Received May 25, 2020, accepted June 15, 2020, date of publication June 19, 2020, date of current version July 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003751

Instance-Level Future Motion Estimation in a Single Image Based on Ordinal Regression and Semi-Supervised Domain Adaptation

KYUNG-RAE KIM[1], YEONG JUN KOH[2], (Member, IEEE), AND CHANG-SU KIM[2], (Senior Member, IEEE)

[1]School of Electrical Engineering, Korea University, Seoul 02841, South Korea
[2]Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, South Korea

Corresponding author: Chang-Su Kim (changsukim@korea.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIP) under Grant NRF-2018R1A2B3003896, in part by the MSIT, Korea, under the ITRC Support Program under Grant IITP-2019-2016-0-00464 supervised by the IITP, and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant 2019R1F1A1062907.

ABSTRACT A novel algorithm to estimate instance-level future motion (FM) in a single image is proposed in this paper. First, the FM of an instance is defined with its direction, speed, and action classes. Then, a deep neural network, called FM-Net, is developed to determine the FM of the instance. More specifically, the multi-context pooling layer is proposed to exploit both object and global context features, and the cyclic ordinal regression scheme is developed using binary classifiers for effective FM classification. Also, the proposed FM-Net is trained in a semi-supervised domain adaptation setting to obtain reliable FM estimation results, even when a source domain in the training process and a target domain in the inference process are different. Extensive experimental results demonstrate that the proposed algorithm provides remarkable performance and thus can be used effectively for computer vision applications, including single object tracking, multiple object tracking, and crowd analysis. Furthermore, the FM dataset, collected from diverse sources and annotated manually, is released as a benchmark for single-image FM estimation.

INDEX TERMS Future motion estimation, cyclic ordinal regression, semi-supervised domain adaptation.

I. INTRODUCTION

Human perception has a capability of forecasting motions accurately, even from a single static image. As illustrated in Figure 1, a human being can proactively predict motion directions and magnitudes of pedestrians from a single image. Such proactive, predictive perceptual capabilities enable us to take desired actions and avoid dangerous situations. To make computing machines achieve a similar level to the human perception, motion understanding and representation have been studied in many computer vision tasks such as optical flow [1]–[3], object tracking [4]–[6], action recognition [7], future frame prediction [8], video interpolation [9], and video compression [10]. However, most conventional techniques depend on temporal information from multiple consecutive frames to estimate motions.

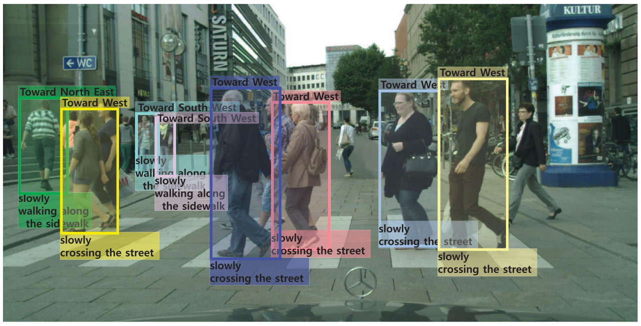
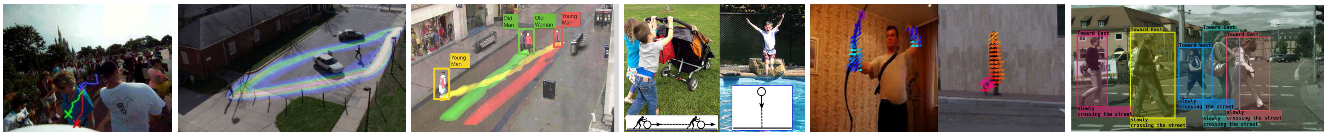

← WC
SATURN
KULTUR
Toward North East
slowly walking along the sidewalk
Toward West
slowly crossing the street
Toward South West
slowly walking along the sidewalk
Toward South West
slowly walking along the sidewalk
Toward West
slowly crossing the street
Toward West
slowly crossing the street
Toward West
slowly crossing the street
Toward West
slowly crossing the street

FIGURE 1. **Seeing a single still image, humans are usually capable of predicting what will happen in the next. To learn such humans' intuition regarding future motion, we develop a deep network, called FM-Net, that predicts the next behaviour of instances in a still image. The above annotations are automatically generated by the proposed FM-Net.**

In this paper, a pioneering algorithm to estimate instance-level future motions (FMs) in a single image is proposed to address the limitation of the aforementioned conventional

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.


Old Man
Old Woman
Young Man
Young Man
Toward East
Toward East
Toward East
Toward West
Toward East
slowly crossing the street
slowly crossing the street
slowly crossing the street
slowly crossing the street
slowly crossing the street

Yagi *et al.* [11] Kitani *et al.* [12] Ma *et al.* [13] Mottaghi *et al.* [14] Gao *et al.* [15] Proposed algorithm

FIGURE 2. Comparison of the proposed FM estimation algorithm with the conventional algorithms [11]–[15].

techniques. The proposed algorithm attempts to challenge the human perception in single-image motion understanding. Future prediction has been studied in some methods [11]–[15], but there are clear differences from the proposed algorithm. As shown in Figure 2, Mottaghi *et al.* [11] predict a pedestrian's future trajectory in a first-person video but require past frames to obtain an accumulative trajectory of the instance. Unlike the method in [11], the proposed algorithm requires only a present frame. On the other hand, some algorithms [12], [13] require both starting and end points to estimate long-term trajectories of instances. Kitani *et al.* [14] define motion scenarios and classify object instances in a single image into one of the pre-defined motion scenarios. However, they focus on scene classification rather than on FM prediction of objects. Gao *et al.* [15] estimate the dense optical flow using only a single image, but their algorithm works only for highly similar motion scenes to training scenes.

Recently, deep neural networks have been extensively adopted for computer vision applications, since they exhibit excellent capabilities for analyzing visual appearance of objects [16]–[25]. Note that humans can predict motions of instances in a single image based on their experience, even unaware of the exact physics. Inspired by this observation, we adopt a deep neural network, called FM-Net, to implement such perceptual capabilities regarding FM. Specifically, we attempt to imitate human beings who recognize object motions by perceiving visual information of the object and its surroundings simultaneously. Thus, the multi-context pooling (MCP), which integrates both object and global context features, is incorporated into DenseNet-121 [16] to learn a unified model for estimating the future direction, speed, and action of an instance. Also, the cyclic ordinal regression (COR) scheme is proposed to train FM-Net effectively.

It has been proven that deep neural networks can achieve remarkable performances, provided that a sufficient number of reliable training examples are available [16]–[25]. Hence, a reliable dataset for single-image FM estimation is essential for learning FM-Net stably. One of the objectives of this paper is to construct a reliable dataset for single-image FM estimation. We construct such a dataset, referred to as FM dataset, which contains a large number of still images containing three kinds of instances: pedestrians, cars, and animals (dogs, cats, and horses). For the pedestrian instances, still images containing pedestrians are collected from the Caltech Pedestrian Detection Benchmark (CPDB) [26], the CityPersons dataset [27], and YouTube [28]. Then, three attributes of FM (*i.e.* direction, speed, and action) are manually assigned to each pedestrian. For the car and animal instances, image examples are assembled from YouTube and labels are also assigned manually to each instance.

However, there might be an overfitting problem to the training data, since the proposed FM-Net is trained in an end-to-end manner. Thus, when a source domain for the training and a target domain for the inference are different, FM-Net may experience performance degradation. For instance, when FM-Net is trained using dog instances only, it may fail to estimate FMs of cat instances. Considering the difficulty of collecting a large number of labeled training examples for a new target domain, we develop a semi-supervised domain adaptation training strategy to improve the generalization performance of FM-Net on new domains. An adversarial training method is developed to perform semi-supervised domain adaptation using three kinds of data: a large number of labeled source domain data, only a limited number of labeled data in the target domain, and a large number of unlabeled data in the target domain.

Experimental results demonstrate that the proposed FM-Net yields remarkable FM estimation results for pedestrian, car, and animal instances despite variations in camera viewpoints and capturing environments, when a sufficient number of labeled training data are provided. Moreover, it is demonstrated that the proposed semi-supervised domain adaptation learning improves FM estimation accuracies, when only a limited number of labeled data for a new domain are available. Furthermore, to demonstrate the applicability of the proposed FM-Net, FM-Net is applied to three computer vision applications: single object tracking, multiple object tracking, and crowd analysis. It is demonstrated that FM-Net makes the conventional single object tracker [4] and multiple object tracker [5] more efficient by reducing search regions. Also, FM estimation results are adopted to analyze the crowd in a single image more effectively.

This work has the following main contributions:

- FM-Net is proposed by incorporating the MCP layer into DenseNet-121 and developing the COR scheme for future direction classification.
- The proposed algorithm estimates FM reliably in diverse scenes and environments.
- The generalization capability of FM-Net on new domains is improved by training FM-Net based on the proposed semi-supervised domain adaptation scheme.
- The efficacy of FM-Net is demonstrated in single object tracking, multiple object tracking, and crowd analysis.

- The FM dataset is released to serve as a benchmark for the interesting research topic of subsequent behaviour estimation in a single still image.

This paper extends the preliminary work [29], by including more results and more analysis on the FM dataset and developing the semi-supervised domain adaptation scheme of FM-Net.

This paper is organized as follows: Section II reviews related work briefly. Section III describes how FM dataset is constructed. Section IV proposes FM-Net, and Section V develops its domain adaptation scheme. Section VI discusses the FM estimation performance of FM-Net, and Section VII shows that FM-Net can be used effectively in vision applications. Finally, Section VIII concludes this paper.

II. RELATED WORK

A. FUTURE MOTION ESTIMATION

There are two types of algorithms for estimating future information: 1) instance-level FM and 2) pixel-level FM.

1) INSTANCE-LEVEL FM ALGORITHMS

First, instance-level FM algorithms are reviewed, which estimate FMs of bounding boxes containing object instances. Some algorithms try to estimate future trajectories of objects [11]–[13], [30]. Mottaghi *et al.* [11] perform future person localization in a first-person video by exploiting trajectory information in past frames. They use three key observations, *i.e.* ego-motion, pose, and scale change streams, as input to a deep neural network to estimate future trajectories. Walker *et al.* [30] use mid-level patches, based on temporal modeling, to determine active objects and predict their future trajectories in a scene. Yagi *et al.* [12] and Ma *et al.* [13], respectively, estimate future trajectories by employing only the information of start and end points of each pedestrian. More specifically, Yagi *et al.* [12] adopt a Markov decision process to predict a trajectory between the provided start and end points. However, their algorithm works on limited scenes only that are used in the training process. Ma *et al.* [13] develop another trajectory estimation method for multiple pedestrians based on game theory. These instance-level algorithms [11]–[13], [30] estimate long-term FM, but they require additional information, such as past frames [11] or starting and end points [12], [13], [30].

Kitani *et al.* [14] estimate future forces and motions of objects from a static image. They learn a neural network to map the image into one of 66 pre-defined scenarios for physical abstraction. Chao *et al.* [31] construct recurrent neural networks to forecast human pose sequences using 3D skeletons from a single frame. However, their method is applicable to limited sports scenes only, which are similar to training data.

2) PIXEL-LEVEL FM ALGORITHMS

Pixel-level FM algorithms [15], [32]–[35] predict dense motion, such as optical flow, from a single image. Pintea *et al.* [32] learn local motion patterns from a set of videos using the structured random forest to estimate flow vectors in a still image. Deep neural networks are employed to estimate pixel-level FMs in a single image [15], [33]–[35]. Walker *et al.* [33] propose a convolutional neural network (CNN) to categorize each pixel into predefined 40 motion clusters. Walker *et al.* [34] also develop a conditional variational autoencoder (VAE), which adopts several distributions of future motion patterns and samples multiple possible future states, to forecast dense trajectories of pixels in a static image. Gao *et al.* [15] design an optical flow estimation network that has an encoder-decoder structure and then employ the dense flow map for action recognition. Li *et al.* [35] develop a spatial-temporal conditional VAE to estimate a set of consecutive future flow maps. By utilizing the predicted flow maps for full-frame synthesis, they facilitate video prediction. These pixel-level algorithms [15], [32]–[35] yield lower-level information (*i.e.* denser motion) than instance-level algorithms. They, however, may be ineffective for images that are dissimilar from the training images.

B. ORDINAL REGRESSION

Ordinal regression is a learning scheme to predict a label (or rank) of an object, where the set of labels has a linear order [36], *e.g.* the set of integers or grades. Several approaches have been tried for ordinal regression [37], such as perceptron learning [38], Gaussian processes [39], and support vector machines (SVMs) [40]. On the other hand, Frank and Hall [41] propose an algorithm to transform a k-class ordinal problem into $k - 1$ binary classification problems. They estimate a rank by combining binary classification results. Li and Lin [42] also reduce ordinal regression into binary classification problems based on extended examples. In this reduction framework, they develop a ranking rule using binary classifiers and combine it with existing algorithms, such as perceptrons or SVMs. Many application methods, such as the age estimators in [43], [44], adopt this reduction framework. Also, Devlaminck *et al.* [45] devise a decomposition scheme for the case of a circular order. In this work, we formulate the FM classification as an ordinal regression problem.

C. SEMI-SUPERVISED DOMAIN ADAPTATION

Semi-supervised domain adaptation adopts a semi-supervised learning method to achieve domain adaptation. Hence, semi-supervised domain adaptation is related to both semi-supervised learning and domain adaption techniques.

Semi-supervised learning aims to train an inference model using a limited number of labeled data and a large number of unlabeled data. Recently, many semi-supervised learning methods have been studied to train deep neural networks efficiently [46]–[49]. Also, various computer vision applications have exploited semi-supervised learning methods to reduce expensive labeling efforts. They include 3D human pose estimation [50], 3D hand pose estimation [51], deraining [52], scene parsing [53], multi-view keypoint detection [54], object detection [55], and skin detection in a single human portrait image [56].

Domain adaptation is a learning task that enables an inference model on a source domain to perform on a target domain, when the data distributions between the two domains are different. Using labeled source domain data, a domain adaptation algorithm attempts to reduce the gap in feature distributions between source and target domains, when target domain data are fully unlabeled (*i.e.* unsupervised domain adaptation [57]–[62]) or have only a few labeled samples (*i.e.* semi-supervised domain adaptation [63]–[67]). Cheng and Pan [66] develop a manifold-based gradient descent algorithm to learn robust models for source and target domains. Saito *et al.* [67] train a domain-invariant classifier using a deep neural network. They adopt adversarial training to maximize the entropy on unlabeled target data while minimizing the entropy with respect to the feature extractor.

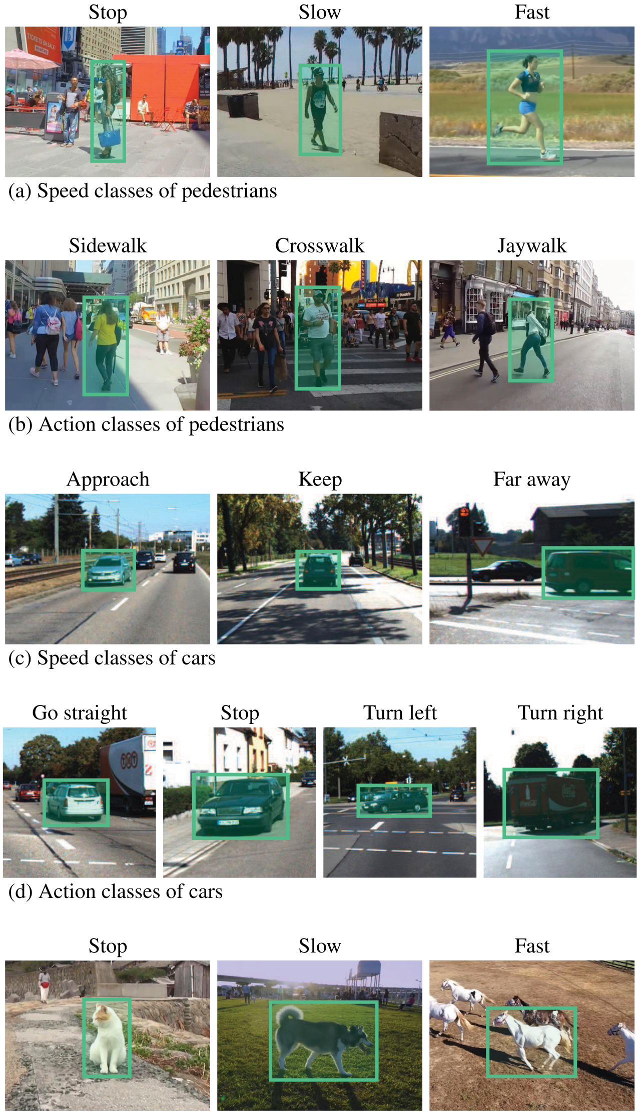
III. FM DATASET

We construct the FM dataset, which annotates pedestrian, car, or animal instances with motion attributes (*i.e.* direction, speed, and action). Note that motion information is often represented by displacement vectors between successive frames. For example, dense optical flow algorithms attempt to predict pixel-level correspondences between video frames. However, it is hard to extract pixel-level motion information precisely and reliably from real-world videos. Instead, through relatively easy annotations, instance-level motions in still images are collected. This instance-level annotation facilitates the construction of a large dataset. The proposed algorithm is not limited to a particular type of instances, so we compose the FM dataset with three kinds of instances: 1) pedestrians, 2) cars, and 3) animals.

A. PEDESTRIANS

In particular, we focus on estimating pedestrians' FMs for the following reasons. First, it is easy to access public datasets, capturing street scenes, and annotate lots of pedestrians. Second, excluding abnormal situations, human behaviour is predictable even in a single image. In other words, pedestrians's FMs can be inferred from semantic contexts in general. Third, humans are often objects of the most interest.

11,342 pedestrian instances are extracted from still images, which are sampled from the CityPersons dataset [27], the Caltech Pedestrian Detection Benchmark (CPDB) dataset [26], and YouTube [28]. For CityPersons and CPDB, the provided bounding boxes of pedestrian instances are used. On the other hand, to collect pedestrian data from YouTube, videos are retrieved using the keywords of 'street,' 'walking,' 'running,' and 'pedestrian.' Then, a small number of videos are ruled out, which include too tiny or severely occluded pedestrians. For such videos, even a human being cannot predict the pedestrians' FMs reliably. Also, overhead videos, captured from cameras looking straight down on the ground, are eliminated. In those videos, body parts under pedestrian shoulders are hardly exposed. In the YouTube videos, however, there are no provided bounding boxes of instances. Thus, the YOLOv3 detector [68] is used to obtain the bounding boxes.



(a) Speed classes of pedestrians

(b) Action classes of pedestrians

(c) Speed classes of cars

(d) Action classes of cars

(e) Speed classes of animals

FIGURE 3. Examples of speed classes and action classes.

For each pedestrian instance, its direction, speed, and action classes are manually labeled by referring to the current frame and nine subsequent frames. The future direction is quantized into one of the four cardinal directions (N, E, S, W) and the four intermediate ones (NE, SE, SW, NW) in the image coordinates. Only these eight quantized directions are used, because they are sufficient in many applications. Moreover, finer quantization makes the annotation difficult and unreliable. For similar reasons, only three speed classes are defined, 'stop,' 'slow,' and 'fast' in Figure 3(a), which contain standing pedestrians, walking pedestrians, and running pedestrians, respectively. Action classes can be defined differently according to the requirements of applications.

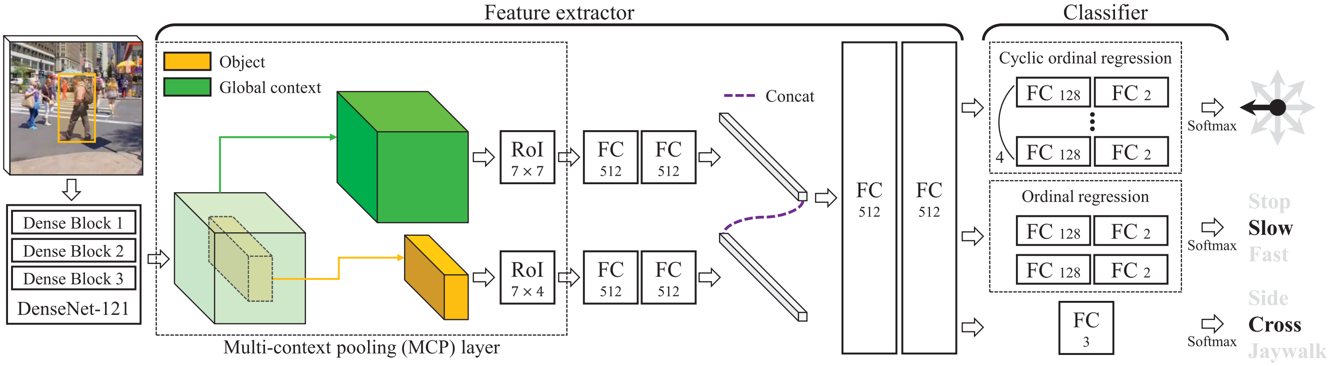


FIGURE 4. The network architecture of FM-Net.

TABLE 1. Statistics of the pedestrian instances in the proposed FM dataset.

Type	Class	No. of Instances
Direction	N	1,017
	NE	1,085
	E	2,190
	SE	1,536
	S	958
	SW	1,143
	W	2,153
	NW	1,689
Speed	Stop	1,994
	Slow	7,419
	Fast	2,358
Action	Sidewalk	6,458
	Crosswalk	2,269
	Jaywalk	3,044
Total		11,771

TABLE 2. Statistics of the car instances in the proposed FM dataset.

Type	Class	# Instances
Direction	N	1,689
	NE	2,312
	E	1,462
	SE	340
	S	295
	SW	4,584
	W	1,853
	NW	3,359
Speed	Approach	2,641
	Keep	13,024
	Far away	230
Action	Go straight	10,849
	Stop	4,896
	Turn left	144
	Turn right	6
Total		15,895

In this work, to monitor pedestrians' behaviour on streets, three action classes of 'sidewalk,' 'crosswalk,' and 'jaywalk' are defined as illustrated in Figure 3(b). Table 1 shows the class distributions of the pedestrian instances in the FM dataset. The entire dataset is split into training and test sets with a ratio of 4 to 1.

B. CARS

For car instances, there are the same 8 directional classes as for pedestrians. In the case of speed, even a human being cannot easily predict the absolute speed of a car in a single image, since the car has a rigid shape. Thus, three speed classes are defined as 'approach,' 'keep,' and 'far away' in Figure 3(c), which represent relative speeds of a car instance with respect to the capturing camera. If the distance between the camera and the instance is decreasing, the speed class is 'approach.' If the camera and the instance move at the same speed in the same direction, the class is 'keep.' Otherwise, the class is 'far away.' Last, four action classes for cars are defined as 'go straight,' 'stop,' 'turn left,' and 'turn right' in Figure 3(d). Then, those three attributes are manually assigned to each car instance. The KITTI object dataset [69], composed of 7,481 training images and 7,518 test images, is used. Since the test images have no bounding box annotations, the training images are only used, which contain 15,895 objects in total. For cars, the network in Figure 4 is trained with 6,526 images with 13,895 objects. The test set consists of 955 images with 2,000 objects. Table 2 shows the class distributions of the car instances in the FM dataset.

C. ANIMALS

Finally, the FM dataset is extended to include animal instances (cats, dogs, and horses) as well. As four-footed animals, they have similar motion characteristics, even though they do not belong to the same family. To construct the FM animal dataset, frames including cats, dogs, and horses are collected from YouTube [28]. Then, the detector, Mask R-CNN [17], is adopted to obtain the bounding boxes. Animals have eight direction classes and three speed classes in the same way as pedestrians do. Figure 3(e) illustrates the three speed classes. The action classification for animals is not performed. 5,516 images, including 6,626 animals, are collected: 5,302 animals are used for training and 1,324 for test. Table 3 represents the class distributions of these animal instances.

TABLE 3. Statistics of the animal instances in the proposed FM dataset.

Type	Class	# Instances
Direction	N	782
	NE	750
	E	922
	SE	810
	S	882
	SW	816
	W	916
	NW	748
Speed	Stop	1,042
	Slow	2,796
	Fast	2,788
Total		6,626

IV. FM-Net

FM-Net is developed to perform the classification of three motion attributes of instances: direction, speed, and action. Let us describe the FM estimation of pedestrian instances. The FMs of cars and animals are estimated in similar ways.

A. FM-Net ARCHITECTURE

Figure 4 is the architecture of FM-Net, which has a single shared feature extractor and three classifiers for direction, speed, and action. FM-Net takes an image patch, in which a pedestrian is located at the center, and yields the three classification results.

It is assumed that, in an image, pedestrians are either located manually or parsed by an object detector. Suppose that a pedestrian has a bounding box with a height h. Then, around the bounding box, the $2h \times 2h$ patch is cropped, which is put into the network. The feature extractor then yields an object feature and a global context feature based on DenseNet-121 [16]. To extract those features from the output of DenseNet-121, the MCP layer is developed, which uses two region of interest (RoI) pooling layers:

- The bounding box is regarded as the RoI for the object feature, which is pooled to spatial resolution 7×4. Note that the object feature conveys the appearance information of a pedestrian with minimal background.
- The $2h \times 2h$ patch is the RoI for the global context feature, pooled to resolution 7×7. This global feature is also important in FM estimation since it conveys overall semantic information about a scene.

The output sizes of the RoI pooling layers are determined empirically. Each RoI pooling layer is followed by two fully-connected (FC) layers.

Then, the classifier performs the three classification tasks, by employing FC layers and softmax layers. More specifically, the object and global context features are concatenated and processed by two FC layers and then by three sub-classifiers for the pedestrian's direction, speed, and action. The three sub-classifiers are designed differently. First, the direction is classified using the COR scheme in Section IV-B. Second, the speed is classified using the linear ordinal regression [42], since the speed classes are in the order of 'stop,' 'slow,' and 'fast.' In other words, 'stop' and 'fast' are more different from each other than 'stop' and 'slow' (or 'slow' and 'fast') are. Third, the 3-way classification of the action is simply performed using an FC layer and a softmax layer, since there is no ordinal relation among the action classes of 'sidewalk,' 'crosswalk,' and 'jaywalk.'

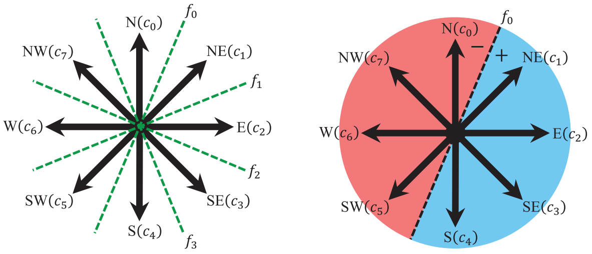


FIGURE 5. Binary classifiers for the cyclic ordinal regression.

B. CYCLIC ORDINAL REGRESSION

As shown in Figure 5, the future direction of a pedestrian is classified into one of the eight directions: N (c_0), NE (c_1), E (c_2), SE (c_3), S (c_4), SW (c_5), W (c_6), NW (c_7). The direction classes have a cyclic order, because N (c_0) is adjacent to both NW (c_7) and NE (c_1). Note that many physical quantities have cyclic orders, *e.g.* 24 hours in a day, longitudes, as well as directions on a plane. Suppose that there are K directional classes in a cyclic order,

$$\mathcal{C} = \{c_0, c_1, \ldots, c_{K-1}\} \tag{1}$$

where K is an even number. In such a case, it is not desirable to apply the K-way classification that does not consider the cyclic order in the loss function. For example, if direction N is misclassified into S, the error is much severer than its misclassification into NE or NW. These errors should be considered differently. Therefore, we propose the COR scheme, by extending the ordinal binary decomposition technique in [42].

Let x be an instance and $y_x \in \mathcal{C}$ be its class. For COR, binary classifiers, $f_0, f_1, \ldots, f_{K/2-1}$, are used. Each binary classifier f_n is defined as

$$f_n(x) = \begin{cases} 1 & \text{if } y_x \in \{c_{(n+1)_K}, \ldots, c_{(n+K/2)_K}\} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $(n)_K$ denotes the modulo operator returning the remainder after the division of n by K. In other words, f_n divides $\mathcal{C}$ into two subsets of the same size (corresponding to the two semicircles), and determines whether the class of x is between $c_{(n+1)_K}$ and $c_{(n+K/2)_K}$ or not. For instance, in Figure 5, f_0 halves the eight directions into the blue and red sides. It outputs 1 if the direction is NE, E, SE, or S, and 0 otherwise.

From (2), it can be shown that

$$f_n = 1 - f_{n+K/2}, \tag{3}$$

$$f_n = f_{n+K}. \tag{4}$$

Due to the symmetry in (3) and the periodicity in (4), all classifiers f_n, $n \in \mathbb{Z}$, are determined by only $K/2$ classifiers, $f_0, f_1, \ldots, f_{K/2-1}$. Note that, in the linear ordinal regression [42], the classes in a line segment is divided into two parts. Therefore, for K-way classification, $K-1$ binary classifiers are required. In contrast, in the proposed COR, a circle is halved into two semicircles, as done in [45]. Consequently, only $K/2$ binary classifiers are needed.

During the training of the classifiers, f_n is assigned a binary ground truth value in (2). On the other hand, in testing, f_n yields a confidence value (*i.e.* softmax probability) between 0 and 1. Using these confidence values of the $K/2$ classifiers, class c_{k^*} of instance x is determined by

$$k^* = \underset{k \in \mathcal{C}}{\operatorname{argmax}} \sum_{n=1}^{K/2} f_{k-n}(x) \tag{5}$$

For example, suppose that $K = 8$ as in Figure 5. Ideally, when x has class c_2,

$$\begin{aligned}\sum_{n=1}^{4} f_{2-n}(x) &= f_1(x) + f_0(x) + f_{-1}(x) + f_{-2}(x) \\ &= f_1(x) + f_0(x) + 1 - f_3(x) + 1 - f_2(x) = 4\end{aligned}$$

which is no less than $\sum_{n=1}^{4} f_{k-n}(x)$ for all k. Thus, its class is declared correctly as c_2. Also, it can be shown that the decision rule in (5) is the maximum likelihood (ML) one [70], if each $f_{k-n}(x)$ represents the probability that x has one of the four directions as defined in (2).

C. SUPERVISED LEARNING OF FM-Net

The MCP layer connects DenseNet-121 and the FC layers using the two RoI pooling layers. Thus, the network in Figure 4 can accept a patch of arbitrary size. However, for effective training and inference, the size of a patch is normalized to 400 × 400 so that it contains a pedestrian at the center whose height is 200 pixels.

The overall loss function is defined as

$$L_{\text{FM}} = L_{\text{Dir}} + L_{\text{Spe}} + L_{\text{Act}} \tag{6}$$

where L_{Dir}, L_{Spe}, and L_{Act} are the losses for the direction, speed, and action classification, respectively. For L_{Dir}, the sum of binary cross entropies [44] is adopted. Specifically,

$$L_{\text{Dir}}(\mathbf{p}, \mathbf{q}) = -\sum_{n=0}^{3} \sum_{i=0}^{1} q_n^i \log p_n^i \tag{7}$$

where $\mathbf{p} = \{p_n^i : i = 0, 1 \text{ and } n = 0, 1, 2, 3\}$ is the softmax probability vector from the four binary classifiers f_n and $\mathbf{q} = \{q_n^i\}$ is the corresponding ground-truth binary vector. L_{Spe} is defined using two binary classifiers in a similar manner. L_{Act} is defined as

$$L_{\text{Act}}(\mathbf{p}, \mathbf{q}) = -\sum_{i=1}^{3} q_i \log p_i \tag{8}$$

where $\mathbf{p} = \{p_i\}$ is the softmax probability vector for the three actions and $\mathbf{q} = \{q_i\}$ is the ground-truth binary vector.

The network is trained via the stochastic gradient descent with a momentum of 0.9 and a batch size of 4 for 20 epochs. The learning rate is 10^{-4} for the first ten epochs and 10^{-5} for the remaining epochs. As initial parameters, the DenseNet-121 model [16] pre-trained on ImageNet [71] is used.

V. DOMAIN ADAPTATION OF FM-Net

When a source domain for training and a target domain for the FM inference are different, FM-Net may fail to estimate the FMs of test instances in the target domain accurately. In this work, we attempt to improve the generalization performance of FM-Net by adapting it to a new target domain in a semi-supervised manner. More specifically, FM-Net is trained in the semi-supervised domain adaptation setting, in which a sufficient number of labeled data are available in the source domain, a limited number of labeled data are in the target domain, and a large number of unlabeled data are in the target domain.

A. SEMI-SUPERVISED DOMAIN ADAPTATION

Let $\mathcal{D}^{\text{S}} = \{(\mathbf{x}_i^{\text{S}}, \mathbf{y}_i^{\text{S}})\}_{i=1}^{m_s}$ denote the set of labeled training data in the source domain, where $\mathbf{x}_i^{\text{S}}$ and $\mathbf{y}_i^{\text{S}}$ are the ith example and its motion label, respectively. Also, m_s is the number of training examples. In the target domain, suppose that we have a limited number of labeled data $\mathcal{D}^{\text{T}} = \{(\mathbf{x}_i^{\text{T}}, \mathbf{y}_i^{\text{T}})\}_{i=1}^{m_t}$, but many unlabeled data $\mathcal{D}^{\text{U}} = \{\mathbf{x}_i^{\text{U}}\}_{i=1}^{m_u}$. Here, note that $m_t \ll m_s$ and $m_t \ll m_u$. The objective is to train FM-Net to adapt the target domain effectively using the three sets $\mathcal{D}^{\text{S}}$, $\mathcal{D}^{\text{T}}$, and $\mathcal{D}^{\text{U}}$. It is obvious that the empirical distribution of data in the source domain $\mathcal{D}^{\text{S}}$ is different from that of data in the target domain $\mathcal{D}^{\text{T}}$ and $\mathcal{D}^{\text{U}}$. Also, as pointed out in [49], [72], even in the same target domain, the limited sampling of labeled data may result in an empirical distribution mismatch between $\mathcal{D}^{\text{T}}$ and $\mathcal{D}^{\text{U}}$. Therefore, we simultaneously minimize the distribution divergence between data in the source domain and the target domain and that between the labeled data and the unlabeled data in the target domain simultaneously, so that the distributions of all three $\mathcal{D}^{\text{S}}$, $\mathcal{D}^{\text{T}}$, and $\mathcal{D}^{\text{U}}$ are well aligned in the embedding space.

The loss function for the semi-supervised domain adaptation of FM-Net is defined as

$$L_{\text{SSDA}} = L_{\text{FM}} + \gamma L_d(\mathcal{D}^{\text{S}}, \mathcal{D}^{\text{T}}, \mathcal{D}^{\text{U}}) \tag{9}$$

where $L_d(\mathcal{D}^{\text{S}}, \mathcal{D}^{\text{T}}, \mathcal{D}^{\text{U}})$ is the distribution divergence loss with a weight parameter $\gamma = 0.1$. Here, L_{FM} is the FM classifier loss function that is defined in (6). The $\mathcal{H}$-divergence [57], [58], [61] is adopted to define $L_d(\mathcal{D}^{\text{S}}, \mathcal{D}^{\text{T}}, \mathcal{D}^{\text{U}})$, which is given by

$$L_d(\mathcal{D}^{\text{S}}, \mathcal{D}^{\text{T}}, \mathcal{D}^{\text{U}}) = d_{\mathcal{H}}(\mathcal{D}^{\text{S}}, \mathcal{D}^{\text{T}} \cup \mathcal{D}^{\text{U}}) + d_{\mathcal{H}}(\mathcal{D}^{\text{T}}, \mathcal{D}^{\text{U}}) \tag{10}$$

where $d_{\mathcal{H}}(\mathcal{D}^{\text{S}}, \mathcal{D}^{\text{T}} \cup \mathcal{D}^{\text{U}})$ is the $\mathcal{H}$-divergence between source and target samples and $d_{\mathcal{H}}(\mathcal{D}^{\text{T}}, \mathcal{D}^{\text{U}})$ is the $\mathcal{H}$-divergence between labeled and unlabeled samples in the target domain.

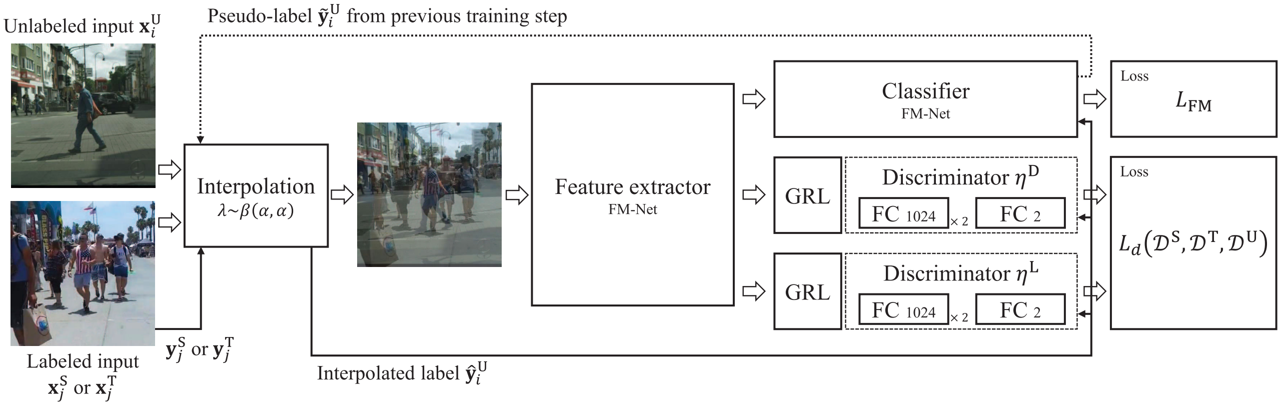

Unlabeled input $\mathbf{x}_i^{\mathrm{U}}$
Pseudo-label $\tilde{\mathbf{y}}_i^{\mathrm{U}}$ from previous training step
Interpolation
$\lambda \sim \beta(\alpha, \alpha)$
Feature extractor
FM-Net
Classifier
FM-Net
Loss
L_{FM}
GRL
Discriminator η^{D}
FC 1024
× 2
FC 2
GRL
Discriminator η^{L}
FC 1024
× 2
FC 2
Loss
$L_d(\mathcal{D}^{\mathrm{S}}, \mathcal{D}^{\mathrm{T}}, \mathcal{D}^{\mathrm{U}})$
Labeled input
$\mathbf{x}_j^{\mathrm{S}}$ or $\mathbf{x}_j^{\mathrm{T}}$
$\mathbf{y}_j^{\mathrm{S}}$ or $\mathbf{y}_j^{\mathrm{T}}$
Interpolated label $\hat{\mathbf{y}}_i^{\mathrm{U}}$

FIGURE 6. The architecture of the proposed FM-Net in the semi-supervised domain adaptation setting.

Following [58], the $\mathcal{H}$-divergence $d_{\mathcal{H}}(\mathcal{D}^{\mathrm{S}}, \mathcal{D}^{\mathrm{T}} \cup \mathcal{D}^{\mathrm{U}})$ is computed as

$$\begin{aligned} d_{\mathcal{H}}(\mathcal{D}^{\mathrm{S}}, \mathcal{D}^{\mathrm{T}} \cup \mathcal{D}^{\mathrm{U}}) &= 2\Bigg(1 - \min_{\eta^{\mathrm{D}} \in \mathcal{H}} \Bigg\{ \frac{1}{m_s} \sum_{\mathbf{x}_i \in \mathcal{D}^{\mathrm{S}}} I[\eta^{\mathrm{D}}(f(\mathbf{x}_i)) = 0] \\ &\quad + \frac{1}{(m_t + m_u)} \sum_{\mathbf{x}_i \in \mathcal{D}^{\mathrm{T}} \cup \mathcal{D}^{\mathrm{U}}} I[\eta^{\mathrm{D}}(f(\mathbf{x}_i)) = 1] \Bigg\}\Bigg) \end{aligned} \tag{11}$$

where $f(\cdot)$ denotes a feature extractor and $\eta^{\mathrm{D}} : f(\mathbf{x}_i) \rightarrow \{0, 1\}$ is a binary discriminator that predicts 1 for the source domain and 0 for the target domain. Also, $I[a]$ is the indicator function, which yields 1 if a statement a is true, and 0 otherwise. Thus, $\frac{1}{m_s} \sum_{\mathbf{x}_i \in \mathcal{D}^{\mathrm{S}}} I[\eta^{\mathrm{D}} f((\mathbf{x}_i)) = 0]$ and $\frac{1}{m_t + m_u} \sum_{x_i \in \mathcal{D}^{\mathrm{T}} \cup \mathcal{D}^{\mathrm{U}}} I[\eta^{\mathrm{D}}(\mathbf{x}_i) = 1]$ represent the prediction error rates of the discriminator η^{D}.

If the empirical distribution mismatch between the source and target domains is to be small, the discriminator η^{D} should be incapable of distinguishing source samples from target ones. In other words, the prediction error rates of the discriminator should be large, and thus the $\mathcal{H}$-divergence $d_{\mathcal{H}}(\mathcal{D}^{\mathrm{S}}, \mathcal{D}^{\mathrm{T}} \cup \mathcal{D}^{\mathrm{U}})$ should be low. Therefore, the prediction error rates of the discriminator are maximized to minimize the divergence. In other words, $\min d_{\mathcal{H}}(\mathcal{D}^{\mathrm{S}}, \mathcal{D}^{\mathrm{T}} \cup \mathcal{D}^{\mathrm{U}})$ is equivalent to

$$\begin{aligned} \max \min_{\eta^{\mathrm{D}} \in \mathcal{H}} \Bigg\{ & \frac{1}{m_s} \sum_{\mathbf{x}_i \in \mathcal{D}^{\mathrm{S}}} I[\eta^{\mathrm{D}}(f(\mathbf{x}_i)) = 0] \\ & + \frac{1}{(m_t + m_u)} \sum_{\mathbf{x}_i \in \mathcal{D}^{\mathrm{T}} \cup \mathcal{D}^{\mathrm{U}}} I[\eta^{\mathrm{D}}(f(\mathbf{x}_i)) = 1] \Bigg\}. \end{aligned} \tag{12}$$

By minimizing $d_{\mathcal{H}}(\mathcal{D}^{\mathrm{S}}, \mathcal{D}^{\mathrm{T}} \cup \mathcal{D}^{\mathrm{U}})$, the feature generator $f(\cdot)$ can be learned to yield the embedding space where the feature distributions of the source and target domains are well matched. Similarly, $d_{\mathcal{H}}(\mathcal{D}^{\mathrm{T}}, \mathcal{D}^{\mathrm{U}})$ is minimized to reduce the distribution gap between labeled and unlabeled samples in the target domain, which is equivalent to

$$\begin{aligned} \max \min_{\eta^{\mathrm{L}} \in \mathcal{H}} \Bigg\{ & \frac{1}{m_t} \sum_{\mathbf{x}_i \in \mathcal{D}^{\mathrm{T}}} I[\eta^{\mathrm{L}}(f(\mathbf{x}_i)) = 0] \\ & + \frac{1}{m_u} \sum_{\mathbf{x}_i \in \mathcal{D}^{\mathrm{U}}} I[\eta^{\mathrm{L}}(f(\mathbf{x}_i)) = 1] \Bigg\} \end{aligned} \tag{13}$$

where $\eta^{\mathrm{L}} : X \rightarrow \{0, 1\}$ is a binary discriminator, which predicts 1 for the labeled data and 0 for the unlabeled data.

To solve the max-min problems in (12) and (13), the adversarial training method is adopted using a gradient reverse layer (GRL) in [61], as shown in Figure 6. For the backpropagation in training, GRL takes gradients and invert the signs of the gradients. However, for the forward propagation, GRL produces the outputs, which are identical to the inputs. Thus, GRL maximizes the errors of the discriminators η^{D} and η^{L}.

B. FM-Net ARCHITECTURE IN DOMAIN ADAPTATION

Figure 6 shows the architecture of FM-Net in the semi-supervised domain adaptation setting. For the encoder, the same FM feature extractor in Figure 4 is used. The output feature is fed into three branches: the FM classifier in Figure 4 and the two discriminators η^{D} and η^{L}. Each discriminator is composed of three FC layers. Note that two GRLs are located between the feature extractor and the discriminators for the adversarial training.

For training, labeled data $\mathcal{D}^{\mathrm{S}} = \{(\mathbf{x}_i^{\mathrm{S}}, \mathbf{y}_i^{\mathrm{S}})\}_{i=1}^{m_s}$ and $\mathcal{D}^{\mathrm{T}} = \{(\mathbf{x}_i^{\mathrm{T}}, \mathbf{y}_i^{\mathrm{T}})\}_{i=1}^{m_t}$, and unlabeled data $\mathcal{D}^{\mathrm{U}} = \{(\mathbf{x}_i^{\mathrm{U}}, \tilde{\mathbf{y}}_i^{\mathrm{U}})\}_{i=1}^{m_u}$ are used. Note that $\tilde{\mathbf{y}}_i^{\mathrm{U}}$ is a pseudo-label of $\mathbf{x}_i^{\mathrm{U}}$, which is estimated by the FM classifier trained in the previous training step. For stable optimization of FM-Net, the cross-set sample augmentation method in [49] is employed, which reconstructs training data by interpolating two kinds of pairs: 1) labeled data in the source domain $\mathcal{D}^{\mathrm{S}}$ and unlabeled data in the target domain $\mathcal{D}^{\mathrm{U}}$ and 2) labeled data $\mathcal{D}^{\mathrm{T}}$ and unlabeled data $\mathcal{D}^{\mathrm{U}}$ in the target domain.

TABLE 4. Classification accuracies (%) according to variations in the MCP layer and the ordinal regression. The last row is the accuracies of the proposed algorithm.

	Object	Global	Direction	Direction+	Speed	Action
	✓		71.58	92.83	84.75	84.57
Without ordinal regression		✓	20.06	40.17	81.03	81.62
(multi-class classification)	✓	✓	73.22	94.37	86.61	84.88
	✓		73.40	**94.87**	86.84	84.57
With ordinal regression		✓	15.71	44.53	81.30	83.30
(proposed)	✓	✓	**74.04**	**94.87**	**88.56**	**86.34**

First, each ith example in $\mathcal{D}^{\mathrm{U}}$ is superposed with randomly sampled jth example in $\mathcal{D}^{\mathrm{S}}$ to generate an interpolated example, given by

$$\hat{\mathbf{x}}_i^{\mathrm{U}} = \lambda \mathbf{x}_j^{\mathrm{S}} + (1-\lambda)\mathbf{x}_i^{\mathrm{U}} \tag{14}$$

$$\hat{\mathbf{y}}_i^{\mathrm{U}} = \lambda \mathbf{y}_j^{\mathrm{S}} + (1-\lambda)\tilde{\mathbf{y}}_i^{\mathrm{U}} \tag{15}$$

$$\hat{z}_i^{\mathrm{D}} = \lambda \cdot 1 + (1-\lambda)\cdot 0 \tag{16}$$

where $\hat{\mathbf{x}}_i^{\mathrm{U}}$ is the interpolated sample, $\hat{\mathbf{y}}_i^{\mathrm{U}}$ is its interpolated label, and $\hat{z}_i^{\mathrm{D}}$ is its interpolated label for the discriminator η_i^{D}. Also, λ is a random variable generated from a prior β distribution, *i.e.* $\lambda \sim \beta(\alpha,\alpha)$ with $\alpha = 0.1$. In this case, the loss for the discriminator η_i^{L} is not computed.

Similarly, the interpolation between $\mathcal{D}^{\mathrm{U}}$ and $\mathcal{D}^{\mathrm{T}}$ is defined as

$$\hat{\mathbf{x}}_i^{\mathrm{U}} = \lambda \mathbf{x}_j^{\mathrm{T}} + (1-\lambda)\mathbf{x}_i^{\mathrm{U}} \tag{17}$$

$$\hat{\mathbf{y}}_i^{\mathrm{U}} = \lambda \mathbf{y}_j^{\mathrm{T}} + (1-\lambda)\tilde{\mathbf{y}}_i^{\mathrm{U}} \tag{18}$$

$$\hat{z}_i^{\mathrm{L}} = \lambda \cdot 1 + (1-\lambda)\cdot 0 \tag{19}$$

$$\hat{z}_i^{\mathrm{D}} = 0 \tag{20}$$

where $\hat{z}_i^{\mathrm{L}}$ is an interpolated label for the discriminator η^{L}.

VI. EXPERIMENTAL RESULTS

This section presents the experimental results of the proposed FM estimation algorithm. First, the performances of FM-Net for pedestrian, car, and animal instances in the FM dataset are evaluated. Second, FM-Net is assessed in the semi-supervised domain adaptation setting, where training and test data are from different domains. Finally, the efficacy of FM-Net is demonstrated on three applications: single object tracking, multiple object tracking, and crowd analysis. To evaluate FM estimation results quantitatively, the classification accuracy (%) is used.

Running Time: With a personal computer with an Intel Core i7-7700K CPU and an NVIDIA GeForce GTX 1080Ti GPU, training for supervised learning of FM-Net takes about 1.96 hours (5.86 minutes per epoch). Also, training for semi-supervised domain adaptation takes about 3 hours (9.05 minutes per epoch). For the test, the FM estimation time is 9.86 milliseconds per instance.

Parameters: For experiments, the parameter h is set to 200, which means that the input size is 400 × 400. Since DenseNet-121 decreases the height and width of an input patch to one-sixteenth, 400 × 400 is a proper size to extract features at the MCP layer. Also, to determine the parameter γ in (9), γ is first constrained to be less than 1, since the loss for FM L_{FM} is more important than L_d. Then, γ is empirically set to 0.1 to yield the best performance. The parameter α is set to 0.1 by following [49].

A. SINGLE-IMAGE FUTURE MOTION ESTIMATION

1) PEDESTRIANS

Table 4 compares the accuracies according to the combination of the MCP layer and the ordinal regression scheme. 'Direction,' 'Speed,' and 'Action' are the classification accuracies of the FM direction, the FM speed, and the FM action, respectively. 'Object' and 'Global' denote the object and global context features, respectively. Thus, the configuration with both 'Object' and 'Global' checked denotes that the proposed MCP layer is used. Also, FM-Net is tested in two ways: with and without ordinal regression. It is observed that the usage of both object and global context features, *i.e.* the output of the MCP layer, improves the 'Direction,' 'Speed,' and 'Action' accuracies regardless of whether ordinal regression is used or not. Furthermore, notice that the proposed ordinal regression scheme with the MCP layer provides the best performances in all three cases of 'Direction,' 'Speed,' and 'Action.' This indicates that the ordinal regression scheme trains the feature extractor of FM-Net more effectively.

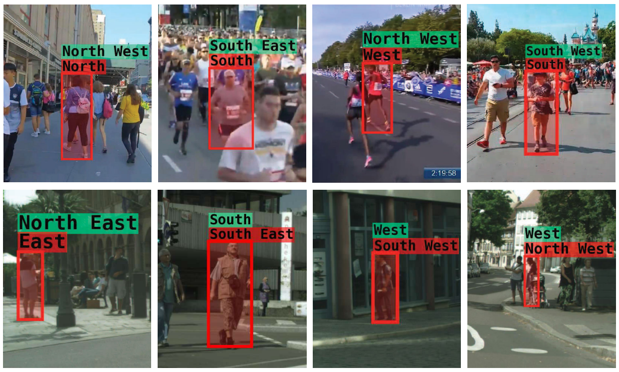


FIGURE 7. Direction classification. Green labels are the ground-truth, while red ones are predicted directions. In these cases, the ground-truth and predicted directions are adjacent to each other.

In Table 4, 'Direction+' means the accuracy when the estimated direction is regarded as correct if it is identical with or adjacent to the ground-truth direction. For example, for the ground-truth direction N, an estimated direction NE, N, or NW is correct in the 'Direction+' accuracy. Figure 7 shows examples in which the ground-truth and predicted

TABLE 5. Comparison of direction classification accuracies (%) of the proposed algorithm with conventional algorithms.

	HOG [73]	ACF [74]	Faster R-CNN [75]	DenseNet-121 [16]	Gao *et al.* [15]	Proposed
Direction	43.99	41.53	<u>46.80</u>	31.32	12.4	**74.04**
Direction+	<u>73.17</u>	69.31	71.31	58.56	38.5	**94.87**

directions are adjacent. In these examples, even a human being cannot easily quantize the true direction into one of the two classes by looking at a single image only. This ambiguity is taken into account to define the metric of 'Direction+.' The proposed algorithm yields the 'Direction+' accuracy that is as high as 94.87%.

As mentioned previously, there is no algorithm that has exactly the same objective as the proposed algorithm. Gao *et al.*'s algorithm [15], which estimates optical flow vectors from a single image, is the most similar to the proposed algorithm. Thus, we obtain optical flow vectors using Gao *et al.*'s algorithm, compute the average of the optical flow vectors in the bounding box of a pedestrian, and regard the average vector as the instance-level future motion vector of the pedestrian. Note that the trained network in [15] is used without re-training on the FM dataset. This is because [15] requires optical flow vectors for its training, but the FM dataset provides only motion attributes for object bounding boxes in sparsely sampled frames. For more comparison, we test various features for pedestrians in Table 5. We implement HOG [73] and ACF [74] to extract handcrafted features. In addition, CNN features are extracted from VGG-16 in the faster R-CNN [75] and DenseNet-121 [16], trained on ImageNet [71], respectively. For these handcrafted or CNN features, an SVM is adopted to train the direction classifier. In Table 5, notice that the proposed FM-Net outperforms these comparison methods significantly. Gao *et al.* fails to predict reliable flow vectors on the FM dataset, yielding very low directional classification accuracies.

Figure 8 shows FM estimation results for images sampled from YouTube. The top three rows illustrate that FM-Net can estimate motions correctly even when scenes are cluttered or crowded. The proposed FM-Net provides correct results by efficiently exploiting clear pose information and sufficient semantic information on the background. On the contrary, the bottom three rows provide failure cases for direction, speed, and action classes. In the first false direction case, it is predicted to be the opposite of the ground-truth direction. The mask on her face confuses the network since there is almost no pedestrian wearing a mask in the training data. In the second false direction case, the appearance of the pedestrian is confusing. In the third case, the predicted direction W is adjacent to the ground-truth direction NW. Note that it is quite challenging to distinguish adjacent directions. The first false speed case is due to occlusion. In the second case, the raised arm causes the incorrect prediction, since it makes the pedestrian seem to walk slowly. In the third case, FM-Net falsely declares 'fast' because his large step is confusing. The first false action case is due to that the pedestrian is walking right beside the sidewalk. In the second case, the removed crosswalk line confuses FM-Net to declare 'sidewalk' incorrectly. In the third case, the pedestrian is falsely predicted to be on a sidewalk, since there are many people surrounding him.

Figure 9 shows the results on images from the CityPersons dataset. In the top three rows, FMs are predicted successfully, even when the pedestrians are far from camera and captured small. On the other hand, the bottom three rows present failure cases. The first and second false direction cases are due to severe occlusion. In the third false direction case, NE is predicted as its adjacent direction E. In the first and second false speed cases, slowly walking people surrounding the target pedestrian lead to the incorrect prediction. In the third false speed case, the pedestrian's small step misleads FM-Net to the 'stop' class. The first false action case is a similar example to the corresponding one in Figure 8. In the second case, the pedestrian is located in boundaries of 'crosswalk,' 'sidewalk,' and 'jaywalk.' In the last case, the action is falsely declared as 'crosswalk' because there are the crosswalk and the traffic light right behind the pedestrian.

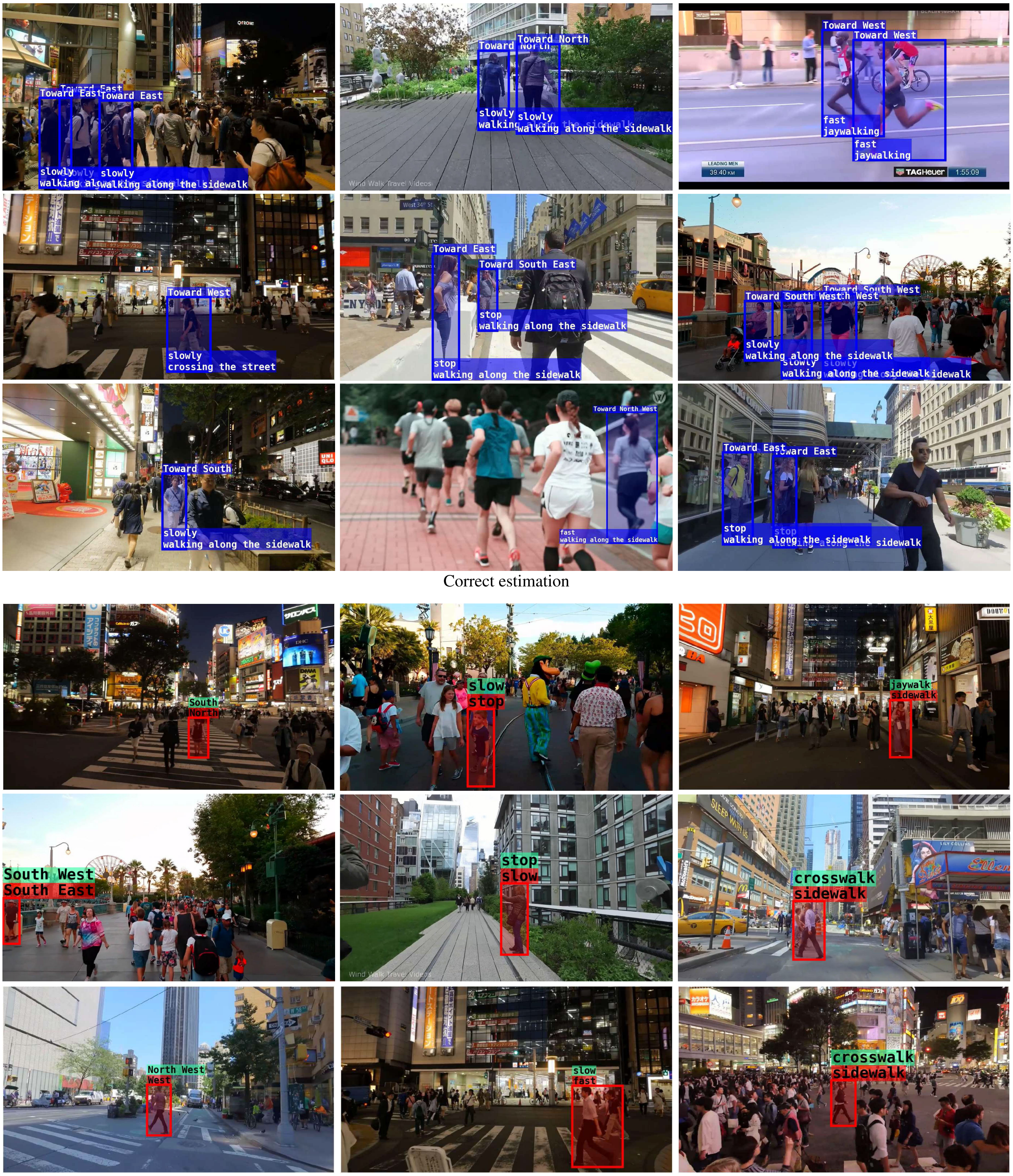
Figure 10 qualitatively shows the efficacy of the proposed MCP layer. Green labels are the results of FM-Net with the MCP layer, while red ones are the results using object features only. The MCP layer exploits global scene information as well as object appearance. In (a) and (b), sidewalk directions provide global contexts, since a pedestrian usually walks along a sidewalk. In (c) and (d), the place information, where the pedestrians are, is used by the MCP layer; people usually stop at the bus stop or walk down the stairs. In (e) and (f), the neighboring crowds are used as global contexts. In (g) and (h), the object information within the bounding boxes is confusing, but the global scene contexts help to estimate the action classes correctly.

TABLE 6. Classification accuracies (%) of the proposed single-image FM estimation on car and animal instances.

	Method	Direction	Direction+	Speed	Action
Cars	Baseline	89.6	98.5	97.0	94.2
	Proposed	**89.9**	**98.7**	**97.4**	**94.5**
Animals	Baseline	86.10	97.05	73.94	-
	Proposed	**87.69**	**97.73**	**74.17**	-

2) CARS AND ANIMALS

Table 6 compares the performances for car and animal instances in the FM dataset. 'Baseline' denotes the results



Correct estimation

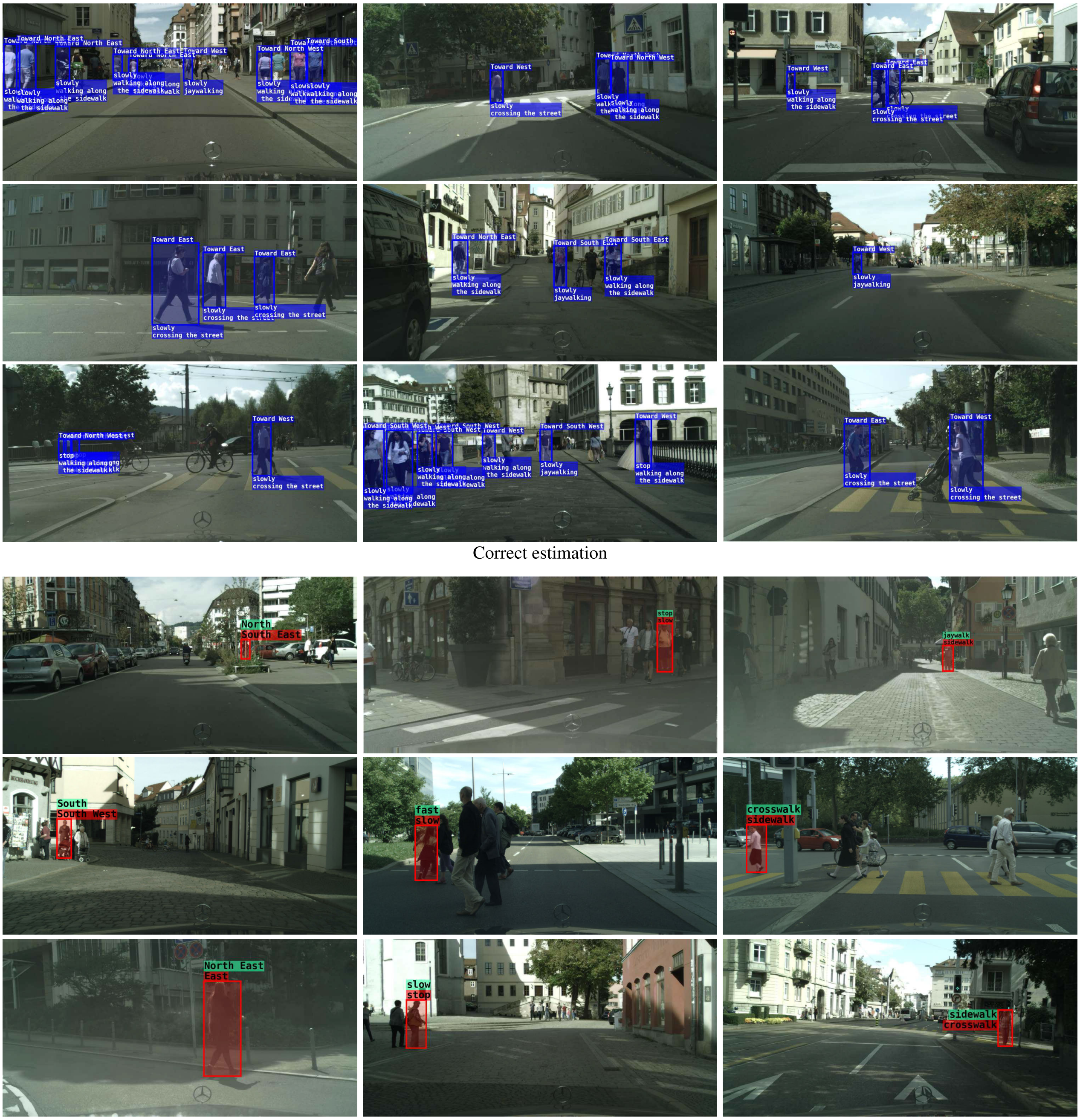
False direction

False speed

False action

FIGURE 8. FM estimation results of YouTube data in the FM dataset. The first three rows present correct estimation results. The bottom three rows show failure cases, where green and red labels are the ground-truth and predicted classes, respectively.

of FM-Net without the proposed ordinal regression scheme, while 'Proposed' is the results of the proposed algorithm. We see that the proposed ordinal regression improves the performances in all classification tasks. The proposed algorithm yields remarkable performances on both car and animal instances, except for the speed classification of animals. The classification of animal speeds is more challenging than that of pedestrian or car instances. Especially, it is often ambiguous to distinguish 'slow' from 'fast' in still images for animals.



Correct estimation

False direction

False speed

False action

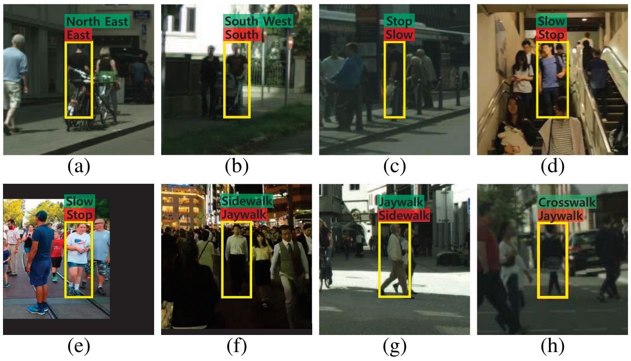
FIGURE 9. **FM estimation results of CityPersons data in the FM dataset. The top three rows present correct estimation results. The bottom three rows show failure cases, where green and red labels are the ground-truth and predicted classes, respectively.**

Figure 11 shows qualitative FM estimation results for car instances. In the correct estimation examples, FM-Net differentiates the 'stop' class from the 'go straight' class correctly based on the scene contexts, although the differentiation is very challenging due to the rigid shapes of cars. The right case of the false direction is due to the partial disappearance of the car. In the right case of the false speed, the speed is classified as 'keep' since the parked car is incorrectly regarded as 'go straight.' In the left case of the false action, the action is misclassified as 'go straight' even though the car does not finish turning right yet. Also, in the right case of the false action, the parked car is falsely predicted to be moving.

For animal instances, Figure 12 presents correct results in the top three rows and failure cases in the bottom three rows. From left to right, the columns contain the results for cats, dogs, and horses, respectively. Some results demonstrate that an animal's head is essential information for its FM prediction. In the first row of the false case, the directions

TABLE 7. Classification accuracies (%) on pedestrian instances according to training methods. In 'LS+LT+UT' and 'LS+LT+UT*,' the semi-supervised domain adaptation is used.

Source domain	Target domain	Training Method	Direction	Direction+	Speed	Action
CityPersons CPDB	YouTube	LS	57.42	87.21	38.24	36.55
		LS+LT	58.02	90.83	64.29	53.08
		LS+LT+UT	61.04	89.63	64.41	54.28
		LS+LT+UT* (Proposed)	**62.12**	**90.95**	**65.02**	**55.37**
YouTube CPDB	CityPersons	LS	56.27	93.26	80.30	65.23
		LS+LT	61.85	93.89	77.56	67.12
		LS+LT+UT	62.07	92.10	79.66	61.22
		LS+LT+UT* (Proposed)	**62.91**	**94.10**	**82.61**	**74.29**



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 10. The efficacy of the proposed MCP layer. Green and red labels indicate estimation results using the MCL layer feature and the object feature only, respectively.

of the dog and the horse are falsely estimated due to the their head directions. Also, in the second row of the failure case, the dog and cat, whose heads are not captured clearly, are falsely predicted to be moving in the opposite direction of the ground-truth. The bottom failure cases show that the animals approaching the camera do not provide sufficient information, such as leg shapes and directions, for predicting FMs reliably.

B. SEMI-SUPERVISED DOMAIN ADAPTATION

In order to evaluate the efficacy of the proposed semi-supervised domain adaptation method, we set training and test data so that they are from different domains. In this test, car instances are excluded, since all cars in the FM dataset are from the same domain of the KITTI object dataset [69].

1) PEDESTRIANS

Note that the pedestrian instances are from the YouTube, CityPersons, and CPDB datasets. Two combinations are considered. First, CityPersons and CPDB are set as the source domain, while YouTube as the target domain. Second, YouTube and CPDB are regarded as the source domain, and CityPersons as the target domain. In both cases, it is assumed that, to train FM-Net, there are a sufficient number of labeled data in the source domain but there are only a limited number of labeled data in the target domain. Specifically, only about 4% of the available data in the target domain, *i.e.* 149 and 123 instances in YouTube and CityPersons, are used to train FM-Net, respectively.

Table 7 lists the classification accuracies for test instances in the target domain according to data combinations and training methods. In Table 7, 'LS' denotes the performances of FM-Net trained using the labeled data in the source domain only, and 'LS+LT' represents the results of FM-Net trained using the labeled data in both source and target domains. Note that the proposed semi-supervised domain adaptation learning is not performed in 'LS' and 'LS+LT.' On the other hand, 'LS+LT+UT' and 'LS+LT+UT*' denote the performances of FM-Net with the semi-supervised domain adaptation using the discriminator η^D only and both discriminators η^D and η^L, respectively. Note that 'LS+LT+UT*', which is the proposed algorithm, yields the highest accuracies for all FM tests.

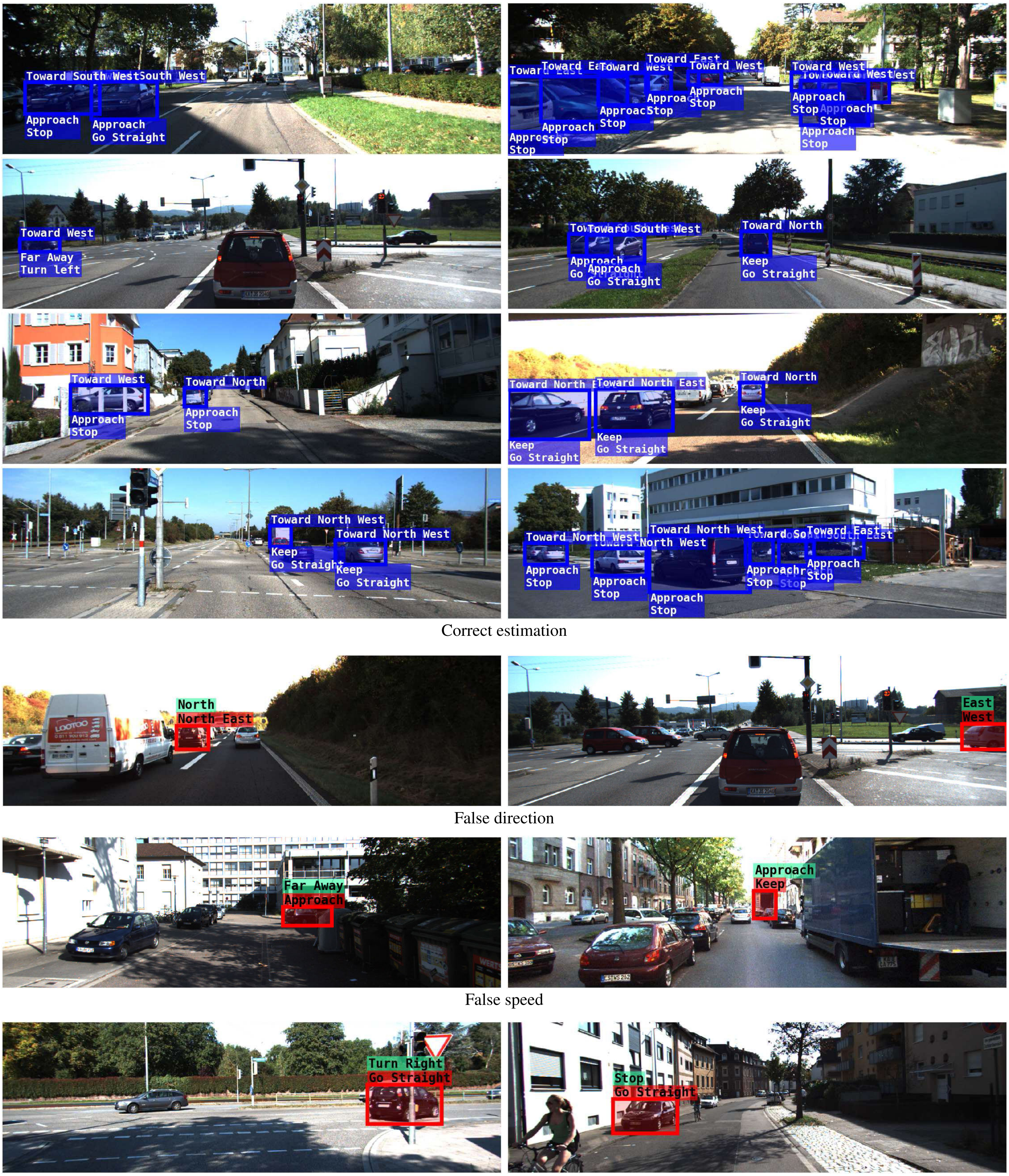
In 'LS+LT+UT' and 'LS+LT+UT*,' the proposed semi-supervised domain adaptation uses the unlabeled data to improve the FM estimation performances on the target domain. Especially, the usage of both discriminators η^D and η^L significantly improves the classification accuracies by aligning feature distributions between the source and target domains and between the labeled and unlabeled data in the target domain simultaneously.

2) ANIMALS

Next, the cat, dog, and horse categories are regarded as different domains, and they are divided into source and target domains. Three combinations for source and target domains are considered, as listed in Table 8. For each target domain, it is assumed that only 45 instances are labeled. It is observed from Table 8 that the adversarial training method for semi-supervised domain adaptation, *i.e.* 'LS+LT+UT*,' enhances the FM estimation accuracies in all classification tasks and in all combinations.

VII. APPLICATIONS

This section introduces three applications of the proposed algorithm: single object tracking, multiple object tracking, and crowd analysis. Using results of FM direction classification and FM speed classification, the conventional single and multiple object trackers [4], [5] can be made more efficient. To demonstrate this, the efficacy of FM estimation in



Correct estimation

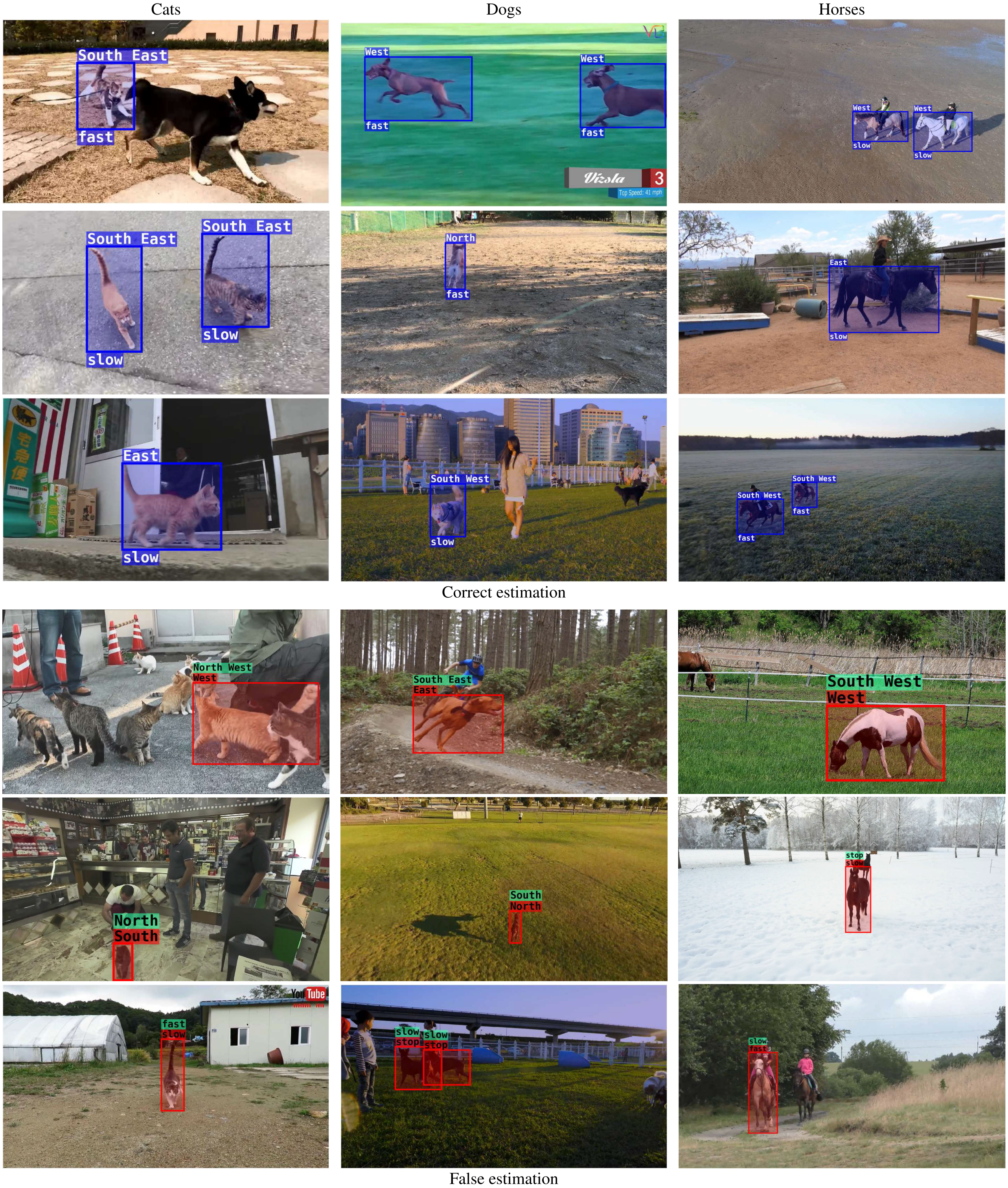
False direction

False speed

False action

FIGURE 11. FM estimation results of car instances. The top four rows show correct results. The bottom three rows are failure cases, where green labels are the ground-truth and red labels are predicted classes.

single and multiple object tracking is evaluated quantitatively. Also, based on FM direction classification, a crowd is partitioned into several clusters and the group direction of each cluster is predicted. Note that, in these three applications, we focus on pedestrian instances. Thus, the FM-Net trained on the FM pedestrian dataset is used in this section.



Correct estimation

False estimation

FIGURE 12. FM estimation results of animal instances. The top three rows show correct results. The bottom three rows are failure cases, where green labels are the ground-truth and red ones are predicted classes.

A. SINGLE OBJECT TRACKING

Single object tracking is a task to estimate the location of a target object in a next frame. In general, single object tracking methods attempt to find the optimal bounding box within a search region in the next frame, which has the most similar appearance to the target object in the current frame. In this regard, FM results of the target object are used to reduce the search range to boost the efficiency of single object tracking.

TABLE 8. Classification accuracies (%) on animal instances according to training methods.

Source domain	Target domain	Training Method	Direction	Direction+	Speed
Dogs, Horses	Cats	LS	74.54	89.12	60.65
		LS+LT	75.46	91.20	59.03
		LS+LT+UT	76.62	90.05	60.88
		LS+LT+UT* (Proposed)	**77.55**	**92.59**	**62.96**
Horses, Cats	Dogs	LS	78.73	**93.16**	65.57
		LS+LT	79.24	92.91	63.54
		LS+LT+UT	**79.49**	**93.16**	67.85
		LS+LT+UT* (Proposed)	**79.49**	**93.16**	**68.35**
Cats, Dogs	Horses	LS	67.00	86.52	64.79
		LS+LT	70.02	90.34	68.81
		LS+LT+UT	71.23	89.13	69.42
		LS+LT+UT* (Proposed)	**74.25**	**91.55**	**69.42**

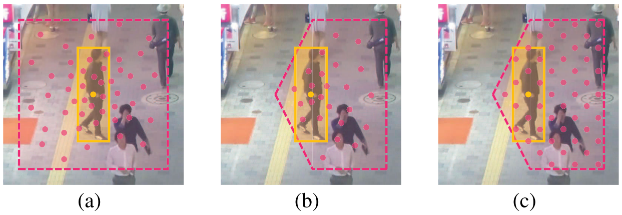


FIGURE 13. Comparison of sampled search points, depicted by red dots, in (a) MDNet, (b) MDNet+FM, and (c) CDT+FM. The points in (a) and (b) are sampled from a Gaussian distribution, while those in (c) are from a uniform distribution.

As a baseline tracker, MDNet [4], which provides competitive performances in several tracking benchmarks [76]–[78], is employed. We follow the details in [4] to sample box candidates from a Gaussian distribution within the search range.

To reduce the search region, predicted FM direction and speed are exploited. More specifically, the search region is narrowed to a fan-shaped area in the predicted FM direction. The angle of the fan-shaped area is set to 135°, which includes the two directions adjacent to the predicted direction. Note that the accuracy of 'Direction+' is higher than 90% in most cases. We use the same maximum distance from the target to a search candidate as the baseline does. Figure 13(a) and (b) compare the sampling strategies of the baseline MDNet and the proposed 'MDNet + FM.' Also, when the FM speed is 'stop,' the four times smaller square than the baseline is adopted because the pedestrian is expected to be not far from the current location in the next frame. The performance of MDNet+FM is evaluated on the object tracking benchmark (OTB) dataset [78] and the temple color 128 (TC128) dataset [79]. Only the sequences whose target objects are pedestrians are used. OTB and TC128 have 22 and 23 such sequences, respectively. After removing duplicated ones, there are 33 pedestrian sequences in total. To measure the tracking performance quantitatively, precision (PR) and success rate (SR) [78] are used.

TABLE 9. Comparison of tracking performances of MDNet [4] and MDNet+FM on the pedestrian sequences in the TC128 and OTB datasets.

Setting	Method	# Samples	PR	SR	fps
I	MDNet	128	0.845	0.589	1.67
	MDNet+FM	85	**0.848**	**0.598**	**2.75**
II	MDNet	192	**0.871**	**0.614**	1.51
	MDNet+FM	128	0.849	0.595	**2.45**
III	MDNet	256	**0.883**	**0.616**	1.41
	MDNet+FM	171	0.830	0.582	**2.24**
IV	MDNet	320	0.875	0.618	1.23
	MDNet+FM	213	**0.893**	**0.623**	**2.15**
V	MDNet	384	0.837	0.584	1.13
	MDNet+FM	256	**0.872**	**0.613**	**1.98**

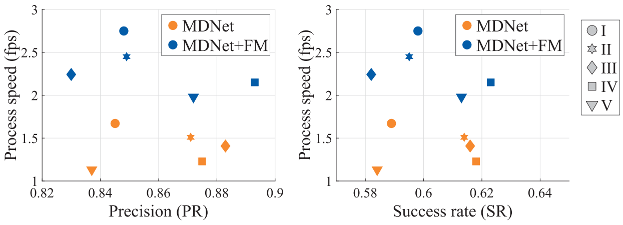


FIGURE 14. PR and SR scores versus tracking speeds on the pedestrian sequences in the TC128 and OTB datasets.

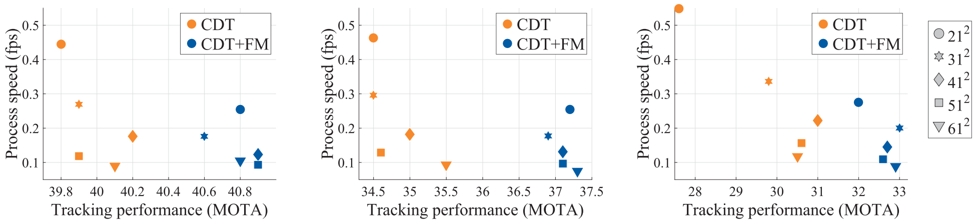
Table 9 compares the performances of MDNet+FM and the baseline MDNet. Note that '# Samples' denotes the number of search candidates. Both MDNet+FM and MDNet adopt the same Gaussian sampling, but MDNet+FM reduces the search region. Consequently, in the same 'Setting' in Table 9, MDNet+FM searches about 33% fewer search candidates than MDNet does. However, MDNet+FM provides comparable or even better tracking performances than MDNet, while improving the tracking speed. It is observed from Table 9 that, in setting IV and V, MDNet+FM provides slightly higher PR and SR scores than MDNet, but it is about 175% faster. Figure 14 plots PR and SR scores versus tracking speeds in terms of frames per second (fps). Notice that MDNet+FM is significantly faster than MDNet at similar PR or SR scores.

B. MULTIPLE OBJECT TRACKING

In multiple object tracking (MOT) sequences [80], objects tend to move slowly and smoothly between consecutive frames. Based on this observation, Bochinski *et al.* [6]

TABLE 10. Comparison of CDT+FM with the baseline CDT on the MOT17 dataset at low video frame rates. The reported fps scores are the processing speeds in frames per second (fps).

Frame rate	# Samples / Method	21^2 MOTA	21^2 fps	31^2 MOTA	31^2 fps	41^2 MOTA	41^2 fps	51^2 MOTA	51^2 fps	61^2 MOTA	61^2 fps
5 fps	CDT	39.8	**0.445**	39.9	**0.270**	40.2	**0.176**	39.9	**0.118**	40.1	**0.090**
	CDT+FM	**40.8**	0.255	**40.6**	0.176	**40.9**	0.123	**40.9**	0.093	**40.8**	0.104
2 fps	CDT	34.5	**0.463**	34.5	**0.300**	35.0	**0.182**	34.6	**0.128**	35.5	**0.094**
	CDT+FM	**37.2**	0.255	**36.9**	0.178	**37.1**	0.131	**37.1**	0.097	**37.3**	0.075
1 fps	CDT	27.6	**0.548**	29.8	**0.337**	31.0	**0.222**	30.6	**0.157**	30.5	**0.117**
	CDT+FM	**32.0**	0.275	**33.0**	0.200	**32.7**	0.145	**32.6**	0.110	**32.9**	0.089



(a) (b) (c)

FIGURE 15. MOTA scores versus tracking speeds on the MOT17 dataset at video frame rates of (a) 5 fps and (b) 2 fps, and (c) 1 fps.

proposed an MOT algorithm, which depends on only the intersection-over-union (IOU) ratio between the bounding boxes of a target object and a search candidate. However, in a low frame rate video (*e.g.* $<$ 10 fps), their algorithm may fail because there can be abrupt changes between frames. To achieve reliable MOT in low frame rate videos, direction and speed results of FM-Net are applied to a more sophisticated MOT algorithm, CDT [5], which is a tracking-by-detection method.

As shown in Figure 13(c), the search region of CDT is narrowed in the same way as MDNet+FM. CDT adopts a uniform distribution for sampling search points. For CDT+FM, the search region is reduced but the number of search points is maintained to increase the sampling density. To assess CDT+FM, the MOT17 benchmark [80] is used. Four out of seven video sequences in MOT17 have camera movements, which make the FM-based reduction of search range invalid. To address this problem, the background motion compensation is performed to all sequences using an affine transformation based on the BRISK keypoint matching [81]. Then, the MOT accuracy (MOTA), which is one of the most comprehensive metrics in the benchmark [80], is computed.

Table 10 compares the performance of CDT+FM with those of CDT. The number of search points is set to 21^2, 31^2, 41^2, 51^2, or 61^2. Since CDT+FM performs the FM prediction and the background compensation, the processing speed of CDT+FM is slower than that of CDT at the same number of search points. However, for all '# Samples' and 'Frame rate,' CDT+FM provides more accurate tracking results. In Figure 15, MOTA scores versus processing speeds are plotted according to the numbers of search points. It exhibits that CDT+FM yields significantly higher MOTA scores than CDT at similar processing speeds.

C. CROWD ANALYSIS

The proposed FM estimation algorithm is also applied to crowd analysis in a single image. By employing the estimated direction of each pedestrian in a crowd, the crowd is partitioned into several clusters and the group direction of each cluster is predicted. For the clustering, the simple k-means algorithm [82] is used. To compute the distance between two instances, the weighted distance $D = D_{\text{Euc}} + \lambda D_{\text{FM}}$ is used, where D_{Euc} is the Euclidean distance between the instances and D_{FM} is the cyclic difference of the directional indices. For example, D_{FM} between adjacent directions is 1, and the maximum D_{FM} is 4 for opposite directions. Also, $\lambda = 40$ is a weight parameter. After the clustering, the group direction of each cluster is obtained. To this end, for each direction, the sum of the directional probabilities of instances within a cluster is computed. Then, the direction, whose sum of the probabilities is maximal, is selected as the group direction.

We capture various crowded scenes using a surveillance camera and detect pedestrians using the YOLOv3 detector [68]. Figure 16 shows some of the crowd analysis results. In each column in Figure 16, the top image shows predicted directions of pedestrians. Even though the scene is crowded, the directions are predicted faithfully. By employing the directional information, the clustering is performed with the number of clusters $k = 5$ in the bottom image. Note that the bottom image is easier to understand than the top image since it conveys information compactly through the data clustering. However, the grouping is not perfect. In Figure 16(b), group 2 contains one pedestrian who is not

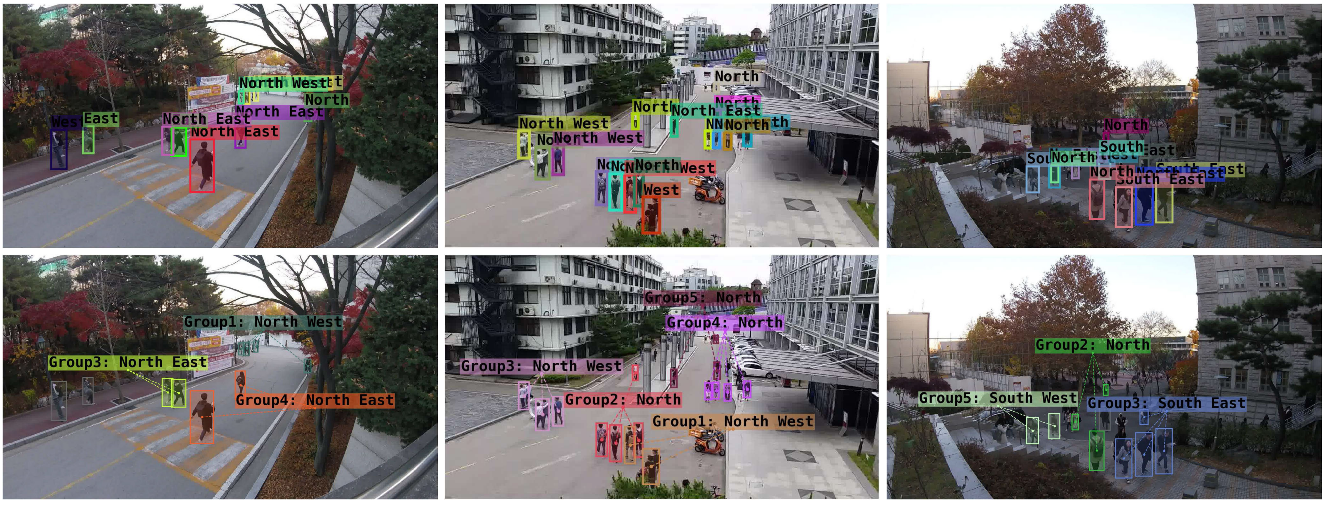


FIGURE 16. Crowd analysis based on FM: In each column, the top image shows predicted directions of instances, and the bottom one is the corresponding clustering result with $k = 5$. If a cluster contains only one pedestrian, its label is not rendered.

near the other three pedestrians in the same group. Also, one person who should be in group 2 is falsely declared to belong to group 1. A more sophisticated clustering technique is required to achieve more meaningful and reliable clustering, which is a future research issue.

VIII. CONCLUSIONS

A novel single-image FM estimation algorithm at the instance level was proposed in this paper. Using the MCP layer, the proposed algorithm extracts object and global context features for faithful FM estimation. The proposed algorithm performs three classification tasks to determine the future direction, speed, and action of an instance. Especially, the COR scheme was proposed for the ordinal regression of future direction. Also, FM-Net was trained in a semi-supervised domain adaptation setting to achieve reliable FM estimation, even when a source domain in the training process and a target domain in the inference process were different. Experimental results demonstrated that the proposed algorithm yields reliable FM estimation performance and can be used for single and multi object tracking and crowd analysis. Moreover, the proposed algorithm can be used for estimating the FMs of cars, cats, dogs, horses, as well as those of pedestrians. It is a future research issue to expand the FM-Net to estimate finer quantized future directions. Another issue is to develop a more sophisticated crowd analysis algorithm using FM estimation.

REFERENCES

[1] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.

[2] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2462–2470.

[3] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A lightweight convolutional neural network for optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8981–8989.

[4] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4293–4302.

[5] H.-U. Kim and C.-S. Kim, "CDT: Cooperative detection and tracking for tracing multiple objects in video sequences," in *Proc. ECCV*, 2016, pp. 851–867.

[6] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.

[7] B. Korbar, D. Tran, and L. Torresani, "SCSampler: Sampling salient clips from video for efficient action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6232–6242.

[8] Y.-H. Kwon and M.-G. Park, "Predicting future frames using retrospective cycle GAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1811–1820.

[9] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3703–3712.

[10] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learning image and video compression through spatial-temporal energy compaction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10071–10080.

[11] T. Yagi, K. Mangalam, R. Yonetani, and Y. Sato, "Future person localization in first-person videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7593–7602.

[12] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *Proc. ECCV*, 2012, pp. 201–214.

[13] W.-C. Ma, D.-A. Huang, N. Lee, and K. M. Kitani, "Forecasting interactive dynamics of pedestrians with fictitious play," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 774–782.

[14] R. Mottaghi, H. Bagherinezhad, M. Rastegari, and A. Farhadi, "Newtonian image understanding: Unfolding the dynamics of objects in static images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3521–3529.

[15] R. Gao, B. Xiong, and K. Grauman, "Im2Flow: Motion hallucination from static images for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5937–5947.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. ICCV*, Oct. 2017, pp. 2961–2969.

[18] J.-T. Lee, H.-U. Kim, C. Lee, and C.-S. Kim, "Semantic line detection and its applications," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3229–3237.

[19] K. Gavrilyuk, A. Ghodrati, Z. Li, and C. G. M. Snoek, "Actor and action video segmentation from a sentence," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5958–5966.

[20] S.-H. Lee, W.-D. Jang, and C.-S. Kim, "Tracking-by-segmentation using superpixel-wise neural network," *IEEE Access*, vol. 6, pp. 54982–54993, 2018.

[21] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

[22] J.-H. Lee and C.-S. Kim, "Monocular depth estimation using relative depth maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9729–9738.

[23] W.-D. Jang and C.-S. Kim, "Interactive image segmentation via backpropagating refinement scheme," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5297–5306.

[24] S.-H. Lee, H.-U. Kim, and C.-S. Kim, "ELF-Nets: Deep learning on point clouds using extended Laplacian filter," *IEEE Access*, vol. 7, pp. 156569–156581, 2019.

[25] K. Lim, N.-H. Shin, Y.-Y. Lee, and C.-S. Kim, "Order learning and its application to age estimation," in *Proc. ICLR*, 2020, pp. 1–20.

[26] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.

[27] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A diverse dataset for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3213–3221.

[28] [Online]. Available: http://youtube.com

[29] K.-R. Kim, W. Choi, Y. J. Koh, S.-G. Jeong, and C.-S. Kim, "Instance-level future motion estimation in a single image based on ordinal regression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 273–282.

[30] J. Walker, A. Gupta, and M. Hebert, "Patch to the future: Unsupervised visual prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3302–3309.

[31] Y.-W. Chao, J. Yang, B. Price, S. Cohen, and J. Deng, "Forecasting human dynamics from static images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 548–556.

[32] S. L. Pintea, J. C. van Gemert, and A. W. M. Smeulders, "Déjà Vu: Motion prediction in static images," in *Proc. ECCV*, 2014, pp. 172–187.

[33] J. Walker, A. Gupta, and M. Hebert, "Dense optical flow prediction from a static image," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2443–2451.

[34] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *Proc. ECCV*, 2016, pp. 835–851.

[35] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Flow-grounded spatial-temporal video prediction from still images," in *Proc. ECCV*, 2018, pp. 600–615.

[36] K. Hrbacek and T. Jech, *Introduction to Set Theory*. New York, NY, USA: Dekker, 1984.

[37] P. A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro, and C. Hervás-Martínez, "Ordinal regression methods: Survey and experimental study," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 127–146, Jan. 2016.

[38] K. Crammer and Y. Singer, "Online ranking by projecting," *Neural Comput.*, vol. 17, no. 1, pp. 145–175, Jan. 2005.

[39] W. Chu and Z. Ghahramani, "Gaussian processes for ordinal regression," *J. Mach. Learn. Res.*, vol. 6, pp. 1019–1041, Dec. 2005.

[40] W. Chu and S. S. Keerthi, "Support vector ordinal regression," *Neural Comput.*, vol. 19, no. 3, pp. 792–815, Mar. 2007.

[41] E. Frank and M. Hall, "A simple approach to ordinal classification," in *Proc. ECML*, 2001, pp. 145–156.

[42] L. Li and H.-T. Lin, "Ordinal regression by extended binary classification," in *Proc. NIPS*, 2007, pp. 865–872.

[43] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung, "Ordinal hyperplanes ranker with cost sensitivities for age estimation," in *Proc. CVPR*, Jun. 2011, pp. 585–592.

[44] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, "Ordinal regression with multiple output CNN for age estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4920–4928.

[45] D. Devlaminck, W. Waegeman, B. Bauwens, B. Wyns, P. Santens, and G. Otte, "From circular ordinal regression to multilabel classification," in *Proc. ECML Workshop*, 2010, p. 15.

[46] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4L: Self-supervised semi-supervised learning," in *Proc. ICCV*, 2019, pp. 1476–1485.

[47] S. Wu, J. Li, C. Liu, Z. Yu, and H.-S. Wong, "Mutual learning of complementary networks via residual correction for improving semi-supervised classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6500–6509.

[48] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5070–5079.

[49] Q. Wang, W. Li, and L. Van Gool, "Semi-supervised learning by augmented distribution alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1466–1475.

[50] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3D human pose estimation in video with temporal convolutions and semi-supervised training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7753–7762.

[51] Y. Chen, Z. Tu, L. Ge, D. Zhang, R. Chen, and J. Yuan, "SO-HandNet: Self-organizing network for 3D hand pose estimation with semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6961–6970.

[52] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3877–3886.

[53] M. Qi, Y. Wang, J. Qin, and A. Li, "KE-GAN: Knowledge embedded generative adversarial networks for semi-supervised scene parsing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5237–5246.

[54] Y. Yao, Y. Jafarian, and H. S. Park, "MONET: Multiview semi-supervised keypoint detection via epipolar divergence," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 753–762.

[55] J. Gao, J. Wang, S. Dai, L.-J. Li, and R. Nevatia, "NOTE-RCNN: NOise tolerant ensemble RCNN for semi-supervised object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9508–9517.

[56] Y. He, J. Shi, C. Wang, H. Huang, J. Liu, G. Li, R. Liu, and J. Wang, "Semi-supervised skin detection by network with mutual guidance," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2111–2120.

[57] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. NIPS*, 2006, pp. 137–144.

[58] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, nos. 1–2, pp. 151–175, May 2010.

[59] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015, pp. 1–11.

[60] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. ICML*, 2015, pp. 1–9.

[61] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, May 2015.

[62] R. Shu, H. H. Bui, H. Narui, and S. Ermon, "A DIRT-T approach to unsupervised domain adaptation," in *Proc. ICLR*, 2018, pp. 1–19.

[63] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 668–675.

[64] T. Yao, Y. Pan, C.-W. Ngo, H. Li, and T. Mei, "Semi-supervised domain adaptation with subspace learning for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2142–2150.

[65] S. Ao, X. Li, and C. X. Ling, "Fast generalized distillation for semi-supervised domain adaptation," in *Proc. AAAI*, 2017, pp. 1–7.

[66] L. Cheng and S. J. Pan, "Semi-supervised domain adaptation on manifolds," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2240–2249, Dec. 2014.

[67] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8050–8058.

[68] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[69] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[70] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2001.

[71] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ''ImageNet: A large-scale hierarchical image database,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[72] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, ''A kernel two-sample test,'' *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012.

[73] N. Dalal and B. Triggs, ''Histograms of oriented gradients for human detection,'' in *Proc. CVPR*, Jun. 2005, pp. 886–893.

[74] P. Dollár, R. Appel, S. Belongie, and P. Perona, ''Fast feature pyramids for object detection,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[75] S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: Towards real-time object detection with region proposal networks,'' in *Proc. NIPS*, 2015, pp. 91–99.

[76] M. Kristan *et al.*, ''The visual object tracking VOT2014 challenge results,'' in *Proc. ECCVW*, 2014, pp. 191–217.

[77] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, R. Pflugfelder, A. Gupta, A. Bibi, A. Lukezic, A. Garcia-Martin, A. Saffari, A. Petrosino, and A. S. Montero, ''The visual object tracking VOT2015 challenge results,'' in *Proc. ICCVW*, Dec. 2015, pp. 1–23.

[78] Y. Wu, J. Lim, and M. H. Yang, ''Object tracking benchmark,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[79] P. Liang, E. Blasch, and H. Ling, ''Encoding color information for visual tracking: Algorithms and benchmark,'' *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.

[80] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, ''MOT16: A benchmark for multi-object tracking,'' 2016, *arXiv:1603.00831*. [Online]. Available: http://arxiv.org/abs/1603.00831

[81] S. Leutenegger, M. Chli, and R. Y. Siegwart, ''BRISK: Binary robust invariant scalable keypoints,'' in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2548–2555.

[82] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA, USA: Kluwer, 1991.

KYUNG-RAE KIM received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2014, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include computer vision and machine learning, especially in the problems of stereo matching and motion estimation.

YEONG JUN KOH (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2011 and 2018, respectively. As an Assistant Professor in March 2019, he joined the Department of Computer Science and Engineering, Chungnam National University. His research interests include computer vision and machine learning, especially in the problems of video object discovery and segmentation.

CHANG-SU KIM (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Seoul National University (SNU). From 2000 to 2001, he was a Visiting Scholar with the Signal and Image Processing Institute, University of Southern California, Los Angeles. From 2001 to 2003, he coordinated the 3D Data Compression Group, National Research Laboratory for 3D Visual Information Processing, SNU. From 2003 and 2005, he was an Assistant Professor with the Department of Information Engineering, Chinese University of Hong Kong. In September 2005, he joined the School of Electrical Engineering, Korea University, where he is a Professor. He has published more than 280 technical papers in international journals and conferences. His research interests include image processing, computer vision, and machine learning. He is a member of the Multimedia Systems and Application Technical Committee (MSATC) of the IEEE Circuits and Systems Society. In 2009, he received the IEEK/IEEE Joint Award for Young IT Engineer of the Year. In 2014, he received the Best Paper Award for the *Journal of Visual Communication and Image Representation* (JVCI). He received the Distinguished Dissertation Award in 2000 for his Ph.D. degree. Also, he is an APSIPA Distinguished Lecturer from 2017 to 2018. He served as an Editorial Board Member of JVCI and an Associate Editor of the IEEE Transactions on Image Processing. He is a Senior Area Editor of JVCI and an Associate Editor of the IEEE Transactions on Multimedia.