Neural Computing and Applications
https://doi.org/10.1007/s00521-019-04200-1

IAPR-MEDPRAI

Granulated deep learning and Z-numbers in motion detection and object recognition

Sankar K. Pal[1] · Debasmita Bhoumik[1] · Debarati Bhunia Chakraborty[1]

Received: 20 July 2018 / Accepted: 11 April 2019

Abstract
The article deals with the problems of motion detection, object recognition, and scene description using deep learning in the framework of granular computing and Z-numbers. Since deep learning is computationally intensive, whereas granular computing, on the other hand, leads to computation gain, a judicious integration of their merits is made so as to make the learning mechanism computationally efficient. Further, it is shown how the concept of z-numbers can be used to quantify the abstraction of semantic information in interpreting a scene, where subjectivity is of major concern, through recognition of its constituting objects. The system, thus developed, involves recognition of both static objects in the background and moving objects in foreground separately. Rough set theoretic granular computing is adopted where rough lower and upper approximations are used in defining object and background models. During deep learning, instead of scanning the entire image pixel by pixel in the convolution layer, we scan only the representative pixel of each granule. This results in a significant gain in computation time. Arbitrary-shaped and sized granules, as expected, perform better than regular-shaped rectangular granules or fixed-sized granules. The method of tracking is able to deal efficiently with various challenging cases, e.g., tracking partially overlapped objects and suddenly appeared objects. Overall, the granulated system shows a balanced trade-off between speed and accuracy as compared to pixel level learning in tracking and recognition. The concept of using Z-numbers, in providing a granulated linguistic description of a scene, is unique. This gives a more natural interpretation of object recognition in terms of certainty toward scene understanding.

Keywords Deep learning · Granular computing · Rough sets · Video tracking · Object recognition · Z-numbers

1 Introduction

Moving object detection, *recognition* and *tracking* find application in several fields of computer vision such as surveillance, security, gesture recognition and intrusion detection. Video tracking is a tedious process due to the bulk of data involved with the video. In tracking, the target objects are associated with consecutive video frames. Detection becomes challenging when the frame rate is high. Moreover, the objects are likely to change their orientation with time, which adds to the complexity of tracking. Furthermore, only tracking the moving objects is not sufficient. Determining the characteristics of the objects is also necessary which leads to object recognition. Various uncertainties and ambiguities make this task of video tracking challenging, and thus the issues are being studied over the years [1]. Video tracking can be supervised or unsupervised. In the supervised approaches, the initial object(s) to be tracked are labeled manually, whereas in unsupervised approach no labeling is needed. The method, we have explained here for detecting and recognizing continuously moving multiple objects in static background, is supervised. Here we have used image processing and machine learning techniques side by side.

Granulation [2] is a basic step of human cognition system. It is a process like self-organization, self-production, morphogenesis, Darwinian evolution that are extracted from natural phenomena. It may be viewed as a process of natural clustering, i.e., replacing a fine-grained universe by a coarse-grained one, more in line with human perception. Clusters or segments so formed by granulation (natural clustering) are

✉ Sankar K. Pal
sankar@isical.ac.in

[1] Center for Soft Computing Research, Indian Statistical Institute, Kolkata 700 108, India

Published online: 02 May 2019

called granules. In other words, granulation is a process of formation and representation of granules, evolved through information abstraction and derivation of knowledge from data. A *granule* is defined as a clump (cluster) of indiscernible objects drawn together, for example, by likelihood, similarity, nearness, or functionality [3].

Granulation leads to information compression, and processing based on the compressed information, rather than the individual data points, may lead to gain in computation time. Depending on the application, granules could be of different types like crisp, fuzzy, and rough-fuzzy. Rough set [4] theory can effectively handle the uncertainties or incompleteness of knowledge arising from the limited distinguishability of objects in the domain of discourse. Object distinguishability and set approximation are the key concepts behind this. Rough set-based granular computing has been applied to image processing [3] and video tracking [5–7] successfully. Here we have used rough set theoretic granular computing in deep learning framework for speedy motion detection and moving object recognition.

Machine learning (ML), a branch of artificial intelligence (AI), basically means learning patterns from examples or sample data. Here the machine is given access to the data and has the ability to learn from it. The data (or examples) could be labeled, unlabeled, or their combination. Accordingly, the learning could be supervised, unsupervised or semi-supervised. Artificial neural networks (ANNs) that have the ability to learn the relation between input and output from examples are good candidates for ML. ANNs enjoy the characteristics like adaptivity, speed, robustness/ ruggedness, and optimality. In the early 2000s, certain breakthroughs in multilayered neural networks (MLP) facilitated the advent of deep learning. Deep learning (DL) means learning in depth in different stages [8]. DL is thus a specialized form of ML which takes ML to the next level in an advanced form. This is characterized by learning the data representations, in contrary to task-specific algorithms.

Deep Learning algorithms/ networks are inspired by the structure and function of the human nervous system, where a complex network of interconnected computation units (nodes) works in a coordinated fashion to process complex information. In order to extract the complex representation from rich sensory inputs, human information processing mechanisms suggest the need of deep (learning) architectures [9]. Such an architecture usually involves a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input. In deep learning architecture, the problem of object recognition can be regarded as a task of labeling different objects in an image with the correct class as well as predicting the bounding boxes with a high probability. Different structures of deep neural networks on this problem have been proposed [10–13].

Convolutional neural network (CNN, or ConvNet) [14] represents one such deep architecture which is most popular for learning with images and video. Like other neural networks, a CNN is composed of an input layer, an output layer, and several hidden layers in between. These layers perform one of the three types, e.g., convolution, pooling, or rectified linear unit (ReLU), of operations on the data. Convolution puts the input images through a set of convolutional filters [15], each of which activates certain features from the images. Pooling [14] simplifies the output by performing nonlinear downsampling, reducing the number of parameters that the network need to learn about. Rectified linear unit (ReLU) [9] allows for faster and more effective training by mapping negative values to zero and maintaining positive values. These three operations are repeated over tens or hundreds of layers, with each layer learning to detect different features. CNNs have been used for motion detection and object recognition [13, 16–23].

Deep learning (DL) has dramatically improved the state of the art in object recognition [9], among other applications. However, since DL relies on sample data (or previous experience), the learning performance depends on the number of such samples. Larger the number is, more accuracy is the performance. Today, we have abundant data; so DL has become a meaningful choice. DL often requires hundreds or thousands of images for the best results unlike the conventional (Shallow) learning. Therefore, DL is computationally intensive and difficult to engineer. It requires a high-performance GPU (Graphical Processing Unit). For example, deep learning networks like single shot detector [13], Faster-R-CNN [11], CFCF [16] provide very fast motion detection and object recognition using GPU.

While deep learning is a computationally intensive process and the aforesaid granular computing paradigm, on the other hand, leads to gain in computation time, it may be appropriate and logical to make their judiciously integrate them so as to make the deep learning framework efficient in terms of computation time requiring only CPU. The proposed study embodies such an attempt, where rough set theoretic granular computing is used in CNN for speedy motion detection and moving object recognition. No attempts have yet been made, to our knowledge, that incorporate the merits of granular computing in deep learning framework.

Further, quantification of performance in image/video processing, and of interpretation and understanding of scenes, where subjectivity is of major concern, has always been a challenging issue. Z-numbers, proposed recently by Zadeh in 2011 [24], provide a framework to quantify the abstraction of semantic information from natural language statements where subjectivity plays an important role in understanding. In other part of the investigation, we have demonstrated how the abstract concept of z-numbers can be used in interpreting a scene with certainty in terms of

recognition of its constituting objects, in natural language. An information measure of a scene is accordingly defined.

The basic block diagram of the proposed granulated deep learning system is shown in Fig. 1 for motion detection and object recognition with linguistic description. Here the method involves recognition of both static objects in the background and moving objects in the foreground separately for scene analysis, using frame difference technique (for detail description, see Sect. 3.3). At first, granulation is performed on the input image frame f_t. Then the object (O_b) and background models (B_g) are computed on the granulated f_t. The O_b and B_g are then fed into a deep learning network (DNN) for recognition of static and moving objects. The output provides a linguistic description of the scene consisting of these objects. Note that, here the input to DNN is a granulated frame and scanning in the convolution layer is done only over the representative pixel of each granule in O_b and B_g. This is unlike the CNN, where the entire image frame is scanned pixel-by-pixel in the convolution layer. This lowers the computation time drastically, compromising little with accuracy.

The novelty of the present investigation mainly lies with:

1. proposing a methodology for granulated deep learning in motion detection and object recognition in video for speedy computation,
2. providing a new linguistic description of object classification results based on Z-numbers that can interpret a scene, in a more natural way with certainty, for its understanding.

This rest of the article is organized in the following way. Section 2 describes a brief introduction to rough sets, image definition, and formation of various granules. In Sect. 3, after explaining the characteristics of conventional convolutional neural network (CNN), we explain the characteristics of proposed granulated deep learning (GDL). This includes the concept, characteristics and relevance of the mechanism of granulated deep learning, and the methodology and algorithms for object tracking and recognition. Section 4 describes the Z-number-based measures for object recognition and scene understanding. Results of object tracking and recognition along with comparisons with some state-of-the-art algorithms are described in Sect. 5. Section 6 provides the conclusions.

2 Granulation techniques

As described before, a granule is a clump (cluster) of objects drawn together, for example, by likelihood, similarity, proximity, or functionality [3]. Using the process of granulation over a data set, granules are extracted where data points having similar characteristics are collected within each granule. In this section, we describe the formation of granules of different kinds, in terms of their size and shape, in images, where gray level similarity, color similarity, spatial similarity and spatio-color similarity among pixels are used in extracting granules (image segments). These are followed by the method of determination of optimal threshold for computing lower-upper approximations of object and background. Before that, we provide the basic concept of rough sets and image definition in this framework.

2.1 Concepts of rough set and image definition

Rough set, as introduced by Pawlak [4], is a formal approximation of a crisp set. Let $A = <U, A>$ be an information system, where U is the universe and A is the set of attributes. Let $B \subseteq A$ and $X \subseteq U$. If $X \subseteq U$, the set $\{x \in U : [x]_B \subseteq X\}$ is known as B-lower approximation of X ($\underline{B}X$), i.e., this set will always be a subset of X. Similarly, the set $\{x \in U : [x]_B \cap X \neq \Phi\}$ represents the B-upper approximations of X in $U(\overline{B}X)$ which will always have a nonzero intersection with X. The roughness of a set X with respect to B is characterized numerically [4] as

$$R_I = 1 - \left(\frac{|\underline{B}X|}{|\overline{B}X|}\right) \tag{1}$$

Let the universe U be an image (I of size $M \times N$) consisting of a collection of pixels. Then if we partition U into a collection of non-overlapping windows of unequal size (of size $m_i \times n_i$, where i reflects the ith granule, say), each window can be considered as a granule G_i. Let T be the threshold for object and background classification. Then the object lower ($\underline{O}_T$) and upper ($\overline{O}_T$) approximations are constructed over all the granules as [3]:

$\underline{O}_T$: The set of the granules with all the pixel values greater than T

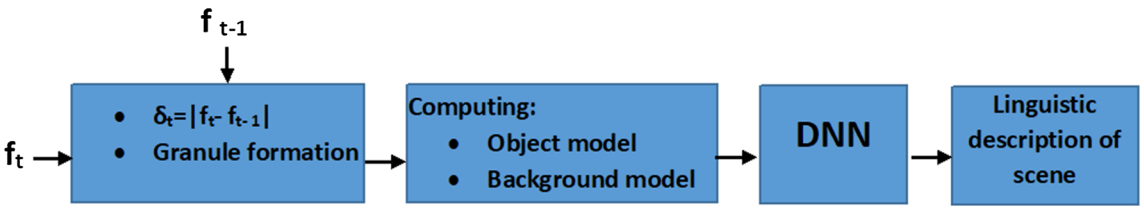

Fig. 1 Overall block diagram of the recognition system

$\overline{O}_T$: The set of the granules with at least one pixel value greater than T

$\underline{B}_T$: The set of the granules with all the pixel values less than T

$\overline{B}_T$: The set of the granules with at least one pixel value less than T

The roughness of object O_T and background B_T is

$$R_{O_T} = 1 - \frac{|\underline{O}_T|}{|\overline{O}_T|} \tag{2}$$

$$R_{B_T} = 1 - \frac{|\underline{B}_T|}{|\overline{B}_T|} \tag{3}$$

2.2 Formation of granules and rough upper-lower approximations

Granulation could be of different types producing granules of equal or unequal size, although unequal granules are more natural for real-life problems. Further, granules may be of various kinds, e.g., crisp granules, fuzzy granules, rough-fuzzy granules, and neighborhood granules. Verification of the proposed algorithm with every type of granule is not feasible. Therefore we have considered three categories of granules in images that broadly covers all possibilities, ranging from equal-sized regular shaped, unequal-sized regular shaped to arbitrary sized and shaped. Granules with arbitrary sized and shaped are more natural and expected to produce better performance, but are computationally intensive. In our study, the granules used are: (i) uniform-sized rectangular granules with spatial similarity, (ii) un-equal-sized rectangular granules with gray level and spatial similarities, and (iii) natural arbitrary-sized/shaped (neighborhood) granules with spatio-color similarity. The granules of first two categories are crisp, whereas the third one is overlapping (fuzzy) in nature. These are described here. Then we explain, in brief, how upper and lower approximations of object and background are computed from the granulated image.

2.2.1 Equal-sized and shaped granules

Let I be an image of size M × N. Let the image be partitioned into a collection of non-overlapping windows of equal size (m × n). Then each window can be considered as a granule.

2.2.2 Unequal-sized and regular-shaped granules

Here, the granulation is based on quad-tree decomposition of the images. A quadrant is further divided into four quadrants if the difference between its maximum and minimum intensity levels exceed a threshold T, where T is the average value of the first and third quartile of the image gray level distribution [5]. The output granules, thus produced, are unequal in size which is more practical for dealing with real-life problems.

2.2.3 Arbitrary-shaped granules

These are formed by region growing technique based on color similarity among the 4-neighbor of a candidate pixel. A granule $N(x)$ centered at a point x_i is formed as [6]:

$$N_{sp-clr}(x_i) = Ux_j \in U :$$
$$x_i \text{ and } x_j \text{ binary connected over}$$
$$|color(x_j) - color(x_i)| < Thr.$$

Thr is the color nearness threshold. One may note that the granules thus produced are of arbitrary size and shape. Moreover, they can be overlapping too. This technique of granule formation is applied on both current frame (f_t) and the difference frame $\delta_t (= |f_t - f_{t-1}|)$. As the granules here may be overlapping in nature, we form a rule base for object-background classification in Table 1. In the rule base, Tempval and RGBval correspond to granular values obtained from δ_t and f_t planes, respectively.

Since the granules may be overlapping in nature, their similarity with the conditional attributes may not be always crisp. To reflect this, the notion of complete belonging (Be), partial belonging (PB) and not belonging (NB) is used in Table 1. It may be noted that rules 5 and 6 are inconsistent. Incorporation of arbitrary-shaped and overlapping granules makes the rule base more natural and robust. Details of the granulation process are mentioned in [6] for video tracking.

Table 1 Rule generation for object-background separation using arbitrary-shaped granules

U	*Tempval*	*RGBval*	Decision
N1	NB	NB	B
N2	NB	Be	B
N3	PB	Be	O
N4	Be	Be	O
N5	Be	NB	B
N6	Be	NB	O
N7	PB	PB	O
N8	NB	PB	B
N9	PB	NB	B
N10	Be	PB	O

2.2.4 Optimum threshold (T_O) for computing lower-upper approximations

After the granulated image plane (I_G) is obtained, we consider different thresholds (T) for object-background separation and determine the one which results in minimum roughness of I_G. That means, given an arbitrary threshold T, we compute the R_{O_T} and R_{B_T} of I_G based on $\underline{O}_T$, $\overline{O}_T$, $\underline{B}_T$, and $\overline{B}_T$, as described in Sect. 2.1. Then we vary the T, and determine the one (T_O) for which the values of R_{O_T} and R_{B_T} of I_G are minimum. Accordingly, $\underline{O}_{T_O}$, $\overline{O}_{T_O}$, $\underline{B}_{T_O}$, and $\overline{B}_{T_O}$ denote the optimal versions of object lower, object upper, background lower and background upper approximations, respectively.

3 Granulated deep learning

Human intelligence and discriminating power is mainly attributed to the massively connected network of biological neurons in the human brain. An artificial neural network (ANN) is a system composed of several simple processing elements (nodes) operating in parallel whose function is determined by network structure, weight, and processing performed at nodes. ANNs are designed in an attempt to mimic the functionality of human brain in order to emulate human performance and thereby function intelligently. ANNs enjoy the characteristics like adaptivity, speed, robustness, ruggedness and optimality. Multi-layer perceptron (MLP) using back propagation of error is a popular neural network model that learn adaptively, updating their connection weights during training. This network can be trained by examples as is often required in real life and sometimes generalized well for some unknown test cases. It consists of multiple layers of simple, two-state, sigmoid processing elements (nodes/neurons) that interact using weighted connection [25]. Since it has the ability to learn the relation between input and output from examples or sample data, MLPs are good candidate for machine learning (ML). As stated before, in the early 2000s, certain breakthroughs in multi-layered neural networks (MLP) facilitated the advent of deep learning (learning in depth in different stages) [8]. DL is a specialized form of ML. Convolutional neural network (CNN) is a popular network for DL with images and video.

It is well established that the learning with deep neural networks is a slow process. Here we explain how the concept and merits of granular computing can be integrated with CNN to speed up its learning mechanism. The effectiveness of the resulting system is demonstrated for motion tracking and scene analysis from videos. Before we describe the proposed granulated learning network, we explain the conventional CNN, in brief.

3.1 Conventional convolution neural network

As explained in Sect. 1, convolutional neural network (CNN) is composed of an input layer, an output layer, and several hidden layers in between. These layers perform one of the three types, e.g., convolution, pooling, or rectified linear unit (ReLU), of operations on the data. These three operations are repeated over tens or hundreds of layers, with each layer learning to detect different features.

The convolution layer of deep learning network basically involves shifting of a sliding window all over the image or frame. Let the input to the network be a $32 \times 32 \times 3$ array of RGB (red–green–blue) pixel values. To explain the functionality of a convolution layer let us imagine a flashlight that is shining over the top left of the image. Let us assume that this flashlight covers a 5×5 pixel area. Let this flashlight slide across all the areas of the input image. The flash light that covers a 5×5 window in the image may be viewed as a filter. The region of the image that is being shined over is called the receptive field. For mathematical matching, the depth of this filter has to be the same as the depth of the input image, where the intensity of the said flash light (filter) is representing a 5×5 array of numbers, called weights; so the dimensions of this filter is $5 \times 5 \times 3$ for RGB components. As the filter is sliding (or convolving), around the input image, it is multiplying the values in the filter with the original pixel values of the image. These multiplications are all summed up and we get a single number. We repeat this process for every location on the input image. Next step would be moving the filter to the right by S unit, then right again by S, and so on, where S is called the Stride. If $S = 1$, after sliding the filter over all the locations, we will find out that what we are left with is a $28 \times 28 \times 1$ array of numbers, which is called an activation map.

After computing the activation map, the next layer is pooling which simplifies the output by downsampling, thereby reducing the number of parameters in the network. Rectified linear unit (ReLU) allows fast training by mapping the negative values to zero and retaining the others as they are. These three operations are repeated several times to obtain a desired (converged) output. A block diagram, showing how the layers are organized (i.e., from input image to output classification), is shown in Fig. 3.

Training a deep neural network from scratch requires enormous labeled (training) data and hence computing power (hundreds of GPU-hours or more). To avoid this, it may be desirable to recollect the training data. In such cases, knowledge transfer or transfer learning between task domains would be helpful [26]. Transfer learning [14] is a technique that shortcuts much of this by taking a piece of a model that has already been trained on a related task and reusing it in a new model. One needs to be careful while

choosing what pre-trained model is to be used. If the problem statement is entirely different from the one on which the pre-trained model was trained, then the prediction would be wrong. For example, a model previously trained for speech recognition would not work if we try to use it to identify objects in images. In our investigation, we have used transfer learning by considering weights from the already trained models of convolutional neural network (CNN) [14]. The trained CNN that we have used is the single-shot detector (SSD) [13] with COCO (common objects in context) data set [27] (described in Sect. 5).

3.2 Granulated deep learning: concepts, characteristics and relevance

The convolution layer of the deep learning network, as described before, basically involves shifting of a sliding window all over the image or frame for convolving (e.g., multiplication and summation). Here, in the proposed model, instead of using a raw pixel-based frame as input to this convolution layer, we make that frame granulated using different approaches, as discussed in Sect. 2.2, considered as input.

If the original image size is 32×32, then after granulation, the image would consist of granules instead of 32×32 pixels. The number of such granules is obviously much less than 32×32. Let us assume that there are n number of granules, named, $g_1, g_2, \ldots g_n$ in a frame. This granulated frame is used as the input of the first convolution layer. The filter region chooses the top left corner of the image, which belongs to granule g_1, as the receptive field. After the first cell value of the activation map (Fig. 2) is computed, we put the same value in the activation map corresponding to the other pixels in g_1. That means, we skip all the remaining pixels which belong to the same granule g_1, and need not compute cell values of the activation map more than once for a particular granule.

Unlike the pixel-based convolution method, where the stride is fixed apriori, here the stride automatically selects only the top left corner of each granule as the receptive field; thereby skipping the remaining pixels. By doing so, the selection of appropriate stride (which is crucial in a conventional CNN) does not arise. Note further that, in the pixel-based method, sliding the filter over all the 32×32 pixels results in a 28×28 array of numbers. But in our method one needs to do the filtering only n times where $n << 32^2$. Thus the computation time is greatly reduced, although some accuracy may be compromised. Figure 3 shows the high level blocks of the Granulated deep learning network. The model is roughly based on the Single Shot Detector (SSD) (the same CNN that we have used for transfer learning during training with SSD in Sect. 3.1). The detailed structure of SSD is available in Ref. [13]. The major difference with that in [13] is that, we implemented it in granulated fashion, where the first convolution layer is granulated (as described before). The input image (granulated) is fed into the granulated convolution layer, followed by max pooling and rectified linear unit (ReLU). Then finally a fully connected layer of the neural network produces the output classification. Use of this network for object recognition and tracking is described in Sect. 3.3 where the granulated convolution layer takes the object model and background model as input.

3.3 Granulated deep learning: object tracking and recognition

Given a scene containing static (background) and moving objects, the recognition algorithm has two parts. First, it recognizes the classes to which these objects belong. Secondly, it tracks the objects in motion. In doing so, the method involves classification of static object and moving objects separately by the granulated deep learning model explained in Sect. 3.2 (Fig. 3). The entire method of

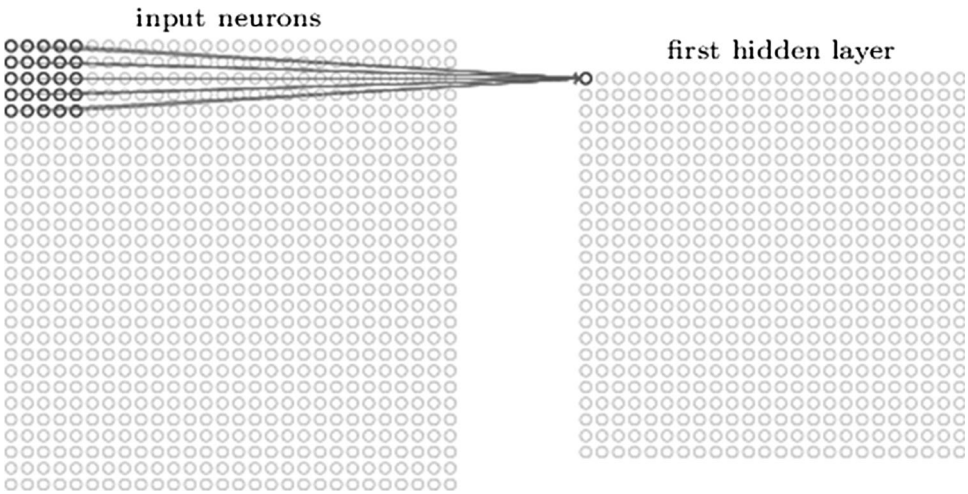

Fig. 2 Activation map

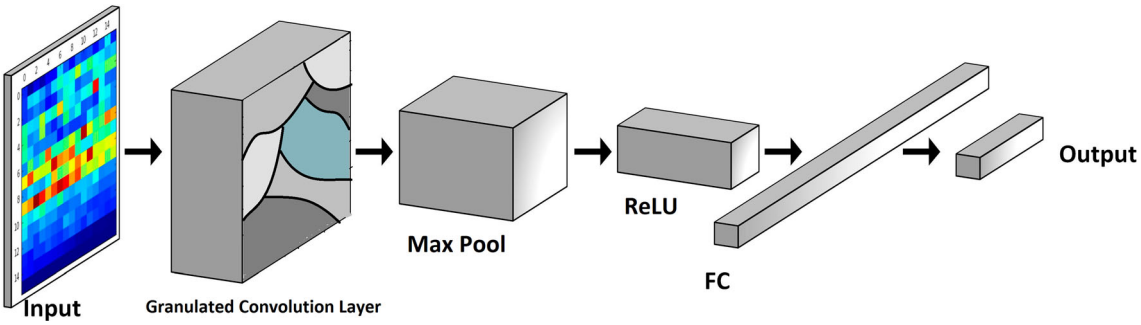

Fig. 3 Layers of granulated deep learning model

tracking and recognition is shown in brief by a block diagram in Fig. 4, which is explained step by step below.

Let f_t and f_{t-1} be the current frame and its immediate previous frame of a video sequence, respectively. Let the frame difference be computed as,

$$\delta = |f_t - f_{t-1}|. \tag{4}$$

δ characterizes the change between two consecutive frames, which represents only the moving portion (pixels) in the frames.

We form granulation on f_t by the methods described in Sect. 2.2. After granulation, the system computes the upper approximation of the background $\overline{B}_T(= U_B)$ over the granulated image, corresponding to the optimum threshold (T_O) as described in Sect. 2.2.4. That means, U_B is the set of granules where at least one pixel value is less than T_O.

The intersection of upper approximated background (U_B) and frame difference (δ) denotes the portion which erroneously may belong to the background despite being a part of the moving object. Therefore, to ensure that the approximated background (U_B) has no part of moving objects, we subtract the intersection of U_B and δ from the U_B, and denote it as the granulated background model (B_G). That is,

$$B_G = U_B - (U_B \cap \delta). \tag{5}$$

The granulated object model (O_B) is obtained as,

$$O_B = f_t - B_G. \tag{6}$$

Let us now explain the task of recognition and tracking. Here recognition concerns with both static (background) and moving objects, and the tracking is done on moving objects. Recognizing static objects (background) is performed only once at the beginning, using the proposed granulated deep learning framework (shown by dashed lines and blocks) with B_G as input. Static objects remain the same throughout the videos. For recognizing moving objects from the input frames $f_i (i = 2, 3, \ldots N$, where N is the number of frames), the granulated object model (O_B) of each frame is taken as input to the said granulated deep learning network. Its output denotes the categories of the moving objects, which are then tracked.

The aforesaid steps are explained in Algorithm 1.

Algorithm 1 : Object recognition and motion tracking using granulated deep learning

Input: Video frames $f_i, \forall i = 1, 2, ..., n'$ where n' is the number of frames in the input video
Output: Frames with recognized objects and bounding box around moving objects

1: Take the current frame f_t and its immediate previous frame f_{t-1}.
2: Calculate the temporal information $\delta = |f_t - f_{t-1}|$.
3: Perform granulation on the frame f_t.
4: Compute the upper background approximation U_B after the quad tree decomposition $U_B = \{\bigcup_i G_i, \forall j = 1, 2, ..., mn\}$ such that $P_j < T$ where $P_j \in G_i$ and T is the threshold for distinguishing object and background.
5: Compute the granulated background model $B_G = U_B - (U_B \cap \delta)$.
6: Feed B_G into the granulated deep-learning framework, proposed in Sec. 4.7, for static (background) object recognition [Step 6 is a one-time process as the background is fixed for all frames].
7: Compute the granulated object model $O_B = f_t - B_G$.
8: Feed O_B into the granulated deep learning framework, described in Sec. 3.2, for moving object(s) recognition with probability of occurrence (P_O).
9: Track the moving object(s) by placing bounding boxes around them.
10: Repeat the process for the next frame f_{t+1} onward.

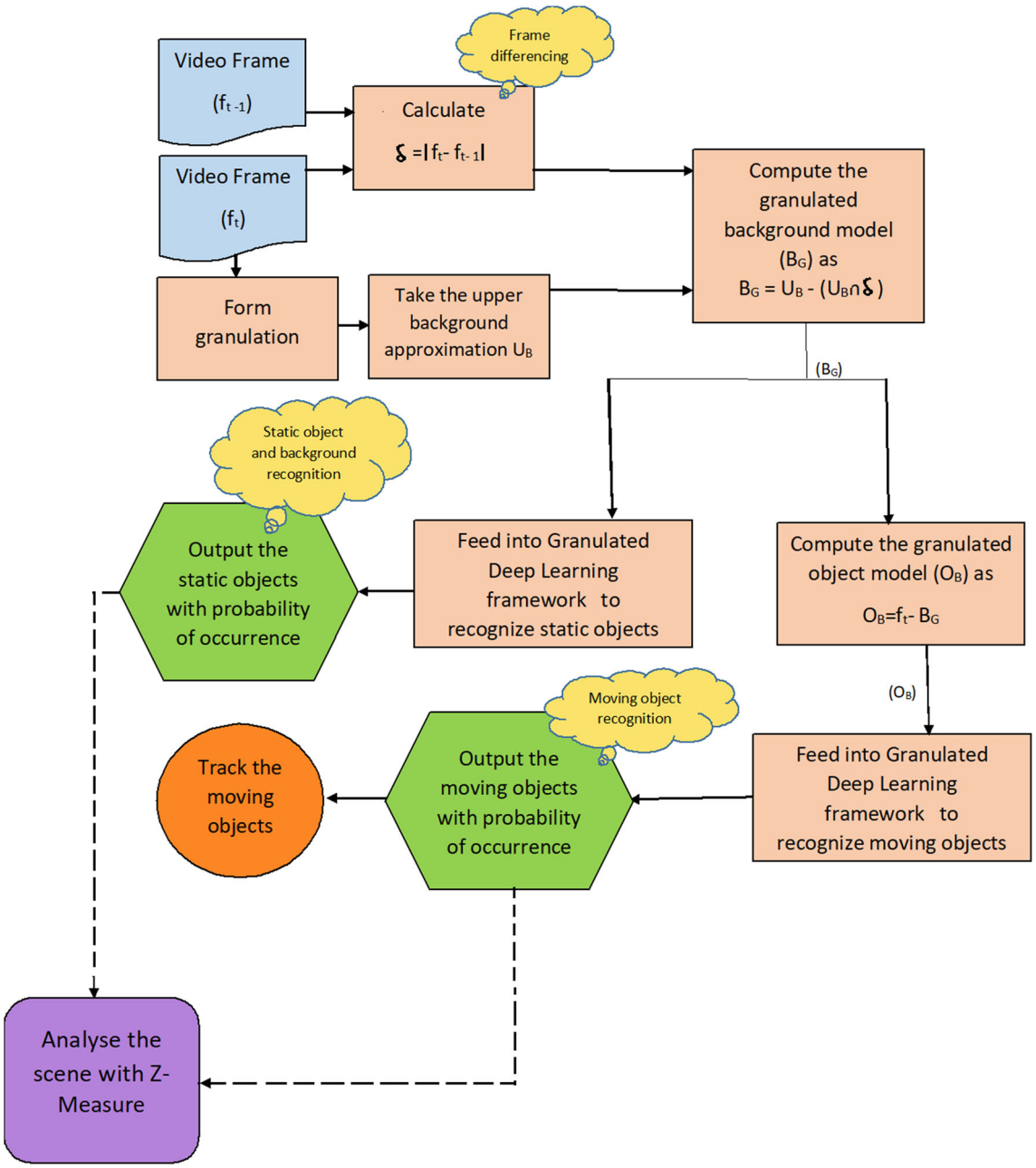

Fig. 4 Block diagram of the proposed granulated deep learning model for object recognition and tracking

The results of recognition of static and moving objects, as obtained in Fig. 4, are used for analysis of the input scene. This is done in a new way in terms of linguistic description using Z-numbers that can interpret a scene, in a more natural way with certainty, for its understanding. This method is described in Sect. 5 along with the performance of tracking and recognition. Before that we define Z-numbers with an example in Sect. 4.

4 Z-number-based description

The Z-numbers [24] provide a new fuzzy-set-theoretic approach to Computing With Words (CWW) [28]. In CWW paradigm, the perceptions are encoded in the words and phrases used to describe events. This is inspired from the remarkable perception-based decision-making ability of the human brain. The concept of Z-number correlates to the issue of certainty of information. A Z-number has two tuples, $Z = (A, B)$. The first tuple is A, which is a constraint, allowed to take on the values of X (a real-valued

uncertain variable, interpreted as the subject of Y). The second tuple, B, is a measure of reliability of the first component. Normally, A and B are described in a natural language, as words or clauses, and are both fuzzy numbers [29].

Example of Z-numbers: Let us consider a statement Y := It takes Jack about 10 min to reach school from his house. Then, X := Distance from Jack's house to school, and $Z =$ <about 10 min, usually >. Here, A is context-dependent while B summarizes the conclusiveness in the relevance of A given X within the context of Y.

In our study, we have used Z-numbers to define measures for object recognition and scene understanding. The granulated neural network predicts the class in which the objects of an unknown scene belongs with certain probability associated with it. Based on this prediction, we have defined some rule bases that would provide the information on A and B of Z-numbers. Details of the methodology and results on scene interpretation are explained in Sect. 5.3.

5 Result on object tracking and recognition

Experiments along with comparisons were conducted to evaluate the effectiveness of the proposed algorithm in (a) motion tracking, and (b) object recognition. For this purpose we have incorporated three types of granulation techniques, (i) equal-sized and shaped granule, (ii) unequal rectangular-shaped granule and (iii) arbitrary-shaped granule, into our granulated deep learning framework. We have used transfer learning by considering weights from the already trained models of convolutional neural network (CNN) [14], as described in Sect. 3.1. The trained CNN that we have used is the single-shot detector (SSD) [13] with COCO (Common objects in context) data set [27], containing 20 classes of objects (+1 for the background). These objects are airplanes, bicycles, birds, boats, bottles, buses, cars, cats, chairs, cows, dining tables, dogs, horses, motorbikes, people, potted plants, sheep, sofas, trains, and tv monitors.

The test data set that we have used to perform our experiments consists of seven different types of video sequences: (1) Cam 131 Sequence (Changing appearance scenario) of ICG Lab [30], (2) Girl sequence of TB -50 (Visual Tracker Benchmark) [31], (3) PASCAL VOC 2007 [31], (4) VOT 2017 [32] (Bolt, Gymnastic1 and Godfather sequence), (5) Jurassic Park Intro [33], (6) PETS 2009 [34], and (7) Changing Size Car [35]. However, to limit the size of the paper, we have shown the results on some of the frames of each of these video sequences. Their comparison with some state-of-the-art algorithms is also provided depending on the data sets. Note that, we have considered these seven types of videos, as the performance results on these sequences by other methods are published and can be used for comparison with that of ours. These video sequences have both single-type objects and multiple-category objects. However, one can use any other videos.

We made the parameters as adaptable as possible. The parameter values are dependent on the nature of the input data. For example, the number of previous frames n is dependent on the speed of the video. It is normally chosen as 7 for the sequences with speed of 15 frames per sec. The object-background threshold value T is initially chosen to be 30 as it was found to be experimentally suitable for most of the data sets. If the set $O_T = \phi$, then reduce T by 5.

One may further note that the deep learning network (Sect. 3.2), where the proposed concept of granulation is used, was trained with 20 classes of objects. But the aforesaid seven types of video sequences that we used as test data do not have all those twenty classes in a particular sequence. Some of them contain objects only of a particular class, whereas some others have objects of multiple classes, but not of all categories. Moreover, some sequences deal only with tracking or recognition, while the others for both tracking and recognition.

5.1 Results of moving object detection and tracking

Let us consider the sequences CAM 131 Changing appearance (chap) scenario, PETS 2009, Jurassic Park and Changing Size Car. Both CAM 131 and PETS 2009 have only one type of class, namely person. Data sets Jurassic Park and Changing Size Car, on the other hand, have multiple types of classes, e.g., chair, dining table, person, bottle, airplane and car. Results of tracking and recognition on these sequences are shown in Figs. 5 and 6 corresponding to objects of single category and multiple categories. Here arbitrary-shaped granules (Sect. 2.2.3) were used in the deep learning network. It is shown that the bounding boxes are decently covering the moving objects in the video frames for both the data sets. Accuracy in recognizing the objects is very high, as depicted therein.

5.2 Comparative study

The comparison is done in terms of time and accuracy between our method of granulated deep learning (with 3×3, quad-tree decomposition, and arbitrary-shaped granules), deep learning without granulation, and other relevant deep learning algorithms such as CFCF [16], CFWCR [17], LOT [18], SCM [19] and SSD [13] as applicable to different data sets. Tables 2, 3, 4 and 5 demonstrate the comparative results. Tables 2 and 3 deal with Cam 131 sequence (Changing appearance scenario) and PASCAL VOT 2017 data, respectively, concerning

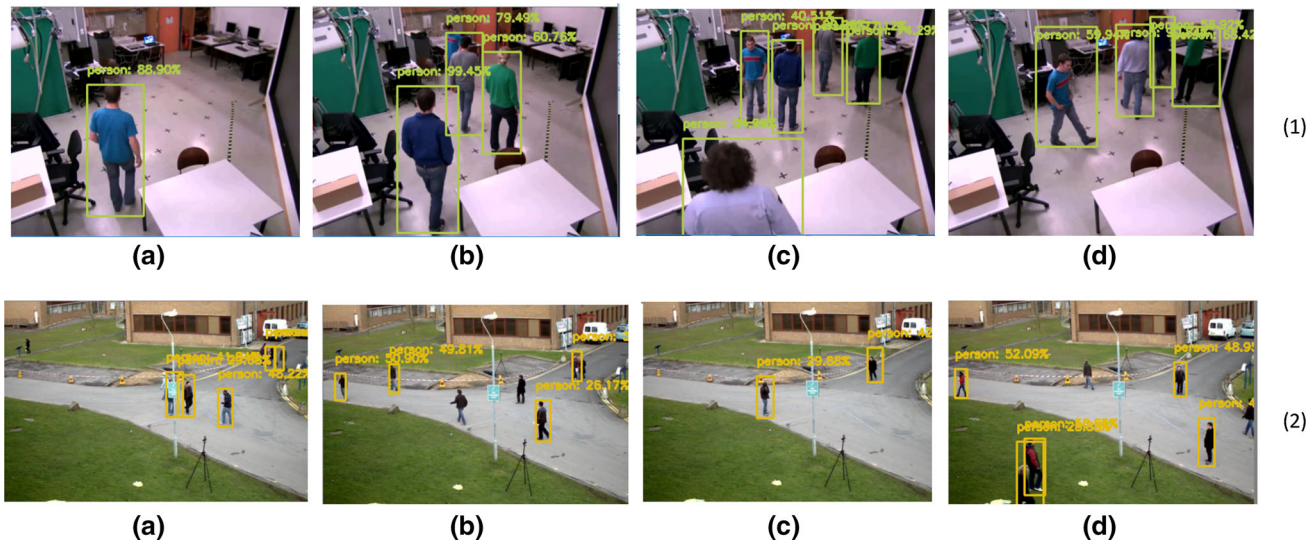

Fig. 5 Results of tracking using rectangular granulation by quad-tree-decomposition: frame nos. 25, 200, 321 and 381 of the Changing appearance (chap) scenario (1) and frame no 105, 194, 424, 530 of the PETS 2009 people tracking data set (2)

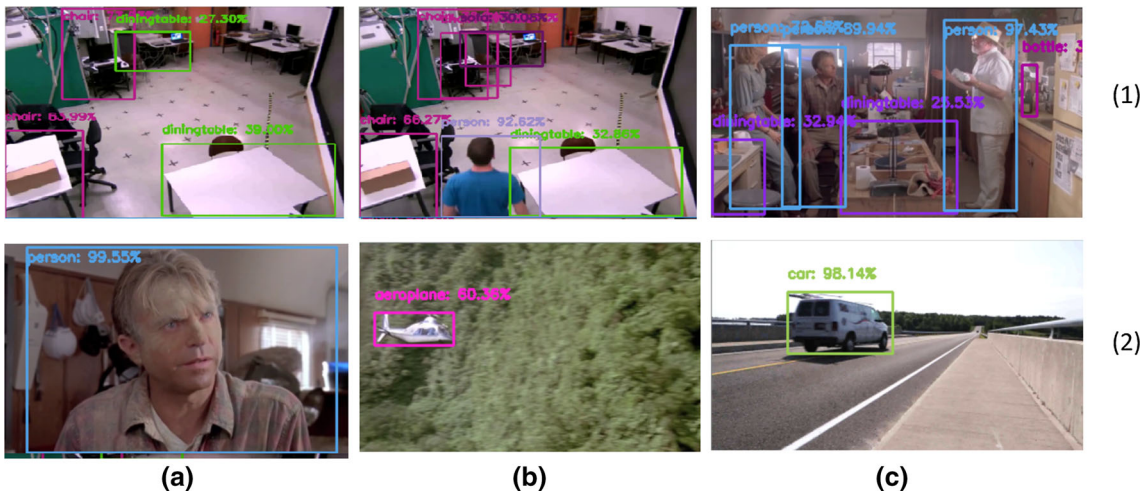

Fig. 6 Results of tracking and recognition for multiple categories of objects: 1(a) and 1(b)- frame nos. 5, 22 of the Changing appearance (chap) scenario, 1(c), 2(a) and 2(b)- frame nos. 12, 25 and 29 of the Jurassic Park Intro data set, and 2(c) frame no. 19 of the Changing Size Car data set (2.c)

Table 2 Time and accuracy comparison for recognition and tracking between our method (granulated deep learning using 3×3 granules, rectangular granules and using arbitrary-shaped granules) and deep learning without granulation

Method	Speed (fps)	Track	Accuracy of detection (%)	Processor (%)
Granulated deep learning using 3×3 granules	2.2	74.6	62.11	CPU
Granulated deep learning using rectangular granules	2	80.1	67.11	CPU
Granulated deep learning using arbitrary-shaped granules	1.89	81.67	68.56	CPU
Deep learning without granulation	1.6	82.25	70.2	CPU

with both tracking and recognition of only one type of objects (person). Table 4 depicts results for only tracking using Girl sequences of TB-50 data with one type of objects. On the other hand, Table 5 deals only with recognition, but it is for multiple classes of objects (viz, cat, dog, train, bird, cup, person, bottle, cow, cycle, dining

Table 3 Time and accuracy comparison for recognition and tracking between CFCF, CFWCR and our methods (granulated deep learning using 3×3 granules, rectangular granules and using arbitrary-shaped granules)

Method	Accuracy (%)	Speed (fps)	CPU/GPU
CFCF	50.9	1.7	CPU
CFWCR	48.4	1.4	CPU
Granulated deep learning using 3×3 granules	41.71	2.1	CPU
Granulated deep learning using rectangular granules	48.1	1.9	CPU
Granulated deep learning using arbitrary-shaped granules	48.59	1.5	CPU

Table 4 Time and accuracy comparison for Tracking between LOT, SCM and our methods (granulated deep learning using 3×3 granules, rectangular granules and using arbitrary-shaped granules)

Method	Accuracy (%)	Speed (fps)	CPU/GPU
LOT	67.6	0.7	CPU
SCM	69	0.51	CPU
Granulated deep learning using 3×3 granules	56.11	1.9	CPU
Granulated deep learning using rectangular granules	67.15	1.6	CPU
Granulated deep learning using arbitrary-shaped granules	67.88	1.2	CPU

Table 5 Time and accuracy for object recognition between SSD and our methods (granulated deep learning using 3×3 granules, rectangular granules and using arbitrary-shaped granules)

Method	Accuracy (%)	Speed (fps)	CPU
SSD	74.3	0.5	CPU
Granulated deep learning using 3×3 granules	54	2.59	CPU
Granulated deep learning using rectangular granules	69	2	CPU
Granulated deep learning using arbitrary-shaped granules	72.3	1.87	CPU

table, horse, airplane, bus and chairs) using the PASCAL VOC data set.

For the purpose of fair comparison of our method with the state-of-the-art algorithms (viz. CFCF [16], CFWCR [17], LOT [18], SCM [19], and SSD [13], we used the same dataset and tasks as used by those authors in their respective studies. Accordingly, the comparing deep learning methods are CFCF and CFWCR in Table 3 for both tracking and recognition using the data PASCAL VOT 2017; LOT and SCM in Table 4 for tracking using the Girl sequence OF TB-50; and SSD in Table 5 for object recognition using PASCAL VOC 2007. In these tables, the accuracy is measured based on the distance between the centroids of the ground truth (provided with the data) and the obtained foreground segment from the respective frames. Time is computed in terms of the number of frames processed per second in the Intel core i5 processor.

As expected, among the various kinds of granulation in the proposed method, the one with arbitrary-shaped granules, characterizing natural granulation, provides the best performance, while the one with 3×3 granules needs the least computation time to process a frame. This is true for all the cases in Tables 2, 3, 4 and 5, thereby demonstrating the effectiveness of the proposed concept of granulated deep learning. The comparing deep learning methods usually provide better accuracy, requiring more computation time, except CFCF (Table 3) which performs better in terms of both time and accuracy than that using arbitrary-shaped granulation.

Let us consider Table 3 as an example for insight analysis, where we made comparison between CFCF, CFWCR and our methods (granulated deep learning using 3×3 granules, rectangular granules and using arbitrary-shaped granules). This is done on the dataset PASCAL Visual Object Tracking 2017, consisting of only one type of objects (human). Here one of our methods, Granulated Deep Learning (GDL) using arbitrary-shaped granules, provides accuracy of 48.59%, whereas the method CFWCR gives accuracy of 48.4%. So our algorithm is giving better accuracy. Moreover, as our claim is to provide speedy algorithm, CFWCR processes 1.4 frames per second, whereas our method takes 1.5 frames per second. So in a longer run, 71.42 sec is needed for 100 frames in CFWCR, whereas our method will need 66.66 sec. Here GDL using arbitrary-shaped granules takes less time, because instead of scanning every pixel as in CFWCR, it does the operation on granulated image in the granulated convolution layer. On the other hand, using the same data set (as in Table 3), CFCF is taking 1.7 frames per second (fps), whereas 1.9 fps is needed by GDL using rectangular granules. So our method is faster, but the accuracy is little less (48.1% vs. 50.9%).

One may note that in our experiment we have dealt with simple indoor and outdoor video sequences. Those

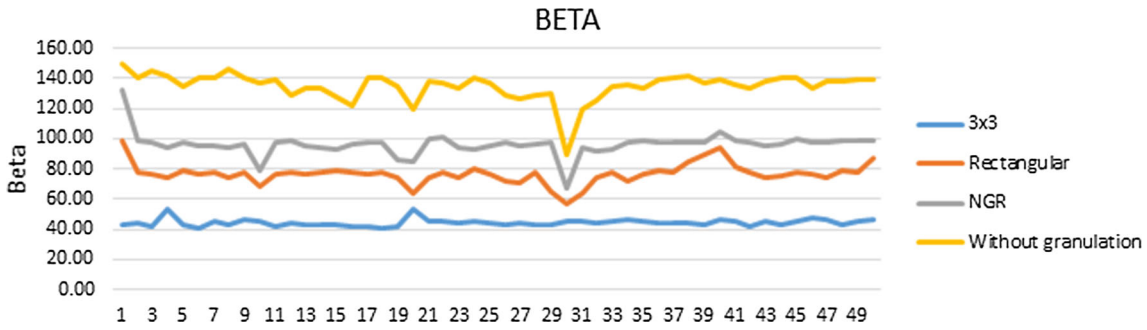

Fig. 7 Variation of Beta index for the 'Changing Appearance' sequence

sequences are supposed to contain very common objects like man, car and chair. It is assumed that the varieties of object(s) present in an unknown sequence will be limited to those of COCO dataset. The data sets that are used here, indeed, contain those categories of objects. It is also assumed that no initial occlusion/ overlapping is present while defining object-background sets.

It may be mentioned that the quality of segmentation of objects in a frame is crucial for their tracking. In a part of our investigation, we have compared the performance of segmentation of all the methods corresponding to Table 2 quantitatively using Beta index [36]. For a given number of object regions, higher the value of Beta, the better is the segmentation. The variation of Beta over frames for the 'Changing Appearance' sequence is shown in Fig. 7. As expected, the arbitrary-shaped granules leading to natural granulation, provides highest Beta index, as compared to the other two types of granulation. As granulation leads to information loss, the conventional deep learning algorithm, involving no granulation, provides better segmentation in terms of Beta index among all, however at the cost of computation time.

5.3 Formulation of Z-number-based measures

In our study, we have used Z-numbers to define measures for object recognition and scene understanding. As described before, the training data set has 20 classes. The granulated neural network predicts the class in which the objects of an unknown scene belongs with certain probability associated with it. Based on this prediction, let us now define some rule bases that would provide the information on A and B of Z-numbers.

Here we define A as the set of classes of similar kind of objects. That is, A = {Animal, Flying Objects, Transport, Two Wheeler, Furniture}. For example: Animal denotes cat, dog, cow and horse, Flying Object denotes airplane and bird, Transport denotes car, bus, train, Two wheeler denotes cycle and motor bike, and Furniture denotes sofa, chair, dining table. Similarly, A can have another possibility as A = {Indoor, Outdoor} where Indoor and Outdoor refer to the location of the aforesaid objects.

B is the set of certainty values. For example B = {Most likely, May be, Not likely at all}. Determining values of B is done in two steps. First, we put some thresholds on P_O (probability of occurrence) of individual object, as obtained from the granulated deep learning network. For example, if P_O of an object is less than 20%, then it is labeled as 'Not Likely' (NL), if the probability is greater than 20% but less than or equal to 70% then it is labeled as 'May Be' (MB), and for greater than 70% it is 'Most Likely' (ML). Having these linguistic values of individual objects, we form different rule bases in the second step, in order to determine the values of B corresponding to each element of A.

Table 6 shows, for example, the rule base, thus formed, for the class "Animal". If all the initial object labels are NL, then the label of the animal class is NL. If all of them are MB then result is ML, otherwise if at least one of them is MB, then the result is MB. And if at least one of them is ML, then the result is ML. We have listed few such representative rules in Table 6 for convenience. Similar rules, derived for Flying Object, Transport, Two Wheeler and Furniture classes, are shown in Tables 7, 8, 9, and 10, respectively.

Table 6 Rule base for animal

Cat	Dog	Cow	Horse	Animal
NL	NL	NL	NL	NL
NL	NL	NL	MB	MB
NL	NL	MB	NL	MB
NL	NL	MB	MB	MB
NL	MB	MB	MB	MB
NL	MB	MB	NL	MB
MB	MB	MB	MB	ML
ML	NL	NL	NL	ML
ML	ML	ML	ML	ML

Table 7 Rule base for flying object

Airplane	Bird	Flying object
NL	NL	NL
NL	MB	MB
MB	NL	MB
MB	MB	ML
NL	ML	ML
ML	ML	ML

Table 8 Rule base for transport

Car	Bus	Train	Transport
NL	NL	NL	NL
NL	MB	NL	MB
MB	NL	MB	MB
MB	MB	MB	ML
MB	MB	ML	ML
NL	NL	ML	ML
ML	ML	ML	ML

Table 9 Rule base for two wheeler

Cycle	Motor bike	Two wheeler
NL	NL	NL
NL	MB	MB
MB	NL	MB
MB	MB	ML
NL	ML	ML
ML	ML	ML

Table 10 Rule base for furniture

Sofa	Chair	Dining table	Furniture
NL	NL	NL	NL
NL	MB	NL	MB
MB	NL	MB	MB
MB	MB	MB	ML
MB	MB	ML	ML
NL	NL	ML	ML
ML	ML	ML	ML

Now for an unknown video frame f_t, its Z-number(s) with A and B, as described before, can be computed to predict the certainty of its constituent objects. Consider, for example, the frame 22 of the Chap scenario [30] (Sect. 5.1). As per the output of granulated neural network, the frame contains sofa with 30%, 4 chairs with percentages 66, 38, 59 and 32, one table with 32% accuracy and a person with 92% accuracy. Z-numbers of the frame f_t are accordingly computed as $Z_O = <\text{Furniture, Most likely}>$ and $Z_O = <\text{Person, Most likely}>$. We have listed in Table 11 the results (Z-number(s) for object recognition) from various frames of different sequences like, Cam 131 Sequence of ICG Lab [30], Jurassic Park [33] and Changing Size Car [35]. One may note that, for objects which are not appearing in the scene, their Z-numbers would contain certainty value as 'Not likely'. Those cases have not been included in Table 11, except the frame number 560, which is shown for illustration.

For scene classification, whether it is indoor or outdoor, the rule base formed is as shown in Table 12. The corresponding Z-numbers of the same set of frames, as used in Table 11, with respect to scene classification are depicted in Table 13. Consider, as an example, the frame number 22 (Table 11), whose Z-measures are computed as $Z_O = <\text{Person, Most likely}>$ and $Z_O = <\text{Furniture, Most likely}>$. According to the rule base (Table 12), if the Furniture class exists in the scene, with a certainty value, then that frame should be classified as Indoor, with the same certainty value. Accordingly, in Table 13, the Z-number for scene classification of this frame is $Z_S = <\text{Indoor, Most likely}>$.

5.4 Significance of Z-number-based measure

The aforesaid description, based on Z-number, provides granulated information of a scene for its understanding. The linguistic description of a frame, as obtained using Z-numbers, has several applications. For example, from their values over frames, one can notice the sudden appearance or disappearance or occlusion of some object(s) in video sequences. Let us consider Table 11 and the frames 490, 498, 553, 554 and 560. In the frame 490, there was no flying object, resulting in $Z_O = <\text{Flying Object, Not likely}>$; and in frame 498, the Z-number was $Z_O = <\text{Flying Object, May Be}>$, i.e., there may be a trace of some flying object. In all the frames up to 553, it was definite that there exists a flying object. In frame 554, the certainty of having a flying object is 'May be'. Finally in frame 560, there was no flying object as $Z_O = <\text{Flying Object, Not likely}>$. From this, one can infer that, the presence, absence or sudden appearance of an object can be noticed automatically using Z-numbers. Accordingly, needful action can be taken, depending on the application, for example, surveillance.

6 Conclusions and discussion

A new approach for motion detection, object recognition and linguistic scene description using deep learning in the framework of granular computing is described. It is shown how the abstract concept of z-numbers can be used to

Table 11 Linguistic description of frames using Z-numbers : Object recognition

Sequence	Frame no	Objects
Chap	22	$Z_O = <$Person, Most likely$>$, $Z_O = <$Furniture, Most likely$>$
Chap	5	$Z_O = <$Furniture, May Be$>$
Chap	12	$Z_O = <$Furniture, Most likely$>$
Jurassic park	490	$Z_O = <$Person, Most likely$>$
Jurassic park	498	$Z_O = <$Flying Object, May Be$>$
Jurassic park	553	$Z_O = <$Flying Object, Most likely$>$
Jurassic park	554	$Z_O = <$Flying Object, May be$>$
Jurassic park	560	$Z_O = <$Flying Object, Not likely$>$, $Z = <$Transport, Most likely$>$
Jurassic park	945	$Z_O = <$Person, Most likely$>$, $Z = <$Vehicle, May Be$>$
Changing car size	19	$Z_O = <$Vehicle, Most likely$>$

Table 12 Rule base for scene classification

Animal	Flying object	Vehicle	Animal	Furniture	Indoor	Outdoor
NL	NL	NL	NL	NL	May be	Not likely
NL	NL	NL	MB	NL	Not likely	May be
MB	NL	NL	NL	NL	Not likely	May be
MB	MB	MB	MB	NL	Not likely	Most likely
MB	MB	MB	ML	NL	Not likely	Most likely
ML	ML	ML	ML	NL	Not likely	Most likely
ML	ML	ML	ML	ML	Not likely	May be
NL	NL	NL	NL	MB	May be	Not likely
NL	NL	NL	NL	ML	Most likely	Not likely

Table 13 Linguistic description of frames using Z-numbers: Scene classification

Sequence	Frame no	Objects
Chap	22	$Z_S = <$Indoor, Most likely$>$
Chap	5	$Z_S = <$Indoor, May Be$>$
Chap	12	$Z_S = <$Indoor, Most likely$>$
Jurassic park	490	$Z_S = <$Indoor, May Be$>$
Jurassic park	498	$Z_S = <$Outdoor, May Be$>$
Jurassic park	553	$Z_S = <$Outdoor, Most likely$>$
Jurassic park	554	$Z_S = <$Outdoor, May be$>$
Jurassic park	560	$Z_S = <$Outdoor, Most likely$>$
Jurassic park	945	$Z_S = <$Outdoor, May Be$>$
Changing car size	19	$Z_S = <$Outdoor, Most likely$>$

quantify the abstraction of semantic information in object classification from a scene, thereby providing a more natural way of interpreting a scene with certainty for its better understanding in natural language.

Various granulation techniques such as 3×3 granules, rectangular granules and unequal-shaped and sized (arbitrary) granules have been used. The theory of rough set is used in defining the background models over the granulated image planes. The method involves separate recognition of static and moving objects. The trained network that we have used is the single-shot detector (SSD) with COCO data set containing 20 classes of objects. Test data has 7 sets of benchmark video sequences, containing 14 types of multiple objects, involved in tasks like, object recognition and/or tracking. The performance in tracking and recognition, with respect to speed and accuracy, is compared with several state-of-the-art deep learning algorithms. The proposed method proves to perform superior to those methods with the said test datasets.

The conventional convolutional neural network (CNN) for deep learning is expensive in terms of time and resource requirement. Incorporation of the proposed concept of granular computing in deep learning reduces the computation time significantly, as it involves scanning only over the granules, instead of each pixel in the input frame. Further, the problem of selecting the appropriate stride, which is crucial in CNN, does not arise here. The method provides a balanced trade-off between speed and accuracy in tracking as compared to pixel level deep learning, and can successfully handle the challenging cases like tracking partial overlapped objects and suddenly appeared objects. Arbitrary-shaped granules, resulting in natural granulation, provides superior performance compared to 3×3 granules and rectangular granules.

The concept of using Z-numbers, in providing a granulated linguistic description of a scene, is unique.

Computation of Z-measure involves the granular information on objects and the certainty in their appearance in the video. This provides an information measure of a scene, by consolidating the classification scores of individual objects belonging to a particular category. This measure can be considered as an index for detecting automatically the change in information in a scene, for example, due to occlusion, sudden appearance and disappearance of objects. Therefore, it promises to have several real-life applications.

Here, we have used COCO data set which deal with simple indoor and outdoor video sequences containing very common objects like man, car etc. to demonstrate the effectiveness of our model. The said model can equally be implemented on larger data sets with same merits and characteristic features.

There are a few assumptions made in our study. These may lead to limitations which could be solved in future. For example, it is assumed that the varieties of object(s) present in a sequence will be limited to those of COCO dataset. The data sets that we are using here falls into these categories. Further, it is assumed that no initial occlusion/ overlapping is present while defining object-background sets. These limitations can be addressed in a further study.

The present investigation primarily demonstrates a way of integrating the concept of granular computing with deep learning networks for speedy computation, and using Z-numbers for quantification of semantic information toward scene understanding. This can be viewed just as a basic module. Further generic variants of these modules can be derived for improved performance depending on the application domain and users' need.

Apart from the aforesaid contributions, the study has enriched the literature of soft computing, particularly for its application in deep learning and z-information measures for scene understanding. In this context, [37, 38] for some new applications of soft computing using neural networks.

Acknowledgements Valuable discussion with Ms. Romi Banerjee is gratefully acknowledged. S.K. Pal acknowledges the INSA Distinguished Professorship. D. Bhunia Chakraborty acknowledges CSIR for providing her Research Associateship.

References

1. Yilmaz A, Javed O, Shah M (2006) Object tracking: a survey. Acm Comput Surv (CSUR) 38(4):13
2. Zadeh LA (1997) Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst 90(2):111–127
3. Pal SK, Uma Shankar B, Mitra P (2005) Granular computing, rough entropy and object extraction. Pattern Recognit Lett 26(16):2509–2517
4. Pawlak Z (2012) Rough sets: theoretical aspects of reasoning about data, vol 9. Springer, Berlin
5. Debarati Chakraborty B, Shankar U, Pal SK (2013) Granulation, rough entropy and spatiotemporal moving object detection. Appl Soft Comput 13(9):4001–4009
6. Chakraborty DB, Pal SK (2016) Neighborhood granules and rough rule-base in tracking. Nat Comput 15(3):359–370
7. Pal SK, Chakraborty DB (2017) Granular flow graph, adaptive rule generation and tracking. IEEE Trans Cybern 47(12):4096–4107
8. Pal SK (2018) Data science and technology: challenges, opportunities and national relevance. In: 14th annual convocation speech: convocation address, National Institute of Technology, Calicut, India, Sept 29
9. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436
10. Erhan D, Szegedy C, Toshev A, Anguelov D (2014) Scalable object detection using deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2147–2154
11. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
12. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
13. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision, Springer, pp 21–37
14. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
15. Kavukcuoglu K, Sermanet P, Boureau YL, Gregor K, Mathieu M, Cun YL (2010) Learning convolutional feature hierarchies for visual recognition. In: Advances in neural information processing systems, pp 1090–1098
16. Erhan Gundogdu A, Alatan A (2018) Good features to correlate for visual tracking. IEEE Trans Image Process 27(5):2526–2540
17. He Z, Fan Y, Zhuang J, Dong Y, Bai H (2017) Correlation filters with weighted convolution responses. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1992–2000
18. Oron S, Bar-Hillel A, Levi D, Avidan S (2015) Locally orderless tracking. Int J Comput Vis 111(2):213–228
19. Zhong W, Huchuan L, Yang MH (2014) Robust object tracking via sparse collaborative appearance model. IEEE Trans Image Process 23(5):2356–2368
20. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L (2014) Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1725–1732
21. Ji S, Wei X, Yang M, Kai Y (2013) 3d convolutional neural networks for human action recognition. IEEE Trans Pattern Anal Mach Intell 35(1):221–231
22. Gan W, Lee MS, Wu CH, Kuo CCJ (2018) Online object tracking via motion-guided convolutional neural network (mgnet). J Vis Commun Image Represent 53:180–191
23. Held D, Thrun S, Savarese S (2016) Learning to track at 100 fps with deep regression networks. In: European conference on computer vision. Springer, pp 749–765
24. Zadeh Lotfi A (2011) A note on z-numbers. Inf Sci 181(14):2923–2932
25. Pal SK, Mitra S (1992) Multilayer perceptron, fuzzy sets, classifiaction. IEEE Trans Neural Netw 3(5):683–697

26. Pan SJ, Yang Q et al (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
27. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision, Springer, pp 740–755
28. Zadeh LA (1996) Fuzzy logic = computing with words. IEEE Trans Fuzzy Syst 4(2):103–111
29. Banerjee R, Pal S (2013) The z-number enigma: a study through an experiment. In: Soft computing: state of the art theory and novel applications, Springer, pp 71–88
30. Possegger H, Sternig S, Mauthner T, Roth PM, Bischof H (2013) Robust real-time tracking of multiple objects by volumetric mass densities. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
31. Visual tracker benchmark data. http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html
32. Kristan M, Matas J, Leonardis A, Vojir T, Pflugfelder Roman, Fernandez Gustavo, Nebehay Georg, Porikli Fatih, Čehovin Luka (2016) A novel performance evaluation methodology for single-target trackers. IEEE Trans Pattern Anal Mach Intell 38(11):2137–2155
33. Jurrasic intro dataset. https://www.youtube.com/watch?v=lc0UehYemQA
34. Ferryman J, Shahrokni A (2009) Pets2009: dataset and challenge. In: 2009 twelfth IEEE international workshop on performance evaluation of tracking and surveillance (PETS-Winter), pp 1–6. IEEE
35. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. https://github.com/mpatacchiola/deepgaze/blob/master/examples/ex_motion_detectors_comparison_video/cars.avi/
36. Pal SK, Ghosh A, Uma Shankar B (2000) Segmentation of remotely sensed images with fuzzy thresholding, and quantitative evaluation. Int J Remote Sens 21(11):2269–2300
37. Taormina R, Chau KW, Sivakumar B (2015) Neural network river forecasting through baseflow separation and binary-coded swarm optimization. J Hydrol 529:1788–1797
38. Wu CL, Chau KW (2011) Rainfall-runoff modeling using artificial neural network coupled with singular spectrum analysis. J Hydrol 399(3–4):394–409