Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Military Object Detection in Defence using Multi-Level Capsule Networks

B Janakiramiaha (bjanakiramaiah@gmail.com)
Pradad V Potluri Siddhartha Institute Of Technology https://orcid.org/0000-0003-3134-0545
Kalyani G
Velagapudi Ramakrishna Siddhartha Engineering College
Karuna A
University College of Engineering kakinada(A)
Narasimha Prasad L V
Institute of Aeronautical Engineering
Krishna M
Sir CR Reddy Engineering College

Research Article

Keywords: Object Recognition , Military Objects , Convolution Neural Network , Capsule Networks , Deep Learning

Posted Date: April 5th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-330732/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Noname manuscript No.
(will be inserted by the editor)

Military Object Detection in Defence using Multi-Level Capsule Networks

B Janakiramaiah[1] · **G Kalyani**[2] · **A Karuna**[3] · **L V Narasimha Prasad**[4] · **M Krishna**[5]

Received: date / Accepted: date

Abstract Automatic target detection plays a major role in automated war operations. The key concept behind automated target detection is military objects recognition from the captured images. For object recognition in the given image, Convolutional Neural Network (CNN) is a powerful classification network. CNN's are location invariants and their performance depends mainly on the size of the training set. The size of the training data is generally available in less proportion for military objects due to its operational and security issues. Hence the performance of CNN may degrade sharply. To address the issue of military objects, a relatively new neural network architecture called Capsule Network (CapsNet) is introduced. Hence, in this article, a variant of CapsNet called Multi-level CapsNet framework is projected for mil-

B. Janakiramaiah
Prasad V. Potluri Siddhartha Institute of Technology
Vijayawada, Andhra Pradesh, india.
E-mail: bjanakiramaiah@gmail.com

G Kalyani
Velagapudi Ramakrishna Siddhartha Engineering College
Vijayawada, Andhra Pradesh, india.
E-mail: kalyanichandrak@gmail.com

A Karuna
University College of Engineering Kakinada(A)
Jawaharlal Nehru Technological University Kakinada,
Andhra Pradesh, India.
E-mail: karunagouthana@gmail.com

L V Narasimha Prasad
Institute of Aeronautical Engineering
Hyderabad, Telangana, India.
E-mail: lvnprasad@iare.ac.in

M Krishna
Sir C R Reddy College of Engineering,
Eluru, Andhra Pradesh, India.
marlapallikrishna@gmail.com

itary object recognition under the case of small training set. The introduced framework of this paper is validated on a dataset of military objects which are collected from the internet. The dataset contains particularly five military objects and the similar civil ones. The proposed framework demonstrates a large improvement of 96.54% of accuracy for military object recognition. Experiments demonstrate that the proposed framework can accomplish a high recognition precision, superior to many other algorithms such as conventional Support Vector Machines and transfer learning based CNNs.

Keywords Object Recognition · Military Objects · Convolution Neural Network · Capsule Networks · Deep Learning

1 Introduction

In the past few decades, wars rely increasingly more upon cutting edge technology, and therefore, the warfare patterns transformed from traditional warfare to informative warfare, which has become the primary type of present day warfare. Quick, proficient, and precise discovery of military objects with the end goal of exact assaults isn't just a fundamental interest for present day warfare, yet in addition an essential component for the improvement of early essential alert systems [1].

Object detection is the foundation for tracking and acknowledging the objects. All the subsequent operations depends on the quality of the detecting the objects in an image. As of now, the generally utilized object identification strategies basically incorporate a few customary ones; for example, feature matching strategy, background displaying technique, edge division strategy, techniques dependent on deep learning, just as strategies dependent on visual remarkable quality. Matching the features technique in the conventional identification methods, [2],[3],[4] has high recognition exactness and precision, however it has low independence and low estimation effectiveness, and its items should be instated physically. Background identification strategies [5],[6],[7] can accomplish programmed separation of items from the background, however the foundation and update of models is tedious and dynamic backgrounds will meddle with the outcomes. Threshold segmentation strategies [8],[9] are advantageous and proficient for circumstances with normal backgrounds and noticeable items, yet the recognition impact under complex conditions isn't acceptable. To summarize, conventional identification calculations have constraints, which make it hard to address the issues of intricate and diverse situations, in real life scenarios. In addition, these techniques are dependent upon manual impedance and their versatile capacities are a long way from being sufficient.

In recent years, there has been a fast and fruitful development on research issues of computer vision. The domain of computer vision is projected to hit significantly with diverse applications from video processing. healthcare and security. Computer vision facilitates the machines with the capacity to see and outwardly sense their general surroundings, like how people utilize their

own eyes. Portions of this achievement have come from implementation and adjusting artificial intelligence techniques, while others from the advancement of new portrayals and models for explicit computer vision issues or from the improvement of effective arrangements.

One sub domain of computer vision that has achieved extraordinary advancement in recent years is object recognition. Object recognition is one type of computer vision that is acquiring force in both the enterprise and consumer networks. Given a set of different objects, object recognition comprises in deciding the area and size of all objects that are available in a picture. Hence, the target of an object recognizer is to discover all objects which occurred at least one with a given object classes paying little heed to scale, area, present, see concerning the camera, halfway impediments, and light conditions. Object recognition is breaking into a wide scope of businesses, with use cases going from individual security to efficiency in the working environment. Facial discovery is one type of it, which can be used as a safety effort to give just certain individuals access to a profoundly characterized zone like defence or military, for instance. It also tends to be utilized to check the quantity of individuals present inside a predefined war territory to consequently change other specialized devices that will help smooth out the time committed to fighting. It can likewise be utilized inside a visual web search tool to help buyers locate a particular thing they're on the chase for,for an instance Pinterest is one illustration of this, as the whole social and shopping stage is worked around this innovation.

In numerous computer vision frameworks, object identification is the principal task being preceded as it permits to get additional data in regards to the recognized item and about the scene. When an object occurrence has been identified like a face, it is conceivable to acquire additional data, like to perceive the particular object i.e., recognizing the subject's face, to follow the item over a sequence of images e.g., to follow the movement of war vehicles in a video, and to remove additional data about the object. Object recognition has been utilized in numerous applications, with the most well-known ones being: interaction between human and computer, robotics, shopper electronics like advanced mobile phones, tracing & tracking the objects in military, search engines and auto driving vehicles. All these applications have various necessities, including: processing time (such as online, real-time and off-line), vigor to faults, and identification in the case of pose changes. While numerous applications consider the location of a single object from a single view, others require the identification of different objects or single object from various perspectives.

The deep learning based techniques for object recognition [10],[11],[12] can be applied to different application scenarios, since they are adaptable and advantageous for displaying from one viewpoint, and are exceptionally differentiated for the discovery and identification of different sorts of objects. This is the motivation behind why they have been applied to a number of applications, like the observing and recognizing the vehicles and walkers. In any case, the detection efficacy of such techniques rely a lot upon the collections,

especially the huge informational indexes and physically marked informational indexes, which further requires a lot of computational assets. The human eye visual mechanism empowers the visual framework to extricate the most fascinating features from the enormous picture information, and thus significantly improving the proficiency of processing the data. Hence, the visual attention strategy has progressively become an interesting issue in the computer vision domain, and in this way dragged the consideration of numerous researchers. For quite a while, numerous scientists have advanced different strategies to acquire significant object, for example, Graph-Based Visual Saliency [13], Frequency Tuned detection [14], regional contrast based object detection [15], cost sensitive Support Vector Machine(SVM) [16], and so forth. Aside from the previously mentioned strategies and their improved variants, numerous new object detection techniques by utilizing deep learning have arisen during the previous two years, for example, Supervised Salient Object Detection [17], recurrent fully connected networks[18], whose standards are to produce saliency maps by the development and preparing of neural networks.

Because of military guidelines on the data to be classified, few efficient investigations have been done in this field in and out of the country. By concerning those works that have been done recently, it is observed that there is a need of efficient methodology or system designed for the task of military objects detection. In order to improve the survivability of weapons and vehicles which are used in the war they will be camouflaged during the non-war time. Consequently, disguise, along with perplexing and alterable war zones, really makes it harder to identify military objects. Considering the qualities and prerequisites of military article recognition tasks, by featuring the impersonation of human visual perception strategy, this paper proposes a methodology for detecting the military objects. The work of this paper explores the following parts: a) introducing a new methodology based on Capsule Networks (CapsNet) of deep learning for detecting the military objects in a given image and b) Collecting a Dataset comprises a sufficient number of military objects to validate the proposed methodology.

2 Review of Literature

Military is an intensely furnished, exceptionally coordinated power essentially deliberated for war. It might comprise branches like an army, air force and naval force. The main aspect in automatic military operations and in mission surveillance is identifying the target automatically. In military operations, sensors can be put on the ground or mounted on automated aeronautical vehicles and automated ground vehicles to collect the data. The primary concern behind Automatic target identification is military objects recognition from the acquired source picture.

The task of detecting the specified objects in a given image has been explored for quite a long time. It is generally viewed as a classification task and numerous sorts of techniques have been intended to address this issue

[19],[20],[21],[22],[23],[24]. In the beginning years, the most acclaimed technique that is utilized to tackle this issue is the SVM [25], which is configured to track down a plane that can isolate the objects of various classes. In most situations, it is elusive to identify the suitable kernels to represent the information to a reasonable space with a task that contains pictures from numerous classes. The primary difficulty is that picture samples are represented with more number of dimensions and even pictures for similar items can show enormous contrasts because of the impact of changes in postures or perspectives on point. To take care of this issue, bag of words [26] is created. It extricates highlights that are invariant from unique pictures and groups them into bag, assuming that a particular word bag addresses a specific semantic idea. Even the re-examined adaptation of a bag of words that contains spatial data actually can't give vigorous features like artificial neural networks.

Concerning neural network strategy, the principal calculation that is broadly utilized in object detection is LeNet [27]. The structure of LeNet is straightforward yet displays great execution in detecting the objects automatically. Later scientists need to extend the organization to get all the more remarkable non-linear learning capacity. A new network was created by researchers called AlexNet [28] by utilizing ReLU activation function and dropouts in the layers. It makes an achievement of overall object detection, by improving the accuracy by over 20% contrasted to the best technique around then. After AlexNet, the techniques are developed in two categories. From one viewpoint, it has been called attention to that deep network structures assumes the main part in improving performance of the network [29]. Common works in this category are VGGNet [30], in which enormous convolutional kernels of size 5 X 5 are supplanted by little portions of size 3 X 3. This change can improve the detection accuracy and lessen the quantity of boundaries simultaneously, making it simpler to prepare a deep network. Then again, analysts built up some more complex neural layers for extracting the features. The most popular ones are GoogLeNet [31] and residual networks [32]. The first network consolidates a few sorts of convolutional parts together to shape another layer, while the later one uses residuals rather than general output by adding an alternate connection from the input to output in every layer. Furthermore, there are a few works like inception residual network which considers both the points in order to improve the detection accuracy of the network [33].

The authors of the article [34] depict various strategies for preprocessing, segmentation, and identification of vehicle estimated protests in LADAR pictures. Five preprocessing techniques are introduced in the article such as median sifting, two 1-D median filters in sequence, Spoke median filter, Donut filter, and Outlier discovery and expulsion. Spoke middle and doughnut filters were essentially not performed well for preprocessing. The remaining filters functioned admirably. Outlier detectors eliminated the outliers while preserving edges and little structures in the image. For segmentation, authors have used four types of area based methods and one method in the category of edge based techniques. The result of segmentation is contributed as input to object detection methodology. In the paper two methodologies are proposed by the

authors. One regular agglomerative grouping approach and one graph based methodology. Finally, the two of them give similar outcomes. Groups with the specified tallness, width, and length inside predefined spans are thought to be potential items. All the techniques were tested by authors on genuine information of different vehicles in various scenes. It is hard to reach any broad determinations. In any case, it appears to be that the area based calculations perform better compared to the edge based ones. Among the locale based techniques, those dependent on morphology or separating activities perform well much of the time.

A framework for detection of military objects is proposed by the authors in [35]. The proposed framework is based on optimal Gabor filtering and deep feature pyramid networks. As an initial step, consolidation of the texture attributes of military items with the necessities of detection task and projected the Fine Region Proposal Network (FRPN). A Gabor filter is planned and screened in the proposed network. They developed the ideal Gabor filter Banks by dissecting the picture energy after Gabor change of certain pictures in the dataset, shorting the time of highlight extraction and lessening the measure of count. At that point, the Renyi limit division strategy is embraced to acquire the region proposition. At last, the Highly Utilized Feature Pyramid Networks (HU-FPN) is proposed to improve the recognition impact of small size items. A base up and a top-down feature pyramid is built in the stage. Through crossover association and reconciliation of highlights at different scales, the component articulation of small objects is enhanced and the identification issue of small items is viably settled. The authors claimed that trial results proved that the strategy proposed has noticeable benefits in exactness, adequacy and small objects recognition when contrasted and the best in class technique, which can make great conditions for the acknowledgement of fast and precise recognition of military items and exact strike under military foundation.

The primary goal of the examination [36] is to help the human administrators, by executing intelligent visual reconnaissance frameworks which help in distinguishing and following dubious or suspected events in the sequence images of a video. The visual observation framework requires quick and hearty strategies for identifying and following moving articles. The authors of the article have explored techniques for distinguishing and following items from unmanned automated vehicles. Moving items were distinguished utilizing versatile background deduction strategy effectively and these identified objects were followed by utilizing Lucas Kanade optical stream tracking, Continuously Adaptive Mean-Shift tracking based methods. The video sequences are changed to tone, immersion and forces which will improve the precision of tracking the objects. In the wake of instating the search window, the shading histogram of the item is registered and saved as a reference. To discover the objects of interest or picture division, histogram back projection21 is used by the authors. The histogram of a picture including the object of interest is made. The following stage is to back-project the histogram over the picture at test where the item should be discovered which is ascertaining the likeli-

hood of whole pixels related to the ground and showing it. The resultant on proper thresholding outputs the ground. The subsequent stage is to compute the item's new size and area utilizing mean-shift technique. CAM Shift depends on mean-shift tracking strategy and was originally proposed to trace human-faces in a user interface. This method has a benefit that it changes the search window adaptively when contrasted with mean-shift tracking.

The authors of article [37] projected a framework called image fused object detector (IFOD). The total framework is made out of four modules: 1) a picture combination module, which can intertwine three diverse sort of pictures into a BGR picture; 2) a feature extractor based on CNN, utilized for removing significant level semantic portrayals from the fused picture; 3) a region of interest (ROI) proposition module controlled on fused picture is used for creating hundreds or thousands of applicant bouncing boxes, for every ROI on highlight map delivered by include extractor module; and 4) a ROI regression and classification is performed to acquire fine bounding boxes and relating class. The authors introduced a novel detection method for the objects of military by intertwining multi-channel CNNs. They join spatial, temporal and thermal data by creating a three-channel picture, and they will be combined as CNN highlight maps in an unsupervised way. The basic concept of the detection methodology is from the quick R-CNN technique and used cross-space move learning strategy to fine tune the CNN model on created multi-channel pictures. The authors tested the proposed technique with the pictures from SENSIAC (Military Sensing Information Analysis Center) data set and contrasted it and the best in class.

Authors of paper [38] introduced a novel methodology for target discovery and classification of objects by clustering the features of the object surface and design highlights extraction. By grouping the surface components, compelling picture segmentation is accomplished and along these lines acquire the design highlights of target objects. Commonplace man-made items including planes, tank, boats, mines and vehicles in normal background can be distinguished. A portion of the picture handling procedure utilized incorporates DCT, Morphological Enhancing, Texture Feature Clustering, Segmentation, and Structure Feature Extraction.

To tackle this issue of military object recognition, a deep transfer learning technique is proposed by the researchers in the article [39]. The principle thought of the methodology comprises two sections: transfer learning for knowledge embedding and blended layer for better feature extraction. It has been demonstrated that the capacity of feature extraction learned in an enormous dataset is useful to related tasks and can be moved to another neural network. The transfer learning is accomplished by fixing the loads of certain layers and afterwards retraining the remaining layers. The vital issue for deep transfer learning is what part ought to be moved and what part ought to be retrained to adjust the network to the new application. This issue is addressed by broad trials, and it is discovered that retraining the last three layers and moving before different layers can arrive at the best exhibition. Moreover, the authors utilized a mixed layer plan to utilize the current data. In each mixed

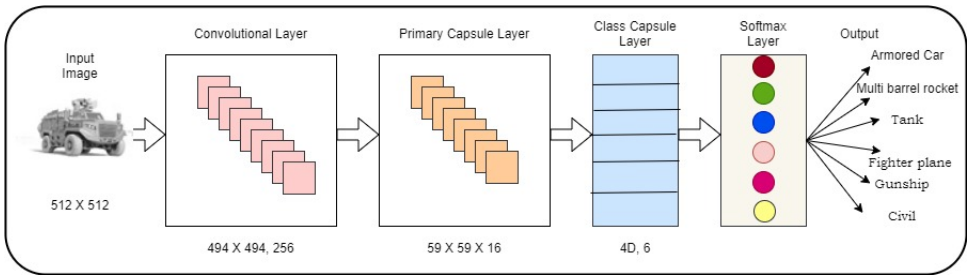

Fig. 1 Multi-level CapsNet Architecture for Military Object Detection

layer, convolution channels in various scales are consolidated together, assisting with adjusting highlights in various scales. The authors proved that by utilizing these two strategies, the proposed strategy shows an enormous improvement in military object detection even though the training data available is very small.

3 Proposed methodology for military object detection

This section explains the methodology of military object detection using capsule networks.

3.1 Proposed Multi-level CapsNet Architecture

For the identification of military objects in the given image, we designed a Multi-level CapsNet which is portrayed in Figure1. The operations convoluted in the implementation of the Multi-level CapsNet architecture are as follows:

- Convolution layer for extracting the features from the input image.
- Primary capsule layer to process the extracted features.
- Class capsule layer on which dynamic routing has been performed to process the features for identification and classification.
- SoftMax layer to convert class capsule layer result into probabilities corresponding to each class considered in military objects.

Six different military objects images are considered as input image for the CapsNet. The input images are applied to preprocessing to elevate the contrast of the picture and to remove the noise in the images. The images are converted to gray scale to consider as input to the proposed multi level CapsNet. All the images are also resized to unique size after converting to the grayscale and preprocessing. Hence, the result of the preprocessing is high contrast smooth gray scale images of size 512 x 512 which are considered as input to the proposed multi-level CapsNet architecture.

3.2 Convolution Layer

Convolutional layers are the ones in which kernels can be applied to the input picture, or to the result of previous convolutional layer. A convolution operation in a convolution layer is a direct activity that includes the multiplication of the set of weights with given input. Given that the strategy was intended for two-dimensional information, the multiplication can be done between an input data of 2D array and a 2D array of weights, called as a kernel or filter. The size of the kernels is very small in size when compared to the input picture size. The kernel is slid across the width and height of the input volume and for each spatial position dot product is applied between the input volume and kernel. The dot product is applied as element wise multiplication among the kernel sized part of the input image and the kernel weights. The element wise multiplications are then added to form a single value. Since it brings about a single value, the activity is regularly alluded to as the "scalar product". Because the kernel is of smaller size than the input volume size, the same kernel is moved multiple times along the input volume size. Particularly, the kernel is applied deliberately to each kernel sized part of the input volume, in a direction left to right, and then top to bottom.

The deliberate use of the same kernel across a picture is an influential thought. In the event that the kernel is intended to identify a particular kind of feature in the given input, at that point the use of that kernel across the whole input picture permits the kernel a chance to find that feature which is exists at any place in the picture. The result of applying the kernel multiple times in 2D array of values gives the filtered input. The result of convolution operation in a convolution layer is called as a feature map. Then the feature map is given as input to a non-linear activation function such as ReLu, tanh and sigmoid to check whether the required feature is present at a specified location of the input image. But in order to identify the type of the picture in the given input image single feature identification is not sufficient. We need to extract more features. Hence, in order to extract the different features that existed in the given input image the number of kernels are applied on the input image. Each kernel gives a separate feature map of the input image. The network can be designed with a number of convolutional filters in sequence by adding more layers and generating more feature maps. The feature maps which are created by deeper layers are more and more abstract features to recognize the objects in the given image.

In designing a network with convolution layers there are four important hyper parameters to be tuned. They are kernel size, kernel count, stride and padding. The generally used kernel size is 3x3. 5x5, 7x7 and 9x9 are likewise utilized relying upon the input image and application. The kernel dimensions are depending on the type of the input image. If the input image is black/white image then 2D kernels are sufficient. If the input image is an RGB image then the used kernels are also of 3D, because the depth of a kernel at a particular layer is equivalent to the depth of its input. The number of kernels in the network is purely dynamic, which is represented as a power of 2 in the range

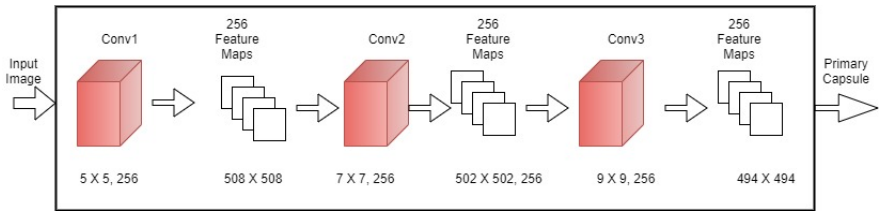

Fig. 2 Convolution Layer of the Multi-Level CapsNet Architecture

32 to 1024. As the number of filters increases, the set of extracted features also increases. Stride: Stride indicates the movement of the kernel on the input image. General set to the default value of 1 which indicates moving only one cell either towards right and bottom. The size of the image reduces by applying the convolution operation. So there should be a limit on the number of times convolution operation can be applied on the input image. To avoid this after every convolution operation the result is again padded to the same size. As the number of convolutional layers increases, the parameters or weights used in the layer also gets increased. For reducing the number of parameters used in convolution layer, Parameter sharing is used. The concept of parameter sharing is weights will be shared by all neurons in a specific feature map. The weight sharing decreases the number of parameters to be maintained in order to maintain efficiency of learning, and good generalization.

Figure 2 shows the design of the convolution layer used in the proposed multi-level CapsNet architecture. The preprocessed input image of size 512 x 512 is considered as input to the convolution layer. The convolution layer is designed with three internal layers of type convolution named as Conv1, Conv2, and Conv3. In all the three layers 256 filters are used with default stride as 1. The filter sizes of 5x5, 7x7 and 9x9 are used in layers in conv1, conv2 and conv3 respectively.

For the first layer (Conv1) the input is of size 512 x 512 and a kernel of size 5x5 with stride 1 is applied. The resulting feature map is of size $[\frac{512-5}{1}+1]$ i.e., [508x508, 256] which is considered as input to the next layer (Conv2). The kernel in Conv2 is also 7x7. Hence, the resulting feature map of Conv2 is of size $[\frac{508-7}{1}+1]$ i.e., [502x502, 256] which becomes input to the next internal layer (Conv3). Due to the kernel of size 9x9 in conv3, the resulting feature map of Conv3 is of size $[\frac{502-9}{1}+1]$ i.e., [494x494, 256]. Hence the final output size of the convolution layer which is considered as input to the next layer of multi-level CapsNet architecture is [494x494, 256]. The number of parameters to be trained in internal layers Conv1, Conv2, and Conv3 are shown in Table 1. A total of 40,448 parameters are tuned in the convolution layer.

3.3 Primary Capsule Layer

The successive layer to the convolution layer is the primary capsule layer. It consists of three distinct processes: Convolution, Reshape function, and Squash

Table 1 Number of parameters trained in Convolution Layer

Name of the Internal Layer	No.of parameters to be trained
Conv1	256*(5*5+1)=6,656
Conv2	256*(7*7+1)=12,800
Conv3	256*(9*9+1)=20,992

function. The input to the primary capsule layer is fed from the convolution layer. The result is an array of feature maps; For example consider that the output is an array of 36 feature maps. Then reshaping function is applied to these feature maps and for an instance it is reshape into 4 vectors of 9 dimensions each (36=4*9) for every location in the image. Now, the last process squashing is applied to guarantee that each vector length is at most one only because the length of every vector indicates the probability of either the object is located or not in the given location of the image. Hence, it should be between 1 and 0. For this purpose Squash function is used in primary capsule layer. This function just ensures that the length of the vector is between 1 and 0 without altering the position information.

In the proposed multi-level CapsNet architecture 16D primary capsules are used and each primary capsule is produced by a small spatial area of the given input image. We designed the primary capsule layer with three levels as shown in Figure 3 by forming sixteen primary capsules based on the result taken from the convolution layer. The extracted capsules of first level are further used to produce another set of sixteen (2nd level) primary capsules which are then used to produce another sixteen (3rd level) primary capsules.

Figure 3 depicts the primary capsule layer of the proposed Multi-level CapsNet Architecture. From the convolution layer the feature map of size [494x494, 256] is taken as input to the primary capsule layer. On the input feature map a convolution is applied with 3x3 kernel and stride as 1. The resulting feature map size is $[\frac{494-3}{1}+1]$ i.e., [492x492, 256]. Subsequent to this, it consists of 16 primary capsules. The task of primary capsules is to take main features identified by the convolution and generate the different combinations of the identified features. The layer has 16 "primary capsules" that work in the same manner of the convolutional layer in their characteristics. Each capsule applies 3x3 kernels (with stride 2) to the 492x492 input volume and hence produces 247x247 output volume. The final output volume size of level 1 primary capsules is 247x247x16 because 16 such capsules are used. The output of the level 1 primary capsules is fed to the level 2 primary capsules. In level two also initially convolution is applied with 3x3 kernel and stride as 1. The obtained result is of size $[\frac{247-3}{1}+1]$ i.e., [245x245, 256]. Then it is passed on to 16 capsules each of 16D. As like in level 1, these 16 capsules also work like convolutions individually. Each capsule applies 3x3 kernels (with stride 2) to the 122x122 volume. The output of the level 2 primary capsules is fed to the level 3 primary capsules. In level three also initially convolution is applied with 3x3 kernel and stride as 1. The obtained result is of size $[\frac{122-3}{1}+1]$

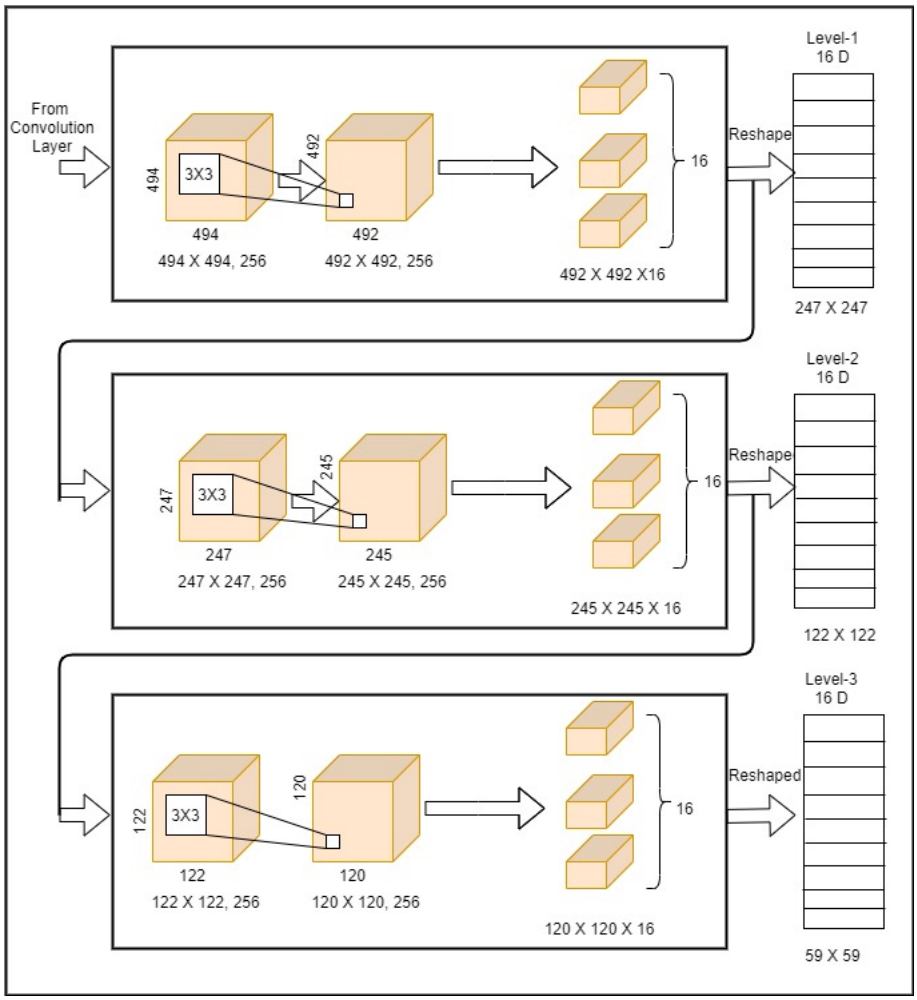

Fig. 3 Primary Capsule Layer of the Mult-level CapsNet Architecture

i.e., [120x120, 256]. Then, it is passed on to 16 capsules each of 16D. As in level 1 and level 2, these 16 capsules also work like convolutions individually. Each capsule applies 3x3 kernels (with stride 2) to the 120x120 volume which produces result of size $[\frac{120-3}{2}+1]$ i.e., [59x59, 256]. The final output volume size of level 3 primary capsules is 59x59x16 i.e., each consists of 16 dimensions.

3.4 Class Capsule Layer

Class capsule layer in multi-level CapsNet is the replacement for max pooling layer of CNN with dynamic routing-by-agreement [40]. The class capsule layer of the proposed multi-level CapsNet architecture is shown in Figure 4. The

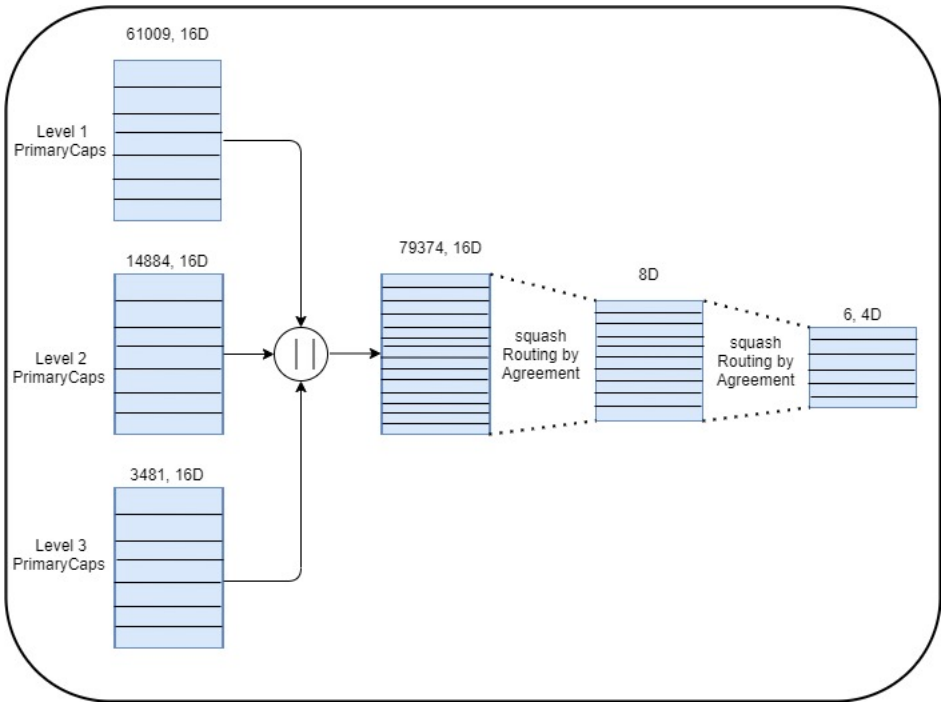

Fig. 4 Class Capsule Layer of the Multi-level CapsNet Architecture

class capsule layer is designed in two levels. For level 1 output of the primary capsule layer is considered as input and for level 2, result of level 1 is considered as input. Squashing and routing by agreement is used in level 1 and level 2 class caps.

The reshaped results of primary capsule layer level 1 level 2 and level 3 are the inputs to the class capsule layer. The first input which is the output of level 1 primary capsules is of size 61009x16D. The second input which is the output of the level 2 primary capsules is of size 14884x16D. The third input which is the output of the level 3 primary capsules is of size 3481x16D. In class capsule layer these inputs are concatenated to form another input of size 79374x16D. For these, class caps are implemented separately which forms two class caps. The level 1 class caps are then considered again to another level class caps which are called level 2 class caps. Between level 1 class caps and level 2 class caps also dynamic routing is implemented.

The process of routing by agreement is illustrated as follows. The i^{th} previous layer (l) capsule output denoted as v_i is considered as input for subsequent layer (l+1) capsules. The j^{th} capsule of that layer takes the input v_i applies the product with corresponding weight w_{ij} between the i^{th} capsule and j^{th} capsule. The resultant is denoted as $v_{j|i}$ which contributes the contribution of i^{th} capsule of (l) layer to the j^{th} capsule of (l+1) capsule.

$$v_{j|i} = W_{ij} * v_j$$

A weighted sum s_j of all the primary capsule predictions for the class capsule j is calculated with the squashing function as:

$$S_j = \sum_{i=1}^{N} C_{ij} * v_{j|i}$$

To ensure that this result is between 0 and 1, squashing function is applied which is computed as follows:

$$SQ_j = \frac{||S_j||^2 * ||S_j||}{1+||S_j||^2 * ||S_j||}$$

The process of dynamic routing by agreement is illustrated in the following Algorithm 1.

Algorithm 1: Process of Routing by agreement

Algorithm 1 :Dynamic Routing

procedure ROUTING($v_{j|i}, m, l$)
  for every l layer capsule i to (l+1) layer capsule j **do**
    $w_{ij} \leftarrow 0$;
  for iteration 1 to m **do**
    for every capsule i ∈ layer l **do**
      c ← softmax(w_i);
    for every capsule j ∈ layer(l+1) **do**
      $S_j \leftarrow \sum_i C_{ij} * v_{j|i}$;
    for every capsule j ∈ layer (l+1) **do**
      $SQ_j \leftarrow squash(S_j)$;
    for every capsule i ∈ layer l and capsule j ∈ layer (l+1) **do**
      $w_{ij} \leftarrow w_{ij} + v_{j|i} * SQ_j$;
  return SQ_j

The class capsule layer requires training of more number of parameters. The trainable parameters as C_{ij} is computed as number of vectors received from Primary capsule layer multiplied with number of vectors required as output. i.e., in level 1 we have three inputs and concatenation of the three is considered for processing.

Hence for level 1:

$$\begin{aligned} &= (61009 + 14884 + 3481) \times 1024) \\ &= 79374 \times 1024 \\ &= 81278976 \end{aligned}$$

for level 2:

$$= 1024 \times 6$$

$$= 6144$$

The total trainable parameters in both the levels are 81,285,120. Parameter training is also required at conversion of scalar values into vector which happen between the capsules, i.e., W_{ij}. The trainable parameters here are computed as follows:

W_{ij} between 16D to 8D capsules + W_{ij} between 8D to 4D capsules

$$= 81278976 \times 16 \times 8 + 6144 \times 8 \times 4$$

$$= 10403708928 + 196608$$

$$= 10403905536$$

Hence the total trainable parameters in class capsule layer are 10,485,190,656.

4 Experimental Investigations

The potential of the proposed model is demonstrated with a dataset of military object images and compared with the predictions of support vector machine and CNN architecture. The implementation of the proposed algorithm is done in python because of the wide availability of the libraries and frameworks for deep learning. To build the deep learning architectures, Keras and TensorFlow are used in the backend. Experiments were done on DELL PowerEdge R740 Server with 2 X Intel Xeon Gold 6226R- 2.9G, 16 C, 32T, 22 M Cache, NVIDIA Quadro RTX8000, 48 GB GDDR6.

4.1 Dataset

To validate the proposed multi-level capsule network based model, the dataset used in the experiments are self-built set with five different military objects (armored car, multi-barrel rocket, tank, fighter plane and gunship) and some general objects (named here as civil) which are similar to the military objects. The created dataset contains 3500 images collected from the Internet. This dataset contains 600 images for each category of the military objects. In the military object dataset, objects are very similar to some military objects but generally normal objects are also included with the name civil. For instance, the civil airplane and the transport aircraft. So, 500 samples of this category are also included to test the algorithms ability of identifying the objects of the category civil. Based on the category, these images are labeled to their respective classes. The details of military objects images in the collected dataset are given in Table 2. The sample object images of six classes are shown in Figure 5.

Table 2 Distribution of military objects images in the dataset.

Image category	Class Label	No.of Images
Armored Car	C0	600
Multi-barrel Rocket	C1	600
Tank	C2	600
Fighter Plane	C3	600
Gunship	C4	600
Civil	C5	500
	Total	**3500**

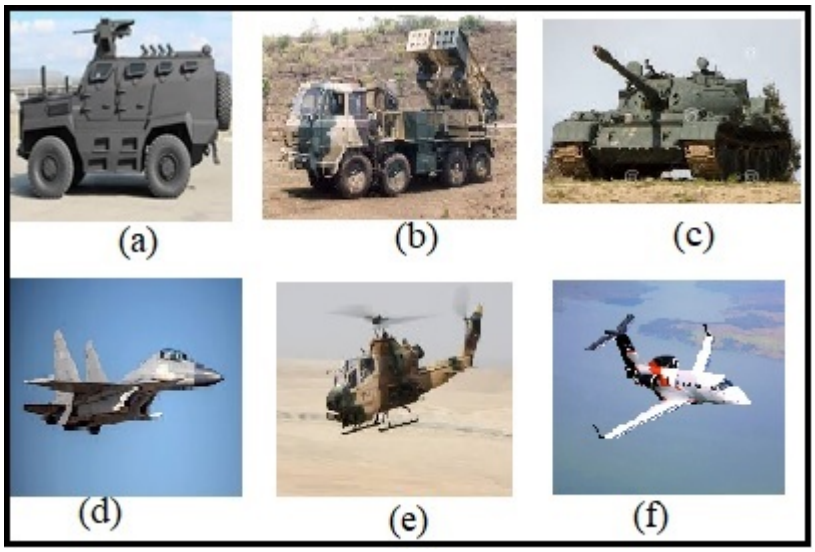

Fig. 5 Sample images in the Collected Dataset.(a) Armored car (b) Multi-barrel Rocket (c) Tank (d) Fighter Plane (e) Gunship (f) general object

4.2 Performance Metrics

For experimentation we used 10-fold cross validation method. The dataset is randomly divided into ten parts. The experimentation has done ten times. Every time one part is considered as testing by considering the remaining four parts as training set. The results of the ten experiments are accumulated for comparison of the results.

When constructing a classification model, estimating how precisely it predicts the correct result is significant. But, this estimation alone is not sufficient as it conveys wrong results in some cases. That is the situation where the additional measures become an integral factor to conclude the more significance estimations of the constructed model.

The performance outcomes that can be evaluated based on confusion matrix are accuracy, precision, specificity, recall or sensitivity, F1-score. For every class the measures are evaluated separately. The average of all classes can be considered as the final value for that measure.

Accuracy is an essential metric for classification models. It is easy to understand and simple to apply for binary and also for multi-class classification problems. Accuracy indicates the proportion of true results in the total number of records tested. Accuracy is effective for assessing the classification model which is constructed from balanced datasets only. Accuracy may interpret wrong results if the given dataset for classification is skewed or imbalanced.

Precision indicates the proportion of the true positives in predicted positives. Another important measure is recall which conveys more information in case if capturing all possible positives is important. Recall indicates the fraction of total positive samples was correctly predicted as positive. Recall is 1 if all positive samples are predicted as positive. If optimal blend of precision and recall is required then these two measures can be combined as a new measure called F1-score. F1-score is the harmonic mean of the precision and recall which lies between 0 and 1. The formulas to evaluate all these measures are shown in Equations 1 to 4.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precition = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - Score = \frac{2 \times Precition \times Recall}{Precition + Recall} \tag{4}$$

In practical a model is to be constructed with precision and recall as 1 which in turn gives F1-score as 1, i.e. a 100% accuracy which is not feasible in classification task. Hence, the constructed classification model should have higher precision with a higher recall value.

4.3 Discussion on Results

The experimentation was done by considering conventional classification method SVM, CNN based model [39] and the proposed Multi level CapsNet based model.

The experiment-1 is conducted by considering SVM as the classification algorithm. The obtained confusion matrix is shown in Table 3. Among the 600 images of each class Armored car, Multi-barrel Rocket, Tank, Fighter Plane, Gunship, and Civil the correctly identified instances are 325, 304, 365, 336 and 362 respectively. 86 armored car objects are identified as multi-barrel rockets and 102 armored car objects are identified as tank by SVM algorithm. 67 and 85 multi-barrel rockets are identified as armored car and tank respectively. 68 and 102 tank object images are classified as armored car and multi-barrel tank

Table 3 Confusion Matrix of Experiment-1: By applying SVM Classification

		Predicted Class					
		Armored car	Multi-barrel Rocket	Tank	Fighter Plane	Gunship	Civil
Actual Class	Armored car	325	86	102	0	0	87
	Multi-barrel Rocket	67	304	85	0	0	144
	Tank	68	102	365	0	0	65
	Fighter Plane	0	0	0	336	105	159
	Gunship	0	0	0	112	362	126
	Civil	5	0	0	102	95	298

Table 4 Observations based on the confusion matrix of Table 3.

	Armored car	Multi-barrel Rocket	Tank	Fighter Plane	Gunship	Civil
TP	325	304	365	336	362	298
TN	2760	2712	2713	2686	2700	2419
FN	275	296	235	264	238	202
FP	140	188	187	214	200	581

Table 5 Confusion Matrix of Experiment-2: By applying CNN-TL

		Predicted Class					
		Armored car	Multi-barrel Rocket	Tank	Fighter Plane	Gunship	Civil
Actual Class	Armored car	434	56	73	0	0	37
	Multi-barrel Rocket	32	357	75	0	0	136
	Tank	24	75	496	0	0	5
	Fighter Plane	0	0	0	392	56	152
	Gunship	0	0	0	105	457	38
	Civil	5	0	0	34	18	443

objects respectively. 105 fighter plane images are misclassified as gunship and 112 gunship objects are misclassified as fighter plane objects.

Based on the confusion matrix of Table 3, the True-Positives (TP), True-Negatives (TN), False-Positives (FP) and False-Negatives (FN) are estimated for each class separately. The estimated values are shown in Table 4. Based on the estimations of Table 4, performance measures accuracy, precision, recall and f-score are evaluated for each class separately. The performance measures of support vector machine classification algorithm are shown in Table 9 and 10. The support vector machine algorithm achieved the highest accuracy of 88.14% for armored car class objects.

The experiment-2 was conducted by considering CNN with Transfer Learning (CNN-TL) from the literature [39] as the classification algorithm. The obtained confusion matrix is shown in Table 5. Among the 600 images of each class Armored car, Multi-barrel Rocket, Tank, Fighter Plane, and Gunship the correctly identified instances are 434, 357, 496, 392 and 457 respectively. 56 armored car objects are identified as multi-barrel rockets and 73 armored car objects are identified as tanks by CNN-TL network. 32 and 75 multi-barrel rocket objects are identified as armored car and tank objects respectively. 24 and 75 tank object images are classified as armored car and multi-barrel tank objects respectively. 56 fighter plane images are misclassified as gunship and 105 gunship objects are misclassified as fighter plane objects.

Table 6 Observations based on the confusion matrix of Table 5.

	Armored car	**Multi-barrel Rocket**	**Tank**	**Fighter Plane**	**Gunship**	**Civil**
TP	434	357	496	392	457	443
TN	2839	2769	2752	2761	2826	2632
FN	166	243	104	208	143	57
FP	61	131	148	139	74	368

Table 7 Confusion Matrix of Experiment-3: By applying the proposed multi-level CapsNet architecture

		Predicted Class					
		Armored car	**Multi-barrel Rocket**	**Tank**	**Fighter Plane**	**Gunship**	**Civil**
Actual Class	**Armored car**	502	29	32	0	0	37
	Multi-barrel Rocket	21	519	33	0	0	27
	Tank	18	35	544	0	0	3
	Fighter Plane	0	0	0	498	45	57
	Gunship	0	0	0	51	488	61
	Civil	5	0	0	24	18	453

Based on the confusion matrix of Table 5, the TP, TN, FP, and FN of experiment-2 are estimated for each class separately. The estimated values are shown in Table 6. Based on the estimations of Table 6, performance measures accuracy, precision, recall and F1-score are evaluated for each class separately based on CNN-TL. The performance measures of CNN-TL network are shown in Table 9 and 10. The CNN-TL based classification has achieved highest accuracy of 93.8% for gunship object class, 93.51% for armored car, 92.8 for tank object class, 90.09% for fighter plane class and 89.31% for multi-barrel rocket class objects

Experiment-3 was conducted by considering the proposed multi-level CapsNet architecture as the classification algorithm. The obtained confusion matrix is shown in Table 7. Among the 600 images of each class Armored car, Multi-barrel Rocket, Tank, Fighter plane, and Gunship, the correctly identified instances are 502, 519, 544, 498 and 488 respectively. 29 armored car objects are identified as multi-barrel rockets and 32 armored car objects are identified as tanks by proposed architecture. 21 and 33 multi-barrel rockets are identified as armored car and tank respectively. 18 and 35 tank object images are classified as armored car and multi-barrel rocket objects respectively. 45 fighter plane objects are misclassified as gunship and 51 gunship objects are misclassified as fighter plane objects.

Based on the confusion matrix of Table 7, the TP, TN, FP, and FN of experiment-3 are estimated for each class separately. The estimated values are shown in Table 8. Based on the estimations of Table 8, performance measures accuracy, precision, recall and F1-score are evaluated for each class separately. The performance measures of proposed architecture are shown in Table 9 and 10. The proposed architecture has achieved on an average an accuracy of 95% for all the five classes military objects.

Table 9 shows the accuracy and precision for each class of the three experiments. Table 10 shows the recall and F1-score for each class of the three

Table 8 Observations based on the confusion matrix of Table 7.

	Armored car	Multi-barrel Rocket	Tank	Fighter Plane	Gunship	Civil
TP	502	519	544	498	488	453
TN	2856	2836	2835	2825	2837	2815
FN	98	81	56	102	112	47
FP	44	64	65	75	63	185

Table 9 Class wise Accuracy and Precision Results of the three experiments

	ACCURACY (%)			PRECISION (%)		
	SVM	**CNN+TL**	**PROPOSED**	**SVM**	**CNN+TL**	**PROPOSED**
Armored car	88.14	93.51	95.94	69.89	87.68	91.94
Multi-barrel Rocket	86.17	89.31	95.86	61.79	73.16	89.02
Tank	87.94	92.8	96.54	66.12	77.02	89.33
Fighter Plane	86.34	90.09	94.94	61.09	73.82	86.91
Gunship	87.49	93.8	95	64.41	86.06	88.57
Civil	77.63	87.86	93.37	33.9	54.62	71

Table 10 Recall and F1-score Results of the three experiments

	RECALL (%)			F1-SCORE (%)		
	SVM	**CNN+TL**	**PROPOSED**	**SVM**	**CNN+TL**	**PROPOSED**
Armored car	54.17	72.33	83.67	61.03	79.27	87.61
Multi-barrel Rocket	50.67	59.5	86.5	55.68	65.63	87.74
Tank	60.83	82.67	90.67	63.36	79.75	90
Fighter Plane	56	65.33	83	58.43	69.32	84.91
Gunship	60.33	76.17	81.33	62.3	80.81	84.8
Civil	59.6	88.6	90.6	43.22	67.58	79.61

experiments. On an average with SVM classification an accuracy of 85.16% is achieved while with CNN-TL and proposed architecture have 91.2% and 95.2% accuracy. The proposed architecture has 86.12% of precision and 85.96% of recall and 85.77% of F1-score on average of all classes.

The box plot analysis of the performance measures accuracy, precision, recall and F1-score are shown in Figure 6, 7, 8 and 9 respectively. Box plots visually illustrate the distribution of numerical data and skewness through displaying the data quartiles (or percentiles) and averages. Box plots visualize the five number summaries of the results of all three experiments. The box plot shows minimum, Q1 (25th quartile), Q2 (50th quartile), Q3 (75th quartile) and max for the individual class results of all the three experiments. From Figure 6 it is clear that the proposed architecture has achieved approximately minimum 93% accuracy and maximum 96.5% accuracy, and 50th quartile as 95% accuracy. Figure 7 shows that the proposed architecture got approximately minimum 87% accuracy and maximum 92% precision, and 50th quartile as 89.5% precision. Figure 8 shows that the proposed architecture got approximately minimum 61% accuracy and maximum 82% recall. Figure 9 shows that the proposed architecture got approximately minimum 85% accuracy and maximum 90% F1-score.

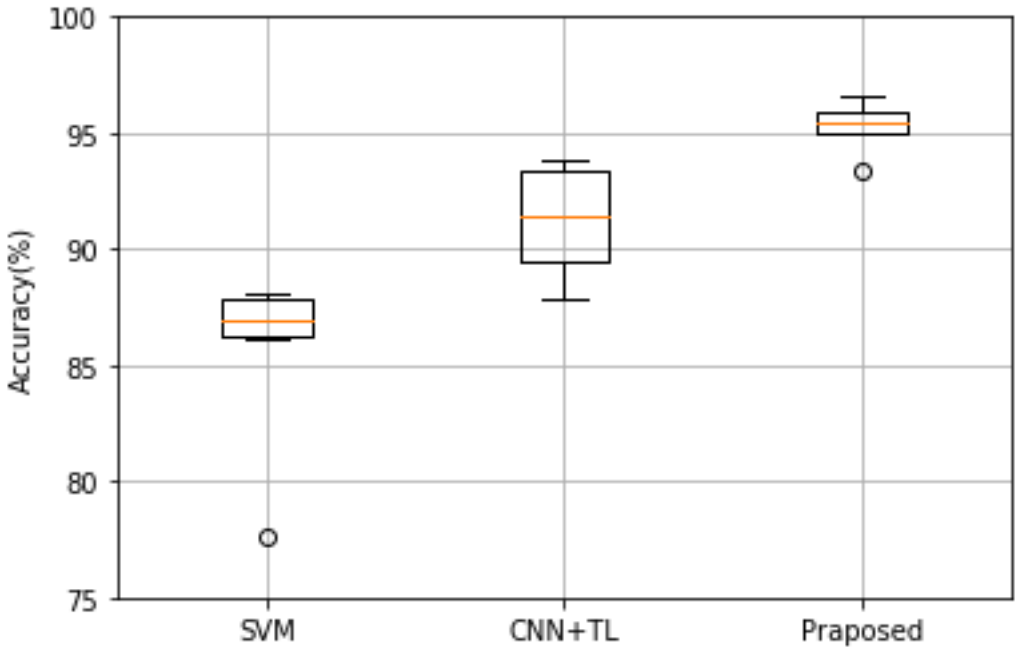

Fig. 6 Comparison of Accuracy with Box Plot Analysis.

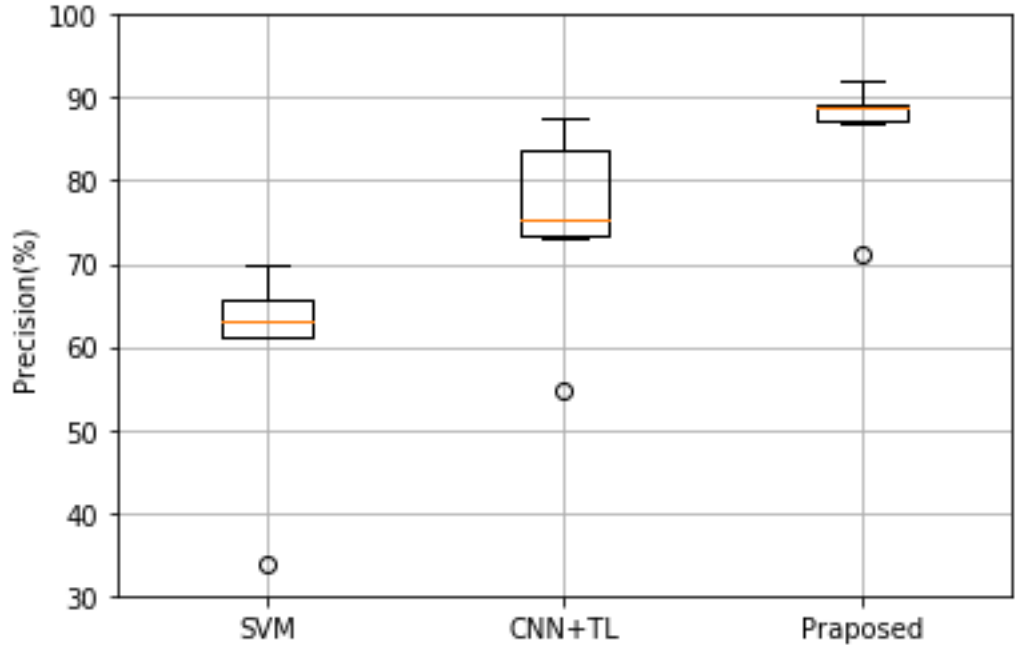

Fig. 7 Comparison of Precision with Box Plot Analysis.

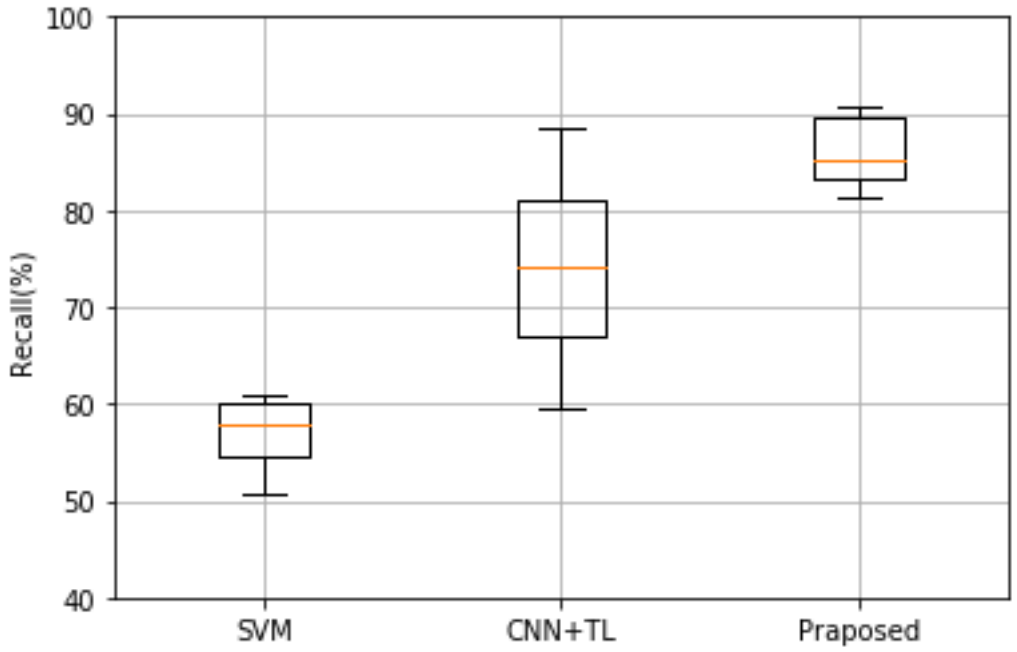

Fig. 8 Comparison of Recall with Box Plot Analysis.

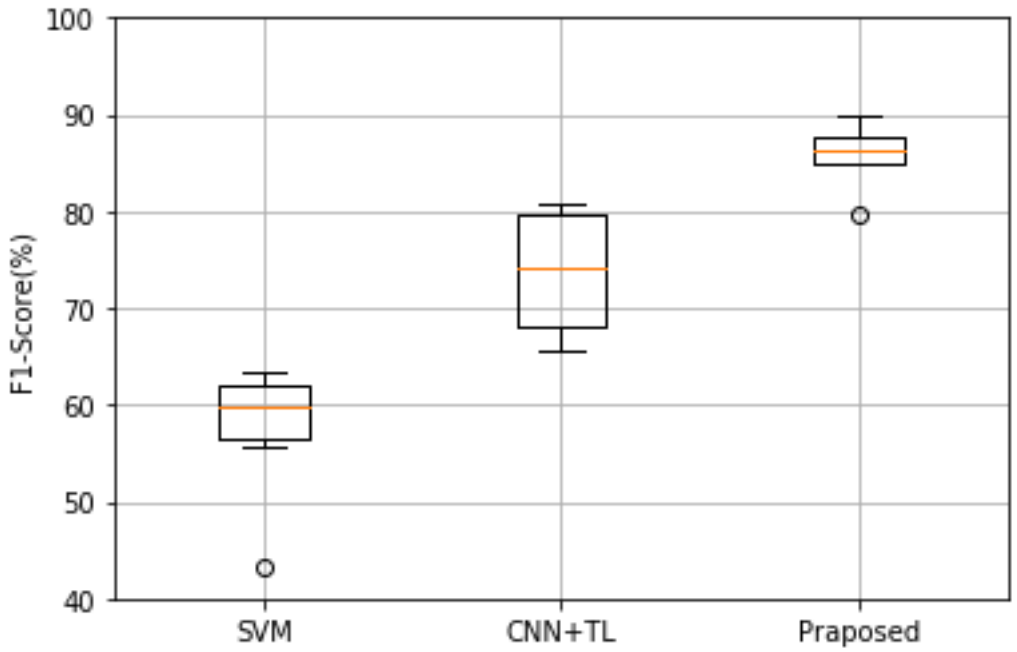

Fig. 9 Comparison of F1-score with Box plot Analysis.

5 Conclusion

The success of war in the military domain depends on automated war strategies followed by the respective departments. Automated target identification plays a vital role in automated war strategies. Detecting the military objects from the captured images is the key point in automated target identification. Deep learning architectures have outperformed results in detecting and classifying the military objects by examining the features in the given input image. Hence, in this article a framework named multi-level CapsNet was introduced from the deep learning domain for classifying the five different types of military objects. The proposed architecture got 96.54% accuracy when compared with the transfer learning based CNN architecture. The work of this article can be enhanced in future by considering other types of military objects also. The same architecture can also be extended for object detection in other branches of military such as navy and air force. In summary, the article projects a deep learning based framework for military object recognition for the purpose of automatic target identification in warfare.

Conflict of interest

The authors declare that they have no conflict of interest.

Contributions

Each author has equally contributed in conceptualization, model building, simulation, and writing of the article.

Corresponding author

Correspondence to B.Janakiramaiah

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Sun, Y.; Chang, T.; Wang, Q.; Kong, D.; Dai, W. A method for image detection of tank armor objects based on hierarchical multi-scale convolution feature extraction. J. Ordnance Eng. 2017, 38, 1681–1691.
2. Dollar, P.; Appel, R.; Belongie, S.; Perona, P. Fast feature pyramids for object detection. IEEE Trans. Pattern Anal. Mach. Intell. 2014, 36, 32–45.

3. H.; Wang, J.; Yuan, Z.; Wu, Y.; Zheng, N.; Li, S. Salient object detection: a discriminative regional feature integration approach. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2083–2090.
4. H. Feature-centric evaluation for efficient cascaded object detection. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), Washington, DC, USA, 27 June–2 July 2004;
5. Li, L.; Huang, W.; Gu, I.Y.-H.; Tian, Q. Statistical modeling of complex backgrounds for foreground object detection. IEEE Trans. Image Process. 2004, 13, 1459–1472.
6. Prasad, D.K.; Rajan, D.; Rachmawati, L.; Rajabally, E.; Quek, C. Video Processing from electro-optical sensors for object detection and tracking in a maritime environment: A Survey. IEEE Trans. Intell. Trans. Syst. 2017, 18, 1993–2016.
7. Savaş, M.F.; Demirel, H.; Erkal, B. Moving object detection using an adaptive background subtraction method based on block-based structure in dynamic scene. Optik 2018, 168, 605–618.
8. Sultani, W.; Mokhtari, S.; Yun, H.B. Automatic pavement object detection using superpixel segmentation combined with conditional random field. IEEE Trans. Intell. Trans. Syst. 2018, 19, 2076–2085.
9. Zhang, C.; Xie, Y.; Liu, D.; Wang, L. Fast threshold image segmentation based on 2D fuzzy fisher and random local optimized QPSO. IEEE Trans. Image Process. 2017, 26, 1355–1362.
10. Druzhkov, P.N.; Kustikova, V.D. A survey of deep learning methods and software tools for image classification and object detection. Pattern Recognit. Image Anal. 2016, 26, 9–15.
11. Janakiramaiah, B., Kalyani, G. Jayalakshmi, A. Automatic alert generation in a surveillance systems for smart city environment using deep learning algorithm. Evol. Intel. (2020).
12. Xu, X.; Li, Y.; Wu, G.; Luo, J. Multi-modal deep feature learning for RGB-D object detection. Pattern Recognit. 2017, 72, 300–313.
13. Schölkopf, B.; Platt, J.; Hofmann, T. Graph-based visual saliency. In Proceedings of the 19th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 4–7 December 2006; MIT Press: Cambridge, MA, USA, 2006; pp. 545–552.
14. Achanta, R.; Hemami, S.; Estrada, F.; Susstrunk, S. Frequency-tuned salient region detection. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–26 June 2009; pp. 1597–1604.
15. Cheng, M.M.; Mitra, N.J.; Huang, X.; Torr, P.H.S.; Hu, S.M. Global contrast based salient region detection. IEEE Trans. Pattern Anal. Mach. Intell. 2015, 37, 569–582.
16. Li, X.; Li, Y.; Shen, C.; Dick, A.; Hengel, A.V.D. Contextual hypergraph modeling for salient object detection. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 3328–3335.
17. Hou, Q.; Cheng, M.M.; Hu, X.; Borji, A.; Tu, Z.; Torr, P.H.S. Deeply supervised salient object detection with short connections. IEEE Trans. Pattern Anal. Mach. Intell. 2018.
18. Wang, L.; Wang, L.; Lu, H.; Zhang, P.; Ruan, X. Saliency detection with recurrent fully convolutional networks. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; Volume 9908.
19. Gao Y, Ma J, Yuille AL (2017) Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples. IEEE Trans Image Process 26(5):2545–2560
20. Garcıa-Laencina PJ, Sancho-Gomez JL, Figueiras-Vidal AR (2010) Pattern classification with missing data: a review. Neural Comput Appl 19(2):263–282
21. Guo X, Li Y, Ling H (2017) Lime: low-light image enhancement via illumination map estimation. IEEE Trans Image Process 26(2):982–993
22. Ma J, Jiang J, Liu C, Li Y (2017) Feature guided gaussian mixture model with semi-supervised em and local geometric constraint for retinal image registration. Inf Sci 417:128–14222.
23. Ma J, Ma Y, Li C (2019) Infrared and visible image fusion methods and applications: a survey. Inf Fusion 45:153–178

24. Semwal VB, Mondal K, Nandi GC (2017) Robust and accurate feature selection for humanoid push recovery and classification: deep learning approach. Neural Comput Appl 28(3):565–574
25. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
26. Kasthuriarachchy BH, Zoysa KD, Premaratne HL (2015) Enhanced bag-of-words model for phrase-level sentiment analysis. In: International conference on advances in ICT for emerging regions, pp 210–214
27. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
28. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: International conference on neural information processing systems, pp 1097–1105"
29. He K, Sun, J (2014) Convolutional neural networks at constrained time cost. In: Computer vision and pattern recognition, pp 5353–5360
30. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
31. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: IEEE conference on computer vision and pattern recognition, pp 1–9
32. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Computer vision and pattern recognition, pp 770–778
33. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inceptionv4, inception-resnet and the impact of residual connections on learning. In: AAAI, pp 4278–4284
34. Palm, Hans Christian, Halvor Ajer, and Trym Vegard Haavardsholm. Detection of military objects in LADAR images.2008.
35. Xiaodong Hu, Peng Zhang, Yi Xiao, Military Object Detection Based on Optimal Gabor Filtering and Deep Feature Pyramid Network, AICS 2019: Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science July 2019 Pages 524–530 https://doi.org/10.1145/3349341.3349462
36. Kamate, Shreyamsh, and Nuri Yilmazer., Application of object detection and tracking techniques for unmanned aerial vehicles., Procedia Computer Science 61 (2015): 436-441.
37. Liu, Shuo, and Zheng Liu.,Multi-channel CNN-based object detection for enhanced situation awareness.arXiv preprint arXiv:1712.00075 (2017).
38. Arya Raj A.K, 2 Radhakrishnan B, A Comparative Study on Target Detection in Military Field Using Various Digital Image Processing Techniques, International Journal of Computer Science and Network, Volume 5, Issue 1, February 2016
39. Yang, Zhi, Wei Yu, Pengwei Liang, Hanqi Guo, Likun Xia, Feng Zhang, Yong Ma, and Jiayi Ma. Deep transfer learning for military object recognition under small training set condition. Neural Computing and Applications 31, no. 10 (2019): 6469-6478.
40. Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. Advances in neural information processing systems. 2017.

Figures

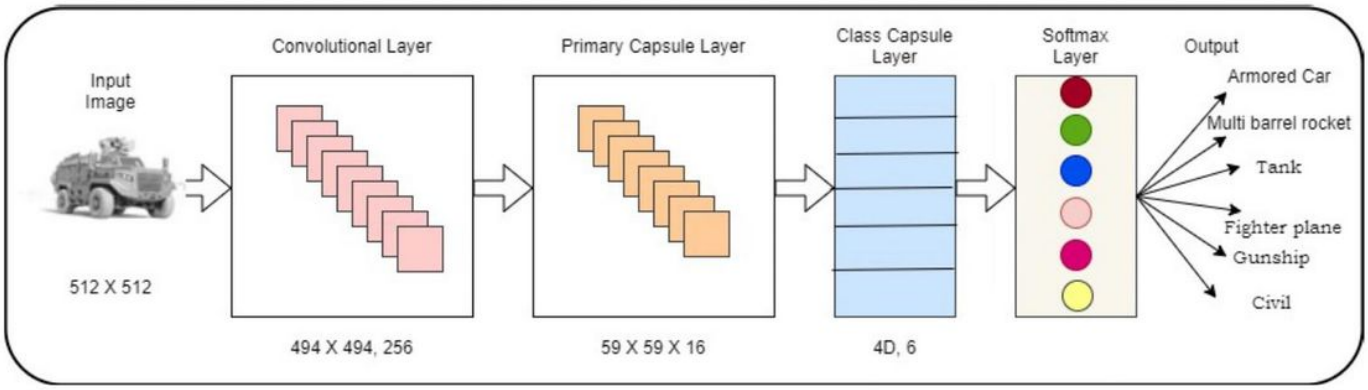

Figure 1

Multi-level CapsNet Architecture for Military Object Detection

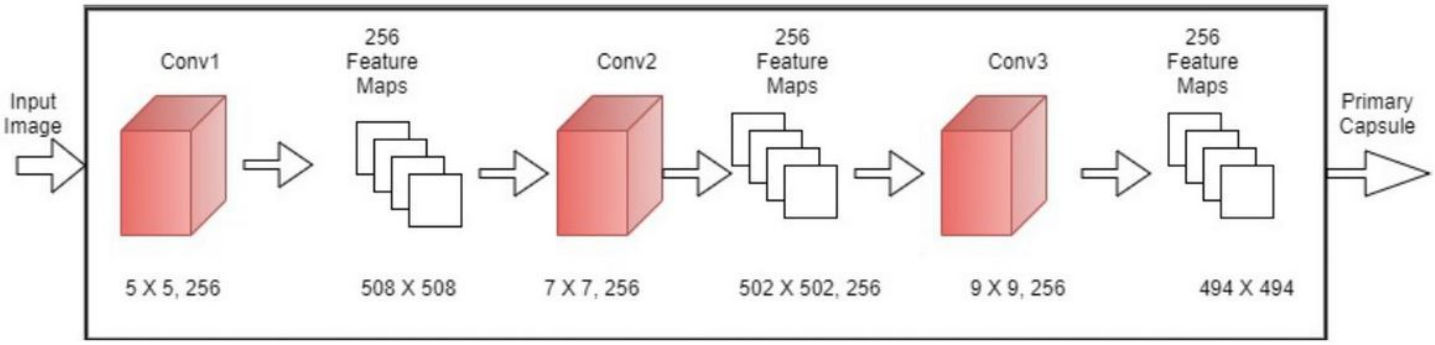

Figure 2

Convolution Layer of the Multi-Level CapsNet Architecture

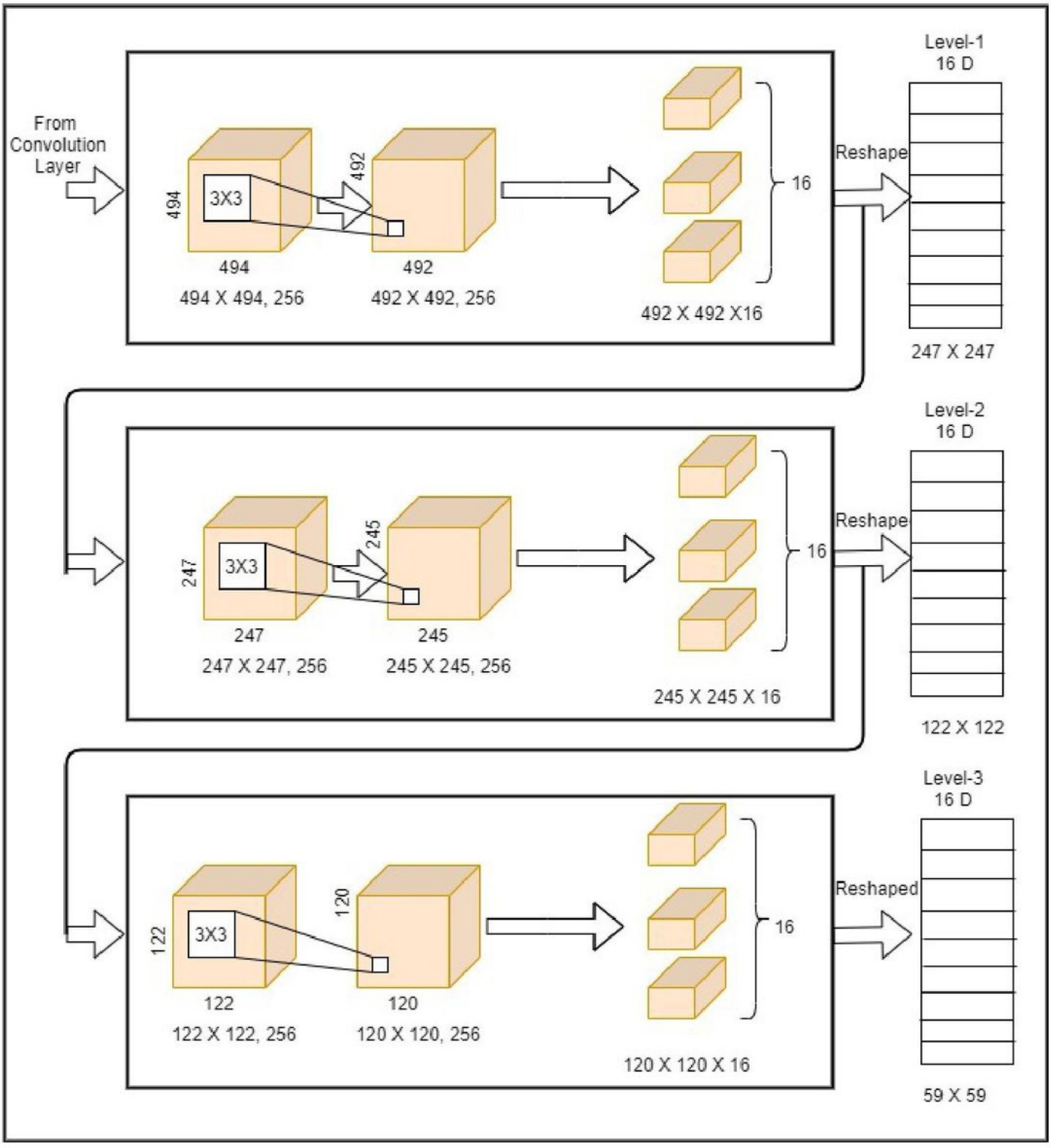

Figure 3

Primary Capsule Layer of the Mult-level CapsNet Architecture

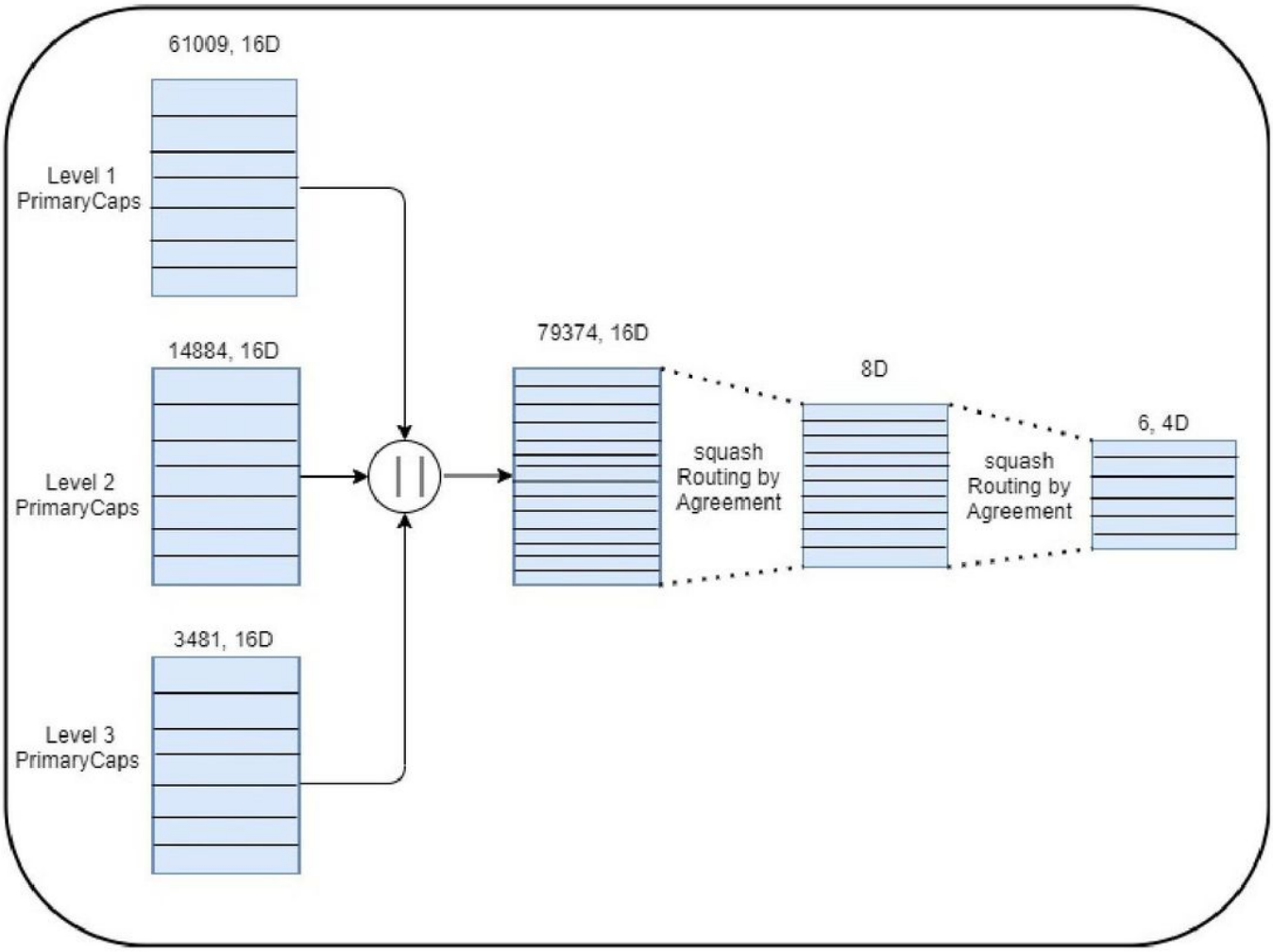

Figure 4

Class Capsule Layer of the Multi-level CapsNet Architecture

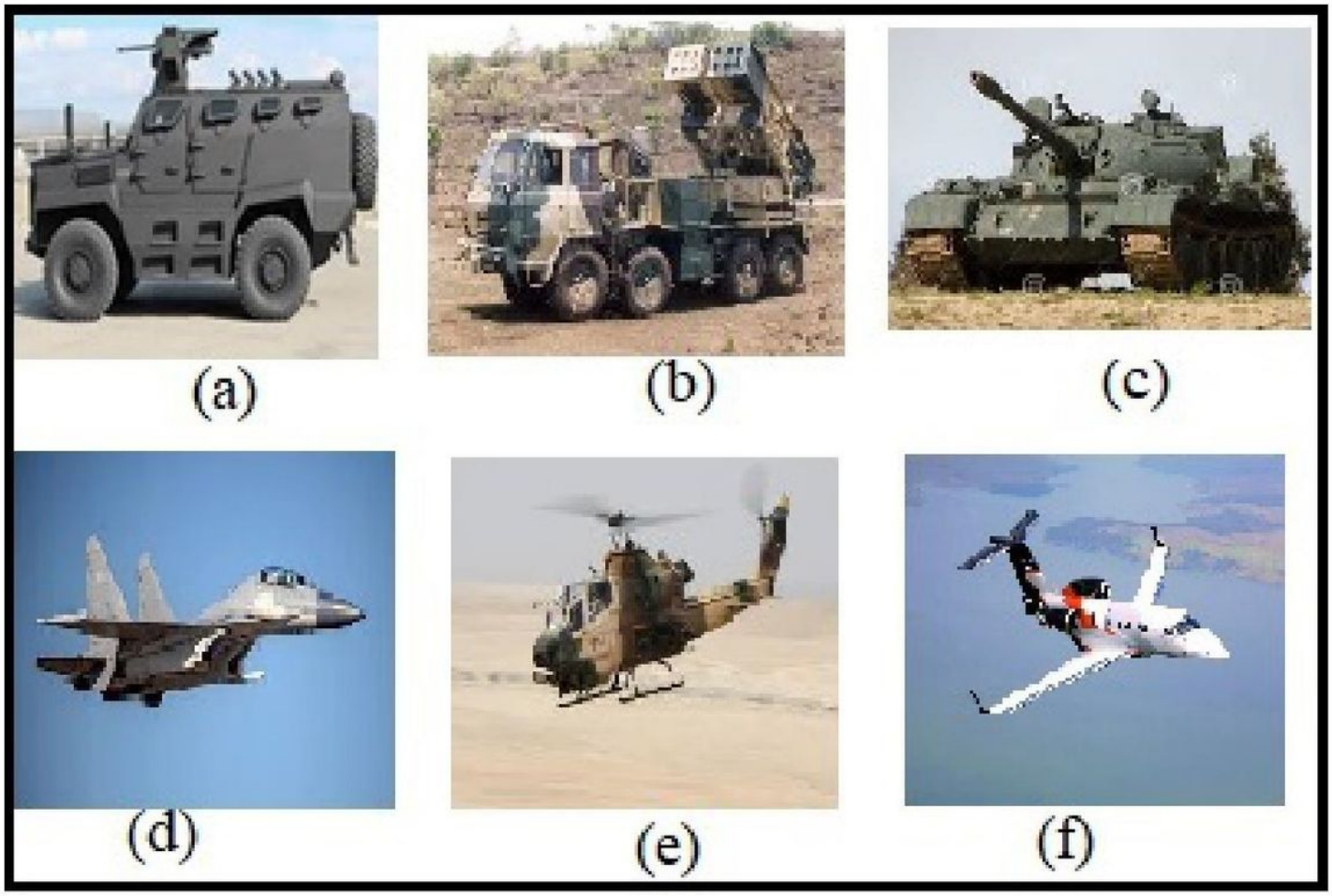

Figure 5

Sample images in the Collected Dataset.(a) Armored car (b) Multi-barrel Rocket (c) Tank (d) Fighter Plane (e) Gunship (f) general object

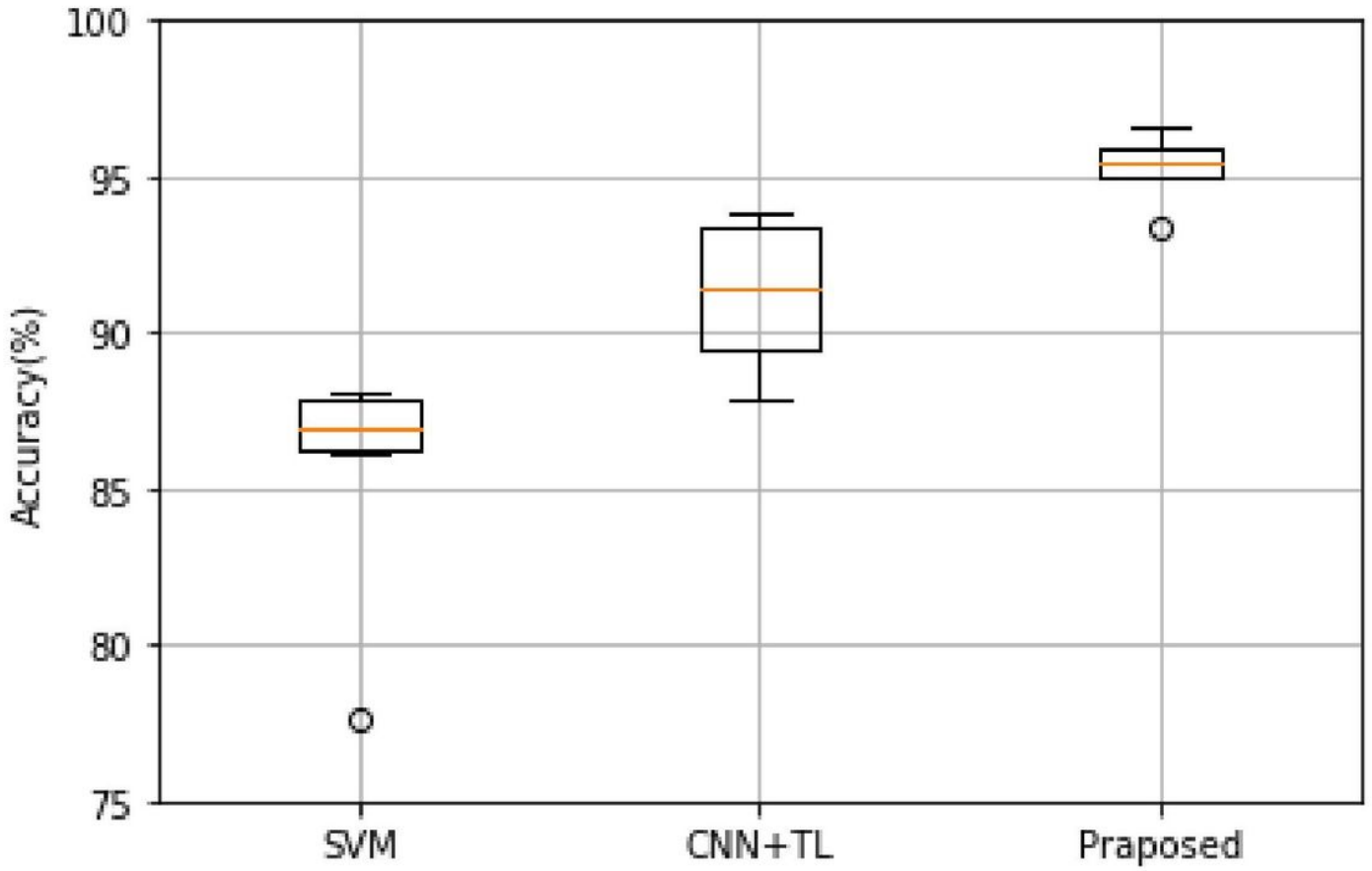

Figure 6

Comparison of Accuracy with Box Plot Analysis.

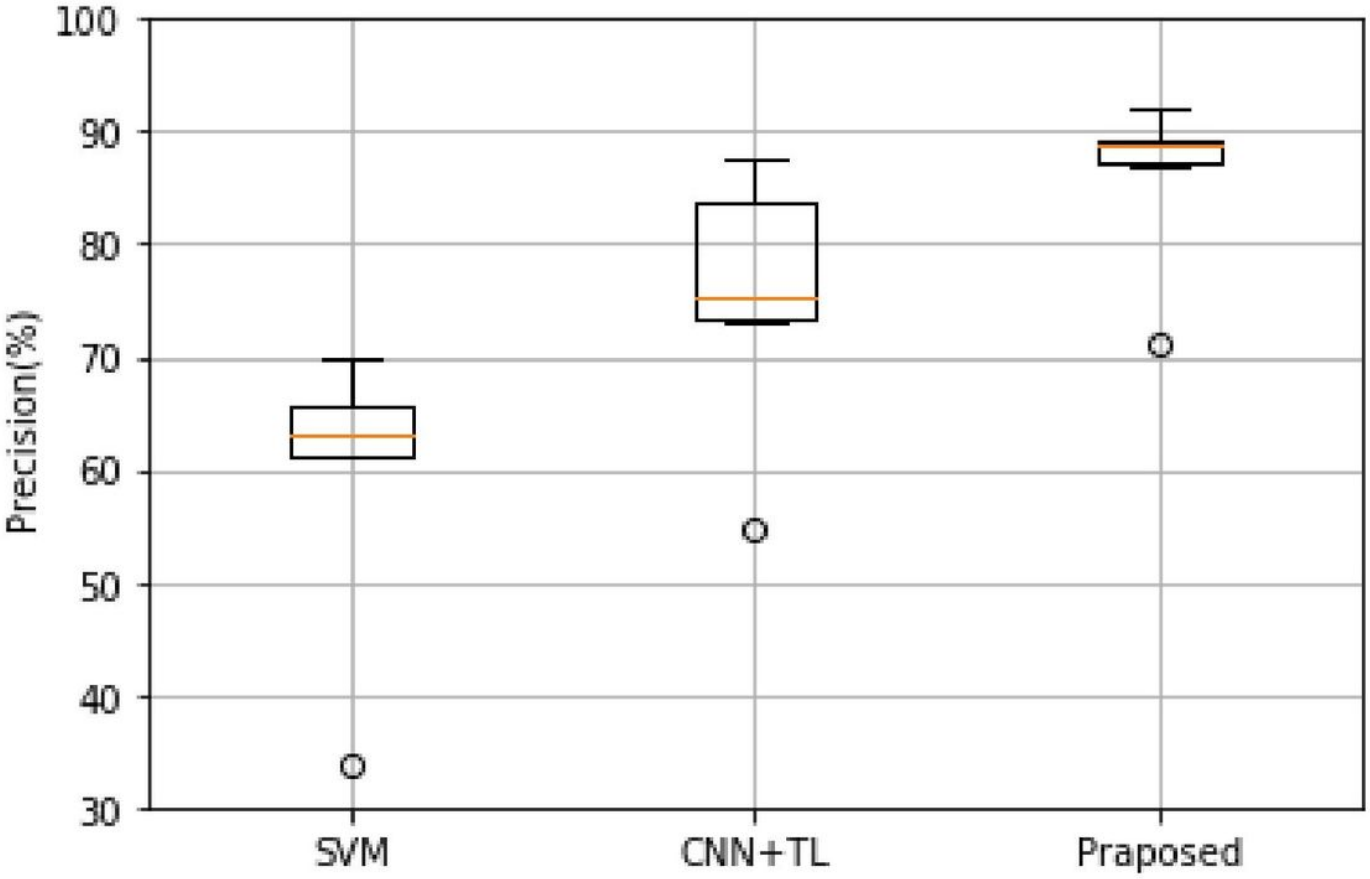

Figure 7

Comparison of Precision with Box Plot Analysis

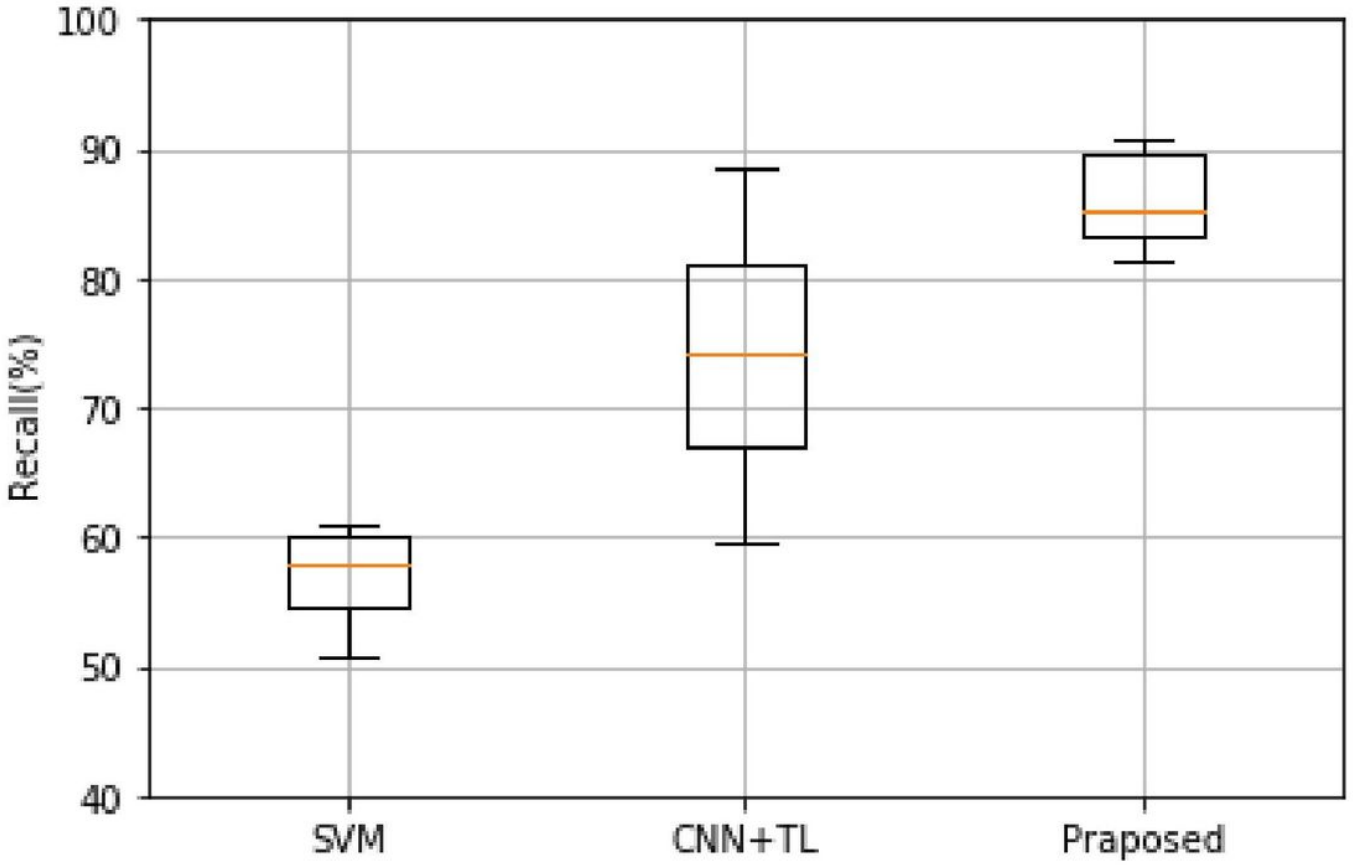

Figure 8

Comparison of Recall with Box Plot Analysis.

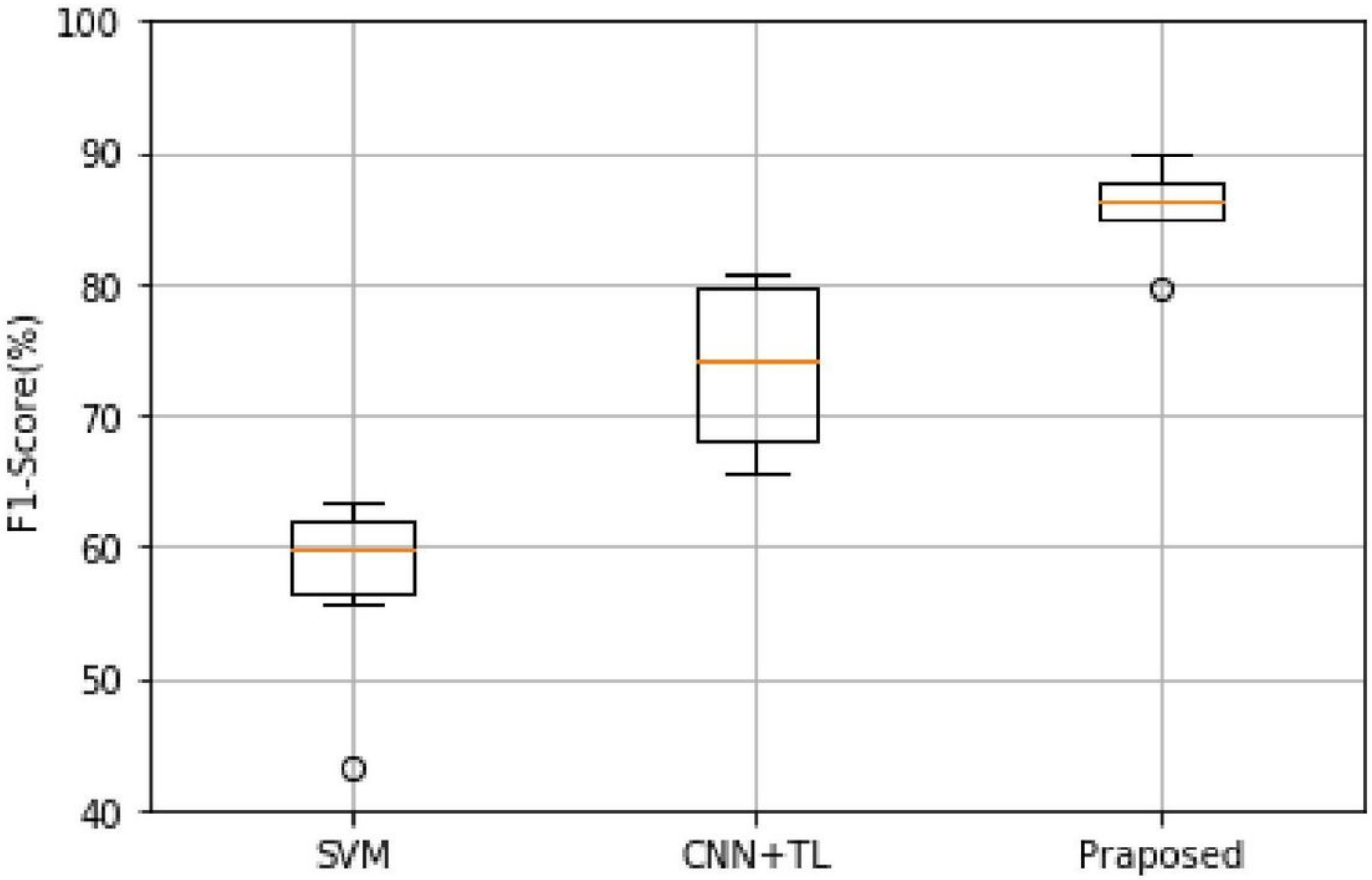

Figure 9

Comparison of F1-score with Box plot Analysis