Full Body Pedestrians Orientation Estimation using Machine Learning

By

Bukhtawar Zamir

CIIT/FA20-RCS-012/WAH

MS Thesis

In

Computer Science

COMSATS University Islamabad

Wah Campus, Pakistan

Session 2020-2022

COMSATS University Islamabad, Wah Campus

Full Body Pedestrians Orientation Estimation using Machine Learning

A Thesis presented to

COMSATS University Islamabad, Wah Campus

In partiall fullfilment

of the requirement of the degree of

MS (Computer Science)

By

Bukhtawar Zamir

CIIT/FA20-RCS-012/WAH

Session 2020-2022

Full Body Pedestrians Orientation Estimation using Machine Learning

A Thesis submitted to the Department of Computer Science as partial fulfillment of the requirement for the award of Degree of MS Computer Science

Name	Registration Number
Bukhtawar Zamir	CIIT/FA20-RCS-012/WAH

Supervisor

Dr. Mudassar Raza
Assistant Professor
Department of Computer Science
COMSATS University Islamabad (CUI), Wah Campus

Co-Supervisor

Dr. Saleha Masood
Assistant Professor
Department of Computer Science
COMSATS University Islamabad (CUI), Wah Campus

Final Approval

This thesis titled

Full Body Pedestrians Orientation Estimation using Machine Learning

By

Bukhtawar Zamir

CIIT/FA20-RCS-012/WAH

Has been approved

For the COMSATS University Islamabad Wah Campus, Pakistan

External Examiner: ____

Supervisor: ____

Dr. Mudassar Raza
Assistant Professor
Department of Computer Science, CUI Wah Campus, Pakistan

Co-Supervisor: ____

Dr. Saleha Masood
Assistant Professor
Department of Computer Science, CUI Wah Campus, Pakistan

Head of Department: ____

Dr. Muhammad Wasif Nisar
Professor
Department of Computer Science, CUI Wah Campus, Pakistan

Author's Declaration

I, Bukhtawar Zamir **(CIIT/FA20-RCS-012/WAH)** hereby declare that I have produced the work presented in this thesis "**Full Body Pedestrians Orientation Estimation using Machine Learning**", during the scheduled period of study. I also declare that I have not taken any material from any source except referred to wherever that amount of plagiarism is within an acceptable range. If a violation of HEC rules on research occurred in this thesis, I shall be liable for punishable action under the plagiarism rules of the HEC.

Date: 29-08-2022

Signature of the Student:

Bukhtawar Zamir

CIIT/FA20-RCS-012/WAH

Certificate

It is certified that Ms. Bukhtawar Zamir (CIIT/FA20-RCS-012/WAH) has carried out all the work related to this thesis under my guidance and supervision at the Department of Computer Science, CUI Wah Campus and the work fulfills and meets the prerequisites for the award of MS degree.

Date: _29-08-2022_______________

Supervisor:

____

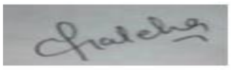
Dr. Mudassar Raza

Assistant Professor

Department of Computer Science,

CUI Wah Campus

Co-Supervisor:



____

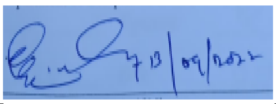
Dr. Saleha Masood

Assistant Professor

Department of Computer Science,

CUI Wah Campus

Head of Department:



____

Dr. Muhammad Wasif Nisar

Professor

Department of Computer Science,

CUI Wah Campus

DEDICATION

I dedicate this thesis to,

My loving Parents especially to my Mother **Asma Farhat** who has sacrificed her comforts for me, Brothers Umair Ghazi and Mohsin Ali and Teachers. Without them I was unable to complete this task.

Their love and prayers helped me to complete this thesis.

I also dedicate my thesis to my friends especially Syeda Sehrish, Mehak, Mian M. Talha, Hira and my seniors who have always helped me to complete this thesis.

ACKNOWLEDGEMENTS

In the name of ALLAH, the Most Gracious and the Most Merciful, all praise and thanks due to ALLAH peace and blessing be upon HIS MESSENGER, without ALLAH's blessings this thesis would not have been possible.

I would like to express my heartfelt appreciation and gratitude to my supervisors **Dr. Mudassar Raza** and **Dr. Saleha Masood**, their vision and execution aimed at creating a structure, definition, and realism around the thesis and fostered the ideal environment for me to learn and grow. I especially pay my heartfelt gratitude to **Dr. Mudassar Raza**. This thesis is a result of his encouragement, his trust on me, and input in the numerous meetings he had with me, despite his busy schedule. I am thankful to my **Supervisors** for their precious time to help me in the completion of my thesis and research work. Finally, I would like to thank my parents and brothers who support me to complete my graduation.

Bukhtawar Zamir

CIIT/FA20-RCS-012/WAH

ABSTRACT

Full Body Pedestrian Orientation Estimation using Machine Learning

In human body orientation estimation (HBOE), body parts play a vital role to build the structure of humans. The estimation task has gotten tremendous attention in previous years due to its applications in almost every field of the real world like medical, surveillance, sports, augmented reality, animations, and many more. Although different approaches for HBOE have been presented, these methods still face obstacles like rapid variation in pose, different viewpoints, camera issues due to weather conditions, etc. The main purpose of this study is to deal with these addressed issues. So much work is being done in this field of research. In this research the dataset called big dataset for body orientation (BDBO) is used. The main purpose of the research is to give high estimation accuracy in short time. The research consists of a few core steps i.e. initial preprocessing, features extraction, features selection and classification. Additionally, the performance of the employed techniques is evaluated and studied to highlight the high results. The first step is image pre-processing in which image sharpening is performed and the resolution of images is enhanced with SRGAN-VGG54. For feature extraction the features from the dataset two CNNs are used in which one is VGG-19, a pre-trained network and the other is a proposed net called BlackNet. The main contribution of this research is BlackNet, which is used to extract the useful features form the dataset for better accuracy. After extracting the features from two CNNs, features are then fused. After feature fusion, features are then passed through the phase of feature selection. For this purpose Whale Optimization Algorithm (WOA) is used for extraction useful and optimal features from fused features. These optimal features are then passed to the state-of-the-art classifiers which are SVM and KNN. A detailed analysis of proposed methodology is given in the sections below to highlight the contribution of this research.

Keywords: Classification, Pre-processing, Pedestrian Orientation, SRGAN-VGG54, VGG-19, WOA.

Table of Content

List of Figures

List of Tables

List of abbreviations

ACC	:	Accuracy
AI	:	Artificial Intelligence
ANN	:	Artificial Neural Networks
AUC	:	Area Under the Curve
CNN	:	Convolutional Neural Network
CV	:	Computer Vision
DL	:	Deep Learning
DNN	:	Deep Neural Network
FE	:	Feature Extraction
FF	:	Feature Fusion
FN	:	False Negative
FP	:	False Positive
FS	:	Feature Selection
F-S	:	F1-Score
FV	:	Feature Vector
HOG	:	Histogram of Gradient
KNN	:	K-Nearest Neighbor
NPV	:	Negative Predicted Values
PPV	:	Positive Predicted Value
PR	:	Precision
PRM	:	Pose Refine Machine
PS	:	Prediction Speed
PSL	:	Partial Least Square
Re	:	Recall
SEN	:	Sensitivity
SEP	:	Specificity
SRGAN	:	Super Resolution Generated Adversarial Network
SSD	:	Single Shot Detector
SVM	:	Support Vector Machine
TN	:	True Negative
TP	:	True Positive
TT	:	Training Time

CHAPTER 1

Introduction

1. Introduction

In computer vision, human body orientation estimation (HBOE) is a crucial process. The most important challenge in the HBOE is to measure the direction of moving pedestrians precisely from a video or image. As a component of the behavior analysis system, accurate HBOE can considerably improve the estimation of human posture [1]. Estimation of the direction of the human body or a specific part of it is important for a variety of activities and healthcare applications [2], counting people [3], detection of a fall [4], and predicting falls in old people [5]. To suggest the walking direction of pedestrians, the body's position is an excellent way to predict what the pedestrian is about to do next in autonomous driving [6]. Basic decisions like route planning and steering control, as well as safety precautions like collision avoidance and accident forewarning, are examples of self-decisions. Human head and full-body orientation estimation are essential issues that are mostly studied in the domain of pedestrian safety and activity prediction [7] and robotic applications [8]. Furthermore, while navigating congested environments and engaging with other pedestrians, individuals have an inherent capacity to predict the future behaviors of other people. For example, avoiding head-on collisions and retaining a safe distance from fellow companions. The capability would allow autonomous devices to operate intelligently in urban contexts by understanding and predicting pedestrian movements [9]. Detecting the teacher's and students' gaze directions and body orientations is critical for determining who is paying much attention to whom. It also gives crucial hints for deciphering their nonverbal, unaware conduct. It is explained how video recordings from a teacher's smartphone may be used to estimate the teacher's and students' gaze directions, as well as their body alignment with Machine Learning (ML) algorithms[10].

With advances in artificial intelligence [11], sensors [12], and control theory, self-driving automobiles are quickly growing. According to a recent analysis of Google's self-driving automobile, 90% of self-driving car failures occur in crowded areas, with 10% of failures caused by the wrong estimation of pedestrian behavior. Crossing the street is one of the most prevalent pedestrian activities, and it is linked to pedestrian safety. Most of the techniques for detecting pedestrian crossing activities are based on their skeletal properties [13, 14]. Inertial localization methods usually predict 3D motion using linear acceleration, angular velocity, and magnetic flux density data from an inertial measurement unit (IMU). The need for correct 3D orientation estimations e.g., roll, pitch, yaw, and quaternion and rotation matrix to accurately convert sensor frame readings to a global reference frame is a well-known flaw that has affected inertial localization. Small mistakes in this component can lead to significant localization errors by making inertial pedestrian localization impractical [15]. Because of their efficiency, existing localization technologies often depend on WiFi, Bluetooth, LiDAR, or camera sensors. WiFi and Bluetooth beacon-based systems are expensive due to the extensive instrumentation of the environment required for proper identification [16]. Whereas LiDAR-based localization is extremely precise, it is both costly and energy-intensive [17]. Recent deep-learning techniques, such as IONet [18] and RoNIN [19], have shown that an IMU can be used to estimate 3D device (or user) motion, but they do not directly address device orientation prediction. Through the use of supervised learning to directly estimate the spatial displacement of the device. These innovative approaches have been able to overcome the problem of drift that traditional inertial localization systems suffer from. Most existing efforts rely on the device's 3D orientation estimations by standard filtering-based procedures which can be erroneous. Before applying a deep rotation, this orientation is usually employed as a first step to rotate the local IMU readings to a common reference frame[20].

Many motion trackers including mechanical [21] and LIDAR [22], and optical movement trackers [23] are created to identify the orientation of the human body. Calculating the 3D postures of several people is more difficult than estimating single human poses. The occlusion generated by neighboring persons adds to the difficulty in estimating multiple persons from a single viewpoint. Because the wider state space, occlusions, and cross-view uncertainty are the key issues in HBOE [24]. The accurate estimation of pedestrians is one of the most difficult tasks in urban areas. Simple tracking and motion models are insufficient to capture human activity, which is very dynamic. Changes in the look of pedestrians e.g., changing garments, sizes, rapid position changes, and variable settings make identifying occluded people a tough problem to deal with [25]. Although convolutional neural networks (CNNs) [26] have made significant progress on difficult datasets containing only clear and high-quality images by implementing the models on real-world problems. In the past few years, object identification techniques consisting of deep learning have been a major inspiration for human activity recognition approaches including Region-based CNN [27], Fast Region-based CNN [28], and Faster Region-based CNN [29] for two-stage approaches and YOLO [30] and SSD [31] for one-stage approaches [32]. The Majority of current pedestrian detection algorithms concentrate on full-body detection. This method performs less well than part detectors when occlusions are also present [33].

Nowadays, the existence of social service robots has become unavoidable to aid or serve people in their daily lives. In some circumstances, a robot is designed to interpret human intent to collaborate with humans [34]. The subjects are divided into numerous categories, including sensor-based, feature utilization, and still or object tracking. Sensor-based types can be divided into four categories such as Laser Range Finder (LRF) [35], RGB camera [36], RGB-D camera [37], and ToF camera [38]. In terms of feature utilization, some studies focused on optical images [39], while others offered a mix of RGB images and depth [40]. Other techniques, such as human body form traits, were also proposed as a solution [38]. Static and moving objects are studied in [41], while the remaining employ static image analyses from the dataset.

The importance of pedestrian detection and categorization in Advanced Driver Assistance Systems (ADAS) has attracted researchers in recent years. Road accidents are the leading cause of death among teenager people under the age of 30, according to the World Health Organization (WHO). Which shows around 1.35 million deaths globally every year with approximately 23% of pedestrians [42]. Pedestrian mistake is the major cause of these incidents. In 2018, WHO revealed that most road deaths are caused in poor and middle-income nations [43] because of their huge population.

The process of predicting pedestrian body orientation is not completely shifted to deep learning algorithms. One of the reasons is the lack of datasets. There are few standard datasets for orientation estimation available publically. These datasets are very small in size, and hence unsuitable for deep neural network training [44].

1.1. Limitations of pedestrian's orientation estimation

The limitations of the orientation found in the literature are as follows:

- Complicated appearance [45] The features of pedestrians are frequently made up of a variety of colors and patterns on their clothing and accessories. Monocular solutions will find it easy to confuse them with the backdrop environment. Monocular solutions have a tougher time detecting orientation-related properties because orientation and behavior are mainly unrelated.

- The short height of pedestrians [46] and the Shorter height of pedestrians make them harder to notice while driving. The smaller height of pedestrians makes it difficult to estimate the orientation correctly.
- Deformation [47] Like other traffic objects pedestrians are changeable and take on a variety of forms and sizes depending on their states and gestures. Several methodologies, such as 3D modeling are ineffective for estimating pedestrian orientation. Furthermore, deformation implies that the same orientation might be associated with a wide range of shape features which is quite ambiguous.

1.2. Research motivation

We initiated this research to improve the accuracy of pedestrian orientation. Our area of research is surveillance in which we have analyzed the orientation of pedestrians. The main objective of this research is to propose a technique that can deal with a huge amount of images for estimation. There is no such dataset available publically for pedestrian orientation. For the proposed methodology, a big dataset was found in the literature that is taken from the authors on a request. As the world population is increasing day by day, managing a large number of people without cameras through humans is impossible within this keeping an eye on the intentions of the human is also very difficult. Orientation estimation through machine learning algorithms helps to estimate intentions. The crowd in the shopping mall, the fall of old people in the hospital, and pedestrians crossing the roads can relate to the importance of orientation estimation. Road accidents are the leading cause of death among young people under 30 years old. Pedestrian mistake is the major cause of these incidents. There must be a technique to solve the issues mentioned above. The reasons listed above are the motivation behind this research.

1.3. Problem statement

Orientation estimation of the pedestrian is very important in so many real-life applications. In the past decade, a lot of work has been done on orientation estimation due to its importance. However, a lot of challenges are still present in this domain. The difficulties and limitations in pedestrian orientation estimation include occlusion, variation in viewpoint, quick pose changes, weather conditions, camera hangings, and the appearance of pedestrians like changing clothes and angles. Due to these challenges, there is a need for improvement in this domain of research. A new approach for better classification will be proposed in our study to increase the accuracy of orientation estimation.

1.4. Research objectives

The research objectives of the proposed study are as follow:

- To propose a new algorithm for pedestrians orientation estimation
- To overcome the existing challenges of the orientation of pedestrians like illumination, low resolution, etc.
- To use optimal feature selection techniques for better classification and accuracy

1.5. Contribution

This research proposes a new framework for the identification and classification of pedestrian orientation estimation. The main contributions of the thesis are as follows:

- A new deep CNN model is proposed which is pre-trained on BlackNet.
- SRGAN-VGG54 is used for pre-processing of pedestrian images. A newly proposed deep CNN network called BlackNet is used with VGG-19 for extracting features. Two feature vectors are generated in this phase. These features are later fused and a feature vector for the fused feature is generated.
- The Whale Optimization Algorithm is used for feature selection to select useful and important features. These selected features are then used for classification on SVM and KNN classifiers.

1.6. Thesis organization

This thesis comprises five chapters, where the detail is mentioned below.

- Chapter 1 describes the introduction of Pedestrian orientation estimation especially using machine learning. It briefly discusses motivation, objective, the contribution of proposed work in research, and the problem statement for the proposed work.
- Chapter 2 presents recent existing literature. The detailed review provides the different image processing and machine learning techniques that are proposed by the researchers.

- Chapter 3 provides the detail of the proposed methodology.
- Chapter 4 explains the experiments and results.
- Chapter 5 includes a conclusion and discussion section in which a summary of the contribution and proposed experiments of the thesis is provided. The appendix and references are listed at the end.

1.7. Summary

In this chapter, a brief introduction to the research topic is given. It is also stated that how pedestrian orientation estimation is important for the world. Applications and limitations are also mentioned. After researching other works, the contribution section is designed to focus on the importance of pedestrian orientation estimation.

Chapter 2

Related Work

2. Related work

Aside from the handcrafted aspects, an end-to-end feature extraction learning system solves the challenges of several classification tasks, and learning based on CNN is proved to be quite effective in recognizing objects [48]. Numerous work has been done on human pose estimation (HPE).

2.1. 2D HPE

HPE is delegated single-person or multi-person agreeing on the total number of assessed persons in the image. In comparison to posing estimation of multi-person, estimating a single person's pose from a given image is much easier [49]. To assess the posture of an individual from an image, there are two popular approaches. 1) Top-down approaches [50]. 2) Bottom-up approaches [51].

2.1.1. 2D pose estimation for single-person

Single-person pipelines that use deep learning methodology may be comprised of two types: regression-based and body part detection-based methods [52]. There are numerous works dependent on the regression system to estimate joints and body parts from images [53] DeepPose by using AlexNet is proposed which is a cascaded deep neural network regressor. As a result, the HPE research framework started to move away from traditional methods to deep learning, specifically convolutional neural networks(CNN) [54]. Figure 2.1 represents the architecture of DeepPose.

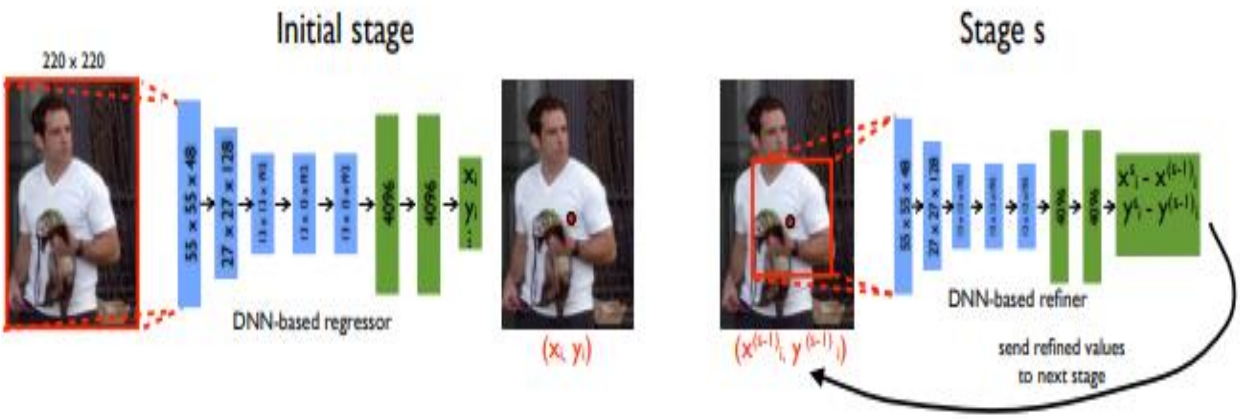


Figure 2.1 Pose regression based on DNN called DeepPose [54].

The objective of body part detection techniques is to estimate the relative position of various joints and parts of the human body [55], which are usually supervised by representations of heatmaps [56]. To address the joint areas and create successful CNN structures for HPE, utilizing heatmaps is a new developing interest [57]. In the case of occluded body parts, Generative Adversarial Networks (GANs) are investigated in pose estimation to produce biologically conceivable posture setups and separate the forecasts with high certainty from low certainty [58]. To give supervision to a network based on adversarial learning using two layered hourglass networks as discriminator and generator is created. The discriminator separates the ground-truth heatmaps from the projected ones. The generator predicts the position of each joint Unlike GAN-based methods, which use the pose estimation network as the generator and the discriminator to provide results [59]. To strengthen HPE models in complex scenarios, authors developed a multi-scale structure-aware neural network that incorporates multi-scale control, multi-scale attribute fusion, structure-aware loss information scheme, and a training process for making key points [60]. Authors created a Deeply Learned Compositional Model, an hourglass-based supervision network, to explain the dynamic and functional relationships between body parts and learn the compositional pattern knowledge in human bodies [61]. Since it was discovered that not all components are similar to one another, a Part-based Branches Network was created to learn representations and part categories instead of a shared depiction for all parts [62]. For successful video-based pose prediction, the authors designed a human pose interpolation module and a main frame proposal network for gathering spatial and temporal information from frames [63].

2.1.2. 2D pose estimation for multi-person

Two primary methodologies for 2D human pose assessment are bottom-up and top-down. The top-down method finds the number and location of people and detects each person [64]. There are two main components in the top-down pipeline, the detector for obtaining individual bounding boxes of the human body and a pose estimator for predicting the positions of key points. A series of projects depend on developing and upgrading the HPE network [65]. The bottom-up pipeline contains two stages: detection of body joints and assembling of joints [66]. The authors presented the earliest detector called DeepCut, which employs a body part detector based on a Fast R-CNN [67]. Authors introduced a Residual Steps Network (RSN) for learning complicated local representations through powerful feature fusion intra-level techniques, and a Pose Refine Machine (PRM) for determining a trade-off for representations of the characteristics on a local and global scale [68, 69].

To solve the occlusion problem in HPE, authors developed a Cascade Pyramid Network (CPN) having two subparts: RefineNet and GlobalNet CPN does well in predicting occluded points, according to their findings [70]. To overcome the problem of occlusion in crowd pose estimation, the authors created an occluded pose dataset and Occluded Pose Estimation and Correction (OPEC-Net) module [71]. Authors crafted two modules, the Spatial & Channel-wise Attention Residual Bottleneck and the Channel Shuffle Module, to improve multi-person pose prediction in occluded scenes by channel-wise and spatial knowledge enhancement [72]. To estimate the human from non-annotated images, the authors have introduced a method called mirror-net which allows for unsupervised image processing without pose specification [73]. To predict the pose of humans in crowded scenarios, the authors have shown remarkable improvements by using simple techniques of data augmentation [74]. To learn the human element classification, a new differentiable Hierarchical Graph Grouping (HGG) approach was suggested by the authors [75]. MultiPoseNet is a multi-task learning paradigm based on the net of a pose residual presented by the authors that can handle keypoint estimation, human recognition, and semantic segmentation tasks all at once [76].

2.2. 3D HPE

Handcrafted features, geometric constraints, and perspective relationships were utilized to predict the 3D pose of humans in the early years. With the advancement in deep learning, the implementation of deep neural networks is increased for image-to-3D HPE in recent years [77]. Estimating the human body or its shape from color images is a difficult task. In [78] authors proposed the novel task without stately limitations of the pose, background, or camera viewpoint to estimate the structure of the human body from multiple color images with off-shelf segmentation. For the estimation of the 3D pose of humans from a single RGB image [79] have proposed a framework for 3D HPE by a single RGB image. A reconstruction network can be combined with any depth map and 2D pose estimator and from a monocular image, which makes the system dynamic and easy to use. For root-relative and 3D HPE [80] proposed metric-scale truncation-hearty (MeTRo) volumetric heatmaps. Rather than being bound to the picture space. They may straightforwardly address the measurement space in which the individual is found and can be anticipated utilizing any completely convolutional network. Figure 2.2 illustrates the method proposed by the authors.

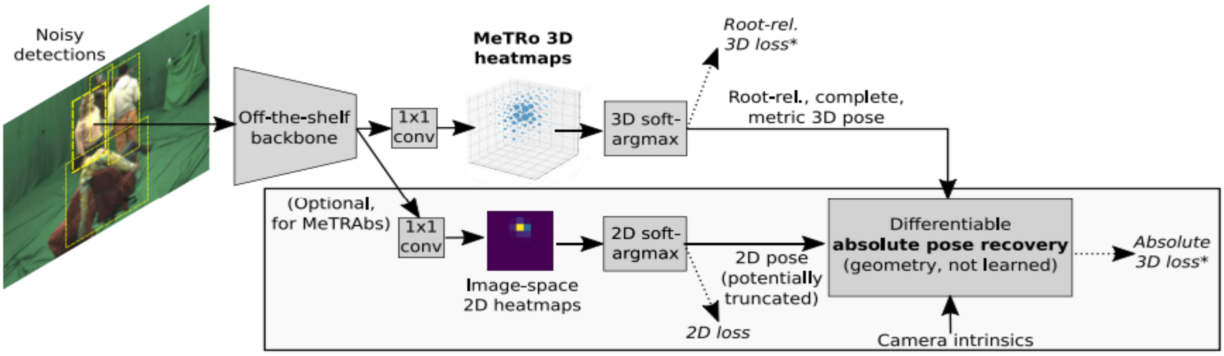


Figure 2.2 Illustration of volumetric heatmap [80]

In the example of MeTRAb, which is a multi-person absolute 3D pose estimation system. 3D joint locations and their relationships, as well as 3D joint rotations using a skeleton body model, are two of the most frequent techniques to illustrate 3D human poses, to extract twist rotations from an image, the author focused on joint appearances and used them successfully in the model. The model predicts twist angles with an average radian error of 0.14 and shows that estimation of twist rotations leads to a more realistic 3D human position [81]. The authors emphasized how occlusion is likely the most challenging barrier for human pose estimation in the outdoors, and they proposed a unique solution: multi-view feature fusion. The strategy is the exact opposite of previous attempts. Even when the joints are obscured in certain views, the suggested approach heatmap of 2D keypoints can consistently detect them [82].

Multiple 3D joint configurations can have the same 2D projection hence 3D pose estimation of the human becomes very difficult. In a recent research graph, neural network (GNN) or pictorial structure model (PSM) are used to reduce ambiguity by combining the character of these two modes the author has proposed a model ContextPose [83]. In outdoor HPE, 3D HPE is limited due to the self-occlusion of joints. To remove the ambiguity between joints the authors have proposed a network that predicts the 3D pose from the 2D pose by capturing the information on spatial structure [84].

2.2.1. 3D HPE methods for single-person

The approaches for 3D HPE for a single person are categorized into model-based and model-free approaches. Most of the time models of the human body are used to estimate 3D human pose [85].To reconstruct 3D human representations, model-free approaches do not use human body models. Model-free techniques, like voxel occupancy grid [86], vertex mesh [87], or implicit surface representation [88], anticipate a 3D body model directly from an image. Direct estimating approaches and 2D to 3D lifting approaches are the two types of methods that may be used. Direct estimate approaches use a 2D image to infer a 3D human position without estimating a 2D pose representation [89]. Methods based on the model [90] determine the parameters of a 3D body model [91], providing a valuable prior on human body shape**.** To solve the tasks in regression-based function for a single person the pose estimation problem is reformulated by the authors as a sequence prediction problem that can be solved efficiently by transformers by avoiding the issues of heatmap-based pose estimation [92]. By gathering local-kinematic parameters with energy-based loss, the authors [93, 94] developed a method for preserving the kinematic structure and looked at 2D component segments using the parent-relative local limb kinematic model. A kinematic latent normalizing flow representation with a differentiable semantic body part alignment loss function (a set of invertible transformations applied to the original distribution) was proposed by the authors [95]. Unlike kinematic models, which create human postures or skeletons, volumetric models can restore human mesh with high quality and provide additional information of human body shape. Authors [96] used SMPL parameters to regress 3D human mesh reconstruction. In the place of predicting SMPL parameters, authors [97] used a Graph-CNN architecture to regress the positions of the SMPL mesh vertices. To improve the flexibility of free-form 3D deformation, authors [98] merged the SMPL model with a hierarchical mesh deformation framework. In the SMPL model, authors [99] introduced a color recovery module to get vertex color via reflection symmetry. In an SMPL-based network which is a self-supervised resolution-aware network, authors [100] presented the contrastive learning technique. To ensure feature and scale consistency, the self-supervised contrastive learning system employs a self-supervision loss and a contrastive feature loss. AMASS, a large-scale motion capture dataset, was used for adversarial training of VIBE, an SMPL-based technique. The VIBE posture regression module used AMASS to distinguish between genuine human movements and anticipated poses. Existing well-trained models may fail when resolution is decreased because low-quality visual material is more prevalent in real-world settings than high-resolution visual material [101]. To recreate the 3D kinematics, the authors [102] used the Adam model. A 3D human representation known as 3D Part Orientation Fields (POFs) was created to encode the 3D orientation of human body parts in 2D space. The authors [103] proposed an orientation keypoints model for 6D HPE that can calculate complete 3-axis joint rotations, including yaw, pitch, and roll. The authors developed a novel Bone-level Skinned Human Mesh Model that decouples bone modelling from identity-specific changes by establishing bone lengths and joint angles [104]. For improved model generalization, the authors [105] updated SMPL to STAR by training with an extra 10,000 scans. The number of model parameters is cut in half compared to SMPL.

2.2.2. 3D HPE methods for multi-person

3D multi-person HPE is consist of Top-Down techniques that use a human detection network to identify single-person zones initially [106]. To determine the unique identification root joint for each individual, the authors [107] suggested Single-stage multi-person Pose Machine (SPM. The dense displacement maps were used to match the body joints to each root joint. However, this approach is constrained and can be used for supervised learning using paired 2D images and 3D posture observations. Regardless of the precision, authors [108] were able to immediately infer an intermediate 3D position of visible body joints. The whole 3D posture is then rebuilt using learned posture priors and global context to infer occluded joints. By using temporal coherence and fitting the kinematic skeleton model, the final 3D posture was refined.

In a multi-person situation, authors [109] developed a distance-based heuristic for interconnecting joints. Specifically, the remaining joints are connected by picking the closest ones in terms of 3D Euclidean distance, starting with the detected heads. Occlusion is another issue with bottom-up techniques. The authors [110] devised an Occlusion-Robust Pose-Maps (ORPM) strategy to include repetition in the formulation of location maps. It enables human association in the heatmaps, particularly for obstructed sceneries, to address this difficulty. Authors [111] presented a frozen network to utilize the common latent space across two distinct modalities represented as a cross-model alignment issue in the absence of paired 2D pictures and 3D posture annotations. The person grouping problem was framed as a binary integer programming (BIP) issue by authors [112]. By addressing the BIP problem, a limb scoring module estimated candidate kinematic connections of identified joints, and a skeleton grouping module combined limbs into skeletons. Body models were employed in a collection of algorithms to solve the association issue, with model parameters optimized to match the model projection with the 2D position. A multi-view consistency constraint was used in the network by Rhodin et al. [113], although it necessitates a substantial quantity of 3D ground-truth training data. Rhodin et al. [114] suggested an encoder-decoder system for learning the geometry-aware 3D latent representation from multi-view pictures and background segmentation without 3D annotations to solve this restriction. When the multi-view camera environment changes, however, the model must be retrained. A multi-view image is used to infer non-rigid 3D deformation parameters and recreate a 3D human body mesh [115, 116]. To match paired multi-view poses for 3D pose reconstruction [117-119] used epipolar geometry and adapted their methods to new multi-view camera environments. Matching each pair of views independently without the cycle consistency constraint can result in inaccurate 3D posture reconstructions. A self-supervised reinforcement learning strategy for selecting a minimal number of views for triangulation reconstruction of the 3D posture is suggested [120]. To prevent inaccurate estimates in each camera view, the authors aggregated all the characteristics in each camera view in 3D voxel space. To locate all persons and estimate their 3D posture, a cuboid proposal network and a pose regression network were created. It is not practicable to employ all perspectives for 3D pose estimation when there are enough viewpoints [121].

Remelli et al. [122] separated feature maps from camera perspectives by encoding pictures of each view into a uniform latent representation. These 2D representations are hoisted to the 3D posture as a lightweight canonical fusion utilizing a GPU-based Direct Linear Transform to speed up the procedure. An iterative processing technique is used to match the 2D postures of every view with the 3D posture while iteratively updating the 3D posture. In contrast to earlier approaches, which have a linear time complexity as the number of cameras increases, their technique has a linear time complexity [123].

2.3. Augmentation

To extend the dataset, Yang Li [124] used three types of data augmentation i.e. translation, rotation, and stretch, which result in reversed orientations from the previous orientations. With the low range of 2D to 3D pose combinations in the data for training, the present estimator for 3D human pose has low generalization capability to newer datasets. Kehong Gong et al. [125] introduce PoseAug which is an auto-augmentation model that increases the diversity of poses to train and improve the generalization of the pose estimator. It presents an augmentor that learns to change multiple geometry aspects of a pose using differentiable operations like posture, body size, viewpoint, and location. The model is applied to the MPI-INF-3DHP dataset and claims 88.6%

accuracy. Jiahang wang et al. [126] Suggested a novel approach called AdvMix to increase stability in various corruptions by evaluating the limitations of current advanced pose estimators to remove noise. Adversarial augmentation is made up of two neural networks which are trained together and against each other. The proposed method is depicted in Figure 2.3. Hanbyul Joo et al. [127] solve this challenge of outdoor HPE by adding a high-quality 3D pose that fits existing 2D datasets through augmentation.

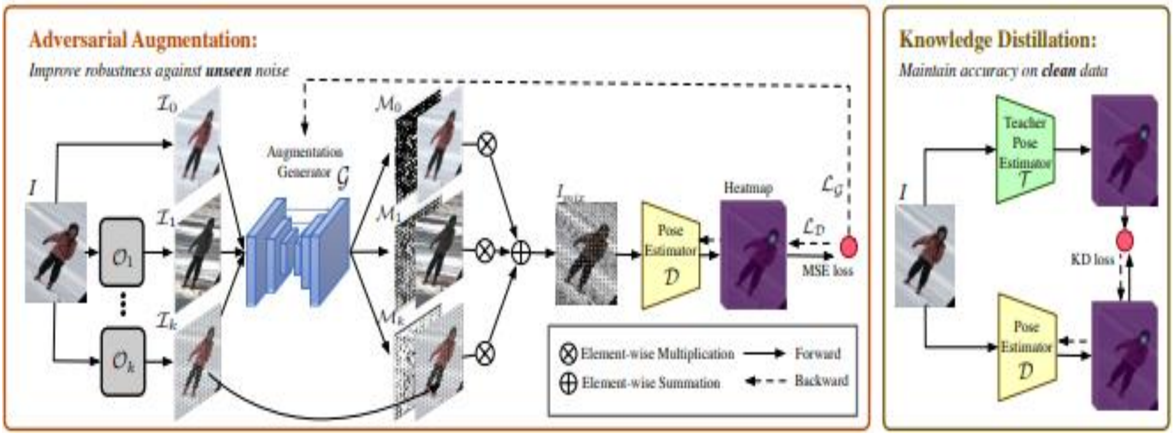


Figure 2.3 Illustration of Advmax Augmentation model [126].

2.4. Preprocessing

To successfully increase the representations of features of pedestrian's heads and bodies, Wei et al. [128] used Gabor Filtering [129] for preprocessing to increase the representation of pedestrian heads and bodies. Rahul et al. [130] performed histogram equalization [131], edge detection [132], and the selection of pedestrians as front point cloud on video data. Y.kohari et al. [133] suggested a technique to predict the orientation of the human body by taking resized and cropped greyscale images of humans as input and two neurons as output. The CNN was trained using a huge synthetic training set against an unchanged background. Siyang Song et al. [134] described a data transformation approach that divides multi-channel action primitive signals into two equal-sized frequency spectrum maps. These spectrum maps are good for feeding into CNNs. Several classification and regression tests were done using the DAIC-WOZ database given by the AVEC 2016 depression challenge to examine the performance of the suggested approaches. There exist many other approaches for image enhancement like SRGAN [135] , CF algorithm [136], Kalman filter [137], Particle filter (PF) [138], ACF [139].

2.5. Feature extraction and feature selection

To define pedestrian appearance information, Vikram Shree et al. [140] offer a feature representation based on the orientation that can be exchanged among sensors. The study proposes a cross-sensor track association technique to accomplish decentralized tracking based on that representation. They applied their methodology to DukeMTMC dataset and claimed 84.79% accuracy. Dennis et al. [141] presented a method for estimating 3D human posture in which the skeletal coordinate system is used to encode 3D human joint locations. As a result, the skeleton estimate is independent of camera settings by allowing it to be utilized in follow-up applications like action recognition that employs temporal data. Based on full body bounding box input, an integrated technique to body and head orientation estimation has been developed. A new model called PedRecNet is presented for feature extraction and claimed accuracy of 77.1% on the MEBOW dataset. Wengefeld et al. [142] presented a merged detection and orientation estimation method that uses the histogram of oriented gradient (HOG) features to find eight classes of upper body orientation and a class of background using HOG features. Fumito shinmura et al. [143] enhance the weights of the features at the silhouette's border of the human body by increasing the size of gradients from depth images in the HOG feature space. As a result, they can perform better on data with a complicated background. Weian Mao et al. [92] turn the posture estimation problem into a series prediction that is efficiently solved by the transformers. Dameng Yu et al. [144] offer a unique method based on high-level semantic characteristics extracted from the positions of human key points. The suggested technique uses a pose estimation algorithm to estimate the positions of human key points which is based on human posture, body motion constraints, and occlusion of body components. The positions of key points are then used to extract high-level semantic characteristics. Figure 2.4 represents the framework of the proposed model.

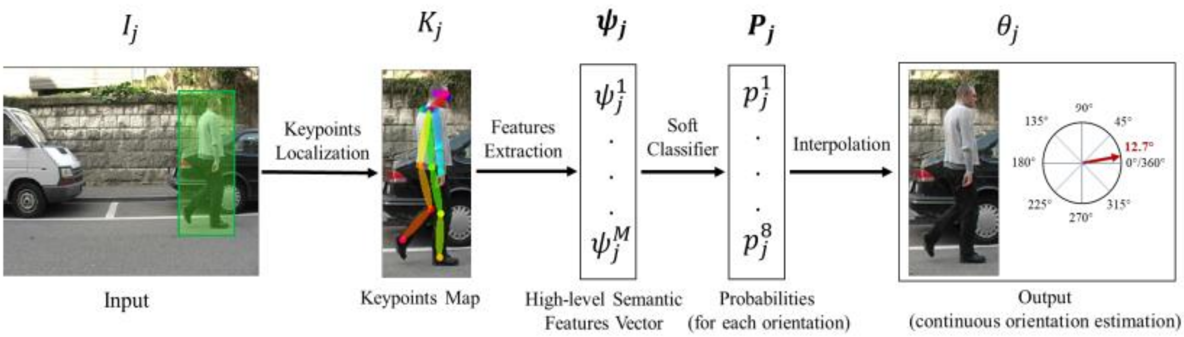


Figure 2. 4 Architecture of proposed model [144].

To accomplish the targeted aims in the article, the authors have used an attention mechanism that picks more important features related to the target by using ResNet-18 [145]. They applied the model to COCO and MPII datasets and claimed the accuracy of 70.9% and 90.4% respectively. Ye Yaun et al. [146] suggested a model called SimPoE that trains a strategy by using the current-frame posture estimate as an input and the next image frame as an output to manage a physically-simulated character and return the next-frame pose estimate by using a hand-crafted layer for feature extraction. Davis Rempe et al. [147] presented a HuMoR model for the Robust estimation of temporal shape and posture. The presented model learns a probability of posture change every time a motion sequence is completed. When different models are applied to video sequences, their results fall short. Inability to manage motion blurriness or pose occlusions are common flaws. Many current monocular 3D posture prediction algorithms focus on one component of the body, ignoring the fact that human motion is communicated through the motion of the body, hand, and face. Yu Rong et al. [148] presented FrankMocap, a quick and correct whole-body 3D posture estimation system that can build 3D faces, hands, and bodies from monocular images captured in the wild by using ResNet-50 as an encoder-decoder structure. Viladimir Gozuv et al. [149] is a wearable sensor-based approach for recovering a human's entire 3D posture from a 3D scan of the surroundings. The proposed approach combines self-localization based on a camera with IMU-based human body tracking. IMUs are attached with body parts and a head-mounted camera. Arjun Gupta et al. [150] used Neural 3D Mesh Renderer [151] to project predicted 3D mesh into a 2D binary mask and compare it with the original image by using feature extraction and matching techniques. Object identification in noisy images is traditionally accomplished using SIFT [152], SURF [153], and ORB [154] but the authors have modified ORB to detect features. Daniel Siectcher et al. [155] used EffiecientNet [156] for feature extraction in their study.

IN Junejo [157] suggested a method by using the color models and used VGG-19 for the task of pedestrian estimation. They experimented on the PETA dataset. Yingying Wang et al. [158] offer a unique deep inertial odometry method that focuses on pedestrian extraction of features. The suggested method was evaluated using the publicly available RoNIN dataset. The performance of the dataset was then tested in real-world scenarios on the CUHK. They claimed an accuracy of 4.98m.

2.6. Classification

To estimate a human orientation, Mudassar Raza et al. [159] used a deep-learning approach to identify the head-pose and full-body orientation of pedestrian and predicts the posture based on appearance. As a fundamental component of deep learning for classification, a supervised deep CNN model is described. The proposed model is trained on two publically available datasets and they claimed 92% accuracy for body orientation and 91% accuracy for head orientation estimation. The proposed framework is shown in Figure 2.5.

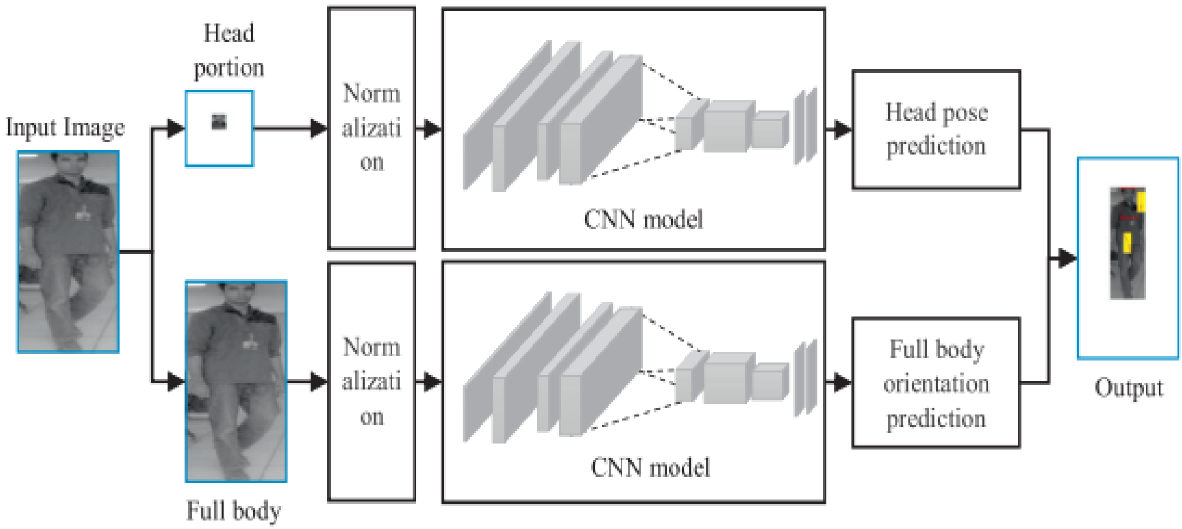


Figure 2. 5 Framework of CNN model proposed by the authors [159].

Zerrouki et al. [160] presented an SVM Hidden Classifiers. Zhanyuan Huang et al. [161] used a neural network for binary classification by utilizing SVM as a classifier for comparison. Norimichi Ukita et al. [162] estimated the poses by Linear SVM. Shile Zhang et al. [163] generated variables of pedestrians from CCTV videos using posture estimation by using keypoint detection. At red-light junctions, pedestrian crossing intentions are classified by using machine learning models i.e. Random Forest (RF), SVM, Gradient boosting (GBM), and Extreme Gradient Boosting (XGBoosting). With data from three junctions, the finest model obtains 92% accuracies and an AUC value of 0.849. Violeta Ana Luz Sosa Leon et al. [164] present a method that employs infrared depth cameras to estimate the position of the body with skeletal joints in an anonymous manner. They created their dataset with 8 body orientations and claimed 90% accuracy. Adria et al. [165] claimed that they have described the first deep learning model for calculating direction directly from video data. A well-known convolutional network is enhanced to give player orientation data by tackling the task as a classification task with classes corresponding to orientation bins and implementing a cyclic loss function. Zixinget al. [166] suggested a shallow neural network classifier to quickly recognize the states of the pedestrian. The classifier is tested on the JAAD dataset and claimed an average accuracy of 81.23 %. Michael Snower et al. [167] Presented keyTrack which is a network based on transformers for binary classification to estimate the pose of multi-person. Table 1 presents the summary of existing methods with datasets and results. Bima Sena et al. [168] represented a technique for estimating human body orientation using depth data from the Kinect camera which contains three one-dimensional distance-based signals that represent the upper body's surface contours, i.e., the upper chest, upper abdomen, and lower abdomen. Instead of utilizing discrete orientations, the authors utilized Support Vector Regression (SVR) for classification and got a mean average error (MAE) of 0.0097. The propped framework is shown in Figure 2.6.

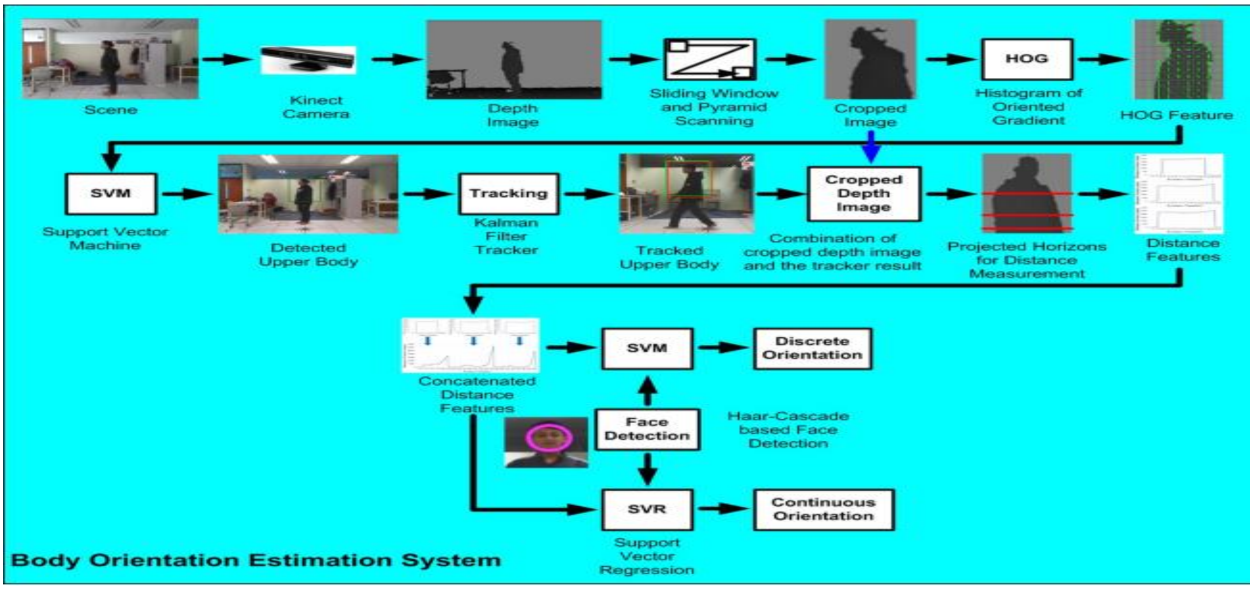


Figure 2.6 Depth image-based HBOE system [168].

Chenhen Zhao et al. [169] presented a FFNet model. Aside from camera images, the model takes into account the 2D and 3D dimensions of pedestrians as two additional inputs. Given input is based on the logical link between them and orientation. Experiments reveal that the suggested model has a 1.72 percent AOS gain over most state-of-the-art models. The model is evaluated on the KITTI dataset. Kataoka et al. [170] created a two-stream net to integrate RGB image sequences with the appropriate optical flow. An SVM classifier based on the network output was used to detect various activities such as crossing, going straight, and turning. Biao Yang et al. [171] solved the pedestrian road crossing problem as a classification problem. The authors used

Resnet3D on the JAAD dataset and claimed 89% accuracy. Safaa Dafrallah et al. [172] introduced a unique technique for estimating pedestrian orientation from a single frame. The proposed approach uses a Capsule Network methodology that has been trained on pedestrian images. Using a single camera positioned on a moving vehicle, a new pedestrian orientation dataset called SafeRoad is produced from real scenes of a city. The suggested technique is then evaluated against the TUD Multiview Pedestrian and Daimler datasets and claimed 83.5% accuracy. Karam M. Abrughalieh et al. [173] solved the orientation estimation as a classification task by dividing the angles from 0 to 360. They created a CNN model and used a dataset created from different publically available dataset of pedestrian orientation estimation.

D. Heo et al [174] adopted a teacher-student learning system to tackle the issue of a small dataset with body orientation estimation. They trained a teacher network with labeled data and then use this network to produce labels for an unlabeled dataset with which the student network is trained. They also turned the challenge into a classification problem by discretizing the output orientation into 45-degree bins. Figure 2.7 shows the orientation bins of the TUD dataset.

Figure 2.7 Eight orientation bins of the TUD dataset from 0 to 315 [174].

Table 2. 1 Summary of existing methods, datasets, and results

Ref.	Year	Methodology	Dataset	Results
[140]	2022	D-MTSMT, a multi-sensor tracking	DukeMTMC	ACC=84.79%
[141]	2022	PedRecNet	MEBOW	ACC=77.1% MAE=16.65
[125]	2021	PoseAug, data Augmentation model	MPI-INP-3DHP	ACC=88.6%

[126]	2021	AdvMix, data Augmentation using two Neural Networks	MPII-C	ACC=90.5%
[127]	2021	EFT, data augmentation	3DWP	ACC=54.2%
[92]	2021	TFpose, ResNet-18 feature extractor	COCO MPII	ACC=70.9% ACC=90.4%
[146]	2021	SimPoE, Current frame as input, next frame as estimated output	Human.6M In-house motion dataset	ACC=57.7% ACC=21.6%
[148]	2021	FrankMocap, Feature extraction	3DPW	ACC=60.0%
[158]	2021	ResNet, raw inertial measurement unit (IMU), DNN	RONIN	4.8m(distance)
[163]	2021	RF,SVM,GBM, XGBoost Classifiers	Images from CCTV	ACC=92%
[164]	2021	F-Formation. Automatic classification	The dataset created by the authors	ACC=90%
[167]	2021	KayTrack, transformer-based network, a binary classifier	PoseTrack2017	ACC=74.0%
[171]	2021	Resnet3D	JAAD	ACC=89%
[172]	2021	CapsNet	TUD, SafeRoad	ACC=83.5%
[174]	2021	CapsNet	TUD	ACC=93.48%
[175]	2021	DCPose, Temporal cues for feature extraction	PoseTrack2017 PoseTrack2018	ACC=79.2% ACC=80.9%
[176]	2021	DetTrack, ResNet-101	PoseTrack2017	ACC=74.1%
[166]	2020	Extended N/NC method to C/NC/LONG classifier	JAAD	ACC=83%
[177]	2020	CapsNet	SafeRoad	ACC=78.95%

2.7. Dataset description

The dataset named big dataset for body orientation (BDBO) [159] is found in the literature. The dataset is not available publically and is given by the authors on request. The dataset contains 34,989 images of the full body. Unique images that are collected are 17,609. There are 8 different classes. Images are taken from 8 angles 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°. Figure 2.8 illustrates the sample images of BDOS at every angle from 0° to 315°. Table 2.2 shows the description of the used dataset.

Figure 2.8 Sample Images of BDOB Dataset At Different Angles [159].

Table 2.2 Description of BDBO with classes and number of images

No.	Class	Number of images
1	0°	3981
2	45°	4313
3	90°	4172
4	135°	4639
5	180°	4588
6	225°	4640
7	270°	4113
8	315°	4543

There are many datasets found in the literature. Most of them are available publically on Kaggle.com and some of them are created by the authors. Table 2.3 shows the summary of publically available datasets.

Table 2.3 Summary of publically available datasets

Ref.	Year	Dataset	Images/Videos	Description
[178]	2018	3DWP	60 videos	It consists of 60 video sequences with 18 models with the variation of cloths
[177]	2020	SafeRoad	5160 images	It consists of 5160 images of pedestrians with 4 orientations i.e. right, left, front, back
[179]	2014	DukeMTMC	36,411 images	It consists of 36411 images of pedestrians having 1812 identities
[180]	2015	COCO	328k images	The dataset contains 328 thousand images having 80 categories and 250,000 human key points.
[181]	2017	PoseTrack	356 videos	It consists of 356 videos and 276k annotations of human body poses
[182]	2016	JAAD	346 videos	The dataset is collected during the drive of 240 hours.

2.8. Performance evaluation

The performance of the proposed technique is tested using different formulas. The confusion matrix is based on four parameters i.e., True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP). These are the common method to explore the results. Table 2.4 shows some of the performance measures.

Table 2. 4 Performance Measures

Methods	Formulas
Accuracy (ACC)	$\frac{TN + TP}{FP + TP + FN + TN}$
TRP/Sensitivity (SE)	$\frac{TP}{FN + TP}$

PRC/Positive Prediction (PPV)	$\frac{TP}{FP + TP}$
Negative Prediction value(NPV)	$\frac{TN}{TN + FN}$
FNR	$\frac{FN}{TP + FN}$
FPR	$\frac{FP}{TN + FP}$

2.9. Summary

In this chapter, the related work is presented about pedestrian orientation estimation. The best results are gathered and presented from the existing work of researchers. The existing work shows how methodologies can be designed from a different perception and increase the results. After this, a dataset is also described which is not available publically. The proposed methodology is implemented on the dataset for verification. Finally, performance measures with their formulas are presented for the evaluation of research work.

Chapter 3

Proposed Methodology

3. Introduction

In this chapter, the proposed methodology is presented with its efficient working and results. The objective of the proposed technique is to enhance the results for the estimation of pedestrian orientation. To achieve the objective, different techniques of image processing are applied in each phase and the best one is chosen for the dataset. A CNN named BlackNet is proposed and used along with pre-trained networks. Whale Optimization Algorithm is used for optimization to select the best features for classification. On the optimized features, classification is performed with SVM and KNN. A Block diagram of the proposed methodology and proposed model architecture with a description is also given in the section.

3.1. Methodology

The proposed work is performed after passing through different levels of data pre-processing, feature extraction, feature fusion, feature selection, and classification. Pre-processing is performed on the acquired data of pedestrian orientation for image enhancement. For this purpose, sharpening is performed on the dataset. On the sharp images, SRGAN-VGG54 is used to increase the resolution of the dataset images. In the phase of feature extraction, two CNNs are used to train the dataset. The one pre-trained models are VGG-19 and the second one is the proposed network model named as BlackNet. Extracted features from these two CNNs are then fused with serial-based fusion and a single fused feature vector is generated. This feature vector is then passed to the next phase of feature selection. For this purpose, Whale Optimization Algorithm (WOA) is applied to the fused feature vector. Testing is performed on selecting the required features using 5 folds and 10 folds on both the KNN classifier and SVM classifier. Because these two classifiers perform well and give higher accuracy as compared to the other classifiers. The results are gathered with other parameters of performance like accuracy, precision, F1 rate, recall, prediction speed, and training speed. Within these parameters, the confusion matrix is also saved for the best results. With the confusion matrix, graphs are also generated for the illustration of best accuracies on each classifier and accuracy comparison between state-of-the-art classifiers. Within this, graphs for features vs training time and features vs accuracy are also generated for better illustration. Following figure 3.1 shows the framework for the proposed methodology.

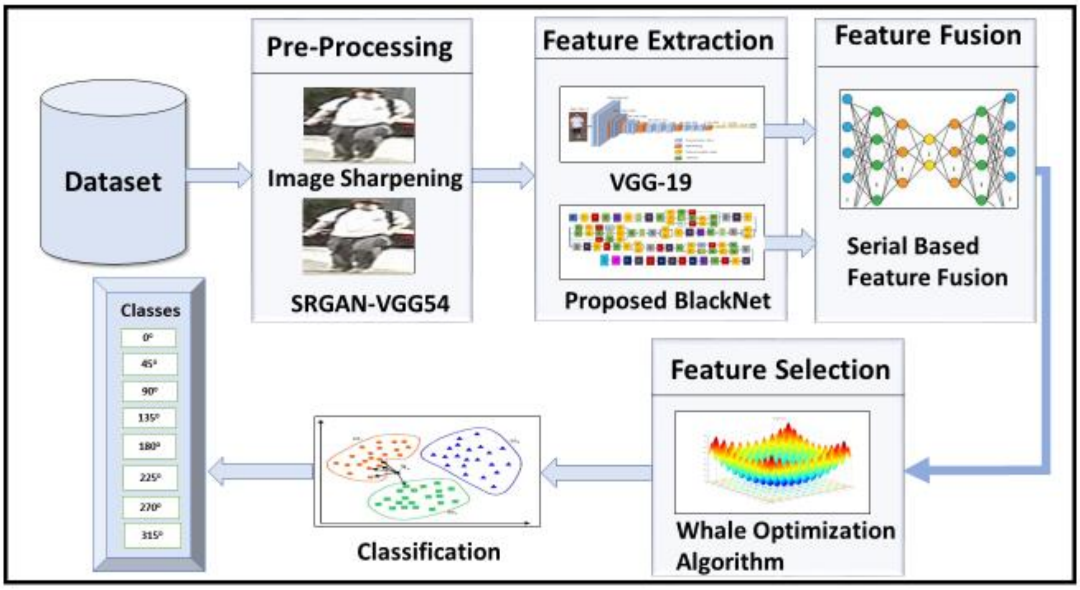


Figure 3.1 Block diagram of the proposed methodology.

3.2. Data acquisition

The dataset is not available publically. The related dataset is found in the literature. To apply the proposed methodology to the related dataset called BDBO, the dataset is taken from the authors on request. The data set contains 8 classes including 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°. After getting the dataset from the authors, it is passed to the next phase of pre-processing.

3.3. Image pre-processing

To enhance the images, sharpening is a technique that enhances the digital images by improving the definition of the edges of the image. The images with poor edges are very dull usually. There is not much of a contrast between the background and the edges. Image sharpening is applied to the images of the dataset. The sharp images are then passed to SRGAN-VGG54 which is used to increase the resolution. SRGAN uses the GAN nature and upscales the image. The purpose of increasing the resolution is to get better accuracy because the high resolution gives high accuracy. The resolution of the original images is 64×64. After image enhancement, it became 128×128 which is the input size of the dataset for the next phase. Figure 3.2 shows the results of preprocessing in which original image, sharp image, and image with an increased resolution by SRGAN-VGG54, and Figure 3.3 shows the pre-processed sample images of every class of the dataset.

Figure 3.2 Result of preprocessing original, sharp, increased resolution image

Figure 3. 3 Illustration of pre-processed sample images of every class of a dataset

3.4. Feature extraction

After applying enhancement techniques to the images of the BDBO, the features are extracted for dimensionality reduction. Feature extraction is the phase that helps to reduce the amount of unnecessary and less useful data from the dataset. In the proposed work, a pre-trained model VGG-19 is used along with the proposed model BlackNet for feature extraction. The used networks are also trained on the dataset.

3.4.1. Proposed BlackNet

The BlackNet is a proposed CNN model which is designed to enhance the results by extracting features according to the requirements. Through BlackNet, 2048 features are achieved from the fc1 layer which is then used in further phases for better performance. The proposed BlackNet has 88 layers with 97 connections. The proposed net contains 27 convolutional layers, 7 leaky ReLu layers, 18 batch normalization, 8 ReLu layers, and 3 pooling layers, 8 dropout layers, 6 additional layers, 3 fully connected layers, 3 layer normalization, 1 softmax, and 1 output layer. In the input layer, the input image size is 227×227×3 with 'zero center' normalization. It will then goes to the convolutional layer with [4 4] stride and [0 0 0 0] padding. Followed by a ReLu layer. Next is a batch normalization layer with 96 channels. A max-pooling layer 3×3 max pooling with stride [2 2] and padding [0 0 0 0]. Next is 2 Grouped convolution layer having 2 groups of 128 5×5×48 convolutions with stride [1 1] and padding [2 2 2 2]. A leaky ReLu with a scale of 0.01 and a dropout layer with a 50% dropout. Next is a Batch normalization with 256 channels. It will then break into 3 convolutional layers as more convolution layers convolve the number of images speedily. These convolutional layers have 3 batch normalization layers with 256 channels. A layer Normalization with 64 channels and a ReLu layer. These 3 convolutional layers are then fused to a single layer using an Additional layer having 3 inputs. A dropout layer with 50% dropout. Then in the next convolution layer, the filter size is updated with 64 3×3×64 convolutions with stride [1 1] and the same padding. The convolution layer is then broken into 2 convolutional layers that are then fused with an additional layer. This layer then proceeds to the leaky ReLu, batch normalization, 3×3 max pooling with stride [2 2], and padding [0 0 0 0]. This batch normalization layer is then beaked into 3 convolutions with 384 1×1×384 convolutions with stride [1 1] and padding [1 1] that are proceeding towards layer normalization, Batch normalization. These 3 branches are then fused with an additional layer. Batch normalization again is beaked to 2 convolution layers proceeding towards layer normalization, and batch normalization. These two branches are then fused with an additional layer proceeding towards 2 dropout layers fully connected layers with 2048 fully connected layers. After a dropout layer with 50% dropout, the fully connected layer has 100 fully connected layers and proceeds towards the softmax layer and a classification layer at the end. Figure 3.3

shows the blocked architecture of the proposed net. While Table 3.1 presents the description of each layer with layer number.

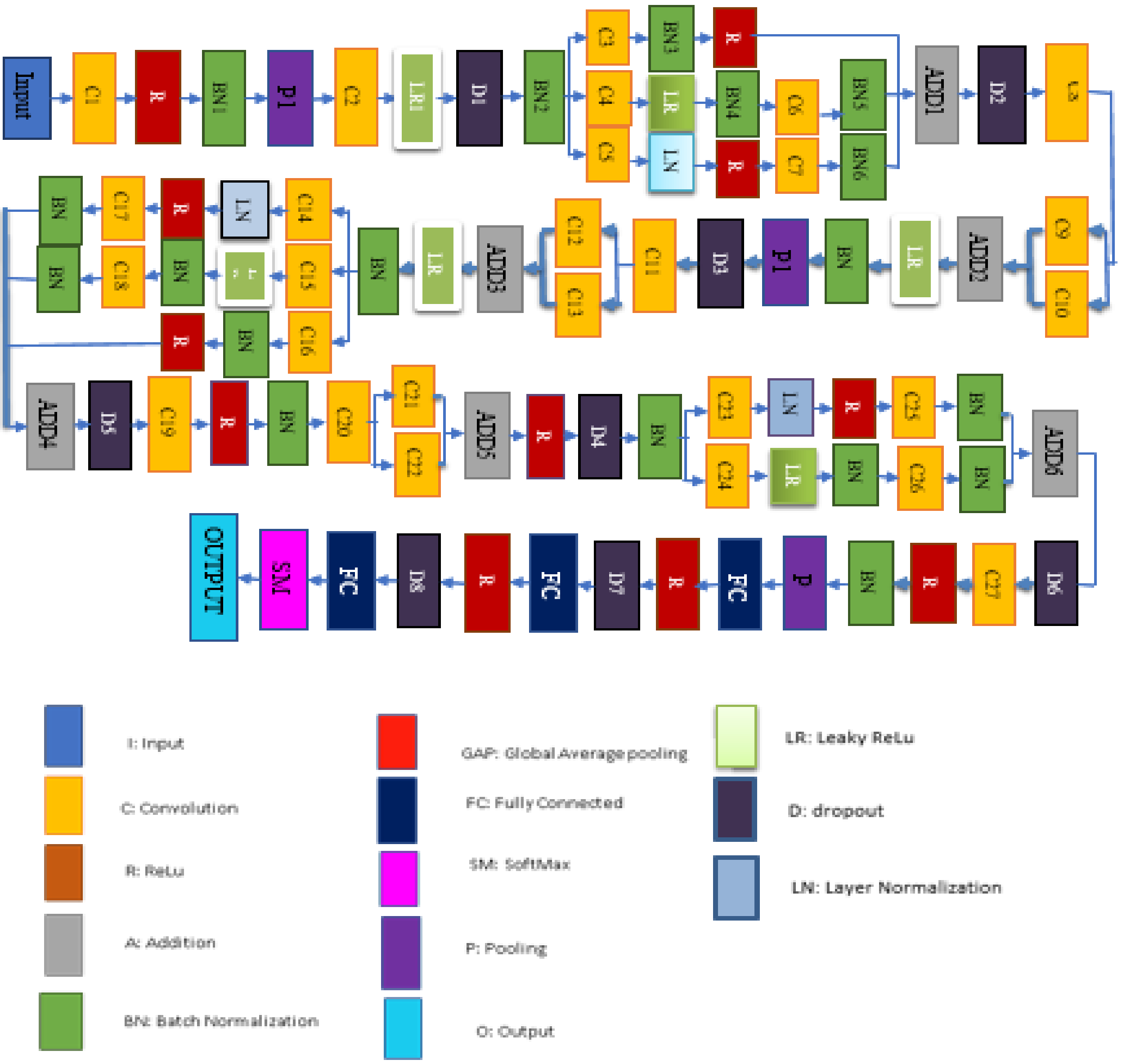


Figure 3. 4 Blocked architecture of the proposed net

Table 3.1 Architecture of Proposed Net

No.	Layer	Type	Description
1	'data'	Image Input	227×227×3 images with 'zerocenter' normalization
2	'C1'	Convolution	96 11×11×3 convolutions with stride [4 4] and padding [0 0 0 0]
3	'relu1'	ReLU	ReLU

4	'BN1'	Batch Normalization	Batch normalization with 96 channels
5	'pool1'	Max Pooling	3×3 max pooling with stride [2 2] and padding [0 0 0 0]
6	'C2'	Grouped Convolution	2 groups of 128 5×5×48 convolutions with stride [1 1] and padding [2 2 2 2]
7	'leakyrelu_1'	Leaky ReLU	Leaky ReLU with scale 0.01
8	'dropout_1	Dropout	50% dropout
9	'BN2'	Batch Normalization	Batch normalization with 256 channels
10	'C3'	Convolution	64 1×1×256 convolutions with stride [1 1] and padding 'same'
11	'leakyrelu_4'	Leaky ReLU	Leaky ReLU with scale 0.01
12	'C4'	Convolution	64 3×3×256 convolutions with stride [1 1] and padding 'same'
13	'C5'	Convolution	256 1×1×256 convolutions with stride [1 1] and padding 'same'
14	'BN3'	Batch Normalization	Batch normalization with 64 channels
15	'C6'	Convolution	256 3×3×64 convolutions with stride [1 1] and padding 'same'
16	'BN4'	Batch Normalization	Batch normalization with 256 channels
17	'layernorm_3_1'	Layer Normalization	Layer normalization with 64 channels
18	'relu_4_1'	ReLU	ReLU
19	'C7'	convolutions	with stride [1 1] and padding 'same'
20	'BN5'	Batch Normalization	Batch normalization with 256 channels
21	'BN6'	Batch Normalization	Batch normalization with 256 channels
22	'relu_3'	ReLU	ReLU
23	'addition_1_1'	Addition	Element-wise addition of 3 inputs
24	'dropout'	Dropout	50% dropout
25	'C8'	Convolution	256 3×3×256 convolutions with stride [1 1] and padding 'same'
26	'C9'	Convolution	256 1×1×256 convolutions with stride [1 1] and padding 'same'
27	'C10'	Convolution	256 3×3×256 convolutions with stride [1 1] and padding 'same'
28	'addition_1'	Addition	Element-wise addition of 2 inputs
29	'leakyrelu_3'	Leaky ReLU	Leaky ReLU with scale 0.01
30	'BN7'	Batch Normalization	Batch normalization with 256 channels
31	'pool2	Max Pooling	3×3 max pooling with stride [2 2] and padding [0 0 0 0]

32	'dropout_2'	Dropout	50% dropout
33	'C11'	Convolution	384 3×3×256 convolutions with stride [1 1] and padding [1 1 1 1]
34	'C12'	Convolution	384 1×1×384 convolutions with stride [1 1] and padding 'same'
35	'C13'	Convolution	384 3×3×384 convolutions with stride [1 1] and padding 'same'
36	'addition_2'	Addition	Element-wise addition of 2 inputs
37	'leakyrelu_2'	Leaky ReLU	Leaky ReLU with scale 0.01
38	'BN8'	Batch Normalization	Batch normalization with 384 channels
39	'C14'	Convolution	64 3×3×384 convolutions with stride [1 1] and padding 'same'
40	'C15'	Convolution	64 1×1×384 convolutions with stride [1 1] and padding 'same'
41	'leakyrelu_5'	Leaky ReLU	Leaky ReLU with scale 0.01
42	'BN9'	Batch Normalization	Batch normalization with 64 channels
43	'layernorm_3_2'	Layer Normalization	Layer normalization with 64 channels
44	'C16'	Convolution	384 3×3×64 convolutions with stride [1 1] and padding 'same'
45	'BN10'	Batch Normalization	Batch normalization with 384 channels
46	'relu_4_2'	ReLU	ReLU
47	'C17'	Convolution	384 1×1×64 convolutions with stride [1 1] and padding 'same'
48	'BN11'	Batch Normalization	Batch normalization with 384 channels
49	'C18'	Convolution	384 1×1×384 convolutions with stride [1 1] and padding 'same'
50	'BN12'	Batch Normalization	Batch normalization with 384 channels
51	'relu_4'	ReLU	ReLU
52	'addition_1_2'	Addition	Element-wise addition of 3 inputs
53	'dropout_3'	Dropout	50% dropout
54	'C19'	Grouped Convolution	2 groups of 192 3×3×192 convolutions with stride [1 1] and padding [1 1 1 1]
55	'relu_1'	ReLU	ReLU
56	'BN13'	Batch Normalization	Batch normalization with 384 channels
57	'C20'	Convolution	384 5×5×384 convolutions with stride [1 1] and padding 'same'
58	'C21'	Convolution	384 1×1×384 convolutions with stride [1 1] and padding 'same'

59	'C22'	Convolution	384 3×3×384 convolutions with stride [1 1] and padding 'same'
60	'addition_3'	Addition	Element-wise addition of 2 inputs
61	'relu_5'	ReLU	ReLU
62	'dropout_4'	Dropout	50% dropout
63	'BN14'	Batch Normalization	Batch normalization with 384 channels
64	'C23'	Convolution	64 1×1×384 convolutions with stride [1 1] and padding 'same'
65	'C24'	Convolution	64 3×3×384 convolutions with stride [1 1] and padding 'same'
66	'leakyrelu_6'	Leaky ReLU	Leaky ReLU with scale 0.01
67	'BN15'	Batch Normalization	Batch normalization with 64 channels
68	'C25'	Convolution	384 3×3×64 convolutions with stride [1 1] and padding 'same'
69	'layernorm_3_3'	Layer Normalization	Layer normalization with 64 channels
70	'relu_4_3'	ReLU	ReLU
71	'C26'	Convolution	384 1×1×64 convolutions with stride [1 1] and padding 'same'
72	'BN16'	Batch Normalization	Batch normalization with 384 channels
73	'BN17'	Batch Normalization	Batch normalization with 384 channels
74	'addition_1_3'	Addition	Element-wise addition of 2 inputs
75	'dropout_5'	Dropout	50% dropout
76	'C27'	Grouped Convolution	2 groups of 128 3×3×192 convolutions with stride [1 1] and padding [1 1 1 1]
77	'relu_2'	ReLu	ReLu
78	'BN18'	Batch Normalization	Batch normalization with 256 channels
79	'pool5'	Max Pooling	3×3 max pooling with stride [2 2] and padding [0 0 0 0]
80	'relu_6'	ReLu	ReLu
81	'fc_1'	Fully Connected	2048 fully connected layer
82	'relu_7'	ReLu	ReLu
83	'drop6'	Dropout	50% dropout
84	'fc_2'	Fully Connected	2048 fully connected layer

85	'drop7'	Dropout	50% dropout
86	'fc_3'	Fully Connected	100 fully connected layer
87	'softmax	Softmax	softmax
88	'class output	Classification	Output cross entropy with 'apple' and 99 other classes

3.4.2. Pre-trained VGG-19

The VGG-19 is a pre-trained network with 47 layers. There is one input layer with 224×224×3, 16 convolution layers with a 3×3 filter in each layer, stride [1 1], and padding [1 1 1 1]. VGG-19 has max Pooling of 2×2 with stride [2 2] and padding [1 1 1 1], 18 ReLu layers. It has 3 fully connected layers and 2 dropout layers with a 50% dropout. The network is pre-trained on ImageNet. VGG-19 is trained on the BDBO dataset. The resulting parameter is 34989×4096 such that 34989 are the images of the dataset and 4096 are the extracted features. Figure 3.4 shows the architecture of VGG-19 CNN.

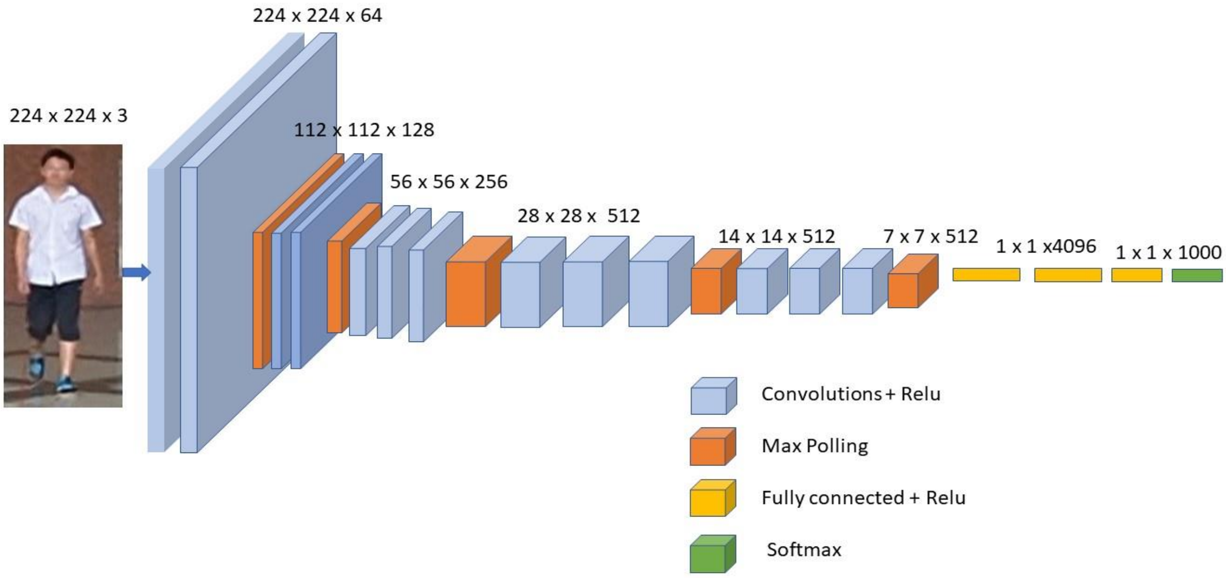


Figure 3.5 Architecture of VGG-19

3.5. Feature fusion and Feature selection

After extracting the features above two deep CNNs, the extracted features are fused using serial-based feature fusion. The 34989×4096 features are extracted from VGG-19, and the 34989×2048 features are extracted from the proposed BlackNet. 34899 represents the total number of images of the dataset used for the experiment while 2048, are the features that are extracted from the proposed BlackNet. After feature extraction, the features are then fused. Hence the resulting vector will get 34989×6144 where 6144 are the total fused features from the above two CNNs. The fusion is performed for better computational analysis from selected features that are optimal. Feature fusion is expressed mathematically as:

$$\int_{X}^{max} X = \sum_{t=1}^{max} A + \sum_{m=1}^{max} B \tag{3.1}$$

$\sum_{t=1}^{max} V$ are the extracted features from the VGG-19.

$\sum_{m=1}^{max} B$ are the extracted features from the proposed BlackNet.

In the equation mentioned above, all the features which are extracted from two CNN models are fused and collected for further analysis.

In the step of feature selection, different unique features are selected from the fused features for getting more optimal results. In the proposed work, Whale Optimization Algorithm (WOA) is used to select the optimal features from 8192 features. Then these optimal features are passed to the next phase of Classification.

3.5.1. Whale optimization algorithm

In 2016, Mirjalili introduced the Whale Optimization Algorithm (WOA) which is a metaheuristic algorithm. The WOA is based on the humpback whale, a very huge mammal on Earth that also includes the finback whale, blue whale, killer whale, and humpback whale. The rare hunting way of the humpback whale is known as bubble-net feeding which is very effective. Encircling prey, spiral updating location, and random search for prey are the three fundamental steps to knowing the WOA. The following steps provide the details of WOA.

Step 1: Encircling prey

Once the location of prey has been determined, the humpback whale would circle it. Equations (3.2) and (3.3) define the encircling prey mechanism of WOA.

$$F = |AX * (j) - X(j)| \tag{3.2}$$

$$X(j+1) = X * (j) - CF \tag{3.3}$$

Where j is the current number of iterations; $X * (j)$ presents the best vector of whale position by a long shot; $X(j)$denotes the current vector of whale position; C and A shows the coefficient of vector and can be calculated by the equations (3.4) and (3.5).

$$C = 2he_1 - h \tag{3.4}$$

$$A = 2e_2 \tag{3.5}$$

Where e_1 and e_2 are the casual numbers$(0,1)$; h is a convergent factor and linearly decreased from 2 to 0; h is calculated through the equation(3.6).

$$h = 2 - 2\, {}^{l}/_{l_{max}} \tag{3.6}$$

Where *l* denotes the current number of iterations; l_{max} denotes the maximum iterations.

Step2: Updating Spiral position

Equation (3.7) describes how a humpback whale updates its spiral position as it swims toward its prey.

$$X(j+1) = X*(j) - F_p\, e^{hl} \cos(2\pi l) \tag{3.7}$$

Where $F_{p=} |X*(j) - X(j)|$ denotes the separation between the whale and prey; h stands for constant and *l* stands for a casual number from$(0,1)$.

It's important to note that the whale must constrict to wrap its prey while swimming in a spiral in its direction. As a result, the probability R t selects the spiral model, while 1-R t selects the encircling prey mechanism. Equation (3.8) illustrates the calculating procedure:

$$X(j+1) = \begin{cases} X*(j) - CF & p < R_t \\ X*(j) - F_p\, e^{hl} \cos(2\pi l) & p \geq R_t \end{cases} \tag{3.8}$$

To lower the value of h, it is designed on the mathematical model to attack prey and approach prey so that C's range decreased along with h during the iteration process. It is said that C is inside a random value when the value of h [-h, h] decreases from 2 to 0. Furthermore, when the value of D is [-1, 1], the next place of the whale can be right now or somewhere else between its prey. When C < 1, the whale attacks its victim.

The humpback whale circles its prey while swimming in a spiral motion. The likelihood of the surrounding prey mechanism and helix position update is set to 0.5 to mimic the whale's hunting habit**.**

Step3: Random search for prey

A whale must change positions while randomly seeking prey to find it. Equations (3.9) and (3.10) shows the procedure of calculation:

$$F = |AX_{rand} - X(j)| \tag{3.9}$$

$$X(j+1) = X_{rand} - CF \tag{3.10}$$

Where X is the position vector of the whale. The whale will be forced to leave its prey to find better prey when C ≥ 1 because a searching agent will update the other whale's positions by the spontaneously searching whale. With this strategy, the algorithm's exploration capabilities may be enhanced, enabling WOA to be searched from all directions.

3.6. Classification

After collecting useful features from all the phases mentioned above, features are then passed through different classification algorithms like such as SVM and its variants including Linear SVM, Coarse Gaussian SVM, Quadratic SVM, Medium Gaussian SVM, Cubic SVM, Fine Gaussian SVM, and KNN algorithms including its variants like Fine KNN, Cubic KNN, Coarse KNN, Cosine KNN, Medium KNN, and Weighted KNN. Different classification techniques are available and used for the classification task. The techniques are Decision trees, Logistic regression, quadratic discriminant analysis, Naive Bayes, linear discriminant analysis, and artificial neural network. But we mentioned selective classification algorithms in the result section because the results on other classifiers are not much effective as compared SVM and KNN. In SVM, the function of SVM can be expressed as:

$$f(x) = \sum_{a=z}^{\infty}(\partial_a g_a V(a_z, z) + \varphi) \tag{3.11}$$

where a_z is a pattern of training, g_z denotes the labels of the classes whose range is denoted as, $g_z \in (+1, -1)$.

However, the functioning of KNN is represented as;

$$f(y) = \| N_K \cup M_i \| / L \tag{3.12}$$

where N_k is the computed nearest neighbor distance, while N_K, M_I are the parameters used to calculate the distance, which must be nearest and absolute to the round-off value.

Chapter 4
Results

4. Introduction

In this section, experiments with results and graphs of the proposed methodology are presented. The experiments are performed on the BDBO dataset. The best results are highlighted which show the highest accuracy given by the classifier on a test case. For the classification task, 12 classification techniques are utilized which include SVM and KNN classifiers and their variants. The results on each classifier are compared with each other. 12 different execution and evaluation measures are calculated including ACC, Total cost, Prediction speed, Sensitivity, Specificity, PPV, NPV, Error, Training speed, Precision, F1 Rate, and Recall rate, to evaluate the proposed algorithm. All the simulations are performed and executed on MATLAB.

4.1. Analysis of results

Different test cases are performed on the dataset by choosing a different number of features. The results are generated through different classifiers and compared with different along with the confusion matrix and ROC of the best results which are obtained on each test case. Experiments are performed using 5 folds. Experiment setup presents all 5 folds experiments which are performed for SVM and KNN classifiers.

4.1.1. Experiment Setup 1 with 5 folds

In the test example, 3000 features from the available features obtained through the suggested work are used for the experiment. As compared to other classifiers KNN and SVM classifiers are effective and efficient classifiers that produce the best results. So these two classifiers with their variants are used to generate the results.

4.1.1.1. Results on 3000 features

By setting the validation into 5 folds, test case 1 is run on 3000 features. After setting these starting values, we compare the performance of SVM and KNN classifiers and the best results are bold in the table below.

Table 4.1 Results on 3000 features (5 folds)

	Sr. #	Classifier	ACC%	PS obs/sec	TT sec	PR %	RE %	F1%
SVM	1	Linear SVM	95	1450	1971	95	95	95
	2	**Cubic SVM**	**95.4**	**1500**	**1952**	**95.1**	**95**	**95.4**
	3	Quadratic SVM	95.1	1700	1960.3	95	95	95.1

	4	Medium Gaussian SVM	94.1	1522	1987	94	94	94.1
	5	Fine Gaussian SVM	93.4	1577	1990	93.1	93	93
	6	Coarse Gaussian SVM	92.5	1500	1997	92	92	92.5
KNN	7	Fine KNN	92	1490	1800	92	92	92.1
	8	Medium KNN	92.1	1498	1811	92	92	92
	9	Cubic KNN	95.1	1590	1948.9	95	95.1	95
	10	Cosine KNN	92.9	1420	1755	93	92	92
	11	Weighted KNN	92.2	1400	2580	92	92	92.2
	12	Coarse KNN	91	1357	2989	91	91	91

Through the above data, it is determined that the Cubic SVM delivers the best outcomes with a 95.4% accuracy rate when the 5 folds approach is applied to 3000 features. It was discovered to be the best overall as well as the best in this experiment.

Along with a detailed table of the result of classifiers, a confusion matrix can also help to understand the results of the best classifiers. Following Figure 4.1 is the confusion matrix of the best classifier on 3000 features.

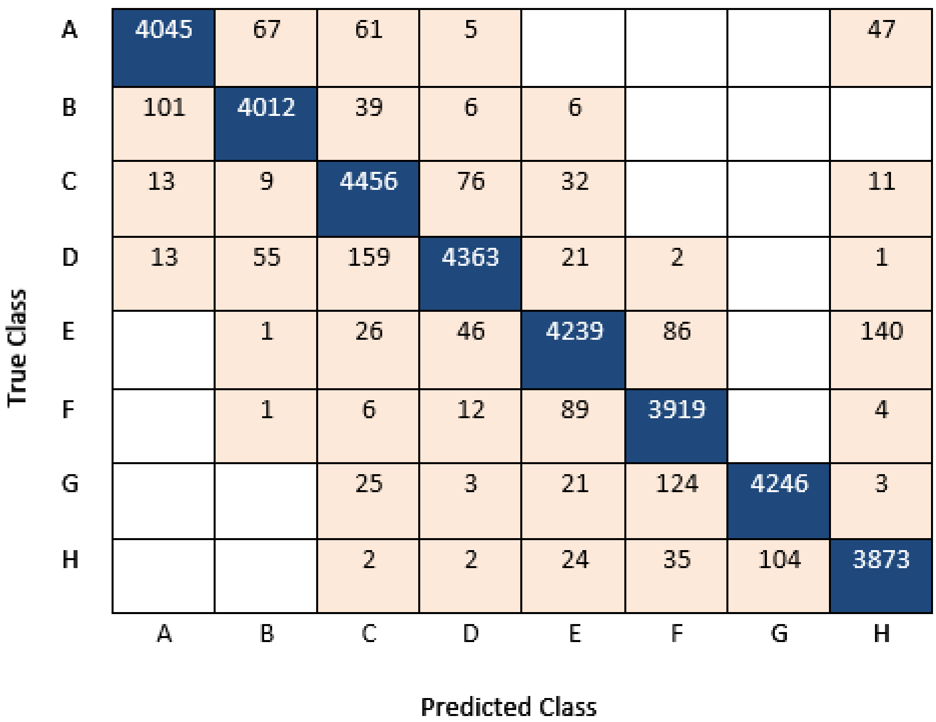


Figure 4. 1 Confusion matrix of best results on Cubic SVM on 3000 Features

With the confusion matrix on 3000 features, a graph is created to represent the accuracy of each classifier on 3000 features and 5 folds. It can be seen from the graph that Cubic SVM has attained maximum accuracy among all other classifiers.

With the confusion matrix, a graph is generated as a comparison between all the classifiers and accuracy on 3000 features. Figure 4.2 presents the graph of all the classifiers and their results.

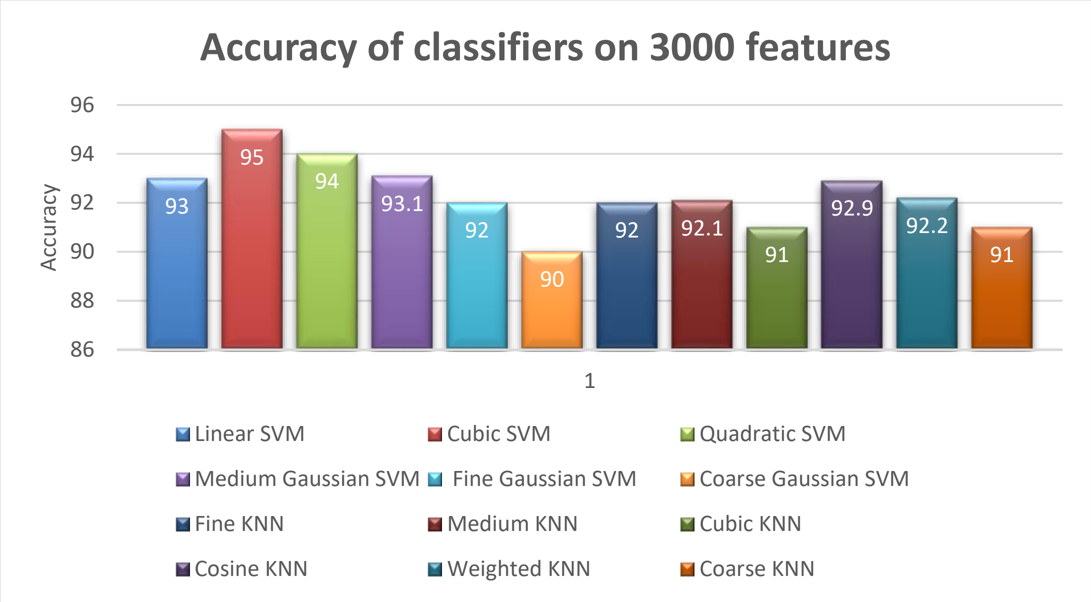


Figure 4. 2 Classifier and accuracy graph on 3000 features

4.1.1.2. Results on 2000 features

By setting the validation into 5 folds, the test case is run on 2000 features. After setting these starting values, we compare the performance of SVM and KNN classifiers and the best results are bold in the table below.

Table 4. 2 Results on 2000 features (5 folds)

	Sr. #	Classifier	ACC%	PS obs/sec	TT sec	PR %	RE %	F1%
SVM	1	Linear SVM	93	1598	1951	92.7	93	93
	2	**Cubic SVM**	**95**	**1550**	**2052**	**94.9**	**95**	**95**
	3	Quadratic SVM	94	1700	1960.3	94	94	93.9
	4	Medium Gaussian SVM	93.1	1522	1987	93	93	93.1
	5	Fine Gaussian SVM	92	1577	1900	91.1	92	92
	6	Coarse Gaussian SVM	90	1500	1878	90	90	90

KNN	7	Fine KNN	92	1490	1800	92	92	92
	8	Medium KNN	92.1	1498	1811	92	92	92
	9	Cubic KNN	91	1590	1948.9	90.8	90	90
	10	Cosine KNN	92.9	1420	1755	92.6	92	92
	11	Weighted KNN	92.2	1400	2580	92	92	92.2
	12	Coarse KNN	91	1357	2989	91	91	91

Through the above data, it is determined that the Cubic SVM delivers the best outcomes with a 95% accuracy rate when the 5 folds approach is applied to 2000 features. The confusion matrix of best results is shown below.

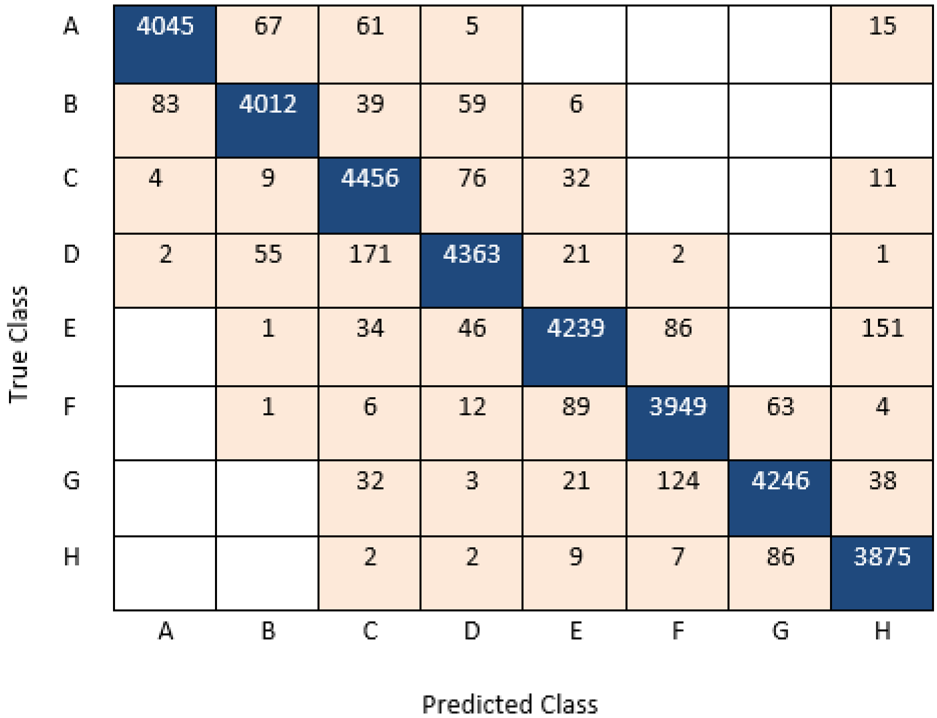


Figure 4. 3 Confusion matrix of Cubic SVM on 2000 Features

The below graph represents the accuracy against each classifier with 2000 features on 5 folds. It can be seen from the graph that Cubic SVM has the highest accuracy, the average accuracy is obtained on Linear SVM and the least accuracy is obtained on Coarse KNN.

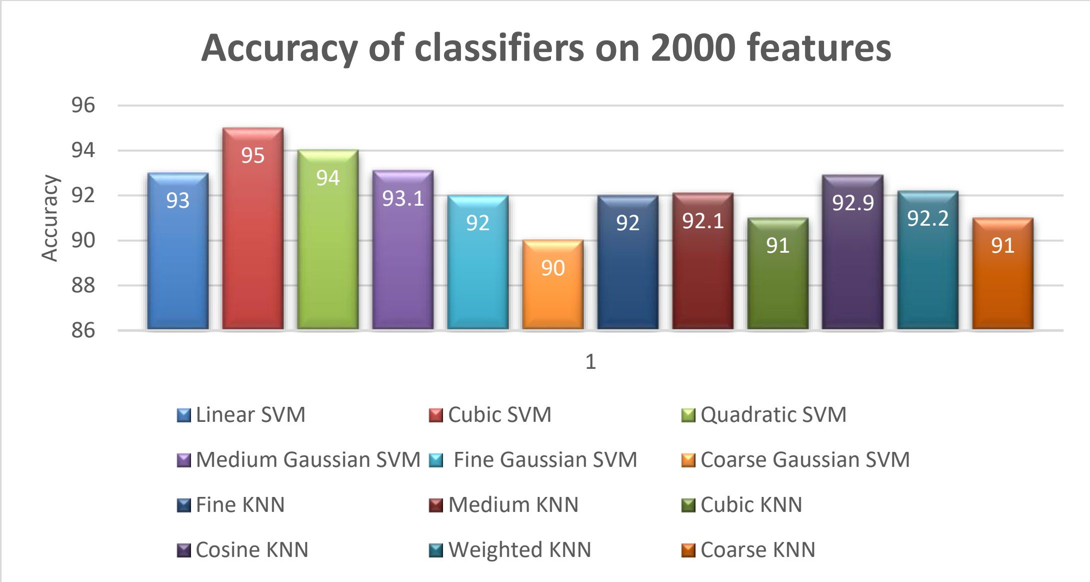


Figure 4.4 Classifier and accuracy graph on 2000 features

4.1.1.3. Results on 1500 features

By setting the validation into 5 folds, the test case is run on 1500 features. After setting these starting values, we compare the performance of SVM and KNN classifiers and the best results are bold in the table below.

Table 4.3 Results on 1500 Features (5 folds)

	Sr. #	Classifier	ACC%	PS obs/sec	TT sec	PR %	RE %	F1%
SVM	1	Linear SVM	91	1198	1951	91	91	91
	2	Cubic SVM	92	1250	1952	92	92	92
	3	**Quadratic SVM**	**94.1**	**1300**	**1960.3**	**94**	**94**	**94.1**
	4	Medium Gaussian SVM	93	922	1987	93	93	93.2
	5	Fine Gaussian SVM	91.9	877	1900	91.1	91	91.9
	6	Coarse Gaussian SVM	90.1	700	1878	90	90	90
KNN	7	Fine KNN	88	690	1800	88	88	88.1
	8	Medium KNN	92.1	898	1811	92	92	92
	9	Cubic KNN	85.1	590	1948.9	85	85.1	85
	10	Cosine KNN	92.9	420	1755	92.8	92	92.6
	11	Weighted KNN	82.2	400	2580	82.1	82	82.2
	12	Coarse KNN	86	357	2989	86	86	86

From the above table, it can be seen that Quadratic SVM is achieving the best results on 5 folds with 94.1% accuracy. The average accuracy is obtained on Cubic SVM and the least accuracy is obtained on weighted KNN. The confusion matrix of the best results is shown below which contains the resultant values for true class and predicted class.

True Class \ Predicted Class	A	B	C	D	E	F	G	H
A	4036	1	16	7	4	151	85	7
B	15	4267	35	28	16	17	19	92
C	85	2	4288	49	3	15	14	4
D	11	2	48	4226	88	33	32	26
E	4	1	7	65	3765	111	21	88
F	11	5	1	11	56	3953	85	117
G	24	41	9	12	14	82	3793	95
H	5	109		7	29	24	133	3765

Figure 4.5 Confusion matrix on Quadratic SVM on 1500 features

The following figure shows the graphical representation of the accuracies of each machine learning algorithm. By the analysis of results for each classifier on 5 folds, the graph is generated in which the highest, average, and least accuracy can be seen on each classifier.

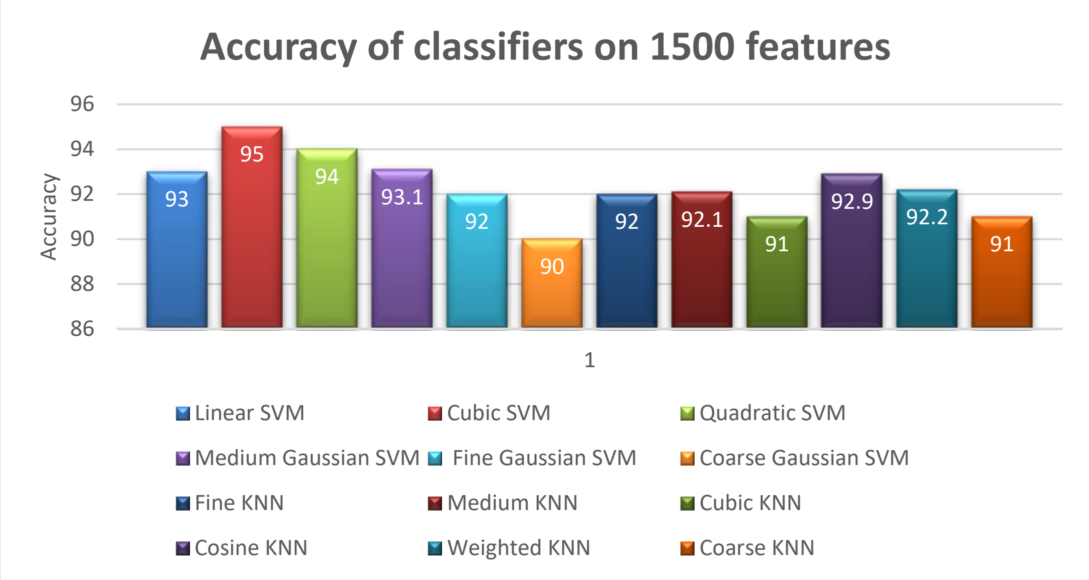


Figure 4.6 Classifier and accuracy graph on 2000 features

4.1.1.4. Results on 1000 features

By setting the validation into 5 folds, the test case is run on 1000 features. After setting these starting values, we compare the performance of SVM and KNN classifiers and the best results are bold in the table below.

Table 4.4 Results on 1000 features 5 folds

	Sr. #	Classifier	ACC%	PS obs/sec	TT sec	PR %	RE%	F1%
SVM	1	Linear SVM	78.8	121	2951	78	78.2	78.8
	2	**Cubic SVM**	**92**	**150**	**2652**	**91.9**	**92**	**92**
	3	Quadratic SVM	89.9	110	2960.3	90	89	89.5
	4	Medium Gaussian SVM	89.7	102	2817	89	89	89.2
	5	Fine Gaussian SVM	88	100	2900	88	88	88
	6	Coarse Gaussian SVM	90	111	1878	90	90	90
KNN	7	Fine KNN	91	117	2520	91	91	9
	8	Medium KNN	86	109	2811	86	85	85.9
	9	Cubic KNN	84	100	2938.9	84	84	84
	10	Cosine KNN	88.2	90	2755	88	88	88.2
	11	Weighted KNN	85	82	2080	85	85	85
	12	Coarse KNN	84.2	80	2119	84	84	84.1

From the above table, it can be seen that Cubic SVM is achieving the best results with 92% accuracy. Average accuracy is obtained on Fine Gaussian SVM which is 88% and the least accuracy is obtained on Cubic KNN which is 84%. Figure 4.7 presents the graph of the classifier and its accuracy.

True Class \ Predicted Class	A	B	C	D	E	F	G	H
A	3885	103	36	32	82	5	20	150
B	133	3815	15	107	13	74	9	6
C	21	4	4321	57	29	7	75	74
D	35	103	135	4259	20	6	70	11
E	99	6	44	19	4012	98	26	180
F	13	87	15	15	112	3779	89	3
G	24	16	128	66	39	91	4267	9
H	80	2	91	8	135	6	10	3649

Figure 4. 7 Confusion matrix on Quadratic SVM on 1000 features

The following graph shows the accuracy of each classifier on 1000 features with 5 folds. In the graphical representation highest accuracy, average accuracy, and least accuracy can be seen.

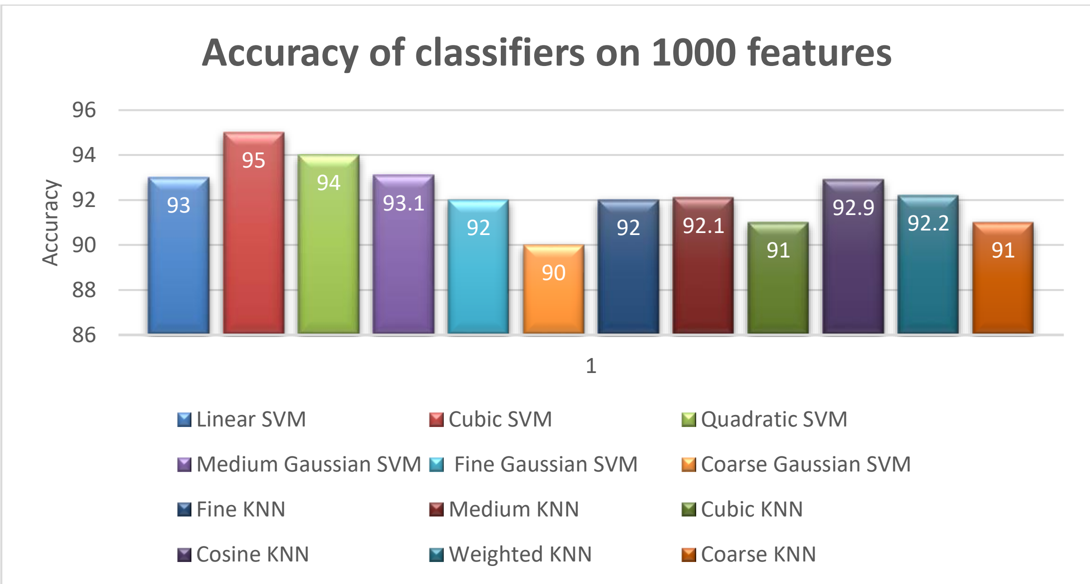


Figure 4.8 Classifier and accuracy graph on 1000 features

4.1.1.5. Results on 500 features

In test case 5**,** by setting the validation into 5 folds, the test case is run on 500 features. After setting these starting values, we compare the performance of SVM and KNN classifiers and the best results are bold in the table below.

Table 4. 5 Results on 500 (5 folds)

	Sr. #	Classifier	ACC%	PS obs/sec	TT sec	PR %	RE%	F1%
SVM	1	Linear SVM	73.6	144	1951	73	73.1	95
	2	**Cubic SVM**	**91.7**	**160**	**1644**	**91.5**	**91**	**91.5**
	3	Quadratic SVM	87.6	151	1560.3	87	87	87.2
	4	Medium Gaussian SVM	88.8	152	1387	88.2	88.1	88
	5	Fine Gaussian SVM	70.4	77	2100	70.1	70	70
	6	Coarse Gaussian SVM	70.8	78	878	70.5	70.2	70
KNN	7	Fine KNN	90	155	2300	90	90	90
	8	Medium KNN	88	149	1811	88	88	88
	9	Cubic KNN	82	115	1948.9	82	82	82
	10	Cosine KNN	68.9	100	1955	68.5	68	68.4
	11	Weighted KNN	70.2	83	2080	70.2	70	70.1
	12	Coarse KNN	75.5	90	1999	75	75.1	75

The above table shows that the Cubic SVM has given the highest accuracy on 500 features. The highest accuracy on Cubic SVM is 91.7%. The average accuracy is obtained by Quadratic SVM which is 87.6% and the Least accuracy is generated by Fine Gaussian SVM which is 70.4% Following figure shows the confusion matrix of the best results on Cubic SVM on 500 features with 5 folds. Confusion matric contains the predicted class and true class.

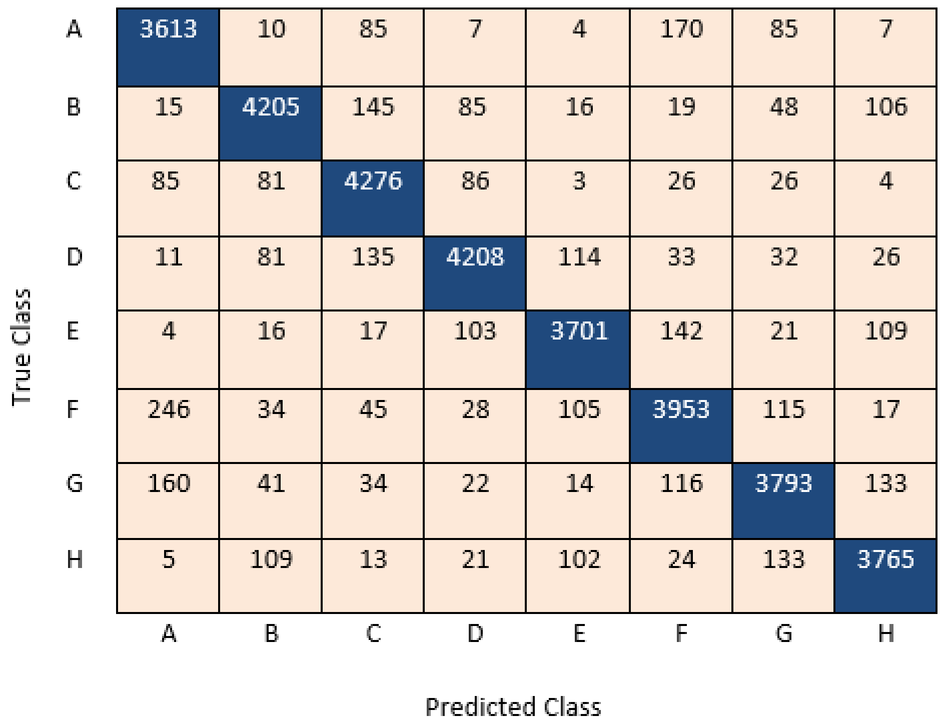


Figure 4. 9 Confusion matrix on Quadratic SVM on 500 features

The following graph shows the comparison of accuracies on each classifier on 500 features with 5 folds.

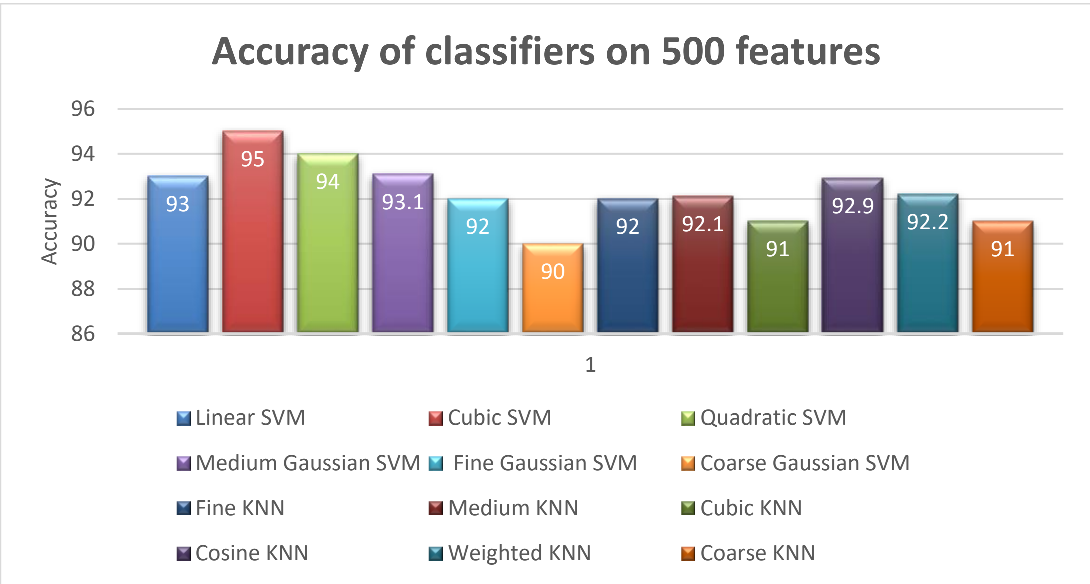


Figure 4. 10 Classifier and accuracy graph on 500 features.

The results are also measured from the perspective of training time. Less training time is considered to be good to generate output. From the following graph, the training time is the least with a low number of features.

Figure 4. 11 Features VS Training time on best Classifiers

Following is a graphical representation of the best accuracy we get from each experiment in experiment setup 1. The graph shows that results are best at a maximum number of selected features.

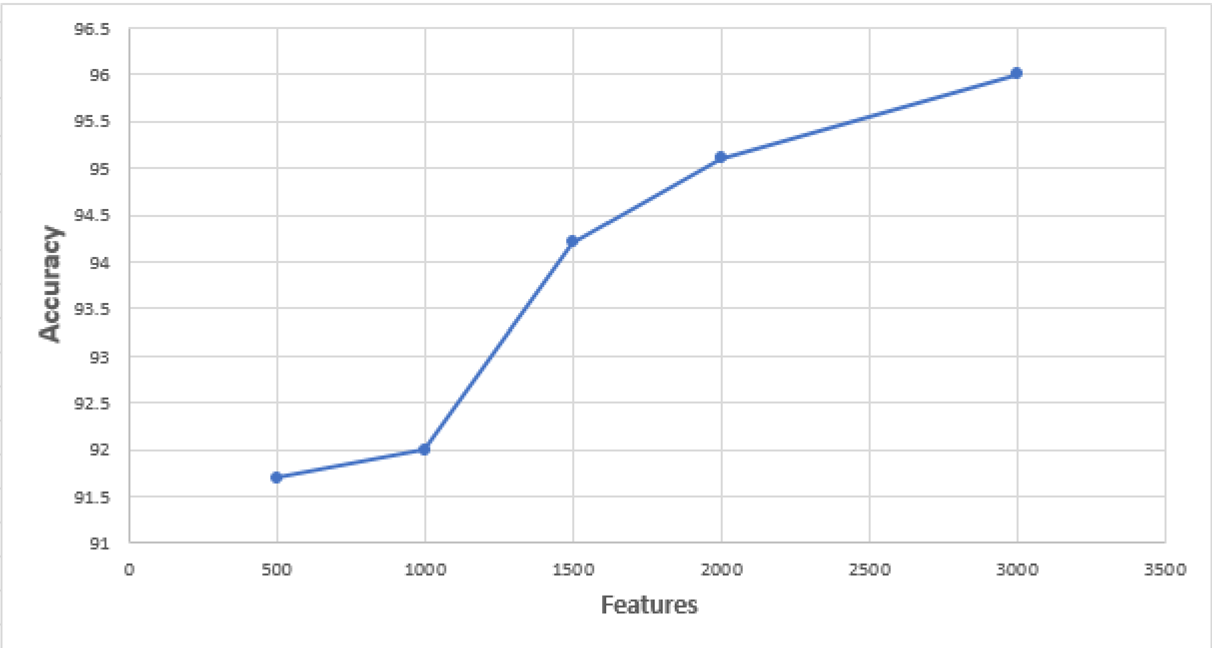


Figure 4. 12 Accuracy VS Features

4.2.2 Experiment setup 2 with 10 folds

In this test example, 3000 features from the available features obtained through the suggested work are used for the experiment with 10 folds. As compared to other classifiers KNN and SVM classifiers are effective and efficient classifiers and are producing the best results. So these two classifiers with their variants are used to generate the results on 10 folds.

4.2.2.1. Results on 3000 features

By setting the validation into 10 folds, test case 2 is run on 3000 features. After setting these starting values, we compare the performance of SVM and KNN classifiers and the best results are bold in the table below.

Table 4.6 Results on 3000 features (10 folds)

	Sr. No	Classifier	ACC%	PS obs/sec	TT sec	PR %	RE%	F1%
SVM	1	Linear SVM	95.7	1650	2071	95.2	95	95.4
	2	**Cubic SVM**	**96**	**1700**	**1952**	**96**	**96**	**96**
	3	Quadratic SVM	95.9	1800	2120	95.1	95	95.9
	4	Medium Gaussian SVM	94.9	1572	2187	94.4	94	94.1
	5	Fine Gaussian SVM	94	1577	1990	94	94	94
	6	Coarse Gaussian SVM	93.1	1500	1997	93	93	93.1
KNN	7	Fine KNN	92.5	1520	1750	92	92	92.5
	8	Medium KNN	92.1	1498	1811	92	92	92
	9	Cubic KNN	95.6	1590	1948.9	95.1	95	95.4
	10	Cosine KNN	92.9	1420	1755	93	92	92
	11	Weighted KNN	92.2	1400	2580	92	92	92.2
	12	Coarse KNN	91.9	1157	3089	91.5	91	91.9

Through the above data, it is determined that the Cubic SVM delivers the best outcomes with a 96% accuracy rate when the 10 folds approach is applied to 3000 features. It was discovered to be the best overall as well as the best in this experiment. The average accuracy on 3000 features is obtained on Coarse Gaussian SVM which is 93.1%. The least accuracy is obtained on Coarse KNN which is 91.9% Along with a detailed table of the result on classifiers, a confusion matrix can also help to understand the results of the best classifiers. Following Figure 4.13 is the confusion matrices of the best classifier on 3000 features.

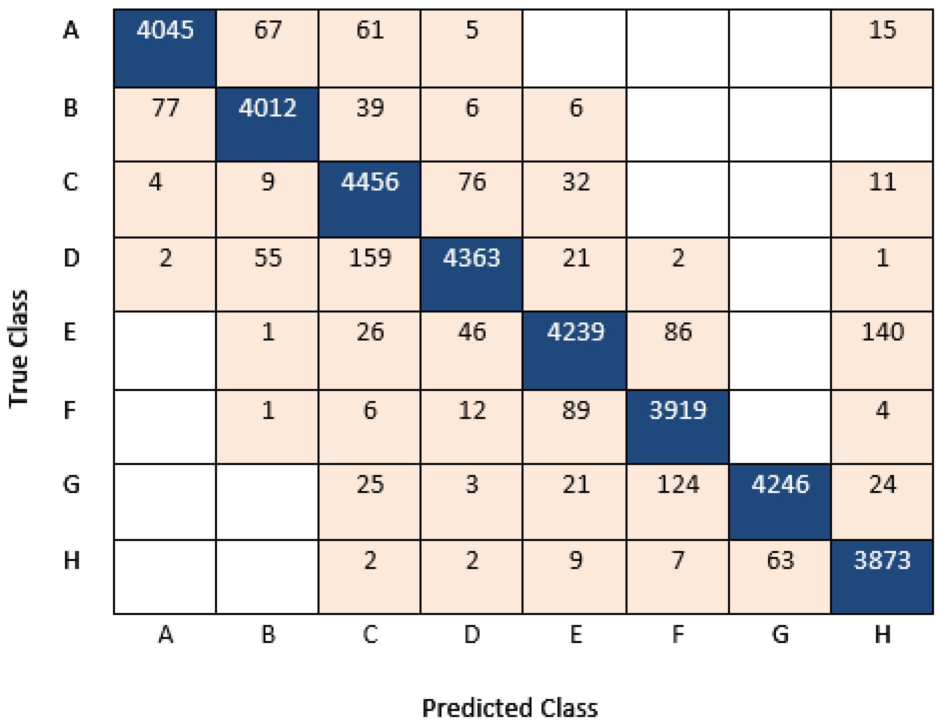


Figure 4. 13 Confusion matrix of best results on Cubic SVM on 3000 Features

With the confusion matrix, a graph is generated as a comparison between all the classifiers and accuracy on 3000 features. The following figure presents the graph of all the classifiers and their results on 10 folds with 3000 features

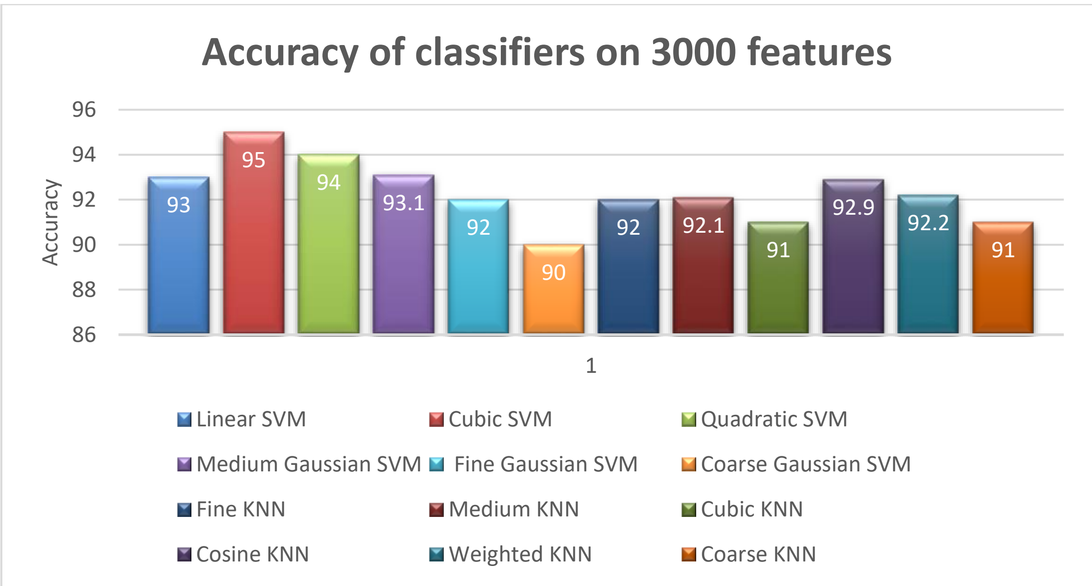


Figure 4. 14 Classifier and accuracy graph on 3000 features on 10 folds

4.2.2.2. Results on 2000 features

By setting the validation into 10 folds, the test case is run on 2000 features. After setting these starting values, we compare the performance of SVM and KNN classifiers and the best results are bold in the table below.

Table 4. 7 Results on 2000 features (10 folds)

	Sr. No	Classifier	ACC%	PS obs/sec	TT sec	PR %	RE%	F1%
SVM	1	Linear SVM	93.3	1498	1951	93.1	93	93.3
	2	**Cubic SVM**	**95.7**	**1880**	**3052**	**95**	**95**	**95.7**
	3	Quadratic SVM	94.5	1800	2660.3	94	94	94.9
	4	Medium Gaussian SVM	93.1	1522	1987	93	93	93.2
	5	Fine Gaussian SVM	92.3	1607	2200	92.1	92	92
	6	Coarse Gaussian SVM	91	1500	3878	91	91	91.
KNN	7	Fine KNN	92	1490	3800	92	92	92
	8	Medium KNN	92.1	1498	4011	92	92	92
	9	Cubic KNN	91	1590	2948.9	91	90.1	91
	10	Cosine KNN	92.9	1420	1755	93	92	92
	11	Weighted KNN	92.2	1400	2580	92	92	92.2
	12	Coarse KNN	91.9	1357	2989	91	91	91

Through the above data, it is determined that the Cubic SVM delivers the best outcomes with a 95.7% accuracy rate when the 5 folds approach is applied to 2000 features. From the table, the average accuracy and the least accuracy can be found. The average accuracy is 93.3% obtained on Linear SVM and the least accuracy is 91% obtained on Coarse Gaussian SVM and Cubic KNN. The confusion matrix of best results is shown below.

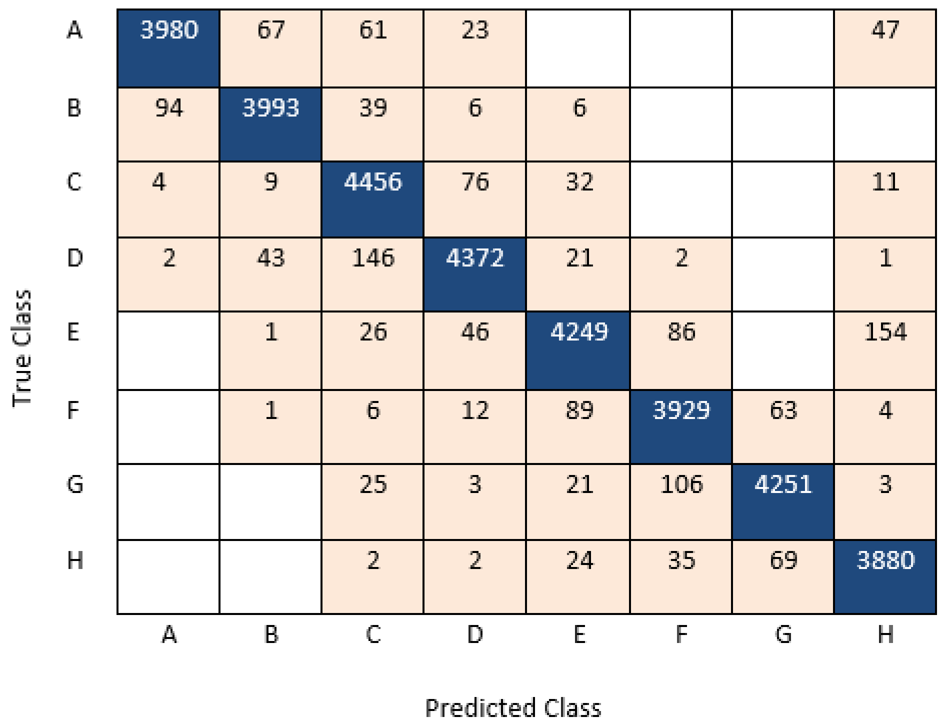


Figure 4. 15 Confusion matrix of Cubic SVM on 2000 Features on 10 folds

The following figure shows the results on each classifier with accuracy on 10 folds.

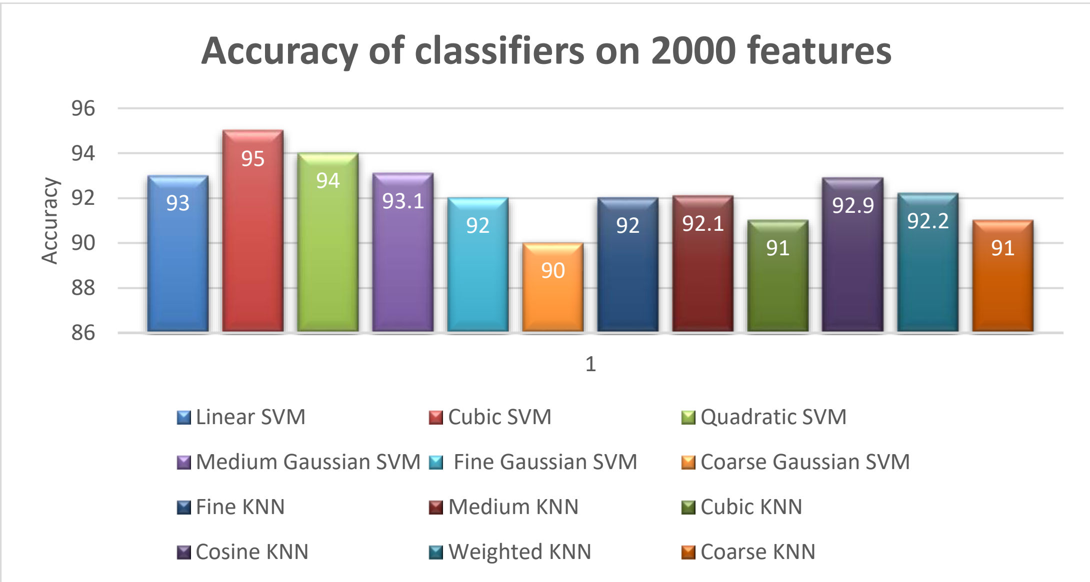


Figure 4. 16 Classifier and accuracy graph on 2000 features on 10 folds

4.2.2.3. Results on 1500 features

By setting the validation into 10 folds, the test case is run on 1500 features. After setting these starting values, we compare the performance of SVM and KNN classifiers and the best results are bold in the table below.

Table 4. 8 Results on 1500 features (10 folds)

	Sr. No	Classifier	ACC%	PS obs/sec	TT sec	PR %	RE%	F1%
SVM	1	Linear SVM	92.5	980	2451	92	92	92.5
	2	Cubic SVM	94.2	950	2152	94	94	94
	3	**Quadratic SVM**	**95**	**1100**	**1960.3**	**95**	**95**	**95**
	4	Medium Gaussian SVM	93.5	1002	1987	93	93	93.2
	5	Fine Gaussian SVM	92	977	1900	92	92	92
	6	Coarse Gaussian SVM	90.7	500	1878	90	92	92.5
KNN	7	Fine KNN	88	490	1800	88	88	88.1
	8	Medium KNN	92.9	498	2811	92.4	92	92
	9	Cubic KNN	85.1	590	1948.9	85	85.1	85
	10	Cosine KNN	92.9	420	1755	92.8	92	92.6
	11	Weighted KNN	85.2	400	3580	85.1	85	85
	12	Coarse KNN	88	398	3189	88	88	88

From the above table, it can be seen that Quadratic SVM is achieving the best results with 95% accuracy. The average accuracy is acquired on Coarse Gaussian SVM which is 90.7% and the least accuracy is 85.1% obtained on Cubic SVM. The confusion matrix of the best results is shown below.

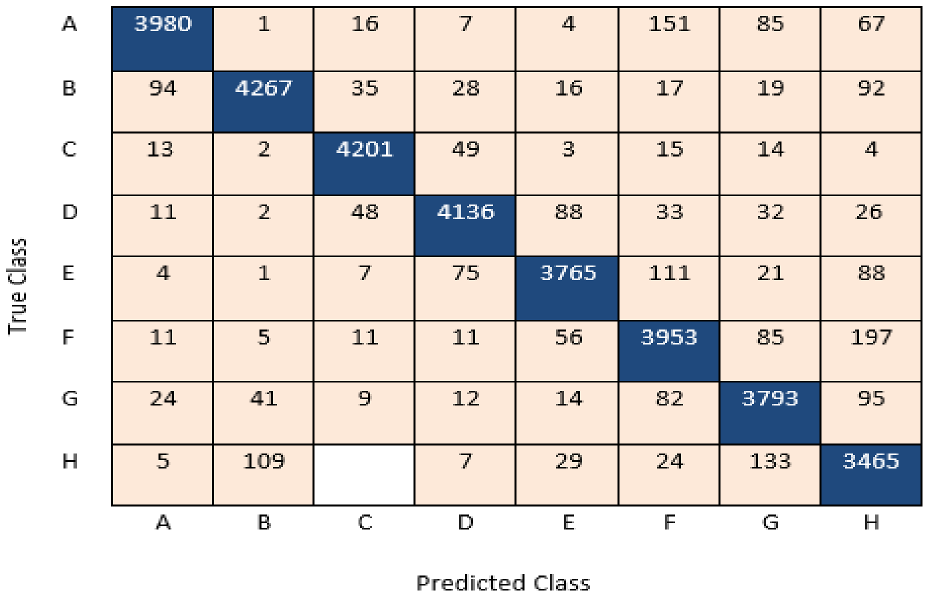


Figure 4. 17 Confusion matrix on Quadratic SVM on 1500 features

The following figure presents the comparison of accuracy against each classifier on 1500 features on 10 folds.

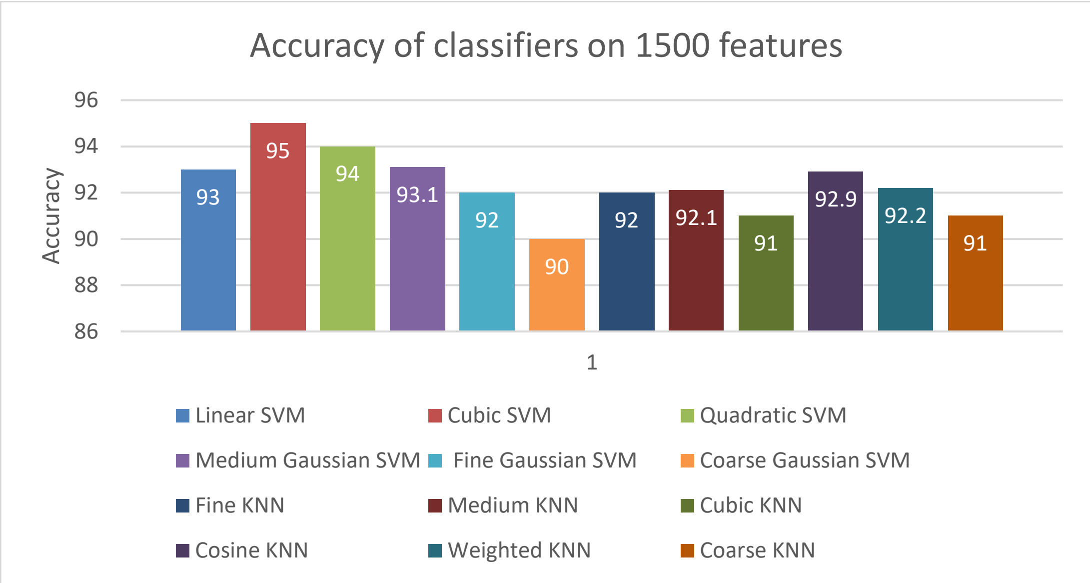


Figure 4. 18 Classifier and accuracy graph on 2000 features on 10 folds

4.2.2.4. Results on 1000 features

By setting the validation into 5 folds, the test case is run on 1000 features. After setting these starting values, we compare the performance of SVM and KNN classifiers and the best results are bold in the table below.

Table 4.9 Results on 1000 features (10 folds)

	Sr . #	Classifier	ACC%	PS obs/sec	TT sec	PR %	RE %	F1%
SVM	1	Linear SVM	85.2	161	1951	85	85	85.2
	2	**Cubic SVM**	**93.5**	**197**	**1652**	**93.2**	**93**	**93**
	3	Quadratic SVM	90	130	1000.3	90	89	89.5
	4	Medium Gaussian SVM	89.7	102	1817	89.5	89	89.2
	5	Fine Gaussian SVM	88	100	1900	88	88	88
	6	Coarse Gaussian SVM	92	111	2008	92	92	92
KNN	7	Fine KNN	91.5	117	2520	91	91	91
	8	Medium KNN	88	109	2811	88	88	88
	9	Cubic KNN	86	90	2938.9	86	86	86
	10	Cosine KNN	89.2	90	2755	89	89	88.2

	11	Weighted KNN	85.5	82	2080	85.1	85	85
	12	Coarse KNN	84.9	90	2119	84.3	84	84.4

From the above table, it can be seen that Cubic SVM is achieving the best results with 93.5% accuracy. The confusion matrix of the best result is shown below.

True Class \ Predicted Class	A	B	C	D	E	F	G	H
A	4110	103	21	32	82	5	20	150
B	133	3815	15	98	13	74	9	6
C	21	4	4321	57	29	7	75	74
D	35	103	135	4297	20	6	70	11
E	99	6	44	19	4012	98	26	210
F	13	87	15	15	112	3779	89	3
G	24	16	108	66	39	91	4267	9
H	80	2	91	8	121	6	10	3649

Figure 4. 19 Confusion matrix on Cubic SVM on 1000 features on 10 folds

The following figure shows the comparison of accuracies of each classifier on 1000 features on 10 folds.

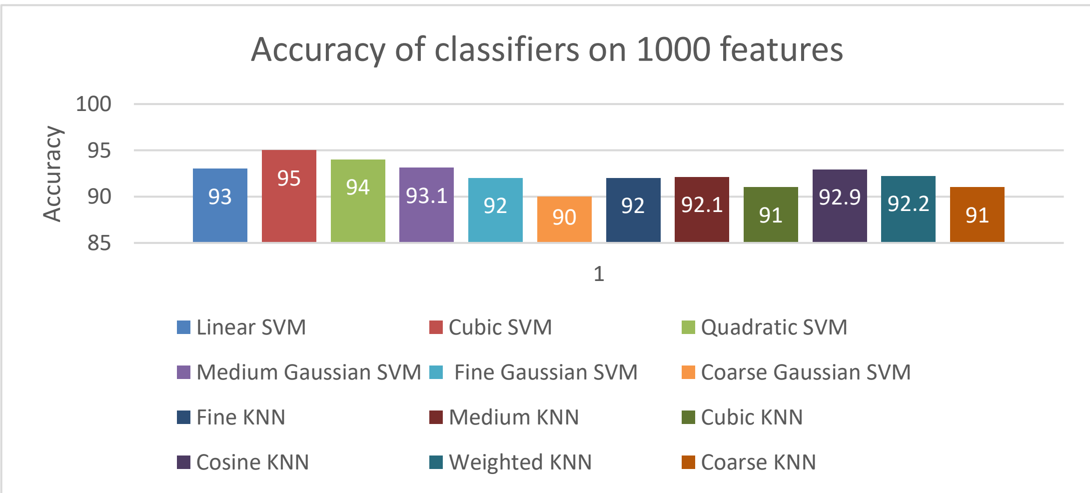


Figure 4. 20 Classifier and accuracy graph on 1000 features

4.2.2.5. Results on 500 features

In test case 5**,** by setting the validation into 5 folds, the test case is run on 500 features. After setting these starting values, we compare the performance of SVM and KNN classifiers and the best results are bold in the table below.

Table 4.10 Results on 500 features (10 folds)

	Sr. #	**Classifier**	**ACC%**	**PS obs/sec**	**TT sec**	**PR %**	**RE %**	**F1%**
SVM	1	Linear SVM	73.6	144	1951	73	73.1	73
	2	**Cubic SVM**	**92.3**	**176**	**944**	**92**	**92.3**	**92.3**
	3	Quadratic SVM	88.6	161	1560.3	88	88	88.6
	4	Medium Gaussian SVM	90.8	152	1387	90.2	90	90
	5	Fine Gaussian SVM	76.4	77	2100	76.1	76	76
	6	Coarse Gaussian SVM	74.8	78	878	74.5	74.2	74
KNN	7	Fine KNN	91.1	155	2300	91	91	90.1
	8	Medium KNN	89	149	1811	89	89	89
	9	Cubic KNN	85	115	1948.9	85	85	85
	10	Cosine KNN	74.9	100	1955	74.5	74	74.4
	11	Weighted KNN	72.2	83	2080	72.2	72	72.1
	12	Coarse KNN	77.5	90	1999	77	77.1	77.5

The above table shows that the Cubic SVM has given the highest accuracy on 500 features. The highest accuracy on cubic SVM is 92.3%. The following figure shows the confusion matrix of the best results on Cubic SVM.

True Class \ Predicted Class	A	B	C	D	E	F	G	H
A	3413	10	85	7	4	170	85	7
B	15	4205	145	85	16	19	48	98
C	41	81	4276	86	3	26	26	4
D	11	81	135	4208	114	33	32	26
E	4	16	17	103	3701	142	21	109
F	246	34	45	28	105	3953	115	17
G	160	41	34	22	14	116	3893	133
H	5	109	13	21	102	24	133	3765

Figure 4. 21 Confusion matrix on Cubic SVM on 500 features

The following figure shows the results of each classifier with 500 features and 10 folds.

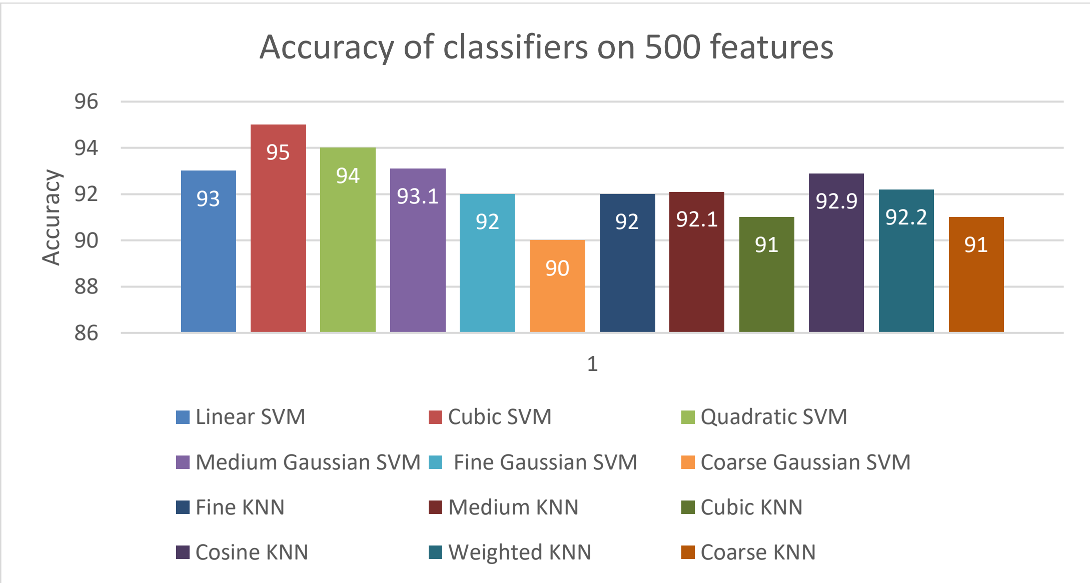


Figure 4. 22 Classifier and accuracy graph on 500 features on 10 folds

The results are also measured from the perspective of training time. Less training time is considered to be good to generate output. From the following graph, the training time is the least with a low number of features.

Figure 4. 23 Features VS Training Time on 10 folds

Following is a graphical representation of the best accuracy we get from each test case in experiment setup 2. The graph shows that results are best at a maximum number of selected features.

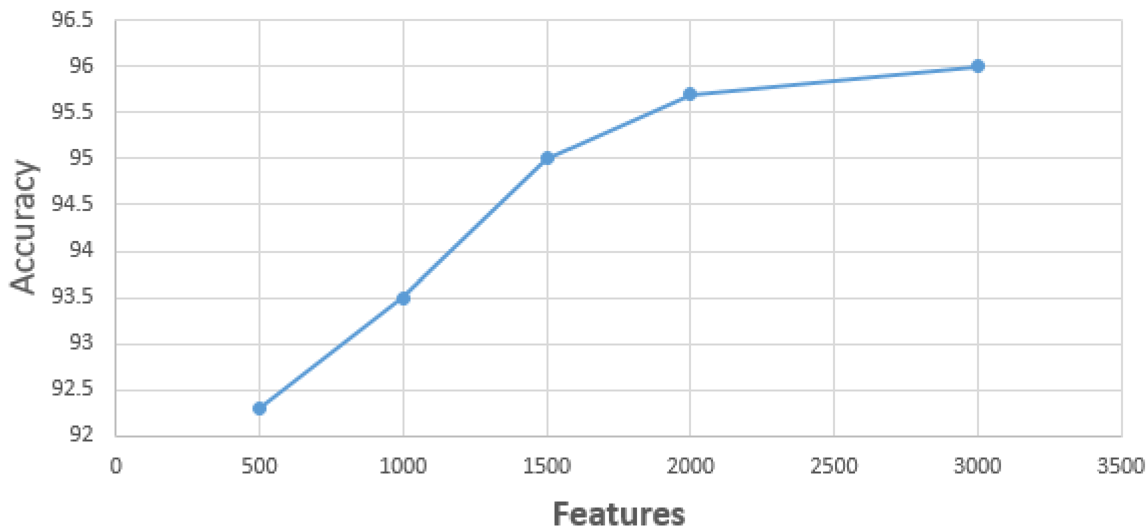


Figure 4. 24 Accuracy vs Features on 10 folds

The following graph represents the comparison between the accuracies of best classifiers on 5 folds and 10 folds. It can be seen through the graph that 10 folds have given high accuracy as compared to the 5 folds. It can also be seen that more number of features have given higher accuracy.

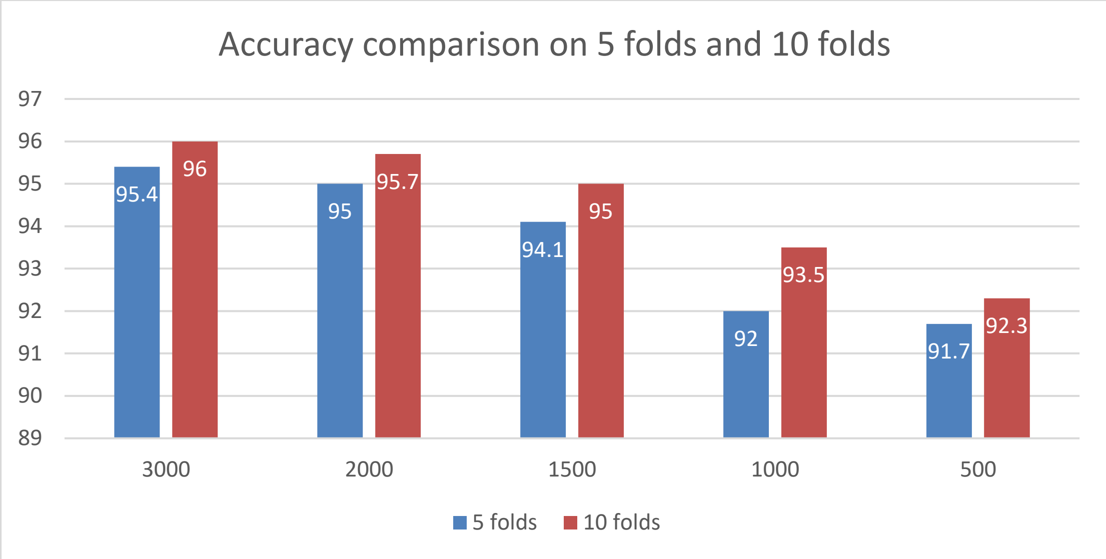


Figure 4. 25 Comparison of accuracies on 5 folds and 10 folds

4.3. Comparison of the proposed methodology with state-of-the-art techniques

Two types of comparisons are performed in this section. The first comparison is carried out with the technique that is applied to the same dataset used in this research work. The technique was proposed in 2018 on a dataset called BDBO. The dataset BDBO is the combination of the TUD-Multiview dataset, CAVIAR dataset, and images taken by the authors. The reason for creating a new dataset is that there is no big dataset available for orientation publically. To train the network on the big dataset, BDBO was created by the authors and still is not available publically. The second comparison of the proposed methodology is carried out with state-of-the-art techniques proposed for orientation estimation. The reason for performing the comparison is that the proposed technique is very general so it will perform well on other datasets too. Table 4.11 and 4.12 presents the comparison of the proposed technique.

Table 4. 11. Comparison with previous technique on BDBO (Direct Comparison)

Ref.	Technique	Dataset	Accuracy
[159]	CNN	BDBO	92%
Proposed	VGG-19+Proposed CNN	BDBO	96%

The following table presents the comparison of the proposed methodology with state-of-the-art techniques. The purpose of the comparison is to highlight the accuracy difference between different methods used for orientation estimation. This is the indirect comparison in which the proposed technique is compared with other techniques having different datasets.

Table 4. 12 Comparison of proposed technique (Indirect Comparison)

Ref.	Year	Accuracy
[140]	2022	84.79%
[141]	2022	77.1%
[171]	2021	89%
[172]	2021	83.5%
[174]	2021	93.48%
[175]	2021	79.2% 80.9%
[166]	2020	83%
[177]	2020	78.95%
[171]	2021	89%
Proposed		**96%**

4.3.1. Discussion

The above tables show the efficiency of the suggested technique. On BDBO the previous results are 92% on a CNN model and the accuracy attained by the proposed model is 96%. The obtained accuracy is also better than other classification results of different techniques.

Chapter 5

Conclusion and Future Direction

5. Conclusion

In the previous decade, a lot of work and research has been done in the field of computer vision. Human orientation estimation is one of the most worked fields of research. In this research, a novel technique is presented for better feature extraction from big data. These features than play a vital role in classification. With the increasing population, it is very difficult to maintain crowded areas without cameras. As the proposed algorithm is a machine learning model so it can estimate the orientation without any human assistance. In the Proposed work, different image processing techniques are applied. Image sharpening and SRGAN-VGG 54 are used in pre-processing. Features are extracted from a pre-trained model VGG-19 and proposed model BlackNet. Extracted features are passed to the feature selection phase for optimized features. The optimized features are then used in the classification process. Classification is performed on SVM and KNN. The highest accuracy is obtained on cubic SVM which is 96% on a maximum number of optimized features.

5.1. Future directions

A lot of techniques and methods are being proposed in recent times for estimation purposes. From the proposed study it can be seen that better accuracies can be attained by using improved and new pre-processing techniques. Pre-processing is the phase in which images are enhanced. The enhanced images play a vital role in extracting the features from the images. Within this classification is the core role in the process of attaining maximum accuracy.

References

[1] D. Zhang, Y. Wu, M. Guo, and Y. Chen, "Deep Learning Methods for 3D Human Pose Estimation under Different Supervision Paradigms: A Survey," *Electronics*, vol. 10, no. 18, p. 2267, 2021.

[2] S. N. Boualia and N. E. B. Amara, "Pose-based human activity recognition: a review," in *2019 15th international wireless communications & mobile computing conference (IWCMC)*, 2019: IEEE, pp. 1468-1475.

[3] Y. Liu, M. Shi, Q. Zhao, and X. Wang, "Point in, box out: Beyond counting persons in crowds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6469-6478.

[4] C.-B. Lin, Z. Dong, W.-K. Kuan, and Y.-F. Huang, "A framework for fall detection based on OpenPose skeleton and LSTM/GRU models," *Applied Sciences*, vol. 11, no. 1, p. 329, 2020.

[5] M. D. N. Handayani, A. H. Sadewa, A. Farmawati, and W. Rochmah, "Anthropometric prediction equations for estimating muscle mass of elderly women," *KEMAS: Jurnal Kesehatan Masyarakat*, vol. 14, no. 2, pp. 195-204, 2018.

[6] M. Fürst, S. T. Gupta, R. Schuster, O. Wasenmüller, and D. Stricker, "HPERL: 3d human pose estimation from RGB and lidar," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021: IEEE, pp. 7321-7327.

[7] B. Noh, H. Park, and H. Yeo, "Analyzing vehicle–pedestrian interactions: Combining data cube structure and predictive collision risk estimation model," *Accident Analysis & Prevention*, vol. 165, p. 106539, 2022.

[8] B. Lewandowski, D. Seichter, T. Wengefeld, L. Pfennig, H. Drumm, and H.-M. Gross, "Deep orientation: Fast and robust upper body orientation estimation for mobile robotic applications," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019: IEEE, pp. 441-448.

[9] Q. Xu, M. Chraibi, and A. Seyfried, "Anticipation in a velocity-based model for pedestrian dynamics," *Transportation research part C: emerging technologies*, vol. 133, p. 103464, 2021.

[10] R. Araya, J. J. F. i. R. Sossa-Rivera, and AI, "Automatic Detection of Gaze and Body Orientation in Elementary School Classrooms," p. 277, 2021.

[11] Y. Cai, L. Dai, H. Wang, L. Chen, and Y. Li, "A novel saliency detection algorithm based on adversarial learning model," *IEEE Transactions on Image Processing*, vol. 29, pp. 4489-4504, 2020.

[12] H. Wang, Y. Yu, Y. Cai, X. Chen, L. Chen, and Q. Liu, "A comparative study of state-of-the-art deep learning algorithms for vehicle detection," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 2, pp. 82-95, 2019.

[13] B. Liu *et al.*, "Spatiotemporal relationship reasoning for pedestrian intent prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3485-3492, 2020.

[14] Z. Fang and A. M. López, "Is the pedestrian going to cross? answering by 2d pose estimation," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018: IEEE, pp. 1271-1276.

[15] S. Shen, M. Gowda, and R. Roy Choudhury, "Closing the gaps in inertial motion tracking," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 429-444.

[16] J. M. Gomez-de-Gabriel, J. A. Fernández-Madrigal, A. Lopez-Arquillos, and J. C. Rubio-Romero, "Monitoring harness use in construction with BLE beacons," *Measurement*, vol. 131, pp. 329-340, 2019.

[17] V. Gokhale, G. M. Barrera, and R. V. Prasad, "FEEL: Fast, Energy-Efficient Localization for Autonomous Indoor Vehicles," in *ICC 2021-IEEE International Conference on Communications*, 2021: IEEE, pp. 1-6.
[18] D. H. Kurniawan *et al.*, "IONET: Towards an Open Machine Learning Training Ground for I/O Performance Prediction," Technical Report, 2021.
[19] S. Herath, H. Yan, and Y. Furukawa, "Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020: IEEE, pp. 3146-3152.
[20] C. Chen, X. Lu, A. Markham, and N. Trigoni, "Ionet: Learning to cure the curse of drift in inertial odometry," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, vol. 32, no. 1.
[21] J. Zhao, "A review of wearable IMU (inertial-measurement-unit)-based pose estimation and drift reduction technologies," in *Journal of Physics: Conference Series*, 2018, vol. 1087, no. 4: IOP Publishing, p. 042003.
[22] C. Zhao, C. Fu, J. M. Dolan, and J. Wang, "L-shape fitting-based vehicle pose estimation and tracking using 3D-LiDAR," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 787-798, 2021.
[23] C.-J. Liang, K. M. Lundeen, W. McGee, C. C. Menassa, S. Lee, and V. R. Kamat, "A vision-based marker-less pose estimation system for articulated construction robots," *Automation in Construction*, vol. 104, pp. 80-94, 2019.
[24] J. Wang *et al.*, "Deep 3D human pose estimation: A review," vol. 210, p. 103225, 2021.
[25] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen, "Repulsion loss: Detecting pedestrians in a crowd," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7774-7783.
[26] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5386-5395.
[27] C. Li, D. Song, R. Tong, and M. J. P. R. Tang, "Illumination-aware faster R-CNN for robust multispectral pedestrian detection," vol. 85, pp. 161-171, 2019.
[28] K. Wang and W. J. I. J. o. A. R. S. Zhou, "Pedestrian and cyclist detection based on deep neural network fast R-CNN," vol. 16, no. 2, p. 1729881419829651, 2019.
[29] J. H. Kim, G. Batchuluun, and K. R. J. E. S. w. A. Park, "Pedestrian detection based on faster R-CNN in nighttime by fusing deep convolutional features of successive images," vol. 114, pp. 15-33, 2018.
[30] W.-Y. Hsu and W.-Y. J. I. t. o. i. p. Lin, "Ratio-and-scale-aware YOLO for pedestrian detection," vol. 30, pp. 934-947, 2020.
[31] J. Yang, W. Y. He, T. L. Zhang, C. L. Zhang, L. Zeng, and B. F. Nan, "Research on subway pedestrian detection algorithms based on SSD model," *IET Intelligent Transport Systems*, vol. 14, no. 11, pp. 1491-1496, 2020.
[32] W. Liu, S. Liao, W. Hu, X. Liang, and X. Chen, "Learning efficient single-stage pedestrian detectors by asymptotic localization fitting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 618-634.
[33] S. Wang, J. Cheng, H. Liu, and M. J. a. p. a. Tang, "Pcn: Part and context information for pedestrian detection with cnns," 2018.

[34] S. Li, L. Zhang, and X. Diao, "Deep-learning-based human intention prediction using RGB images and optical flow," *Journal of Intelligent & Robotic Systems,* vol. 97, no. 1, pp. 95-107, 2020.

[35] O. N. Tepencelik, W. Wei, L. Chukoskie, P. C. Cosman, and S. Dey, "Body and Head Orientation Estimation with Privacy Preserving LiDAR Sensors," in *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021: IEEE, pp. 766-770.

[36] D. Mehta *et al.,* "XNect: Real-time multi-person 3D motion capture with a single RGB camera," *Acm Transactions On Graphics (TOG),* vol. 39, no. 4, pp. 82: 1-82: 17, 2020.

[37] M. Hanif *et al.,* "A novel and efficient multiple RGB images cipher based on chaotic system and circular shift operations," *IEEE Access,* vol. 8, pp. 146408-146427, 2020.

[38] R. W. A. Saputra, B. S. B. Dewantara, and D. Pramadihanto, "Human Body's Orientation Estimation Based On Depth Image," in *2019 International Electronics Symposium (IES),* 2019: IEEE, pp. 266-271.

[39] J. D. Bermudez, P. N. Happ, R. Q. Feitosa, and D. A. Oliveira, "Synthesis of multispectral optical images from SAR/optical multitemporal data using conditional generative adversarial networks," *IEEE Geoscience and Remote Sensing Letters,* vol. 16, no. 8, pp. 1220-1224, 2019.

[40] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, "The best of both modes: Separately leveraging rgb and depth for unseen object instance segmentation," in *Conference on robot learning,* 2020: PMLR, pp. 1369-1378.

[41] B. S. B. Dewantara, F. Ardilla, and A. A. Thoriqy, "Implementation of depth-HOG based human upper body detection on a mini pc using a low cost stereo camera," in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT),* 2019: IEEE, pp. 458-463.

[42] W. H. Organization, "Global status report on road safety 2018 World Health Organization," 2018.

[43] W. H. Organization, "Global status report on road safety 2018: summary," World Health Organization, 2018.

[44] N. Aafaq, A. Mian, W. Liu, S. Z. Gilani, and M. Shah, "Video description: A survey of methods, datasets, and evaluation metrics," *ACM Computing Surveys (CSUR),* vol. 52, no. 6, pp. 1-37, 2019.

[45] R. Wang, Y. Cui, X. Song, K. Chen, and H. Fang, "Multi-information-based convolutional neural network with attention mechanism for pedestrian trajectory prediction," *Image and Vision Computing,* vol. 107, p. 104110, 2021.

[46] C. Lin, J. Lu, G. Wang, and J. Zhou, "Graininess-aware deep feature learning for pedestrian detection," in *Proceedings of the European conference on computer vision (ECCV),* 2018, pp. 732-747.

[47] J. Hua, Y. Shi, C. Xie, H. Zhang, and J. Zhang, "Pedestrian-and vehicle-detection algorithm based on improved aggregated channel features," *IEEE Access,* vol. 9, pp. 25885-25897, 2021.

[48] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition,* vol. 77, pp. 354-377, 2018.

[49] M. Fieraru, A. Khoreva, L. Pishchulin, and B. Schiele, "Learning to refine human pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 205-214.

[50] Z. Su, M. Ye, G. Zhang, L. Dai, and J. J. a. p. a. Sheng, "Cascade feature aggregation for human pose estimation," 2019.

[51] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy, "Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 269-286.

[52] A. Toshev and C. D. J. C. Szegedy, "Human pose estimation via deep neural networks’," pp. 1653-1660.

[53] D. C. Luvizon, H. Tabia, and D. Picard, "Human pose regression by combining indirect part detection and contextual information," *Computers & Graphics,* vol. 85, pp. 15-22, 2019.

[54] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653-1660.

[55] G. Moon, J. Y. Chang, and K. M. Lee, "Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image," in *Proceedings of the ieee/cvf international conference on computer vision*, 2019, pp. 10133-10142.

[56] Y.-A. Chung, Y. Belinkov, and J. Glass, "Similarity analysis of self-supervised speech representations," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021: IEEE, pp. 3040-3044.

[57] B. Artacho and A. Savakis, "Unipose: Unified human pose estimation in single images and videos," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7035-7044.

[58] Y. Liu, Y. Xu, and S.-b. Li, "2-D human pose estimation from images based on deep learning: a review," in *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2018: IEEE, pp. 462-465.

[59] C.-J. Chou, J.-T. Chien, and H.-T. Chen, "Self adversarial training for human pose estimation," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2018: IEEE, pp. 17-30.

[60] L. Ke, M.-C. Chang, H. Qi, and S. Lyu, "Multi-scale structure-aware network for human pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 713-728.

[61] W. Tang, P. Yu, and Y. Wu, "Deeply learned compositional models for human pose estimation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 190-206.

[62] W. Tang and Y. Wu, "Does learning specific features for related parts help human pose estimation?," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1107-1116.

[63] Y. Zhang, Y. Wang, O. Camps, and M. Sznaier, "Key Frame Proposal Network for Efficient Pose Estimation in Videos," in *European Conference on Computer Vision*, 2020: Springer, pp. 609-625.

[64] C.-z. Guan, "Realtime multi-person 2d pose estimation using shufflenet," in *2019 14th International Conference on Computer Science & Education (ICCSE)*, 2019: IEEE, pp. 17-21.

[65] C. Zheng *et al.*, "Deep learning-based human pose estimation: A survey," *arXiv preprint arXiv:2012.13392*, 2020.

[66] Z. Tian, H. Chen, and C. J. a. p. a. Shen, "Directpose: Direct end-to-end multi-person pose estimation," 2019.

[67] Z. Tian, H. Chen, and C. Shen, "Directpose: Direct end-to-end multi-person pose estimation," *arXiv preprint arXiv:1911.07451*, 2019.
[68] Y. Cai *et al.*, "Learning delicate local representations for multi-person pose estimation," in *European Conference on Computer Vision*, 2020: Springer, pp. 455-472.
[69] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2334-2343.
[70] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7103-7112.
[71] L. Qiu *et al.*, "Peeking into occluded joints: A novel framework for crowd pose estimation," in *European Conference on Computer Vision*, 2020: Springer, pp. 488-504.
[72] K. Su, D. Yu, Z. Xu, X. Geng, and C. Wang, "Multi-person pose estimation with enhanced channel-wise and spatial information," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5674-5682.
[73] T. Nakatsuka, K. Yoshii, Y. Koyama, S. Fukayama, M. Goto, and S. J. a. p. a. Morishima, "MirrorNet: A Deep Bayesian Approach to Reflective 2D Pose Estimation from Human Images," 2020.
[74] T. Golda, T. Kalb, A. Schumann, and J. Beyerer, "Human pose estimation for real-world crowded scenarios," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2019: IEEE, pp. 1-8.
[75] S. Jin *et al.*, "Differentiable hierarchical graph grouping for multi-person pose estimation," in *European Conference on Computer Vision*, 2020: Springer, pp. 718-734.
[76] M. Kocabas, S. Karagoz, and E. Akbas, "Multiposenet: Fast multi-person pose estimation using pose residual network," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 417-433.
[77] T. Xu and W. J. a. p. a. Takano, "Graph Stacked Hourglass Networks for 3D Human Pose Estimation," 2021.
[78] A. Sengupta, I. Budvytis, and R. J. a. p. a. Cipolla, "Probabilistic 3D Human Shape and Pose Estimation from Multiple Unconstrained Images in the Wild," 2021.
[79] A. Diaz-Arias, M. Messmore, D. Shin, and S. J. a. p. a. Baek, "On the role of depth predictions for 3D human pose estimation," 2021.
[80] I. Sárándi, T. Linder, K. O. Arras, and B. Leibe, "Metrabs: Metric-scale truncation-robust heatmaps for absolute 3d human pose estimation," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 1, pp. 16-30, 2020.
[81] C. Nakatsuka, J. Xu, and K. Tasaka, "Learning Joint Twist Rotation for 3D Human Pose Estimation from a Single Image," 2021.
[82] Z. Zhang, C. Wang, W. Qiu, W. Qin, and W. J. I. J. o. C. V. Zeng, "Adafuse: Adaptive multiview fusion for accurate human pose estimation in the wild," vol. 129, no. 3, pp. 703-718, 2021.
[83] X. Ma, J. Su, C. Wang, H. Ci, and Y. J. a. p. a. Wang, "Context Modeling in 3D Human Pose Estimation: A Unified Perspective," 2021.
[84] X. Huang, J. Huang, and Z. J. I. A. Tang, "3D Human Pose Estimation With Spatial Structure Information," vol. 9, pp. 35947-35956, 2021.

[85] A. Arnab, C. Doersch, and A. Zisserman, "Exploiting temporal context for 3D human pose estimation in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3395-3404.
[86] G. Varol *et al.*, "Bodynet: Volumetric inference of 3d human body shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 20-36.
[87] W. Zeng, W. Ouyang, P. Luo, W. Liu, and X. Wang, "3d human mesh regression with dense correspondence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7054-7063.
[88] S. Saito, T. Simon, J. Saragih, and H. Joo, "Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 84-93.
[89] G. Moon and K. M. J. a. p. a. Lee, "I2L-MeshNet: Image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image," 2020.
[90] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3D human pose and shape via model-fitting in the loop," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2252-2261.
[91] G. Pavlakos *et al.*, "Expressive body capture: 3d hands, face, and body from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10975-10985.
[92] W. Mao, Y. Ge, C. Shen, Z. Tian, X. Wang, and Z. J. a. p. a. Wang, "Tfpose: Direct human pose estimation with transformers," 2021.
[93] J. N. Kundu, S. Seth, M. Rahul, M. Rakesh, V. B. Radhakrishnan, and A. Chakraborty, "Kinematic-structure-preserved representation for unsupervised 3D human pose estimation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, no. 07, pp. 11312-11319.
[94] J. N. Kundu, S. Seth, V. Jampani, M. Rakesh, R. V. Babu, and A. Chakraborty, "Self-supervised 3D human pose estimation via part guided novel image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6152-6162.
[95] A. Zanfir, E. G. Bazavan, H. Xu, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "Weakly supervised 3d human pose and shape reconstruction with normalizing flows," in *European Conference on Computer Vision*, 2020: Springer, pp. 465-481.
[96] H.-Y. F. Tung, H.-W. Tung, E. Yumer, and K. J. a. p. a. Fragkiadaki, "Self-supervised learning of motion capture," 2017.
[97] N. Kolotouros, G. Pavlakos, and K. Daniilidis, "Convolutional mesh regression for single-image human shape reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4501-4510.
[98] H. Zhu, X. Zuo, S. Wang, X. Cao, and R. Yang, "Detailed human shape estimation from a single image by hierarchical mesh deformation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4491-4500.
[99] J. N. Kundu, M. Rakesh, V. Jampani, R. M. Venkatesh, and R. V. Babu, "Appearance Consensus Driven Self-supervised Human Mesh Recovery," in *European Conference on Computer Vision*, 2020: Springer, pp. 794-812.

[100] X. Xu, H. Chen, F. Moreno-Noguer, L. A. Jeni, and F. De la Torre, "3d human shape and pose from a single low-resolution image with self-supervised learning," in *European Conference on Computer Vision*, 2020: Springer, pp. 284-300.
[101] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5253-5263.
[102] D. Xiang, H. Joo, and Y. Sheikh, "Monocular total capture: Posing face, body, and hands in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10965-10974.
[103] M. Fisch and R. J. a. p. a. Clark, "Orientation Keypoints for 6D Human Pose Estimation," 2020.
[104] H. Wang, R. A. Güler, I. Kokkinos, G. Papandreou, and S. Zafeiriou, "BLSM: A Bone-Level Skinned Model of the Human Mesh," in *European Conference on Computer Vision*, 2020: Springer, pp. 1-17.
[105] A. A. Osman, T. Bolkart, and M. J. J. a. p. a. Black, "STAR: Sparse Trained Articulated Human Body Regressor," 2020.
[106] J. Zhen *et al.*, "SMAP: Single-Shot Multi-Person Absolute 3D Pose Estimation," in *European Conference on Computer Vision*, 2020: Springer, pp. 550-566.
[107] X. Nie, J. Feng, J. Zhang, and S. Yan, "Single-stage multi-person pose machines," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6951-6960.
[108] D. Mehta *et al.*, "XNect: Real-time multi-person 3D motion capture with a single RGB camera," vol. 39, no. 4, pp. 82: 1-82: 17, 2020.
[109] M. Fabbri, F. Lanzi, S. Calderara, S. Alletto, and R. Cucchiara, "Compressed volumetric heatmaps for multi-person 3d pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7204-7213.
[110] D. Mehta *et al.*, "Single-shot multi-person 3d pose estimation from monocular rgb," in *2018 International Conference on 3D Vision (3DV)*, 2018: IEEE, pp. 120-130.
[111] J. N. Kundu, A. Revanur, G. V. Waghmare, R. M. Venkatesh, and R. V. Babu, "Unsupervised Cross-Modal Alignment for Multi-Person 3D Pose Estimation," in *European Conference on Computer Vision*, 2020: Springer, pp. 35-52.
[112] A. Zanfir, E. Marinoiu, M. Zanfir, A.-I. Popa, and C. J. A. i. N. I. P. S. Sminchisescu, "Deep network for the integrated 3d sensing of multiple people in natural images," vol. 31, pp. 8410-8419, 2018.
[113] H. Rhodin *et al.*, "Learning monocular 3d human pose estimation from multi-view images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8437-8446.
[114] H. Rhodin, M. Salzmann, and P. Fua, "Unsupervised geometry-aware representation for 3d human pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 750-767.
[115] J. Liang and M. C. Lin, "Shape-aware human pose and shape reconstruction using multi-view images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4352-4362.
[116] M. Habermann, W. Xu, M. Zollhofer, G. Pons-Moll, and C. Theobalt, "Deepcap: Monocular human performance capture using weak supervision," in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5052-5063.
[117] M. Kocabas, S. Karagoz, and E. Akbas, "Self-supervised learning of 3d human pose using multi-view geometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1077-1086.
[118] A. Kadkhodamohammadi, N. J. M. V. Padoy, and Applications, "A generalizable approach for multi-view 3d human pose regression," vol. 32, no. 1, pp. 1-14, 2021.
[119] H. Qiu, C. Wang, J. Wang, N. Wang, and W. Zeng, "Cross view fusion for 3d human pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4342-4351.
[120] C. Sminchisescu, "Domes to drones: Self-supervised active triangulation for 3d human pose reconstruction," 2019.
[121] H. Tu, C. Wang, and W. J. a. p. a. Zeng, "Voxelpose: Towards multi-camera 3d human pose estimation in wild environment," 2020.
[122] E. Remelli, S. Han, S. Honari, P. Fua, and R. Wang, "Lightweight multi-view 3d pose estimation through camera-disentangled representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6040-6049.
[123] L. Chen, H. Ai, R. Chen, Z. Zhuang, and S. Liu, "Cross-view tracking for multi-human 3d pose estimation at over 100 fps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3279-3288.
[124] Y. Li, L. Xin, D. Yu, P. Dai, J. Wang, and S. E. Li, "Pedestrian trajectory prediction with learning-based approaches: A comparative study," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019: IEEE, pp. 919-926.
[125] K. Gong, J. Zhang, and J. Feng, "Poseaug: A differentiable pose augmentation framework for 3d human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8575-8584.
[126] J. Wang, S. Jin, W. Liu, W. Liu, C. Qian, and P. Luo, "When human pose estimation meets robustness: Adversarial algorithms and benchmarks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11855-11864.
[127] H. Joo, N. Neverova, and A. Vedaldi, "Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation," in *2021 International Conference on 3D Vision (3DV)*, 2021: IEEE, pp. 42-52.
[128] W.-Y. Hsu, "Automatic pedestrian detection in partially occluded single image," *Integrated Computer-Aided Engineering*, vol. 25, no. 4, pp. 369-379, 2018.
[129] L. He, C. Liu, J. Li, Y. Li, S. Li, and Z. Yu, "Hyperspectral image spectral–spatial-range Gabor filtering," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4818-4836, 2020.
[130] R. Raman, P. K. Sa, S. Bakshi, and B. Majhi, "Kinesiology-inspired estimation of pedestrian walk direction for smart surveillance," *Future Generation Computer Systems*, vol. 108, pp. 1008-1026, 2020.
[131] D. Vijayalakshmi and M. K. Nath, "A novel contrast enhancement technique using gradient-based joint histogram equalization," *Circuits, Systems, and Signal Processing*, vol. 40, no. 8, pp. 3929-3967, 2021.
[132] M. Versaci and F. C. Morabito, "Image edge detection: A new approach based on fuzzy entropy and fuzzy divergence," *International Journal of Fuzzy Systems*, vol. 23, no. 4, pp. 918-936, 2021.

[133] Y. Kohari, J. Miura, and S. Oishi, "Cnn-based human body orientation estimation for robotic attendant," in *IAS-15 Workshop on Robot Perception of Humans*, 2018.

[134] S. Song, L. Shen, and M. Valstar, "Human behaviour-based automatic depression analysis using hand-crafted statistics and deep learned spectral features," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, 2018: IEEE, pp. 158-165.

[135] Y. Jin, Y. Zhang, Y. Cen, Y. Li, V. Mladenovic, and V. Voronin, "Pedestrian detection with super-resolution reconstruction for low-quality image," *Pattern Recognition,* vol. 115, p. 107846, 2021.

[136] G. Gao, "Design of online retail commodity recommendation system based on Item-CF algorithm," in *2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT)*, 2021: IEEE, pp. 460-464.

[137] Z. Huang, X. Zhu, Y. Lin, L. Xu, and Y. Mao, "A novel WIFI-oriented RSSI signal processing method for tracking low-speed pedestrians," in *2019 5th International Conference on Transportation Information and Safety (ICTIS)*, 2019: IEEE, pp. 1018-1023.

[138] Q. Xu, H. Wu, J. Wang, H. Xiong, J. Liu, and K. Li, "Roadside pedestrian motion prediction using Bayesian methods and particle filter," *IET Intelligent Transport Systems,* vol. 15, no. 9, pp. 1167-1182, 2021.

[139] O. M. Lwin and T. L. L. Thein, "Detection and Indication of Pedestrian Crossing on the Road Using ACF AND KALMAN Filter," *International Journal Of All Research Writings,* vol. 3, no. 4, pp. 62-66, 2020.

[140] V. Shree, C. Diaz-Ruiz, C. Liu, B. Hariharan, and M. Campbell, "Orientation-Discriminative Feature Representation for Decentralized Pedestrian Tracking," *arXiv preprint arXiv:2202.13237,* 2022.

[141] D. Burgermeister and C. Curio, "PedRecNet: Multi-task deep neural network for full 3D human pose and orientation estimation," *arXiv preprint arXiv:2204.11548,* 2022.

[142] T. Wengefeld, B. Lewandowski, D. Seichter, L. Pfennig, and H.-M. Gross, "Real-time person orientation estimation using colored pointclouds," in *2019 European Conference on Mobile Robots (ECMR)*, 2019: IEEE, pp. 1-7.

[143] F. Shinmura *et al.*, "Estimation of driver's insight for safe passing based on pedestrian attributes," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018: IEEE, pp. 1041-1046.

[144] D. Yu, H. Xiong, Q. Xu, J. Wang, and K. Li, "Continuous pedestrian orientation estimation using human keypoints," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019: IEEE, pp. 1-5.

[145] X. Ou *et al.*, "Moving object detection method via ResNet-18 with encoder–decoder structure in complex scenes," *IEEE Access,* vol. 7, pp. 108152-108160, 2019.

[146] Y. Yuan, S.-E. Wei, T. Simon, K. Kitani, and J. Saragih, "Simpoe: Simulated character control for 3d human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7159-7169.

[147] D. Rempe, T. Birdal, A. Hertzmann, J. Yang, S. Sridhar, and L. J. Guibas, "Humor: 3d human motion model for robust pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11488-11499.

[148] Y. Rong, T. Shiratori, and H. Joo, "Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1749-1759.

[149] V. Guzov, A. Mir, T. Sattler, and G. Pons-Moll, "Human poseitioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4318-4329.

[150] A. Gupta and L. Carlone, "Online monitoring for neural network based monocular pedestrian pose estimation," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020: IEEE, pp. 1-8.

[151] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3907-3916.

[152] Q. Zhao *et al.*, "A CNN-SIFT hybrid pedestrian navigation method based on first-person vision," *Remote Sensing*, vol. 10, no. 8, p. 1229, 2018.

[153] R. Khemmar, M. Gouveia, B. Decoux, and J.-Y. Ertaud, "Real time pedestrian and object detection and tracking-based deep learning. application to drone visual tracking," 2019.

[154] F. Liu, J. Zhang, J. Wang, H. Han, and D. Yang, "An UWB/vision fusion scheme for determining pedestrians' indoor location," *Sensors*, vol. 20, no. 4, p. 1139, 2020.

[155] D. Seichter, B. Lewandowski, D. Höchemer, T. Wengefeld, and H.-M. Gross, "Multi-task deep learning for depth-based person perception in mobile robotics," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020: IEEE, pp. 10497-10504.

[156] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019: PMLR, pp. 6105-6114.

[157] I. N. Junejo, "A deep learning based multi-color space approach for pedestrian attribute recognition," in *Proceedings of the 2019 3rd international conference on graphics and signal processing*, 2019, pp. 113-116.

[158] Y. Wang, H. Cheng, C. Wang, and M. Q.-H. Meng, "Pose-Invariant Inertial Odometry for Pedestrian Localization," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-12, 2021.

[159] M. Raza, Z. Chen, S.-U. Rehman, P. Wang, and P. Bao, "Appearance based pedestrians' head pose and body orientation estimation using deep learning," *Neurocomputing*, vol. 272, pp. 647-659, 2018.

[160] N. Zerrouki and A. Houacine, "Combined curvelets and hidden Markov models for human fall detection," *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 6405-6424, 2018.

[161] Z. Huang, Y. Liu, Y. Fang, and B. K. Horn, "Video-based fall detection for seniors with human pose estimation," in *2018 4th International Conference on Universal Village (UV)*, 2018: IEEE, pp. 1-4.

[162] N. Ukita and Y. Uematsu, "Semi-and weakly-supervised human pose estimation," *Computer Vision and Image Understanding*, vol. 170, pp. 67-78, 2018.

[163] S. Zhang, M. Abdel-Aty, Y. Wu, and O. J. I. T. o. I. T. S. Zheng, "Pedestrian crossing intention prediction at red-light using pose estimation," 2021.

[164] V. A. L. S. León and A. Schwering, "Detecting socially occupied spaces with depth cameras: evaluating location and body orientation as relevant social

features," in *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*: IEEE, pp. 1-8.
[165] A. Arbués-Sangüesa, A. Martín, P. Granero, C. Ballester, and G. J. a. p. a. Haro, "Learning football body-orientation as a matter of classification," 2021.
[166] Z. Wang and N. Papanikolopoulos, "Estimating pedestrian crossing states based on single 2d body pose," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020: IEEE, pp. 2205-2210.
[167] M. Snower, A. Kadav, F. Lai, and H. P. Graf, "15 keypoints is all you need," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6738-6748.
[168] B. S. B. Dewantara, R. W. A. Saputra, and D. Pramadihanto, "Estimating human body orientation from image depth data and its implementation," *Machine Vision and Applications,* vol. 33, no. 3, pp. 1-19, 2022.
[169] C. Zhao, Y. Qian, and M. Yang, "Monocular pedestrian orientation estimation based on deep 2D-3D feedforward," *Pattern Recognition,* vol. 100, p. 107182, 2020.
[170] H. Kataoka, Y. Satoh, Y. Aoki, S. Oikawa, and Y. Matsui, "Temporal and fine-grained pedestrian action recognition on driving recorder database," *Sensors,* vol. 18, no. 2, p. 627, 2018.
[171] B. Yang, W. Zhan, P. Wang, C. Chan, Y. Cai, and N. Wang, "Crossing or not? context-based recognition of pedestrian crossing intention in the urban environment," *IEEE Transactions on Intelligent Transportation Systems,* 2021.
[172] S. Dafrallah, A. Amine, S. Mousset, and A. Bensrhair, "Monocular pedestrian orientation recognition based on Capsule Network for a Novel Collision Warning System," *IEEE Access,* vol. 9, pp. 141635-141650, 2021.
[173] K. M. Abughalieh and S. G. Alawneh, "Predicting pedestrian intention to cross the road," *IEEE Access,* vol. 8, pp. 72558-72569, 2020.
[174] D. Heo, J. Y. Nam, and B. C. Ko, "Estimation of pedestrian pose orientation using soft target training based on teacher–student framework," *Sensors,* vol. 19, no. 5, p. 1147, 2019.
[175] Z. Liu *et al.*, "Deep dual consecutive network for human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 525-534.
[176] M. Wang, J. Tighe, and D. Modolo, "Combining detection and tracking for human pose estimation in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2020, pp. 11088-11096.
[177] S. Dafrallah, Z. Sabir, A. Amine, S. Mousset, and A. Bensrhair, "Pedestrian walking direction classification for Moroccan road safety," in *Proc. Int. Conf. Ind. Eng. Oper. Manage.*, 2020, pp. 1-8.
[178] T. Von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 601-617.
[179] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3754-3762.
[180] S. Rostianingsih, A. Setiawan, and C. I. Halim, "COCO (creating common object in context) dataset for chemistry apparatus," *Procedia Computer Science,* vol. 171, pp. 2445-2452, 2020.

[181] M. Andriluka *et al.*, "Posetrack: A benchmark for human pose estimation and tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5167-5176.

[182] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Joint attention in autonomous driving (JAAD)," *arXiv preprint arXiv:1609.04741*, 2016.