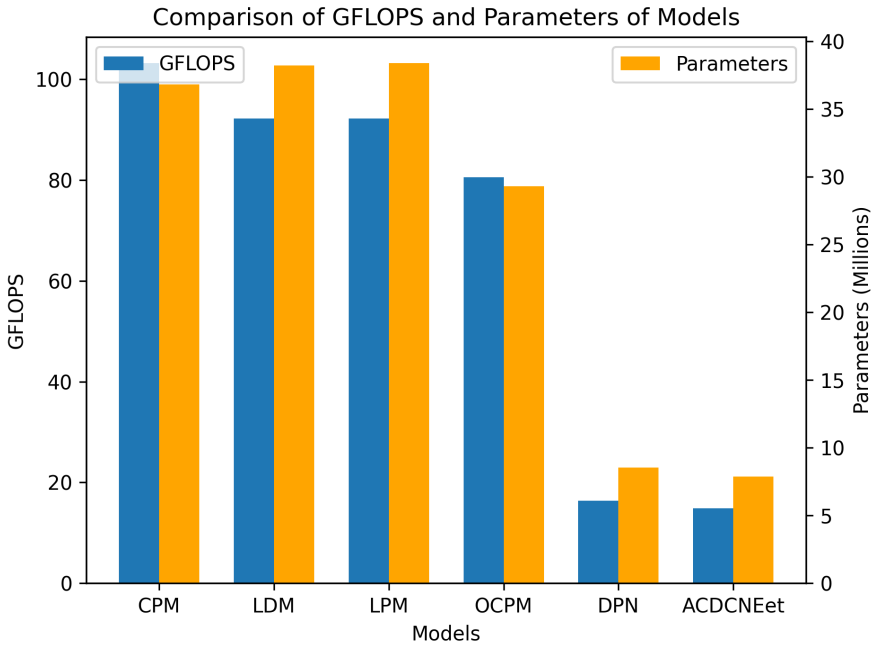


Figure 7.8: Parameters and GFLOPs comparison of ACDCNet with other models.

clarity is compromised, obtaining higher resolution and more detailed image versions was deemed necessary to improve interpretation and analysis. The model cannot accurately predict keypoints in some scenarios because the hands are highly occluded. To highlight these circumstances the circle red keypoints represent ground-truth keypoints.

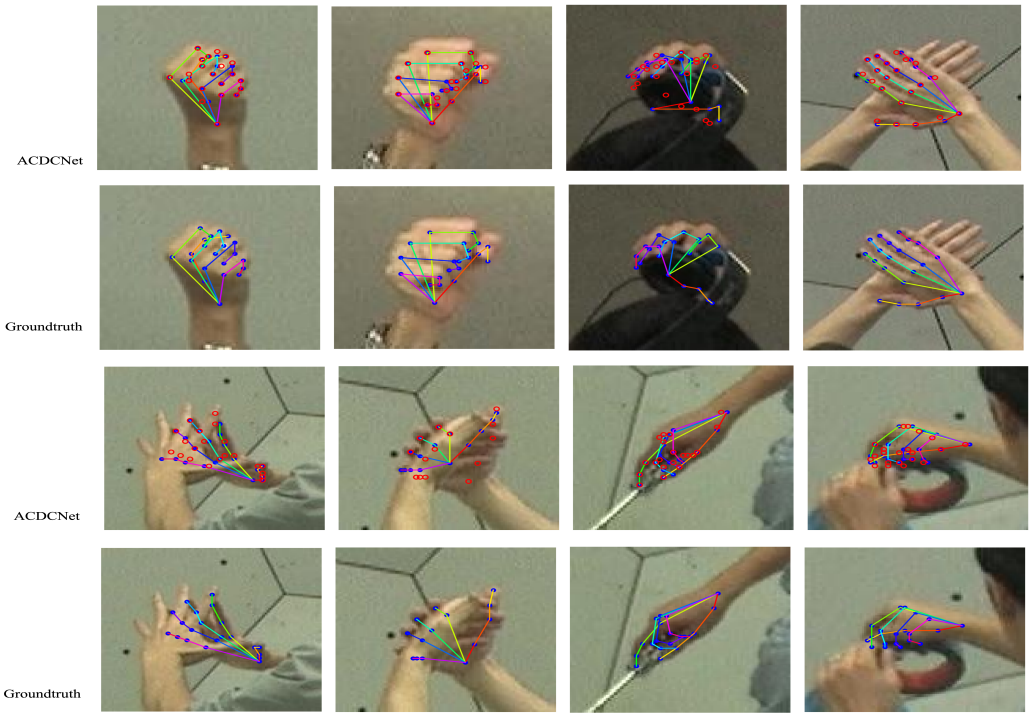


Figure 7.9: ACDCNet visual illustration on random test images.

7.4 Ablation Study

An ablation study was conducted to show the impact of integrating SE and GC blocks. In this study, we trained ACDCNet to determine their respective contributions with and without these blocks. Surprisingly, models with SE and GC blocks were consistently better than those without. As can be seen in Table 7.2 and PCK comparison in Figure 7.10, the inclusion of SE and GC blocks significantly improves network performance by maintaining the tradeoff between computational complexity and accuracy, as shown in

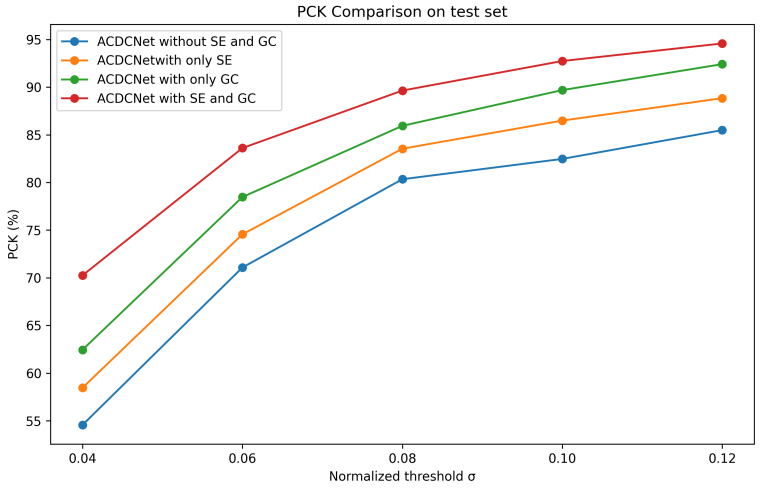


Figure 7.10: PCK comparison of our ACDCNet with and without SE and GC blocks.

Figure 7.11. These blocks contribute to accuracy and demonstrate the effectiveness of using contextual information and recalibration mechanisms for each channel.

Table 7.2: Performance Comparison of Different Models

Model	PCK					Ave	Para (M)	GFLOPs
	σ 0.04	σ 0.06	σ 0.08	σ 0.10	σ 0.12			
Without SE and GC	54.56	71.08	80.34	82.47	85.49	74.98	6.12	12.00
With SE	58.46	74.56	83.54	86.49	88.84	78.97	6.98	12.58
With GC	62.45	78.47	85.94	89.69	92.41	81.62	7.02	14.15
ACDCNET	**70.25**	**83.61**	**89.64**	**92.74**	**94.58**	**86.16**	**7.88**	**14.89**

7.5 Discussion and Analysis

Our proposed ACDCNet is an essential advancement in 2D HPE. By integrating innovative components such as the modified EfficientNet B0 framework,

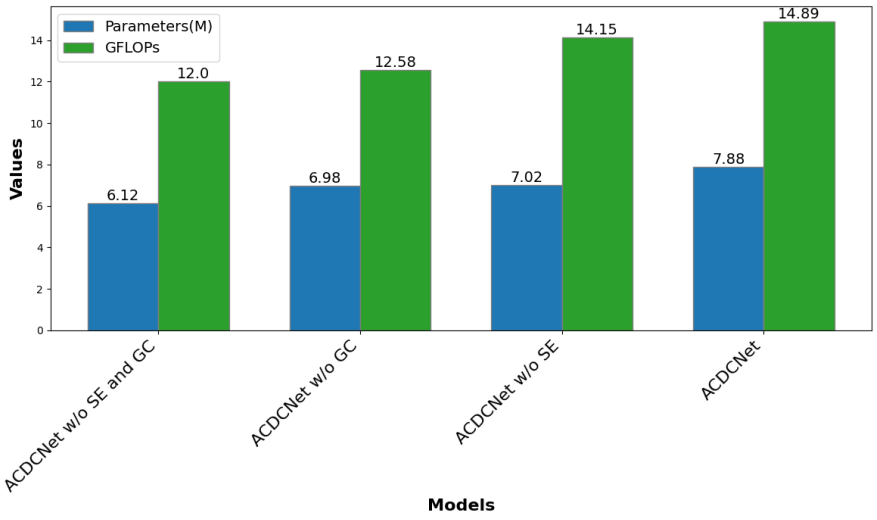


Figure 7.11: Parameter and GFLOps comparison of our ACDCNet with and without SE and GC blocks.

SE block, and GC block, ACDCNet achieves significant accuracy while maintaining computational efficiency. Our experimental results show that ACDCNet outperforms state-of-the-art models at various thresholds, demonstrating robustness and consistency in accurately predicting keypoints. The model performs better with significantly fewer parameters, indicating a more rational and efficient architecture. Qualitative evaluations emphasize the robustness of ACDCNet in various scenarios, effectively handling occlusions and complex backgrounds. Visualizations further confirm the importance of SE and GC blocks in capturing complex details and contextual information, improving overall model performance.

The ablation studies performed to confirm the importance of SE and GC blocks in improving network performance. By dynamically recalibrating the importance of each channel and improving context aggregation, these blocks contribute to improving model accuracy without compromising computational efficiency. While ACDCNet represents a significant step towards lightweight models in the field of 2D HPE, it is important to acknowledge its limitations. Despite our efforts to reduce computational complexity, the

model still requires more computational resources than what is typically available for real-time processing on edge devices. This limitation highlights the ongoing challenge in the field to balance high accuracy with the constraints of real-time implementation, especially on resource-limited platforms.

Overall, ACDCNet shows promising potential for various applications in human-computer interaction, augmented and virtual reality, and provides a balanced approach between accuracy and efficiency in 2D HPE tasks. However, the computational requirements still pose challenges for real-time implementation on edge devices. This limitation opens avenues for future research, focusing on further optimizing the model architecture and exploring hardware-specific optimizations to enable real-time performance on a wider range of devices. In conclusion, while ACDCNet represents a significant advancement in the field of 2D HPE, striking a balance between model accuracy and real-time performance on edge devices remains an important area for future work. The insights gained from this research provide a solid foundation for further innovations in creating highly accurate, computationally efficient HPE models suitable for a broader range of real-world applications.

Chapter 8

Conclusion and Future Work

In conclusion, this dissertation has made substantial progress in fulfilling the primary objectives outlined at the outset: to enhance model efficiency without compromising accuracy and to advance feature extraction and geometrical understanding in the domain of 2D hand pose estimation (HPE). This dissertation proposed several innovative deep-learning architectures designed to address the complex challenges inherent in HPE. By exploring efficient neural network architectures, including variants such as VGG and the EfficientNets family, we have successfully streamlined model complexity while preserving high levels of accuracy. These architectures represent a significant breakthrough, demonstrating the feasibility of balancing computational efficiency and precise hand pose estimation. Moreover, our methodologies have yielded tangible advancements in feature extraction and geometrical understanding, critical components for accurate HPE. By integrating attention mechanisms into our models, we have empowered them to discern complex spatial details amidst cluttered backgrounds and occlusions. These mechanisms guide the model's focus toward salient hand regions while suppressing irrelevant information, resulting in improved feature extraction and enhanced model performance.

Incorporating global contextual modules has further enriched our models' understanding of the scene, enabling them to capture essential contextual cues crucial for precise pose estimation. By considering the relationships

and dependencies between different hand regions within the scene, our models gain a holistic perspective, enhancing their ability to infer hand poses in diverse real-world scenarios accurately. Furthermore, deformable convolutions have revolutionized our approach to handling geometric complexities inherent in hand movements. By allowing the model to adapt its receptive fields and sampling locations based on the spatial context, deformable convolutions enable more accurate feature extraction and localization, even in varying hand shapes and orientations.

Our contributions have led to approximately **3**$\times$ lower parameters and approximately **5**$\times$ lower Gflops compared to the state-of-the-art model while maintaining a 3.88% heigher accuracy. Although it is not yet suitable for real-time applications, our model represents a significant step forward, offering a more efficient and accurate approach to 2D HPE. In this context, real-time performance would typically require processing at least 30 frames per second (fps), with many applications aiming for 60 fps or higher for smoother interaction. These advancements hold immense potential for transformative impact across many fields, including human-computer interaction, robotics, and augmented reality, propelling technological progress and enhancing societal well-being globally.

In considering the future research directions for this dissertation, several promising routes to explore using advanced deep learning architectures and techniques to enhance feature extraction for hand pose estimation are identified. One key direction could be investigating transformer-based models, such as Vision Transformers (ViT), building upon the attention mechanisms and advanced machine learning algorithms discussed earlier in our dissertation. Transformers have demonstrated impressive performance in various computer vision tasks by effectively capturing long-range dependencies and global context. Adapting transformer architectures for hand pose estimation could improve feature extraction and complex hand structure modeling.

Additionally, integrating convolutional attention mechanisms, like Coordinate Attention, can enable the model to selectively focus on the most informative spatial regions and feature channels. This can help the model extract and emphasize the relevant features for accurate hand pose estimation. Developing hierarchical feature extraction architectures, where the model learns features at multiple scales or resolutions, can also be a fruitful direction. Techniques like feature pyramid networks or multi-scale feature aggregation can allow the model to capture features at different levels of granularity, better representing the intricate details of the hand.

Exploring the use of equivariant representations, such as those obtained through group convolutions or steerable CNNs, can ensure that the model's feature extraction is invariant to certain transformations (e.g., rotation, scaling) of the hand pose. This can improve the model's generalization and robustness. Combining the deep learning model with differentiable rendering techniques, as explored in methods like HAMR, can enable end-to-end training and better integration of the 3D hand structure into the feature extraction process. Furthermore, investigating unsupervised or self-supervised feature learning approaches, such as contrastive learning or generative adversarial networks, can help extract more robust and generalizable features from the hand images without relying solely on labeled data. Finally, exploring hybrid architectures that combine the strengths of different deep learning models, such as integrating convolutional neural networks with recurrent neural networks or graph neural networks, can more effectively capture both local and global hand pose features.

A crucial extension of this research is the adaptation of our proposed methods to 3D Hand Pose Estimation (HPE). This transition presents both challenges and opportunities to enhance the applicability and accuracy of our models. To extend our work to 3D HPE, we can explore several approaches: integrating depth information alongside RGB data, developing

2D-to-3D lifting techniques, adapting our architectures for end-to-end 3D estimation, utilizing multi-view approaches for more accurate 3D reconstruction, incorporating temporal information for stable pose estimation across video sequences, and exploring volumetric representations of the hand. These 3D extensions, combined with the advanced deep learning techniques mentioned earlier, have the potential to address a wider range of applications, including virtual reality interactions, detailed hand tracking for motion capture, and more accurate gesture recognition in 3D space.

By incorporating these advanced deep learning techniques and architectures, and extending our work to 3D HPE, we can further enhance our models' feature extraction capabilities, leading to improved accuracy, robustness, and generalization across diverse hand pose scenarios in both 2D and 3D spaces. These advancements can significantly impact applications ranging from human-computer interaction and augmented reality to sign language recognition and gesture-based interfaces, providing more comprehensive and versatile solutions for hand pose estimation across various dimensions and use cases.

References

[1] Michael Cheng-Tek Tai. "The impact of artificial intelligence on human society and bioethics". In: *Tzu chi medical journal* 32.4 (2020), pp. 339–343.

[2] Julie A Jacko. "Human computer interaction handbook: Fundamentals, evolving technologies, and emerging applications". In: (2012).

[3] Mafkereseb Kassahun Bekele and Erik Champion. "A comparison of immersive realities and interaction methods: Cultural learning in virtual heritage". In: *Frontiers in Robotics and AI* 6 (2019), p. 91.

[4] Tiberiu Boros and Stefan Daniel Dumitrescu. "A "small-data"-driven approach to dialogue systems for natural language human computer interaction". In: *2017 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. IEEE. 2017, pp. 1–6.

[5] Enrique Fernández-Rodicio et al. "Modelling multimodal dialogues for social robots using communicative acts". In: *Sensors* 20.12 (2020), p. 3440.

[6] Xiao Li et al. "A critical review of virtual and augmented reality (VR/AR) applications in construction safety". In: *Automation in construction* 86 (2018), pp. 150–162.

[7] Leyla Khaleghi et al. "Multi-view video-based 3D hand pose estimation". In: *IEEE Transactions on Artificial Intelligence* (2022).

[8] Muneeb Ur Rehman et al. “Dynamic hand gesture recognition using 3D-CNN and LSTM networks”. In: *Computers, Materials & Continua* 70.3 (2021).

[9] Bosang Kim et al. “Mesh Represented Recycle Learning for 3D Hand Pose and Mesh Estimation”. In: *arXiv preprint arXiv:2310.12189* (2023).

[10] Dong Uk Kim, Kwang In Kim, and Seungryul Baek. “End-to-end detection and pose estimation of two interacting hands”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 11189–11198.

[11] Farnaz Farahanipad et al. “A survey on GAN-based data augmentation for hand pose estimation problem”. In: *Technologies* 10.2 (2022), p. 43.

[12] Rui Jin and Jianyu Yang. “Domain adaptive hand pose estimation based on self-looping adversarial training strategy”. In: *Sensors* 22.22 (2022), p. 8843.

[13] Yang Liu et al. “Internet+: A light network for hand pose estimation”. In: *Sensors* 21.20 (2021), p. 6747.

[14] Sungheon Park, Jihye Hwang, and Nojun Kwak. “3d human pose estimation using convolutional neural networks with 2d pose information”. In: *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*. Springer. 2016, pp. 156–169.

[15] Shuaibing Wang et al. “A Survey of Deep Learning-based Hand Pose Estimation”. In: *2022 IEEE 8th International Conference on Cloud Computing and Intelligent Systems (CCIS)*. IEEE. 2022, pp. 331–340.

[16] Takehiko Ohkawa, Ryosuke Furuta, and Yoichi Sato. “Efficient annotation and learning for 3d hand pose estimation: A survey”. In: *International Journal of Computer Vision* 131.12 (2023), pp. 3193–3206.

[17] Taeyun Woo et al. "A survey of deep learning methods and datasets for hand pose estimation from hand-object interaction images". In: *Computers & Graphics* 116 (2023), pp. 474–490.

[18] Theocharis Chatzis et al. "A comprehensive study on deep learning-based 3D hand pose estimation methods". In: *Applied Sciences* 10.19 (2020), p. 6850.

[19] Chenfei Zhu et al. "SARN: Shifted Attention Regression Network for 3D Hand Pose Estimation". In: *Bioengineering* 10.2 (2023), p. 126.

[20] Sani Salisu et al. "A Survey on Deep Learning-Based 2D Human Pose Estimation Models." In: *Computers, Materials & Continua* 76.2 (2023).

[21] Xiaozheng Zheng et al. "Joint-Aware Regression: Rethinking Regression-Based Method for 3D Hand Pose Estimation." In: *BMVC*. 2021, p. 344.

[22] Umar Iqbal et al. "Hand pose estimation via latent 2.5 d heatmap regression". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 118–134.

[23] Markus Oberweger and Vincent Lepetit. "Deepprior++: Improving fast and accurate 3D hand pose estimation". In: *Proceedings of the IEEE international conference on computer vision Workshops*. 2017, pp. 585–594.

[24] Xingyi Zhou et al. "Model-based deep hand pose estimation". In: *arXiv preprint arXiv:1606.06854* (2016).

[25] Tomas Simon, Hanbyul Joo, and Yaser Sheikh. "Hand Keypoint Detection in Single Images using Multiview Bootstrapping". In: *CVPR* (2017).

[26] Xiao Sun et al. "Cascaded hand pose regression". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 824–832.

[27] Xinghao Chen et al. “Pose guided structured region ensemble network for cascaded hand pose estimation”. In: *Neurocomputing* 395 (2020), pp. 138–149.

[28] Liuhao Ge et al. “Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3593–3601.

[29] Shanxin Yuan et al. “Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4866–4874.

[30] Xiaozheng Zheng et al. “HaMuCo: Hand Pose Estimation via Multiview Collaborative Self-Supervised Learning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 20763–20773.

[31] James Steven Supančič et al. “Depth-based hand pose estimation: methods, data, and challenges”. In: *International Journal of Computer Vision* 126 (2018), pp. 1180–1198.

[32] James S Supancic et al. “Depth-based hand pose estimation: data, methods, and challenges”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1868–1876.

[33] Jonathan Tompson et al. “Real-time continuous pose recovery of human hands using convolutional networks”. In: *ACM Transactions on Graphics (ToG)* 33.5 (2014), pp. 1–10.

[34] Xianghan Wang et al. “Cfam: Estimating 3d hand poses from a single rgb image with attention”. In: *Applied Sciences* 10.2 (2020), p. 618.

[35] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. “Using a single rgb frame for real time 3d hand pose estimation in the

wild". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 436–445.

[36] Christian Zimmermann and Thomas Brox. "Learning to estimate 3d hand pose from single rgb images". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 4903–4911.

[37] Bugra Tekin et al. "Direct prediction of 3D body poses from motion compensated sequences". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 991–1000.

[38] Shih-En Wei et al. "Convolutional pose machines". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4724–4732.

[39] Tianhong Pan, Zheng Wang, and Yuan Fan. "Optimized convolutional pose machine for 2D hand pose estimation". In: *Journal of Visual Communication and Image Representation* 83 (2022), p. 103461.

[40] Deying Kong et al. "Adaptive graphical model network for 2D handpose estimation". In: *arXiv preprint arXiv:1909.08205* (2019).

[41] Deying Kong et al. "Rotation-invariant mixed graphical model network for 2D hand pose estimation". In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2020, pp. 1546–1555.

[42] Yifei Chen et al. "Nonparametric structure regularization machine for 2D hand pose estimation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 381–390.

[43] Shiqiang Yang et al. "Hand pose estimation based on improved NSRM network". In: *EURASIP Journal on Advances in Signal Processing* 2023.1 (2023), pp. 1–15.

[44] Chengde Wan et al. “Dense 3d regression for hand pose estimation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5147–5156.

[45] Tianhong Pan and Zheng Wang. “High-resolution network with an auxiliary channel for 2D hand pose estimation”. In: *Multimedia Tools and Applications* (2023), pp. 1–12.

[46] Alejandro Newell, Kaiyu Yang, and Jia Deng. “Stacked hourglass networks for human pose estimation”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer. 2016, pp. 483–499.

[47] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[48] Xizhou Zhu et al. “An empirical study of spatial attention mechanisms in deep networks”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 6688–6697.

[49] Jie Song et al. “Thin-slicing network: A deep structured model for pose estimation in videos”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4220–4229.

[50] Jonathan J Tompson et al. “Joint training of a convolutional network and a graphical model for human pose estimation”. In: *Advances in neural information processing systems* 27 (2014).

[51] Mazhar Javed Awan et al. “Image-based malware classification using VGG19 network and spatial convolutional attention”. In: *Electronics* 10.19 (2021), p. 2444.

[52] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[53] Chaity Banerjee, Tathagata Mukherjee, and Eduardo Pasiliao Jr. "An empirical study on generalizations of the ReLU activation function". In: *Proceedings of the 2019 ACM Southeast Conference*. 2019, pp. 164–167.

[54] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. "Activation functions in neural networks". In: *Towards Data Sci* 6.12 (2017), pp. 310–316.

[55] Diganta Misra. "Mish: A self regularized non-monotonic activation function". In: *BMVC* (2020).

[56] Tomas Simon et al. "Hand keypoint detection in single images using multiview bootstrapping". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 1145–1153.

[57] Sartaj Ahmed Salman, Ali Zakir, and Hiroki Takahashi. "Cascaded deep graphical convolutional neural network for 2D hand pose estimation". In: *International Workshop on Advanced Imaging Technology (IWAIT) 2023*. Vol. 12592. SPIE. 2023, pp. 227–232.

[58] Görkem Algan and Ilkay Ulusoy. "Image classification with deep learning in the presence of noisy labels: A survey". In: *Knowledge-Based Systems* 215 (2021), p. 106771.

[59] Zhuang Liu et al. "A convnet for the 2020s". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11976–11986.

[60] Yu Li, Zhuoran Shen, and Ying Shan. "Fast video object segmentation using the global context module". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer. 2020, pp. 735–750.

[61] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[62] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.

[63] Yu Li, Zhuoran Shen, and Ying Shan. "Fast video object segmentation using the global context module". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer. 2020, pp. 735–750.

[64] Jifeng Dai et al. "Deformable convolutional networks". In: *Proceedings of the IEEE Conference on Computer Vision*. 2017, pp. 764–773.

[65] Feng Chen et al. "Adaptive deformable convolutional network". In: *Neurocomputing* 453 (2021), pp. 853–864.

[66] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[67] Sartaj Salman., Ali Zakir., and Hiroki Takahashi. "Deformable Pose Network: A Multi-Stage Deformable Convolutional Network for 2D Hand Pose Estimation". In: *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: VISAPP*. INSTICC. SciTePress, 2024, pp. 814–821. ISBN: 978-989-758-679-8. DOI: 10.5220/0012569000003660.

Publication Lists

Related Publications

Journal

1. **Sartaj Ahmed Salman**, Ali Zakir, and Hiroki Takahashi. "SDFPoseGraphNet: Spatial Deep Feature Pose Graph Network for 2D Hand Pose Estimation." Sensors 23, no. 22 (2023): 9088. (Chapter 3)

2. **Sartaj Ahmed Salman**, Ali Zakir, and Hiroki Takahashi. " Attention-Driven Contextual Feature-Enhanced Deformable Convolutional Based Network for 2D Hand Pose Estimation .“ Submitted to Franklin Open Journal (Under Review). (Chapter 7)

International Conference

1. **Sartaj Ahmed Salman**, Ali Zakir, and Hiroki Takahashi. "Cascaded deep graphical convolutional neural network for 2D hand pose estimation." In International Workshop on Advanced Imaging Technology (IWAIT) 2023, vol. 12592, pp. 227-232. SPIE, 2023. (Chapter 3)

2. **Sartaj Ahmed Salman**Salman, Ali Zakir, and Hiroki Takahashi. "Adopting ConvNeXt and contextual representation for enhanced feature integration in compact CPM for 2D hand pose estimation". In International Workshop on Advanced Imaging Technology (IWAIT) 2024, vol. 13164, pp. 367-372. SPIE. (Chapter 4)

3. **Sartaj Ahmed Salman**, Ali Zakir, Gibran Benitez-Garcia, and Hiroki Takahashi. "ACENet: Attention-Driven Contextual Features-Enhanced Lightweight EfficientNet for 2D Hand Pose Estimation." In 2023 38th International Conference on Image and Vision Computing New Zealand (IVCNZ), pp. 1-6. IEEE, 2023. (Chapter 5)

4. **Sartaj Ahmed Salman**, Ali Zakir, and Hiroki Takahashi. "Deformable Pose Network: A Multi-Stage Deformable Convolutional Network for 2D Hand Pose Estimation." In Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: VISAPP, 814-821. ISBN 978-989-758-679-8. ISSN 2184-4321. (2024). (Chapter 7)

Not Related Publications

Journal

1. Zakir, Ali, **Sartaj Ahmed Salman**, and Hiroki Takahashi. "SOCA-PRNet: Spatially Oriented Attention-Infused Structured-Feature-Enabled PoseResNet for 2D Human Pose Estimation." Sensors 24, no. 1 (2023): 110.

International Conference

1. Zakir, Ali, **Sartaj Ahmed Salman**, and Hiroki Takahashi. "SAHF-LightPoseResNet: Spatially-Aware Attention-Based Hierarchical Features Enabled Lightweight PoseResNet for 2D Human Pose Estimation." In International Conference on Parallel and Distributed Computing: Applications and Technologies, pp. 43-54. Singapore: Springer Nature Singapore, 2023.

2. Zakir, Ali, **Sartaj Ahmed Salman**, Gibran Benitez-Garcia, and Hiroki Takahashi. "AECA-PRNetCC: Adaptive Efficient Channel Attention-based PoseResNet for Coordinate Classification in 2D Human Pose." In 2023 38th International Conference on Image and Vision Computing New Zealand (IVCNZ), pp. 1-6. IEEE, 2023.

3. Zakir, Ali, **Sartaj Ahmed Salman**, Gibran Benitez-Garcia, and Hiroki Takahashi. "EBA-PRNetCC: An Efficient Bridge Attention-Integration PoseResNet for Coordinate Classification in 2D Human Pose Estimation." In Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: VISAPP, 133-144. ISBN 978-989-758-679-8. ISSN 2184-4321. (2024)

4. **Sartaj Ahmed Salman**, Lian, Z., Saleem, M., and Zhang, Y. (2020, December). Functional connectivity based classification of adhd using different atlases. In 2020 IEEE International Conference on Progress in Informatics and Computing (PIC) (pp. 62-66). IEEE.

5. Saleem, M., Lian, Z., **Sartaj Ahmed Salman**, Tabassum, I., Badar, L. T., and Zakir, A. (2020, December). Strength and similarity guided gsr based network to diagnose adhd. In 2020 IEEE International Conference on Progress in Informatics and Computing (PIC) (pp. 185-189). IEEE.