electronics

Article

Real-Time Pose Estimation Based on ResNet-50 for Rapid Safety Prevention and Accident Detection for Field Workers

Jieun Lee [1], Tae-yong Kim [2], Seunghyo Beak [3], Yeeun Moon [4] and Jongpil Jeong [1,*]

Department of Smart Factory Convergence, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon 16419, Republic of Korea; lu3873@g.skku.edu (J.L.); skywin94@naver.com (T.-y.K.); bjh1205@g.skku.edu (S.B.); mye1113@skku.edu (Y.M.)

* Correspondence: jpjeong@skku.edu; Tel.: +82-10-9700-6284 or +82-31-299-4267

Abstract: The present study proposes a Real-Time Pose Estimation technique using OpenPose based on ResNet-50 that enables rapid safety prevention and accident detection among field workers. Field workers perform tasks in high-risk environments, and accurate Pose Estimation is a crucial aspect of ensuring worker safety. However, it is difficult for Real-Time Pose Estimation to be conducted in such a way as to simultaneously meet Real-Time processing requirements and accuracy in complex environments. To address these issues, the current study uses the OpenPose algorithm based on ResNet-50, which is a neural network architecture that performs well in both image classification and feature extraction tasks, thus providing high accuracy and efficiency. OpenPose is an algorithm specialized for multi-human Pose Estimation that can be used to estimate the body structure and joint positions of a large number of individuals in real time. Here, we train ResNet-50-based OpenPose for Real-Time Pose Estimation and evaluate it on various datasets, including actions performed by real field workers. The experimental results show that the proposed algorithm achieves high accuracy in the Real-Time Pose Estimation of field workers. It also provides stable results while maintaining a fast image processing speed, thus confirming its applicability in real field environments.

Keywords: OpenPose; pose estimation; ResNet-50; computer vision; field workers; rapid safety prevention; accident detection

Citation: Lee, J.; Kim, T.-y.; Beak, S.; Moon, Y.; Jeong, J. Real-Time Pose Estimation Based on ResNet-50 for Rapid Safety Prevention and Accident Detection for Field Workers. *Electronics* **2023**, *12*, 3513. https://doi.org/10.3390/electronics12163513

Academic Editors: Ming Liu and Dah-Jye Lee

Received: 21 June 2023
Revised: 28 July 2023
Accepted: 16 August 2023
Published: 19 August 2023

1. Introduction

In recent years, the field of real-time human pose estimation has rapidly advanced, driven by the introduction of cutting-edge techniques such as deep learning which has significantly improved the field of real-time human pose estimation [1–3]. This technology has direct implications for enhancing the safety management of field workers by serving as a valuable tool for accident prevention and rapid response in the workplace [1,2].

Libraries such as OpenPose, which leverage convolutional Nnural networks (CNNs), provide real-time extraction of features from multiple human poses, hands, feet, and faces, enabling behavioral recognition for ensuring worker safety [4]. Moreover, the adoption of ResNet-50, a powerful deep learning model, has further improved the implementation of real-time pose estimation [5]. By harnessing the capabilities of ResNet-50, we aim to achieve real-time pose estimation for workers, not only improving accident prevention but also fostering an efficient work environment, reducing worker fatigue, and enhancing overall productivity. Continuous advancements in real-time pose estimation technologies are expected to play a crucial role in creating safer and more efficient work environments for field workers.

It is crucial for fieldwork environments to have safety practices that prevent health incidents while also allowing for the quick detection of and response to such incidents. In particular, construction sites and factories pose substantial risks to workers, and on-site accidents can lead to severe injuries or even loss of life. Existing safety prevention and

accident detection systems largely operate in static situations, while using video-based motion detection or sensors to detect accidents which may not effectively capture real-time worker poses and take immediate preventative actions. Moreover, the limited ability to accurately estimate or monitor worker pose in real time means that worker safety may not be guaranteed, accidents may not be responded to quickly when using such methods. To address this limitation, this paper proposes a new real-time pose estimation technique that utilizes OpenPose based on ResNet-50, a 50-layer residual network model [6]. By employing the deep learning and image processing capabilities of ResNet-50, this technique allows for the accurate and real-time estimation of worker poses, facilitating proactive safety measures and accident prevention during work.

Real-time pose estimation with OpenPose based on ResNet-50 offers numerous benefits for improving worker safety and preventing serious accidents. First, it enables swift responses in dynamic work environments, detecting worker pose changes and dangerous behaviors in real time and issuing immediate warnings or safety measures. Second, the superior image processing capabilities of ResNet-50 contribute to precise worker pose estimation, enabling the identification of potential hazards during work in advance. Moreover, OpenPose based on ResNet-50 enhances both worker safety and efficiency by monitoring and analyzing worker poses in real time, providing insights to improve work methods, tool usage, and overall work efficiency. Consequently, this paper proposes a real-time pose estimation technique using OpenPose based on ResNet-50 to support prompt safety prevention and accident detection for field workers and minimize risks in the work environment.

The idea for this study was born out of efforts to address worker safety in the field. The safety of workers working in the field should always be the top priority, and accident prevention and rapid response to accidents are very important aspects of this goal. The idea of developing a safety prevention and accident detection system based on real-time pose estimation using ResNet-50 was then derived with this in mind.

The idea generation process in this research is as follows: First, safety problems that may occur in the field were identified. Considering that workers working in the field are exposed to various risk factors, we explored various ways to address these risks. In this process, we identified the need for real-time pose estimation. Second, we reviewed existing research and technology. Real-time pose estimation technology based on deep learning is already being used in various fields, so we explored how it could be applied in the field. Third, we evaluated the applicability of ResNet-50. ResNet-50 is known to be a model with excellent image classification performance. We investigated whether it can be used for real-time pose estimation, and in this process, we also confirmed the possibility of developing a real-time pose estimation system based on ResNet-50. Fourth, to verify the practicality of the research, the performance of the proposed model was compared with those of various other models. Through this process, we were able to objectively evaluate the performance of the theory and derive improved performance for human pose estimation based on this theory.

Workers in the field perform different types of tasks in various work environments, and these tasks involve risks of safety accidents that can cause serious injury or loss of life [7]. To avoid these risks and prevent accidents in advance, there is a need for a system that can estimate and analyze worker poses in real time. In this paper, we propose a system for worker safety prevention and accident detection by combining the OpenPose algorithm and ResNet-50 to accurately estimate worker poses. This system can quickly detect the current state of a worker and the hazardous situation in the work environment. The contributions of this paper are as follows:

- Real-time pose estimation: This paper proposes a method that allows for the pose of a worker to be estimated in real time. This is an important contribution to estimating the pose of workers in the field and quickly detecting dangerous situations.
- Accuracy: The method proposed in this paper shows high accuracy in pose estimation through a combination of OpenPose algorithm and ResNet-50. This aids the real-time detection of a worker's status by accurately estimating a worker's pose.

- Safety prevention and accident detection: The method proposed in this paper monitors the pose of workers in real time and also detects safety hazards that may occur in the work environment. This helps workers perform their tasks in a safer environment while also enabling rapid response and rescue in the case of accidents.
- Building an AI-based safety system: The proposed system can be utilized to enhance safety in an enterprise or field. By analyzing and evaluating the posture and behavior of workers in real time, the posture estimation technology based on AI can contribute to building a more systematic and efficient safety management system.
- Efficient resource utilization: In this study, ResNet-50 was applied to enable real-time pose estimation at a faster speed compared to the existing OpenPose. This improved speed and accuracy can be seen as an important contribution to resource efficiency. Especially when estimating the pose of many workers simultaneously in real-time in a large work environment, the efficiency of the system is crucial.
- Optimization for industrial settings: By using a dataset of workers in an industrial setting to train and evaluate the model, this study yields optimized results for real-world work environments, which is an important contribution that increases its applicability in real-world settings.
- Future applications: The techniques in this study can serve as a basis for safety and accident prevention for field workers, and future applications can be actively researched. For example, it could be applied to posture estimation and accident prevention systems not only in industrial sites but also in various fields such as healthcare, sports, and security.

Ultimately, the objective of this thesis is to propose a system that quickly estimates the pose of a worker for safety prevention and accident detection based on real-Time pose estimation. Therefore, to achieve this goal, the proposed system is designed to achieve high accuracy and a fast real-time processing speed. This is expected to be an important research work that can ensure the safety of field workers and contribute to accident prevention.

In the present work, we conducted a study on "Real-Time Pose Estimation technique based on ResNet-50". The rest of this paper is organized as follows: Section 2, Related Work, discusses Open-Pose, multi-person pose estimation, and ResNet; Section 3, ResNet-50-based real-time pose estimation, describes the overall structure, feature extraction stages, and operation flow; Section 4, Experiment and Result, describes the experimental environments, performance metrics, and results; and Section 5, Conclusion, concludes the paper by describing the results.

2. Related Work

2.1. OpenPose

OpenPose is a library maintained by Carnegie Mellon University that enables the real-time extraction of the location of the body, hand, face, and other points from the motion of multiple people in a photo or video [8]. The software uses computer vision and machine learning to learn information about individual people as it processes video frames. This makes it possible to track multiple people simultaneously. In contrast to the traditional Haar algorithm, OpenPose allows one to analyze motion from a variety of angles and orientations without being limited to frontal images [9].

OpenPose is a human pose estimation technique that locates a person's face, body, and joints such as those of the hands and feet. Previous pose estimation methods used a top-down approach to detect people and then estimate their pose. By contrast, OpenPose has evolved into a state-of-the-art method that first detects the body parts of all people in the image (nose, left elbow, etc.) and then assembles joints based on connectable pairs between body parts in a bottom-up approach to ultimately estimate their pose. OpenPose uses a pre-trained neural network to predict a Part Affinity Field (PAF), which is a two-dimensional coordinate system consisting of a heat map of body parts, their locations, and orientation information in an input image. The number of joints has previously been

limited to 15, but the algorithm is currently being refined to extract up to as many as 18 and even 25 joints [10].

In this way, OpenPose can serve as a useful method for real-time 2D pose estimation of multiple people.

2.2. Multi-Person Pose Estimation

Existing multi-person estimation methods can be divided into three categories: top-down, bottom-up, and the more recent single-step methods [11].

2.2.1. Top-Down Method

Top-down methods first use an object detector to obtain the bounding box of a person object in the image. Individual person instances are then cut out of the bounding box to perform first-person pose estimation. There have been various important works examining this method, such as Hourglass, RMPE, CPN, SimpleBaseline, and HRNet. In general, top-down methods are slower to infer. This method divides the multi-person estimation task into two steps: person detection and single-person pose estimation. Rather than cropping the region of interest (RoI) in the original image, Mask RCNN uses the RoI alignment operation to extract the features of the RoI from the detector's feature map, which substantially improves the inference speed. Top-down methods also rely heavily on the detector performance [11].

2.2.2. Bottom-Up Method

Bottom-up methods are instance-agnostic, as they detect all keypoints and then group them into individual keypoints. Most existing bottom-up methods typically focus on how to connect detected keypoints to keypoints that belonging to the same person. However, OpenPose uses PAFs to associate keypoints from the same instance. Next, associative embedding creates a detection heat map and tagging map for each body joint, grouping keypoints with similar tags into a single person. PersonLab groups keypoints by manually learning a 2D offset field for each keypoint pair. Compared to PifPaf's top-down approach to grouping keypoints into full-body poses, the bottom-up approach is generally more efficient, because it has a simpler pipeline that shares convolutional operations [11]. However, the process after grouping is heuristic and involves many tricks, so it often performs worse than top-down methods [12].

2.2.3. Single-Stage Methods

To avoid the limitations of the above top-down and bottom-up methods, we propose a single-step method that uses a dense regression of a set of pose candidates on the spatial locations where each candidate consists of keypoint locations of the same person. The single-step method adopts a parallel processing mode to perform multi-scale noise filtering through a single traverse according to the information about each point in the point cloud [13]. SPM proposes a structured pose representation to incorporate the positional information of human instances and body joints. CenterNet proposes that the regressed keypoint locations should be matched with the closest keypoints detected in the keypoint heatmap due to the weak regression results. Point Set Anchors uses a deformable convolution to improve predefined pose anchors to mitigate the difficulty of feature misalignment. FCPose and InsPose use dynamic instance-aware convolution to solve the problem of multi-person estimation, thereby achieving a better trade-off between accuracy and efficiency than other single-step methods. While these approaches can perform competitively, they are not fully end-to-end optimized and still require heuristic post-processing such as NMS or keypoint localization [11].

Among these various multi-person estimation algorithms, we chose OpenPose for the following reasons: One of the objectives of this research is real-time pose estimation. OpenPose is known to be an algorithm that provides highly accurate real-time estimation [14]. Therefore, we chose OpenPose because it can estimate the pose of a field worker

in real time and provide stable results. Moreover, OpenPose is already a widely used and validated algorithm in a wide variety of research and application areas [15]. Finally, as mentioned in the title of this paper, we used ResNet-50 and integrated it with OpenPose for real-time pose estimation. OpenPose is a flexible algorithm that can be used with various neural network architectures, while ResNet-50 is a neural network architecture that achieves excellent performance in image classification and feature extraction tasks [16]. Therefore, by combining the excellent feature extraction capabilities of ResNet-50 with the multi-human pose estimation capabilities of the OpenPose algorithm, we were able to perform multi-human pose estimation in real time.

2.3. ResNet

ResNet, which stands for Residual Network, is a type of neural network that can be considered to be the backbone of many computer vision tasks. Compared to other neural networks, ResNet models can train networks with up to 150 layers. It is not easy to train a CNN because the gradients vanish; vanishing gradients is a problem that causes gradient backward propagation, while the value of the gradient factor can eventually become extremely small after repeated multiplication. The ResNet model is one of the best models for solving this problem. There are different types of ResNet models depending on the number of layers. These are the starting point for transfer learning. Therefore, the deeper the network model goes into the layers, the worse the model performs. It is not easy to increase the network depth simply by stacking layers individually [17].

Skip connections are also known as shortcut connections. As the name suggests, they can be used to skip certain layers of a neural network and feed the output of one layer as input to the next. Skip connections are used to solve problems depending on the type of model. In a ResNet, skip connections are used to solve the degradation problem, while this technique is used for feature extraction in DenseNets. ResNet models are designed to solve image classification problems. They operate under the concept of matrix addition, where data from an earlier layer are passed to a deeper layer. This procedure does not support any other additional parameters, as the results of the previous layer are added to the next layer [17].

There are several types of CNNs based on residual networks. Based on the layer model, they can be categorized into different models, such as ResNet- 18, ResNet-34, ResNet-50, ResNet-101, ResNet-152, ResNet-164, and ResNet-202. Shortcut connections make it easy to convert simple networks into residual networks. A straight network can be the result of a VGG-16 model. ResNets have fewer filters and less complexity. ResNet-34 achieves performance that is nearly twice that of other CNNs. The ResNet-50 architecture consists of five layers. Each ResNet-50 layer consists of different convolution and ID blocks. Each convolution block also contains three convolution layers and one ID block [17].

The ResNet-18 model is an 18-layer deep CNN that is derived by training over one million images from the dataset [18]. ResNet-34 is a 34-layer CNN-based model that is also a pre-trained image classification model [19]. The ResNet-34 model is trained on a dataset of 100,000+ images across 200+ classes. It is superior to traditional networks in that the residuals from each layer can be input and further reused by subsequent connected layers. The ResNet-34 model uses two implementation rules. For similar types of feature images, several filters are doubled if the number of similar filters and the size of the feature map are half their original number and size, respectively. This is undertaken to preserve the time complexity of each layer. In total, there are 34 weighted layers in the ResNet-34 model [17]. ResNet-50 is a CNN model proposed to solve the gradient vanishing and gradient explosion problems [20]. The ResNet-50 model architecture consists of five layers, each with a convolution block and an ID block [21]. The first part of the input preprocessing contains the convolutional layer, and the second through fifth parts are all composed of bottleneck building blocks [22]. Each convolution block has three convolution layers and one ID block. There are millions of trainable parametric elements in a ResNet-50 model. There are over 20 million parameters that can be used to train a ResNet-50 model. The

difference between the two models is the design of the building blocks. Depending on the time required to train the layers, these parameters were modified with the intention to hold for time reasons. Compared to the two-layer stack of ResNet-34, a three-layer stack is used in ResNet-50. Because of this advantage, the accuracy of ResNet-50 is higher than that of ResNet-34 [17,18]. ResNet-101 is a 101-layer CNN [23]. The model has been trained on more than 1 million images and can also classify different types of images into 1000 object categories. For better resolution, 224 × 224 images are used [17]. ResNet-101 can be built with three or more blocks compared to ResNet-50. ResNet-152, which is also known as a CNN, is a deeper model with 152 layers [24]. It again introduces the concept of skip connections, connecting the next layer to the previous layer without input variations. This model is a type of artificial neural network that is composed of pyramidal cells. ResNet-164 is a type of CNN with 164 layers, and this model is also trained using more than 100,000 images along with other class objects [17,25]. All models in the ResNet have flexible layers, meaning they can be used to efficiently solve a variety of complex problems. All the different nomenclature used in ResNet shows that it can work on different layers. For example, the naming of ResNet-50 indicates that it can operate on 50 different neural network layers. In this ResNet model, shortcuts connections are added to an otherwise simple network. The uniqueness shortcut was used directly because the dimensions of the inputs and outputs are similar to each other. As the dimensionality increases, there are two options for inclusion. The first option is to use additional published data for dimensional expansion so that one can perform an identity mapping with the shortcut. The other field data option is to use projection shortcuts for dimensional matching [17].

Out of all these different ResNet models, we chose ResNet-50 because of its efficiency, time, and generalization. ResNet- 50 has the appropriate depth and capacity to perform well in many computer vision tasks [6,26]. ResNet-50 has more trainable parameters than the smaller model, ResNet-18, which therefore allows for richer representation learning [6,27]. However, it is lighter than the deeper ResNet-101 or ResNet-152 models, which are more efficient in terms of memory and computational resources [28]. We chose ResNet-50 because deeper or larger ResNet models require more computational resources and memory, which can be time consuming, and because the initial models are likely to be less accurate [6]. ResNet-50 is also widely used as a pre-trained model on large datasets such as ImageNet [29]. This means that ResNet-50 has learned common features from a variety of image data and can be transferred to other tasks [29]. Therefore, we chose ResNet-50 because we can expect high performance for the pose estimation task by exploiting the generalizability of pre-trained models [30].

We also used ResNet-50 instead of VGG19 for the following reasons: VGG19 and ResNet-50 are both models that are often used in common computer vision tasks. However, ResNet- 50 can sometimes outperform VGG19. ResNet-50 is composed of deep neural networks and can perform well when used with deeper networks because it can solve the gradient vanishing problem by introducing residual connections [6,29,31]. Therefore, we chose ResNet-50 instead of VGG19 because ResNet-50 is expected to achieve higher performance. ResNet-50 is also a lighter model than VGG19. Although ResNet-50 has 50 layers, it has the advantage of having a relatively small number of parameters. This can reduce computational resources and memory usage, which is beneficial for resource-constrained tasks such as real-time pose estimation. Therefore, ResNet-50 provides greater speed and efficiency when performing real-time tasks such as real-time pose estimation. Finally, ResNet-50 is compatible with OpenPose [14]. OpenPose is a comprehensive solution for multi-person pose estimation, and we chose ResNet-50 for our study because it allows us to easily take advantage of the various features of OpenPose for pose estimation tasks.

3. ResNet-50 Based Real-Time Pose Estimation

3.1. Overall Structure

Pose estimation models estimate human poses based on input images, and such estimation requires accurate and diverse human image datasets. The COCO dataset [32] and

the MPII dataset [33] are representative datasets that fulfill this requirement, thus providing images of people taken in various environments and common information extracted from these images. By training a model on these datasets, the model can learn to accurately estimate human poses in a variety of situations. Figure 1 shows the overall pipeline. The overall pipeline structure consists of seven steps: Input Image, Feature Extraction, Convolutional Layer, Part Confidence Maps, Part Affinity Fields, Bipartite Graph Matching, and Results.

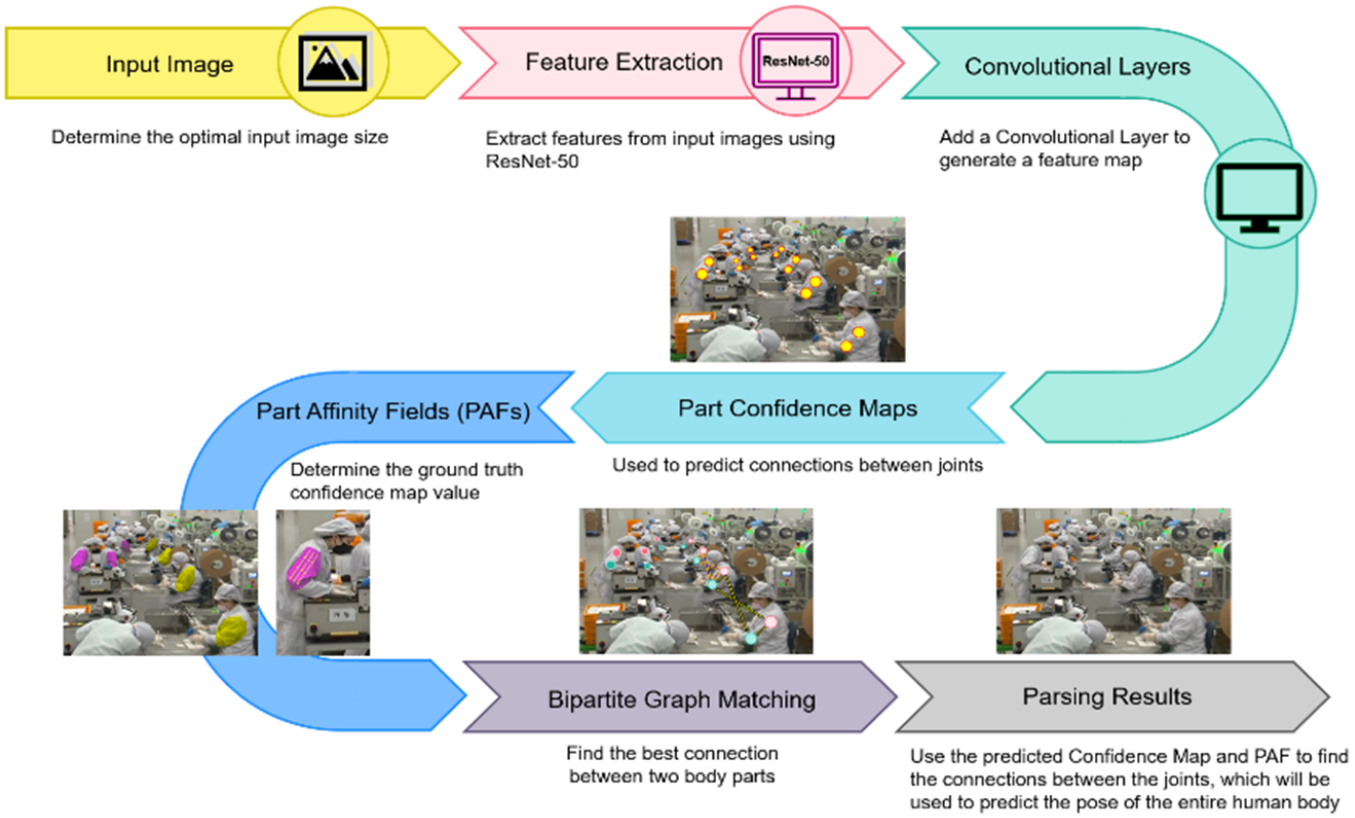

Figure 1. Overall pipeline.

The first step is Input Image. This step utilizes human images and joint information collected from the COCO dataset [32] and MPII dataset [33] to train the model. Based on the knowledge learned from these datasets, the model will have the ability to predict the position of the human joints in the input image. However, the input images can be of varying sizes and resolutions. This affects the performance and accuracy of the model. It is difficult to accurately predict joint positions in small images, and processing time can be long in large images. Therefore, we perform an experiment to determine the optimal input image size. The experiment consists of evaluating the performance of the model by resizing the image with different input image sizes. Based on the experimental results, we determine the optimal input image size to be 256 $\times$ 256 and resize the images for all COCO [32] and MPII datasets [33] to that size. Once the optimal input image size is determined, the pose estimation model is trained on that size. The Input Image step also includes preprocessing and is a very important step that determines the performance and accuracy of the model.

The second step is Feature Extraction. In this step, ResNet-50 is used to extract features from the input image. ResNet-50 consists of a deep neural network that can process the input data at multiple layers to extract more abstract features. The extracted features are then passed to the convolutional layer to generate a feature map that is extracted from the input image. The convolutional layer is one of the components of a deep learning network that applies convolutional operations to the input data to generate a feature map. Convolutional layers are effective for learning spatial features in image processing, and

they are mainly used in CNNs. At this point, the feature map extracted from the input image can effectively represent the features of the input image, thus enabling the model to more accurately recognize the features of the target in the input image. This extracted feature map significantly affects the performance of the model because the feature map must correctly preserve the information in the input image. Therefore, it is important to determine the optimal ResNet-50 configuration and parameter settings during the feature map stage, as failing to extract important information from the input image may result in poor model performance.

After generating the Feature Map, the third step requires the addition of a convolutional layer to generate the Part Affinity Fields and Part Confidence Maps. The PAFs and Part Confidence Maps are used to predict the connection information between each body part and joint. The added convolutional layer extracts the PAFs and Part Confidence Maps from the input Feature Map. To elaborate, a PAF is a vector connecting each joint that is used to predict the connection information between each joint, while the Part Confidence Map is a heat map representing the joint position in the input image, which is used to accurately predict the position of each joint. A heat map is a graphical representation of the intensity or density of a particular object or feature in an input image. Such maps are mainly used in computer vision to visualize the location or probability distribution of objects or features of interest, and they mainly use color to represent information, where the higher the intensity of the object or feature, the brighter the color. PAFs and Part Confidence Maps are important factors that substantially affect the performance of the model and require optimal configuration of the convolutional layer and the parameter settings of the convolutional layer. Therefore, research and experiments should be conducted in this step to optimize the convolutional layer required to generate PAFs and Part Confidence Maps.

Part Confidence Maps, which are heat map representations of the location of each body part, are used by a model trained on an artificial neural network to locate a specific body part in an input image. In this case, a Part Confidence Map contains information about the location of a specific body part per map, while the value of a pixel in the map indicates the confidence level of that location. For example, if a location in a Part Confidence Map has a high pixel value, the location is more likely to be a specific human body part. Each Part Confidence Map takes a color image as input and uses distributions to predict joint locations instead of x and y coordinate values. By using distributions to predict joint locations, more accurate predictions can be made by averaging the predicted values from multiple locations rather than predicting a location for a specific coordinate. This approach is known to be more accurate than typical prediction methods. The model also must account for multi-person cases, so each Part Confidence Map can have multiple values, thus allowing it to predict the joint positions of multiple people in the input image. In this step, a Part Confidence Maps is generated for each body part, which is later used to predict the actual joint position.

In this step, a Part Confidence Map representing the position of each body part is generated, followed by PAFs representing the connection information between two joints. PAFs play a very important role in human pose estimation. In this step, the connection information between each joint is extracted; this will later be used to connect the joints. PAFs are maps of the connecting lines between two joints, with each layer predicting a different pair of joints. Each layer considers the pairs of joints it connects and uses different information to extract the connection information between the joints. This extracts the connection information between joints from the input image, and this connection information allows for the accurate detection of each joint. PAFs, like Part Confidence Maps, are generated using the convolutional layer, and they utilize a variety of information to extract the connection information between joints from the input image. For example, the Part Confidence Map generated in the previous step is used to predict the position of each joint, and the connection information between the joints is extracted based on this.

This Bipartite Graph Matching step is a very important step in human pose estimation. In this process, we use PAFs and Part Confidence Maps to find the best connections between

the generated limbs. To find the best connections, we use Maximum-Weight Bipartite Graph Matching, which simultaneously considers the connectivity between all body parts to identify the best connections. This results in the detection of a very large number of limbs, which can be called a K-Partite Graph. The purpose of pose estimation is to find the best connections in this K-Partite Graph. Once the best connections are found, pose estimation is possible for each limb, and the overall human body pose can be estimated. In the Bipartite Graph Matching step, PAFs and Part Confidence Maps are used to generate limbs and find the best connections. To achieve this, we use the Maximum-Weight Bipartite Graph Matching algorithm to find the best connections. This is a very important step in estimating the results of pose estimation.

In the final step of pose estimation, the results step synthesizes all the information generated in the previous steps to estimate the human pose. It synthesizes the results of the Part Confidence Maps, PAFs, and Bipartite Graph Matching to generate both the position of each joint and the connections between them. Here, it is important to connect each limb to complete the human skeleton, which predicts the position of each joint in the input image and uses it to generate the connection information between each joint. This skeleton is then used to estimate the pose of the person.

In summary, Figure 1 shows the process of predicting human poses by estimating joint positions and connections between joints from input images. For this purpose, ResNet-50 is used to extract features from the input image, while a convolutional layer is added to generate PAFs and Part Confidence Maps. Then, the Part Confidence Map is generated, and Bipartite Graph Matching is conducted to find the optimal connection to complete the human skeleton. After completing these steps, we can estimate the human pose from the input image.

3.2. Feature Extraction Stages

Overall, the ResNet-50 network consists of an input layer, five stages, and an output layer. Each stage is composed of several residual blocks of different sizes and widths, as shown in Figure 2. We can see in the figure that each stage is composed of several residual blocks of different sizes and widths. Residual refers to the difference or error between the input and output in a neural network, while the residual block is an important component of the ResNet (Residual Neural Network) architecture. Residual blocks are introduced to mitigate the vanishing gradient problem that occurs with the increasing depth of the network. The vanishing gradient problem refers to the phenomenon wherein, during the training of a deep learning model, the gradient value becomes progressively smaller as the gradient is calculated using the back-propagation algorithm. As the gradient value becomes smaller in the lower layers of the neural network, the weights of those layers may either not be updated or become unstable during backpropagation to the higher layers, thus causing the model to fail to converge and learning to slow down. This residual block solves these problems by introducing a skip connection between the input and output. The skip connection works by adding the input directly to the output. As a result, the output of the residual block is the result of adding the residual to the input. This can be expressed as a formula as follows:

$$y = F(x) + x$$

where x is the input and F(x) is the output after passing through the residual block. This structure allows the residual block to pass the input as it is and to learn with only the error or modified information. As mentioned previously, each stage in ResNet-50 consists of multiple residual blocks. A residual block consists of a series of convolutional layers, batch normalization, and activation functions, and it must be designed so that the inputs and outputs have the same dimensionality. Therefore, if necessary, a dimensionality-matching transformation can be performed using a 1×1 convolution. ResNet-50 uses these residual blocks to construct a network, starting from the input layer and continuing to connect the output layer through five stages; through this structure, a deep network can be built as

shown in Figure 2, which has better performance than other previous network structures. In a typical deep learning model, when passing the input data through multiple layers to the output layer, the information may be continuously changed and lost; by contrast, in the shortcut connection, the input data are directly connected to the output layer even in the middle layer of the network, which prevents loss of information on the input data, which plays a major role in determining the performance of the model.

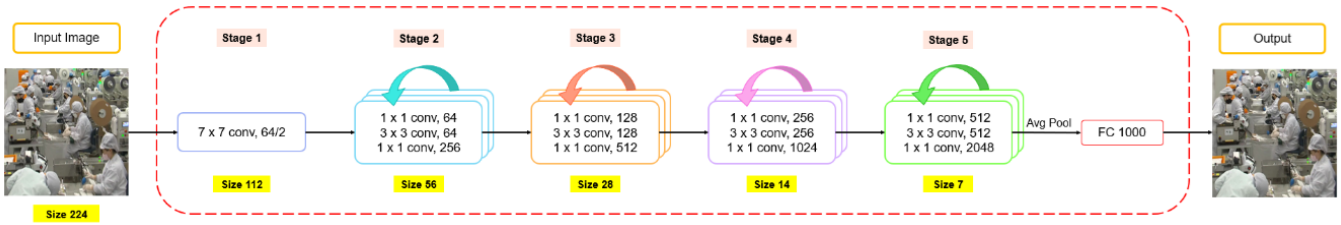

Figure 2. Stage of feature extraction.

As shown in Figure 2, each stage builds several residual blocks in sequence, where the input values are taken as they are through shortcut connections and where only the remaining residual information—F(x)—is added. This makes it easier for the neural network to learn and build a deeper network.

Stage 1 uses a 7 × 7 convolutional filter to extract a feature map from the Input Image, and this stage uses 64 convolutional filters, each of which performs a convolutional operation on the input image. The stride value is set to 2, which is the interval at which the convolution filters are applied to the input image, while skipping without overlapping. Here, stride refers to the interval at which the filter (kernel) is applied, i.e., how far the filter moves over the input image. If the stride is 2, then the filter moves by 2 pixels, and the size of the output Feature Map is determined by dividing the size of the input image by the stride value. In this case, since the stride is set to 2 in Stage 1, the size of the input image is halved. Therefore, here, we halve the size of the input image and generate 64 feature maps.

Stage 2 is a residual block consisting of 1 × 1 conv, 64/3 × 3 conv, 64/1 × 1 conv, 256. In this stage, the input image is downsampled by 1/4 and 256 feature maps are generated. Here, downsampling refers to reducing the spatial resolution of the input data, i.e., reducing the size of the input image to reduce the size of the output. This allows for the use of larger filters in the next stage and also helps improve the performance of the model by allowing it to process more information. The first layer in this section, a 1 × 1 conv layer, expands the input feature map from 64 channels to 256 channels. A 3 × 3 conv layer is then used to keep the number of channels in the input feature map at 64. Finally, a second 1 × 1 conv layer expands the input Feature Map from 64 channels to 256 channels. After all of these operations, the input image is downsampled by a quarter, ultimately resulting in 256 feature maps. The residual block is also repeated in this section. The residual block is structured with a shortcut connection between the input and output values, while the shortcut connection works in such a way that the input value is added to the output value immediately after passing through the layer.

Stage 3 is responsible for reducing the input size by 1/8. To achieve this, a 1 × 1 convolution is first used to reduce the number of channels in the input data to 128. Next, we use a 3 × 3 convolution to process the feature map and reduce the number of output channels back to 128. We then use a 1 × 1 convolution to increase the number of output channels to 512. This reduces the size of the input data by 1/8, ultimately resulting in 128 feature maps in total. Stage 3 reduces the size of the input data while increasing the number of feature maps. This allows the model to extract more complex and diverse features from the input image.

Stage 4 works on a 14 × 14 feature map. In this step, there are three residual blocks, each consisting of the following: First, a 1 × 1 Convolution operation is performed to reduce the number of channels in the input Feature Map. Next, a 3 × 3 convolution operation

is performed, and a 1×1 convolution operation is then performed again to increase the number of channels. The output Feature Map is then added to the input feature map to obtain the final output. This structure is called a Residual Connection. Through this process, different features are extracted from the input feature map to ultimately obtain a more complex feature map. The Feature Map generated here in Stage 4 is 14×14 in size and has 1024 channels in total. The 14×14 feature map generated in Stage 4 is converted to a 7×7 feature map by downsampling. In this process, the size of the feature map is halved with a stride of 2×2 steps using two convolution layers and Max Pooling. The resulting 2048-channel Feature Map is then used to calculate the average value for each channel using Average Pooling to create a one-dimensional vector. The purpose of pooling is to reduce the number of optimization parameters. Depending on the purpose, pooling can be divided into two forms: Max Pooling and Average Pooling. Max Pooling emphasizes features by selecting the largest value within a certain size. By contrast, Average Pooling calculates and uses the average of the values within a certain size.

Average Pooling is a technique used in CNNs to downsample a feature map. It involves averaging the values in each rectangular block of the feature map to output a new, smaller feature map. Each rectangular block is typically 2×2 or 3×3 in size and is represented as a grid-like division of the feature map to perform Average Pooling. This grid is set to a constant size, and each rectangular block contains the values from that area. In other words, Average Pooling is a technique used to generate a new, smaller feature map by calculating the average of the values in each rectangular block. Meanwhile, Average Pooling includes less important factors than Max Pooling, while still allowing for the use of variance and mean to more easily determine object location. After passing through the convolution layer in Stage 5, the Feature Map is down-sampled using Average Pooling and fed into the FC Layer with 1000 neurons. The FC layer is a type of fully-connected neural network layer, meaning that every neuron in the layer is connected to every neuron in the previous layer. The output of the FC layer is a vector with a length of 1000, which represents the predicted class probability for the input image, where each layer is downsampled and batch normalized before finally passing through the average pool layer and the FC layer to ultimately produce the final prediction. In the feature extraction part, we use downsampling and pooling to obtain a feature map with important features extracted from the input image. This helps in the next step, which is to accurately estimate the pose of the field worker and detect hazardous situations.

3.3. Operation Flow

The operation flow of this method consists of Real-Time Video Detection, Feature Extraction, Convolutional Layer, Part Confidence Maps, PAFs, Bipartite Graph Matching, and Results (Real-Time Field Worker's Pose Estimation), as shown in Figure 3. Moreover, if the pose is not detected, it returns to the Real-Time Video Detection step and starts over.

The first stage is Real-Time Video Detection, which detects the worker's behavior in real time. It recognizes the worker by receiving real-time video input from a camera or video. After recognizing the worker, it proceeds to the feature extraction stage to extract features from the input video or camera. The ResNet-50-based neural network architecture is then used to generate a feature map from the input data. This feature map contains important information obtained from the input data, which can accurately estimate the worker's pose and help improve worker safety and prevent safety incidents. Therefore, this step can be thought of as separating the worker's actions and extracting features for each action.

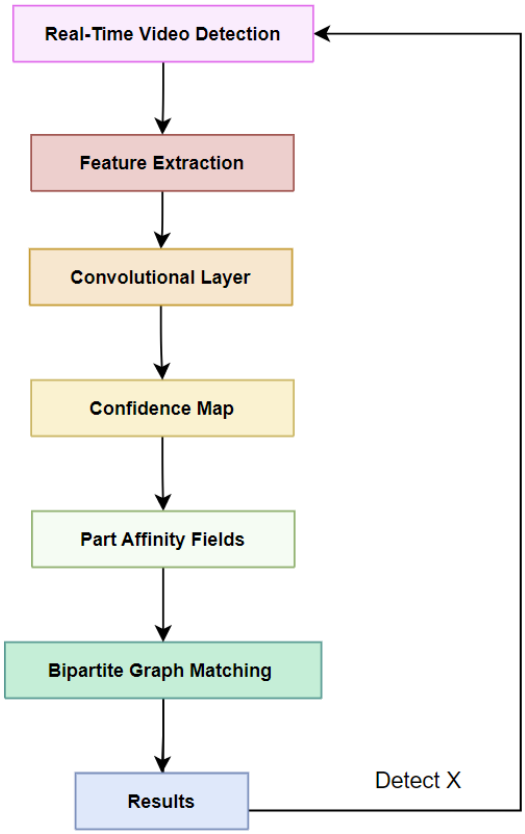

Figure 3. Operation flow.

The next step involves performing a convolution operation on the feature map created in the Feature Extraction step. In this process, we extract various features of the worker's behavior and apply filters to obtain higher-level abstracted features. In other words, this step involves obtaining a good understanding of the worker's behavior and extracting detailed features of the behavior. With the filtered dataset, the Part Confidence Map shows the confidence values for each part of the worker's behavior. This Part Confidence Map provides information about the presence and accuracy of each part of the worker's behavior. After the Part Confidence Maps step, the next step is the PAF step. The PAFs contain information about the connections between the parts of the worker to estimate the pose of the worker pose by connecting the parts of the worker to each other. The Part Confidence Maps and PAFs are used to find the best connections. The Part Confidence Map estimates the position of each part, while the PCFs provide information about the connections between parts. With information, the Bipartite Graph Matching algorithm can determine the best connections by taking the location and connectivity of each part into account. As a result, the Bipartite Graph Matching step estimates the worker's pose by exactly matching each part. The algorithm uses Part Confidence Maps and PCFs to find the best connections, and it accurately estimates the location and connectivity of each part of the worker pose. The Results step provides Real-Time, on-the-spot worker pose estimations. This stage builds on the work conducted in the previous Bipartite Graph Matching step to produce an accurate estimate of the worker pose. The estimated worker pose information is provided in real time and can be used to assess the safety status of the worker and detect hazardous situations. In this way, this technology can monitor the behavior of field workers in real time and accurately estimate their pose to aid in safety prevention and accident detection. Moreover, the pose estimation results can be visualized and provided as needed, including the location of each part of the worker, connection information, and details of the worker's motion, so the worker's motion can be verified in real time and necessary actions can be

taken. If the worker's pose is not accurately estimated, the system returns to the beginning of the Real-Time Video Detection stage to detect the worker's motion and returns to the first step to detect the worker's motion.

3.4. Scenarios

The real-time posture estimation technique based on ResNet-50 proposed in this paper aims to enhance safety measures and accident prevention for field workers in various high-risk scenarios. How the technology proposed in the paper is effective in these risky situations and the specific scenarios in which field workers face potential risks can be seen as follows.

3.4.1. Construction Sites

Construction sites are known for their inherent risks due to tasks involving working at heights, moving heavy objects, and operating machinery. Workers are exposed to potential dangers such as falls from elevated locations and injuries from malfunctioning equipment. Our real-time pose estimation system, utilizing ResNet-50-based OpenPose, enables accurate and rapid pose estimation for field workers. By continuously analyzing the workers' body postures and movements, the system can detect hazardous actions, such as improper ladder usage or incorrect lifting techniques. Through real-time monitoring, it provides timely feedback to both workers and safety personnel, allowing them to correct unsafe behaviors and prevent potential accidents. Additionally, the system maintains a historical record of pose data, facilitating the identification of patterns that contribute to accidents. This data-driven approach enables the implementation of targeted safety training and measures to reduce the occurrence of workplace incidents.

3.4.2. Oil and Gas Industry

The oil and gas industry involves working in environments where hazardous situations such as explosions, fires, and toxic substance leaks pose significant threats to field workers' safety. Workers are at risk of coming into contact with flammable materials or experiencing functional impairments due to exposure to hazardous chemicals. Our real-time pose estimation technology based on ResNet-50 can serve as an essential safety monitoring system in such high-risk environments. By accurately estimating workers' poses in real time, the system can identify unsafe behaviors and improper use of safety equipment. It provides instant alerts to workers and supervisors, guiding them to adhere to safety protocols and maintain correct working postures. Moreover, the system's historical pose data analysis can provide valuable insights into past incidents and identify potential risk factors. This allows for proactive risk management, implementing targeted safety measures to prevent accidents and ensure a safer working environment for all personnel.

3.4.3. Power Supply Industry

The power supply industry presents its workforce with unique risks, including electric shocks, falls from elevated structures, and fire hazards. Field workers in this industry face potential dangers when dealing with high-current devices or working on unsafe electrical installations. Real-time pose estimation system, based on ResNet-50, plays a vital role in mitigating these risks. By continuously monitoring workers' movements, the system can identify potentially unsafe actions, such as improper handling of live wires. Through real-time alerts and notifications, it enables workers and supervisors to take immediate corrective actions, preventing accidents and ensuring a safer work environment. Additionally, the system's historical data analysis allows for the identification of patterns and trends related to accidents and near-miss incidents. This data-driven approach facilitates targeted safety training and the implementation of proactive safety measures, further enhancing the overall safety of field workers in the power supply industry.

In conclusion, our real-time pose estimation technique based on ResNet-50 presents a significant advancement in safety management and accident prevention for field workers

operating in high-risk scenarios. By addressing specific scenarios in construction sites, the oil and gas industry, and the power supply industry, we have demonstrated the versatility and practicality of our proposed technology. The accurate estimation of workers' poses in real-time and continuous analysis of their movements empower our system to identify potential hazards promptly. This real-time monitoring, coupled with historical data analysis, enables the proactive implementation of targeted safety measures, effectively minimizing the occurrence of workplace incidents. These achievements directly align with our research objectives, showcasing the potential impact of our system on worker safety and accident prevention. By providing real-time feedback and alerts to both workers and supervisors, our system promotes safe working behaviors and fosters a proactive safety culture within industrial settings.

4. Results

4.1. Experiment Environments

The available 2D body pose estimation libraries support most pipelines, thus allowing users to provide their own frame readers (e.g., video, image, or camera streaming) and displays to visualize the results, and to generate output files with the results (e.g., JSON or XML files). Moreover, traditional face and body keypoint detectors are not combined, meaning that different libraries are required for each purpose. OpenPose overcomes all of these issues. It runs on a variety of platforms, including Ubuntu v22.04, Windows 11 Home, Mac OS Ventura, and embedded systems (e.g., NVIDIA Tegra TX2). It also supports a wide range of hardware and devices. Users can choose to input from images, video, webcam, and IP camera streaming. They can also choose whether to save the results to disk, enable or disable each detector (body, feet, face, hands), normalize pixel coordinates, control how many GPUs to use and how they are used, skip frames for faster processing, and more [4].

As can be seen in Table 1, the development environment of this study is as follows. OS: Windows 11 Home; CUDA: v11.8; cuDNN: v8.2.4; OpenCV: v4.5.4; Python: v3.9.13; GPU: NVIDIA GeForce RTX 3070Ti Laptop GPU.

Table 1. Development environment.

	OS	CUDA	cMa + ke	cuDNN	OpenCV	Python	Visual Studio	GPU
Development Environment	Windows 11 Home	v11.8	v3.26.3	v8.2.4	v4.5.4	v3.9.13	2019 Enterprise	NVIDIA GeForce RTX 3070Ti Laptop GPU

In this paper, we evaluate our proposed method using three benchmarks. First, the MPII Human Multi-Person dataset, which consists of 3844 training and 1758 test groups and contains images of multiple people interacting with 14 body parts in highly detailed poses [33]. Second, the COCO keypoint challenge dataset is a dataset that requires the detection of 17 keypoints (body parts) for each person [4,32]. Finally, the Field Workers dataset, which we built in this thesis, consists of a subset of 15,000 annotations from the COCO keypoint dataset and contains a variety of scenarios that occur in real-world workplaces. These datasets are used to solve problems in different scenarios and were utilized to evaluate the performance of the proposed method. According to the authors' previous research, these benchmarks have shown different image processing speeds and accuracy [4,32]. In addition, these datasets collect images covering a variety of real-world problems and include different scenarios such as crowding, scale variation, occlusion, and contact. We have experimentally demonstrated that the method proposed in the paper works effectively in real-world applications, and these benchmarks demonstrate the results compared to the existing OpenPose, AlphaPose, and DensePose as mentioned in the paper. In addition, Figure 4 shows the qualitative results of the algorithm.

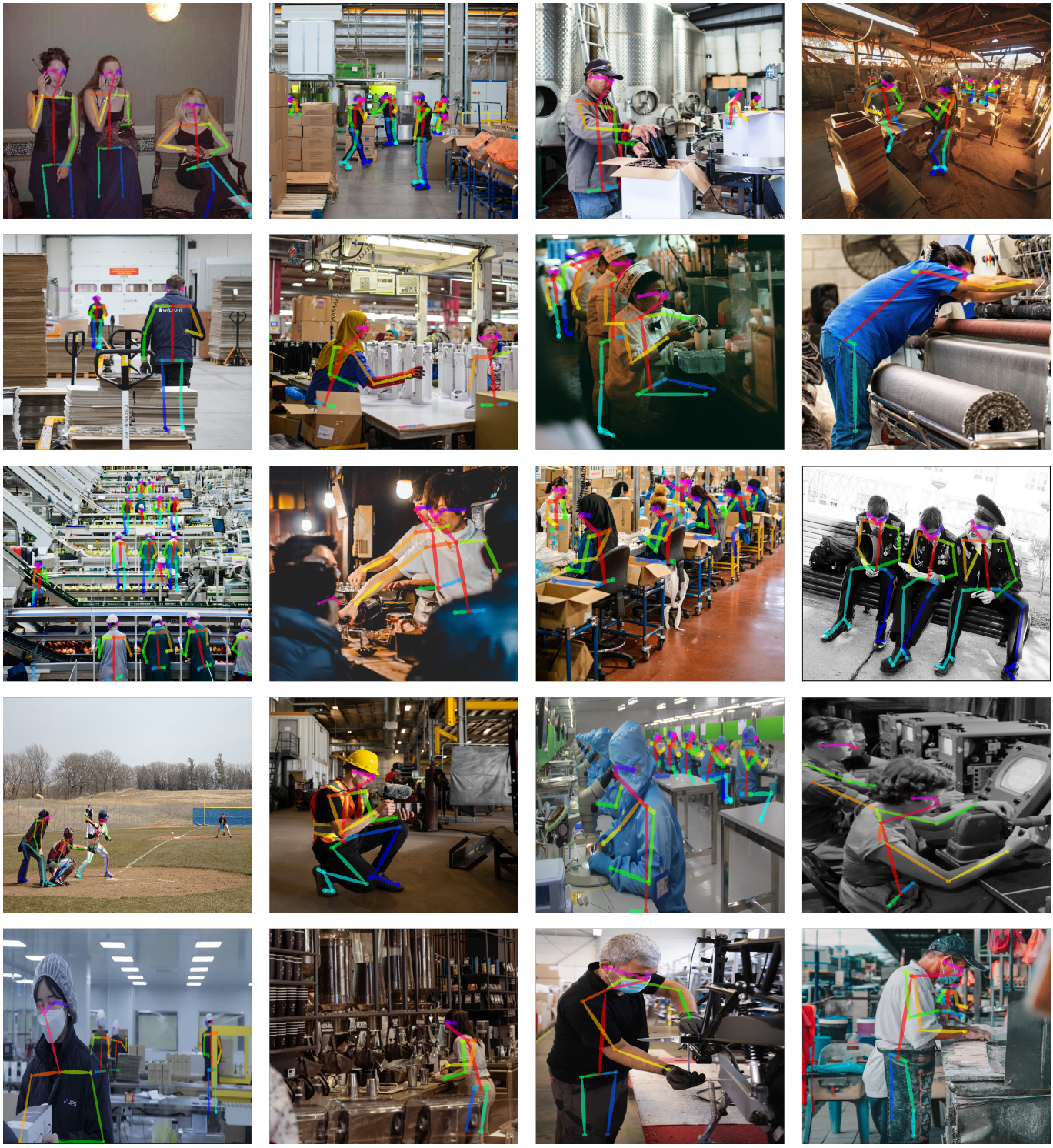

Figure 4. Results.

4.2. Performance Metrics

In this study, various performance metrics are used to evaluate the real-time pose estimation based on the ResNet-50 model. These performance metrics assist in quantitatively assessing and comparing the performance of the real-time pose estimation technique using the ResNet-50 architecture.

Average Precision (AP) is a metric commonly used in multi-class classification tasks such as object detection. It calculates the area under the precision-recall curve to obtain the

average precision for multiple classes. AP takes into account class imbalances and provides a comprehensive evaluation of performance for all classes.

Mean Average Precision (mAP) is also used in multi-class classification problems. Similar to AP, it calculates the area under the precision-recall curve for each class and then computes the mean of all class AP values to obtain the final mAP.

Accuracy is a metric that represents the proportion of correctly predicted poses by the pose estimation model. It is generally used to measure how well the predicted poses align with the actual poses. The accuracy is computed as follows (where *TP* = True Positive; *TN* = True Negative; *FP* = False Positive; *FN* = False Negative; *TP* + *TN* = Number of correct predictions; *TP* + *FN* + *FP* + *TN* = Total number of predictions):

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN.} \quad (1)$$

Precision is the ratio of true positive predictions to all positive predictions made by the model. *Recall*, on the other hand, is the ratio of true positive predictions to all actual positive instances. *Precision* and *Recall* are calculated using the following formulae [10]:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN.} \quad (3)$$

Precision and *Recall* are complementary metrics and a model is evaluated positively if both values are high.

F1 Score is the harmonic mean of *Precision* and *Recall* and is used to consider both *Precision* and *Recall* simultaneously. It provides a comprehensive evaluation of a model's prediction performance. *F1 Score* is calculated using the following formula:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall.} \quad (4)$$

Image processing time refers to the time required takes to process a given input image and output a pose estimation result. It can be affected by various factors, including the resolution of the input image, the complexity of the algorithm, and the hardware used. Several techniques and methods have been studied to improve the image processing time of real-time pose estimation sytems [2,34]. For instance, a monocular image-based pose estimation model using deep learning [2] is one of the methods that can minimize the image processing time while achieving accurate results. Additionally, hardware accelerators can be utilized to parallelize and expedite image processing tasks. In this study, we switched from OpenPose with VGG19 to ResNet-50 to reduce the image processing time. As VGG19 is known to be a complex model, adopting ResNet-50, a lighter model compared to VGG19, allowed us to increase the image processing speed. This alteration significantly improved the image processing time of the real-time pose estimation system. These performance metrics are employed to quantify and compare the performance of ResNet-50-based real-time pose estimation and play a vital role in evaluating the research results.

4.3. Results

In this study, we evaluated the performance of a real-time pose estimation technique using ResNet-50-based OpenPose and compared it to the performance of the original OpenPose, AlphaPose, and DensePose for rapid safety incident prevention and accident detection among field workers.

Image processing time is an important factor in real-time pose estimation techniques. Figure 5 presents the results of measuring the time each model took to process a given set of images. The algorithms used for comparison are the traditional OpenPose, AlphaPose, and DensePose. AlphaPose is a computer vision algorithm that specializes in multi-person

recognition and pose estimation, while recognizing multiple people in an input image or video and estimating the body parts and joint positions of each person. It ranks first on the MPII dataset [33] and shows high performance on the COCO dataset [32], and it is based on YOLOv3, which is an algorithm that provides both high speed and accuracy [35]. DensePose is an algorithm that divides the human body into fine-grained units through density-based segmentation and estimates the pose at each unit, while it also generates a high-resolution pose map by estimating the exact position and orientation of human body parts in the input image. It has the ability to map each human pixel in an RGB image to a 3D surface, and it is an algorithm that can be used to solve the problem of segmenting parts and instances within an object [36]. In our experiments, we measured the processing time of each model when 10, 20, 30, 40, and 50 images were input. OpenPose based on ResNet-50 showed respective image processing times of 3.05 s, 3.41 s, 3.7 s, 4.22 s, and 4.5 s. On the other hand, the image processing times for traditional OpenPose were 3.87 s, 4.42 s, 5.2 s, 5.73 s, and 5.9 s, respectively; those for AlphaPose were 5.92 s, 6.6 s, 7.12 s, 6.63 s, and 6.85 s, respectively; and those for DensePose were 6.21 s, 6.81 s, 7.39 s, 7.64 s, and 8.02 s, respectively. Taken together, the average image processing time for ResNet-50-based OpenPose is 3.78 s, while the average image processing times are 4.84 s for traditional OpenPose, 6.16 s for AlphaPose, and 7.37 s for DensePose. From these results, we can see that OpenPose based on ResNet-50 has a faster image processing speed than other pose estimation algorithms.

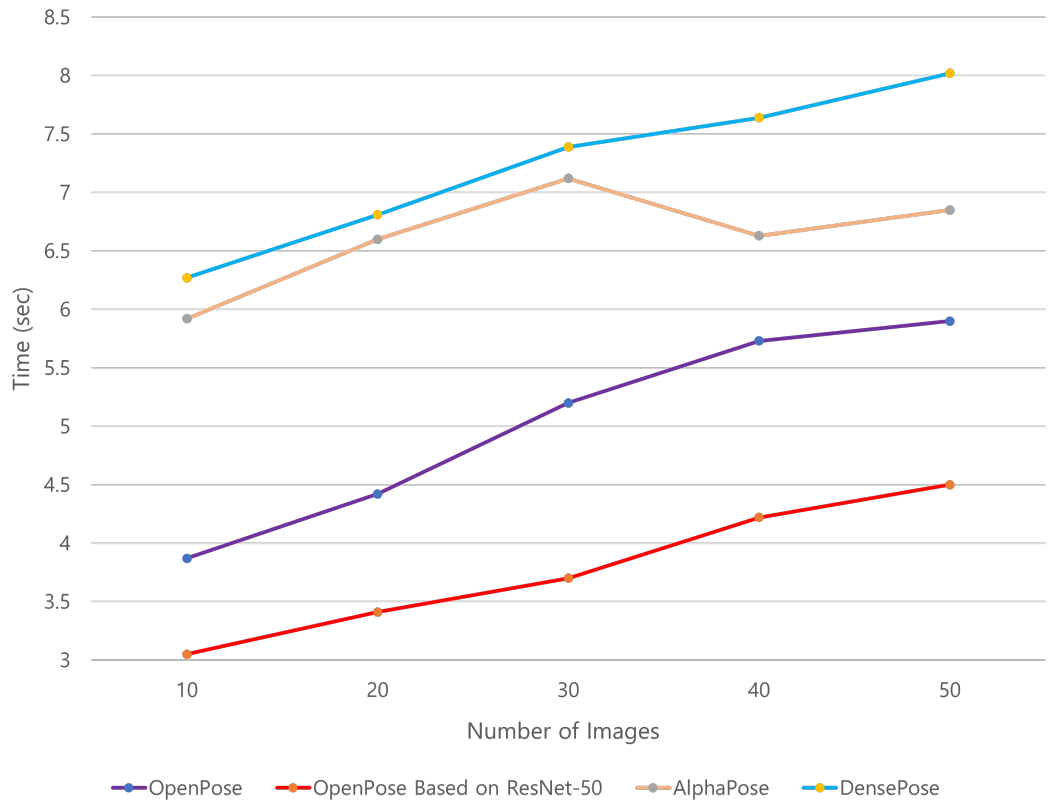

Figure 5. Image processing speed.

To provide a more detailed explanation of the results, we identified specific factors that could influence the accuracy of pose estimation results, resulting in individual body parts being omitted from the estimation. These factors include the following: first, the human arms being obscured by other objects or a dark background, leading to inaccurate recognition of the left shoulder; second, low-resolution input images or the presence of noise, which can cause pose estimation algorithms to produce inaccurate results; third, dataset imbalances or a lack of diversity, causing difficulties for the pose estimation model

in recognizing specific body parts. Such omissions of individual body parts can significantly impact the accuracy of the results, particularly in applications where high reliability is required, such as safety prevention and accident detection. In Figure 5, various colors are used to visualize the pose estimation results, representing the different body parts of humans as points. Each color corresponds to a specific body part, and these points collectively form the human pose in the image. The pose estimation technique identifies and localizes these body parts, enabling the system to recognize key points on the human body accurately.

For better visual understanding, the colors are assigned to specific body joints as follows:

- Red: Line connecting the head and neck;
- Orange: Line connecting the right shoulder and right elbow;
- Yellow: Line connecting the right elbow and right wrist;
- Lime Green: Line connecting the left shoulder and left elbow;
- Green: Line connecting the left elbow and left wrist;
- Cyan: Line connecting the right thigh, right calf, and right foot;
- Blue: Line connecting the left thigh, left calf, and left foot;
- Pink: Line connecting the nose and right eye;
- Purple: Line connecting the nose and left eye.

These color-coded points effectively illustrate the human body's pose and its spatial relationships in the image. By examining the accuracy and alignment of these colored points, one can evaluate the effectiveness of the pose estimation algorithm in estimating human poses in real-time.

We also evaluated the performance in terms of accuracy. Figure 6 compares the results of ResNet-50-based OpenPose and the original OpenPose. The results of ResNet-50-based OpenPose and traditional OpenPose on the right side of Figure 6 show that ResNet-50-based OpenPose accurately estimated the feet of the seated worker on the far left of the image. These results show that ResNet-50-based OpenPose provides higher accuracy and faster image processing speed than traditional OpenPose. This is useful for rapid safety prevention and accident detection for field workers, and shows that ResNet-50-based OpenPose is an effective algorithm for real-time pose estimation, outperforming traditional OpenPose and other algorithms in terms of image processing speed and accuracy.

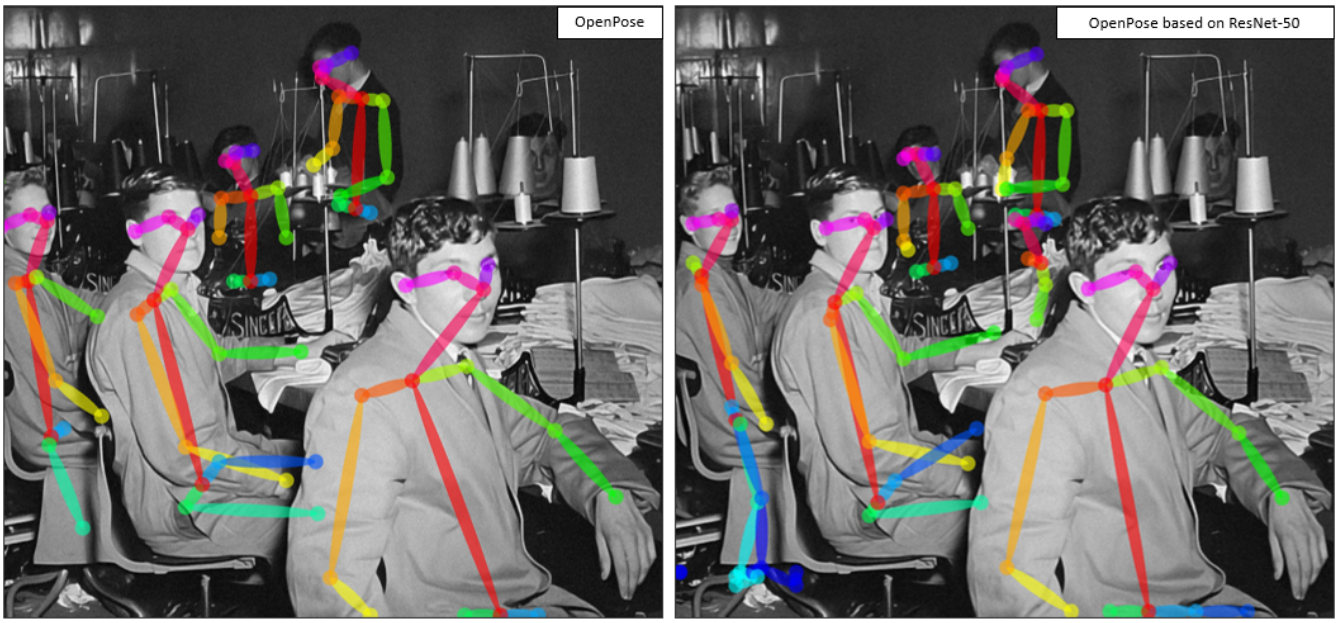

Figure 6. Compare image results.

In addition, through the Precision–Recall curve graph, Figure 7, is the Precision–Recall curve graph of each model, and Figure 8 is the combined Precision–Recall curve of each model. From these graphs, we can clearly see that OpenPose based on ResNet-50 has higher Precision and Recall values than other models. Precision is a measure of the percentage of results predicted to be positive that are actually positive, while Recall is a measure of the percentage of true positives that are correctly predicted to be positive. Therefore, high Precision and Recall values mean that the model correctly predicts positives and misses fewer positives. This indicates that OpenPose based on ResNet-50 provides better performance as a real-time pose estimation technique.

Finally, the evaluation metrics of each model can be seen in Table 2, where ResNet-50-based OpenPose shows higher mAP, AP, Accuracy, F1 Score, Precision, and Recall values compared to other models. This further emphasizes that OpenPose based on ResNet-50 can be used as an effective solution for real-time safety accident prevention and accident detection for field workers. The excellent performance of the model is expected to help it quickly detect potentially dangerous situations that may occur in the field and take timely action.

This study demonstrates that ResNet-50-based OpenPose can be utilized for rapid safety accident prevention and accident detection of field workers, which leads to the conclusion that it can be useful for rapid safety prevention and accident detection of field workers. Therefore, this study emphasizes that ResNet-50-based OpenPose is an effective algorithm for real-time pose estimation and outperforms other algorithms besides traditional OpenPose in terms of image processing speed and accuracy. This study further demonstrates that ResNet-50-based OpenPose can be used for rapid safety prevention and accident detection of field workers.

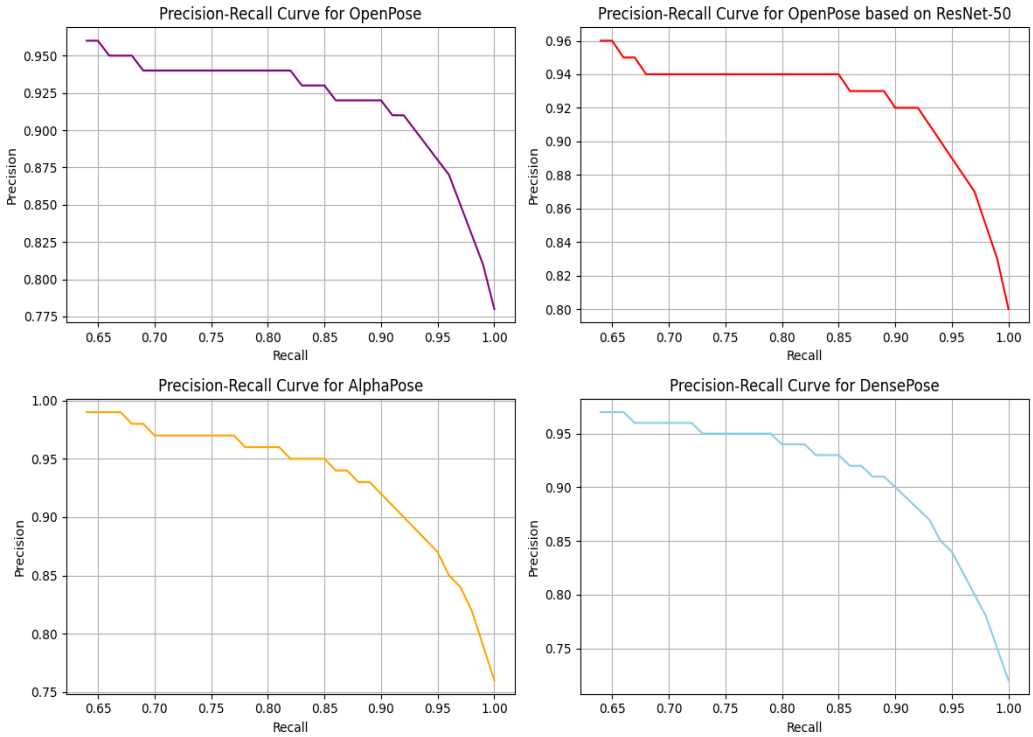

Figure 7. Precision–Recall curve of each model.

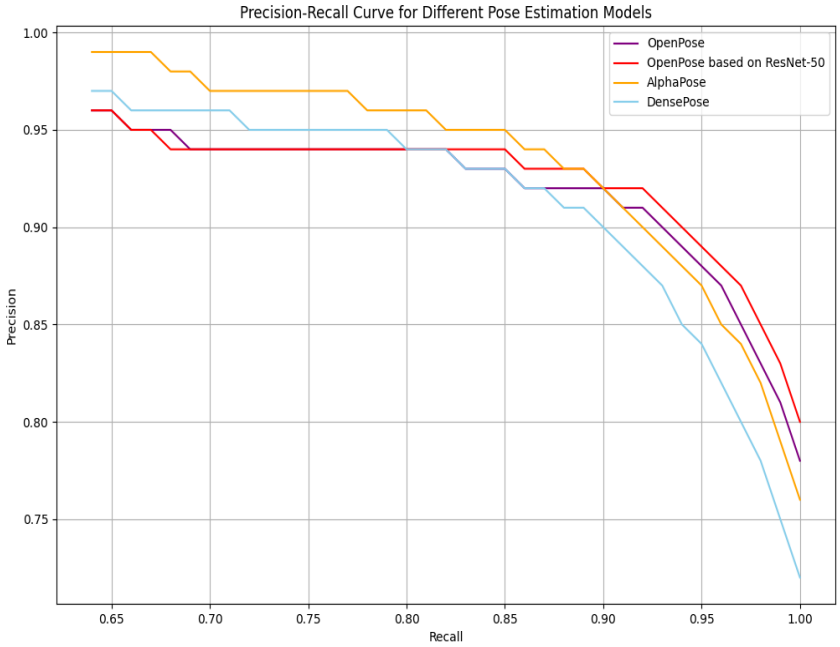

Figure 8. Overall Precision–Recall curve.

Table 2. Performance metrics value.

Model	AP	mAP	Accuracy	F1-Score	Precision	Recall
OpenPose	0.88	0.89	0.90	0.87	0.88	0.86
OpenPose based on ResNet-50	0.91	0.92	0.92	0.90	0.91	0.89
AlphaPose	0.88	0.89	0.91	0.88	0.87	0.89
DensePose	0.86	0.87	0.88	0.85	0.86	0.84

5. Conclusions

This study aims to explore a real-time pose estimation technique using ResNet-50-based OpenPose for rapid safety prevention and accident detection among field workers. The primary objective is to efficiently estimate the pose in real time, contributing to the prevention of safety incidents and quick accident detection. To achieve this, extensive experiments and evaluations were conducted, resulting in the development and evaluation of the proposed real-time pose estimation technique based on ResNet-50. We have made important contributions and conclusions according in this work. First, we developed a real-time pose estimation method using OpenPose based on ResNet-50. This method combines computer vision algorithms and deep learning techniques for multiple-human pose estimation. Using the excellent image feature extraction capabilities of ResNet-50 architecture along with the multi-joint pose estimation capabilities of the OpenPose algorithm, it is possible to estimate the pose of field workers in real time faster than the existing OpenPose. Second, this study verifies the performance of ResNet-50-based OpenPose through evaluation. The evaluation focused on image processing time and accuracy, and excellent results were obtained in comparison to other pose estimation algorithms. OpenPose based on ResNet-50 proved to be effective for the real-time pose estimation of field workers by providing fast image processing speed and high accuracy. In conclusion, this study demonstrated that the ResNet-50-based real-time pose estimation technique can be extremely helpful for rapid safety incident prevention and accident detection among field workers.

In this study, a real-time pose estimation technique using ResNet-50-based OpenPose was developed and evaluated for the rapid safety prevention and accident detection of field workers. The experimental results showed that OpenPose based on ResNet-50 has high

accuracy and a fast image processing speed. Therefore, this study verified the importance of real-time pose estimation technology in improving safety among field workers and preventing accidents. In future research, we plan to pursue the following directions: first, we will conduct research aiming to optimize deep learning architectures and algorithms to further improve the performance of real-time pose estimation techniques; second, we will evaluate the performance of pose estimation in different environments and investigate its applicability in the real world; third, we will conduct an empirical study by integrating the real-time pose estimation technique with a safety prevention system targeted toward field workers. This will further enhance the safety of field workers and provide a practical way to quickly respond to accident detection.

Author Contributions: Conceptualization, J.L. and J.J.; methodology, J.L. and T.-y.K.; software, J.L. and T.-y.K.; validation, J.L. and J.J.; formal analysis, J.L.; investigation, J.L. and T.-y.K. and S.B. and Y.M.; resources, J.J.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, J.J.; visualization, J.L. and T.-y.K.; supervision, J.J.; project administration, J.J.; funding acquisition, J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the SungKyunKwan University and the BK21 FOUR (Graduate School Innovation) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was supported by the SungKyunKwan University and the BK21 FOUR (Graduate School Innovation) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF). Moreover, this work was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Creative Consilience Program (IITP-2023-2020-0-01821) supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Marchellus, M.; Park, I.K. Human Motion Prediction with Deep Learning: A Survey. In Proceedings of the Korean Society of Broadcast Media Engineering Conference, Seoul, Republic of Korean, 23 June2021; pp. 183–186
2. Choi, J. A Study on Real-Time Human Pose Estimation Based on Monocular Camera. Domestic Master's Thesis, Graduate School of General Studies, Kookmin University, Seoul, Republic of Korea, 2020.
3. Zarkeshev, A.; Csiszár, C. Rescue method based on V2X communication and human Pose Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; Volume 63, pp. 1139–1146.
4. Cao, Z.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 172–186. [CrossRef]
5. Kocabas, M.; Karagoz, S.; Akbas, E. Multiposenet: Fast Multi-Person Estimation using pose residual network. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 417–433.
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
7. Park, J.H. A Study on Recognition and Analysis of Industrial Worker Risk Situation. Ph.D. Dissertation, Baejae University Graduate School, Daejeon, Republic of Korea, 2022.
8. Lin, C.-B.; Dong, Z.; Kuan, W.K.; Huang, Y.F. A framework for fall detection based on OpenPose skeleton and LSTM/GRU models. *Appl. Sci.* **2020**, *11*, 329. [CrossRef]
9. Yoo, H.R.; Lee, B.-H. OpenPose-based Child Abuse Detection System Using Surveillance Video. *J. Korea Telecommun. Soc.* **2019**, *23*, 282–290.
10. Younggeun, Y.; Taegeun, O. A Study on Improving Construction Worker Detection Performance Using YOLOv5 and OpenPose. *J. Converg. Cult. Technol. (JCCT)* **2022**, *8*, 735–740.
11. Shi, D.; Wei, X.; Li, L.; Ren, Y.; Tan, W. End-to-end Multi-Person Estimation with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
12. El Kaid, A.; Brazey, D.; Barra, V.; Baïna, K. Top-down system for multi-person 3D absolute pose estimation from monocular videos. *Sensors* **2022**, *22*, 4109. [CrossRef]

13. Zheng, Z.; Zha, B.; Zhou, Y.; Huang, J.; Xuchen, Y.; Zhang, H. Single-stage adaptive multi-scale point cloud noise filtering algorithm based on feature information. *Remote Sens.* **2022**, *14*, 367. [CrossRef]
14. Cao, Z.; Simon, T.; Wei, S.E.; Sheikh, Y. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1302–1310.
15. Chen, Y.; Wang, Z.; Peng, Y.; Zhang, Z.; Yu, G.; Sun, J. Cascaded pyramid network for Multi-Person Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7103–7112.
16. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
17. Chhabra, M.; Kumar, R. An Efficient ResNet-50 based Intelligent Deep Learning Model to Predict Pneumonia from Medical Images. In Proceedings of the 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 7–9 April 2022; pp. 1714–1721.
18. Odusami, M.; Maskeliūnas, R.; Damaševičius, R.; Krilavičius, T. Analysis of features of Alzheimer's disease: Detection of early stage from functional brain changes in magnetic resonance images using a finetuned ResNet18 network. *Diagnostics* **2021**, *11*, 1071. [CrossRef] [PubMed]
19. Shahwar, T.; Zafar, J.; Almogren, A.; Zafar, H.; Rehman, A.U.; Shafiq, M.; Hamam, H. Automated detection of Alzheimer's via hybrid classical quantum neural networks. *Electronics* **2022**, *11*, 721. [CrossRef]
20. Li, X.X.; Li, D.; Ren, W.X.; Zhang, J.S. Loosening Identification of Multi-Bolt Connections Based on Wavelet Transform and ResNet-50 Convolutional Neural Network. *Sensors* **2022**, *22*, 6825. [CrossRef]
21. Fulton, L.V.; Dolezel, D.; Harrop, J.; Yan, Y.; Fulton, C.P. Classification of Alzheimer's disease with and without imagery using gradient boosted machines and ResNet-50. *Brain Sci.* **2019**, *9*, 212. [CrossRef] [PubMed]
22. Zhang, R.; Zhu, Y.; Ge, Z.; Mu, H.; Qi, D.; Ni, H. Transfer learning for leaf small dataset using improved ResNet50 network with mixed activation functions. *Forests* **2022**, *13*, 2072. [CrossRef]
23. Nasirahmadi, A.; Sturm, B.; Edwards, S.; Jeppsson, K.H.; Olsson, A.C.; Müller, S.; Hensel, O. Deep learning and machine vision approaches for posture detection of individual pigs. *Sensors* **2019**, *19*, 3738. [CrossRef] [PubMed]
24. Pérez-Pérez, B.D.; Garcia Vazquez, J.P.; Salomón-Torres, R. Evaluation of convolutional neural networks' hyperparameters with transfer learning to determine sorting of ripe medjool dates. *Agriculture* **2021**, *11*, 115. [CrossRef]
25. Altameem, A.; Mahanty, C.; Poonia, R.C.; Saudagar, A.K.J.; Kumar, R. Breast cancer detection in mammography images using deep convolutional neural networks and fuzzy ensemble modeling techniques. *Diagnostics* **2022**, *12*, 1812. [CrossRef] [PubMed]
26. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, Real-Time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. *Comput. Vision–Eccv.* **2016**, *14*, 630–645.
29. Mascarenhas, S.; Agarwal, M. A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In Proceedings of the International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), Bengaluru, India, 22–24 December 2021; pp. 96–99.
30. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. DHow transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3320–3328.
31. Ikechukwu, A.V.; Murali, S.; Deepu, R.; Shivamurthy, R.C. Shivamurthy. ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images. *Glob. Transit. Proc.* **2021**, *2*, 375–381. [CrossRef]
32. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755.
33. Andriluka, M.; Pishchulin, L.; Gehler, P.; Schiele, B. 2d human pose estimation: New benchmark and state of the art analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3686–3693.
34. Park, J.-H.; Ha, O.-K.; Jun, Y.-K. Analysis of Image Processing Techniques for Real-Time Object Recognition. *Proc. Korea Comput. Inf. Soc.* **2017**, *25*, 35–36.
35. Fang, H.S.; Li, J.; Tang, H.; Xu, C.; Zhu, H.; Xiu, Y.; Li, Y.L.; Lu, C. Alphapose: Whole-body regional Multi-Person Estimation and tracking in Real-Time. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 7157–7173. [CrossRef] [PubMed]
36. Güler, R.A.; Neverova, N.; Kokkinos, I. Densepose: Dense human Pose Estimation in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.