Doctoral Thesis

Feature Extraction for Single Shot Multibox Object Detector

単画像多矩形物体検出のための特徴抽出

November 25, 2022

TUERSUNJIANG YIMAMU

Department: Manufacturing Engineering

Kitami Institute of Technology

Feature Extraction for Single Shot Multibox Object Detector

TUERSUNJIANG YIMAMU, Graduate School of Engineering, Manufacturing Engendering Major, Kitami Institute of Technology, Kitami, Japan.
E-mail: tursunjanemam@gmail.com

Abstract:

Recently, object detection based on deep convolutional neural networks (CNNs) have achieved remarkable result and successfully applied many real-world applications. However, scale variation problem in multiscale object detection still is challenging problem, especially for small objects. Concerning the above problem, we proposed a new detection network with an efficient feature fusion module based on SSD using VGG-16 as backbone called Multi-path Feature Fusion Single Shot Multi-box Detector (MF-SSD). The proposed feature fusion module consists of two newly designed modules with dilated convolution, which fuses features from shallow layers (mainly contain boundary information) to higher level features (mainly contain semantic reach information) without reducing the original resolution of the feature map. We have conducted experiments on three datasets to explicate the efficacy of our proposed detector. The proposed MF-SSD with input size 512×512 achieved 81.5% mAP and 34.1 % mAP on PASCAL VOC test set and MS COCO test-dev, respectively. Experimental results show the proposed feature fusion module can improve both semantic and boundary information for object detection.

Table of Contents

1. Introduction

Object detection can be applied by wide applications due to its having momentous research parts in computer vison. Classifying a particular object category and the need to find out location information by using bounding boxes are the essence of object detection. In recent years, the introduction of deep convolutional neural networks (CNNs) is one of main turning points in object detector due to its dominant importance of extracting features. We can categorize CNN based object detector into two types of frameworks including the one stage detector, e.g., Faster R-CNN [1], R-FCN [2] and SPP-Net [3], and the two-stage object detector, e.g., YOLO [4], SSD [5] and RetinaNet [6] etc.

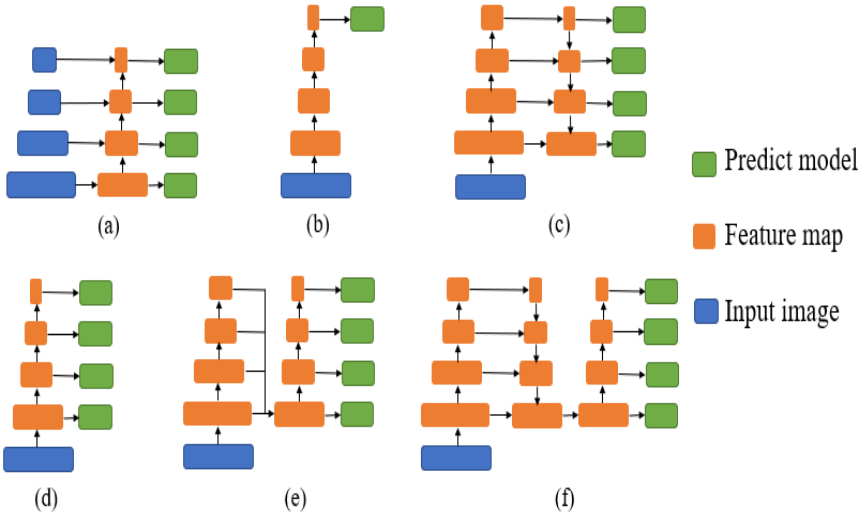


Fig. 1. *(a) Different scale features are generated using an image pyramid, which is time consuming. (b) only single scale feature is generated to detect objects, which leads lower accuracy and is adopted Faster R-CNN [1] and R-FCN [2]. (c) Different scale features are generated by a feature fusion method to improve feature map from top layer to down layer. (d) different scale features are generated using a feature pyramid, which is adopted by SSD (e) different scale features from different layers are fused by concatenation approach, which is adopted by FSSD [15]. (f) Different scale features from different layers are generated to further improve contextual information for object detection, which achieves high performance but relatively lower speed.*

However, scale variation in object detection is still a challenging problem for both methods. We can clearly see from Fig. 1 that various approaches have been introduced and carried out by researchers for handling the scale variation problem. CNN based object detector such as SSD, Faster RCNN extracts useful information from the input images by using various backbone networks such as VGG [7], ResNet [8]. The backbones, which are used in detector and pretrained by ImageNet [9], are the main body of detector for extracting features. The design of novel classifiers for getting higher score adopts ImageNet on which most of the classifiers are trained. VGG connects multiple layers which consist of convolution layers and max pooling layers with different kernel size to build a deeper network without residual connection. GoogLeNet [10], a deeper and wider neural network, consists of inception modules which enhance feature extraction at different scales using convolution layers with different kernel size. ResNet architecture consists of “bottleneck” design which is using skip connections to jump over some layers

with residual sum operation in each stage. DenseNet [11] densely connects several layers, through dense blocks, where we connect all layers with each other.

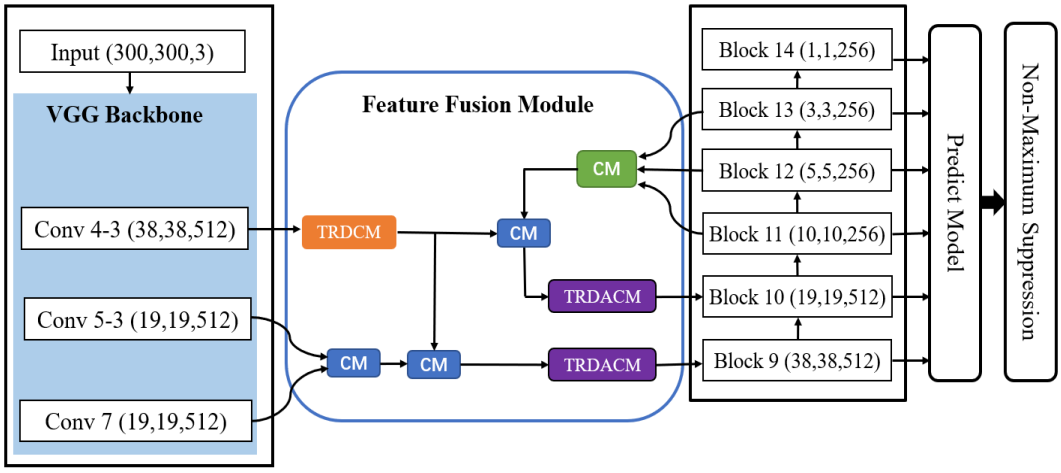


Fig. 2. *The architecture of Multi-path Feature Fusion Single Shot Multi-box Detector (MF-SSD).*

To mitigate scale variation problem, feature fusion strategies have been proposed such as Image pyramids [12], FPN [13], DSSD [14] and FSSD [15]. As shown in Fig. 1. (a), Image pyramids generate different scale feature maps using CNN with different scale images, which is computationally expensive. Anchors with various size are generated by Fig. 1. (b) like structures by using single feature map. This kind of method has a limitation to detect various size of objects due to the fixed receptive fields. Fig. 1. (c) like structure has been applied by FPN, DSSD, SharpMask [16] methods which fuse different scale features using element wise summation, while DSSD obtains features from top layers to fuse features from down layers by using Deconvolutions. As shown in Fig. 1. (d), feature pyramid is adopted by SSD to detect objects with different scales. In SSD, the extracted features of small objects usually are obtained from Conv 4-3 layer and the extracted features of large objects usually are obtained from Conv 8 layer. Fig. 1. (e) like structure has been adopted by FSSD [15] which fuses features through concatenation approach which gives us a chance to fuse more layers form last part of the detector without losing any information.

Shallower layers usually contain boundary information including angles, lines and curves. The lack of semantic rich information in shallower layers causes lower accuracy on small objects for a deeper network. Extracting semantic features in deep layer in which there are sufficient information with larger receptive fields and lower resolutions provides a good condition for object classification. Extracting features in shallow layers in which there are spatial-rich information with small receptive fields provides a good condition for object localization or vector regression. It is vital that high resolution representations are made available to small object detection. There are poor high-resolution features available in the deeper layer, which made model hard to detect small objects.

We can expand kernel size with original weights using a dilated convolution [17] which samples sparsely at different locations and increases receptive field with same computational cost. Dilated convolution avoids the negative effect of down sampling operation, and it is possible to have features with high resolution in deep layers by using dilated convolution. However, large object detection without enough receptive fields is difficult. DetNet [18] uses dilated convolution to design a specific detection backbone.

1.1. Contribution

In this research, to mitigate problems pointed out above, we propose Multi-path Feature Fusion Single Shot Multi-Box Detector (MF-SSD) by adding an efficient feature fusion module which embed two newly designed modules to extract sufficient features. We provided the architecture of MF-SSD in Fig.2. We employ dilated convolution with different dilation ratios to design Two-branch Residual dilated Convolution Module (TRDCM) and Two-branch Residual dilated Add Convolution Module (TRDACM), which enlarges receptive field without extra computational cost. We have conducted numerous experiments on MS COCO [19]， PASCAL VOC2007, and PASCAL VOC2012 [20] datasets to explicate the efficacy of our proposed detector. In the proposed feature fusion module, we not only extract features with sufficient boundary information, which is beneficial for vector regression, but also extract features with contextual information which is beneficial for object classification. Using the proposed feature fusion module, we improve a lot of performance compared with original SSD, especially for small objects. In summary, we highlighted our main contributions as follows:

1. A new object detection framework, MF-SSD, is proposed to handle multiscale problem, especially for small objects.
2. We newly designed an efficient feature fusion module to extract sufficient information with various types to improve better performance for object detection compared with the conventional SSD. The proposed feature fusion module contains two newly designed modules including the TRDCM and TRDACM. Through these two modules, we improved contextual information with a sufficient feature.
3. We have explicated the efficacy of our proposed MF-SSD through numerous experiments.

2. Related work

2.1. Traditional object detection

The whole processing of traditional object detection consists of more steps compared to other frameworks. The first step is generating candidate regions with input images which can be slid by sliding-windows with certain step size, which is called a region selector. The next step is extracting useful features of the candidate regions. The feature extractor mainly uses SIFT [21], HOG [22], etc. Whether we extracted sufficient information in the above step will affect the performance of the classification; the final stage is to identify object category. Before CNN based object detection algorithm, DPM [23] was widely studied method in this field.

2.2. Object detection based on deep neural network

2.2.1 Two-state object detection: The two-stage framework is accurate and also relatively complicated compared to one-stage framework. There are many methods including Selective Search [24], Edge Boxes [25], and RPN [26] to produce region proposals in the first step. In the second step, the SVM or CNN based algorithm refines and classifies the proposals generated in the previous stage. Features in the region of interest (RoI) can be extracted by using the selective search, which is adopted R-CNN [27]. Different objects which are classified by SVM indicated under a RoI by bonding boxes. SPPnet [3] demonstrated that feature maps can be generated efficiently on a single image scale by using region-based detectors. Faster R-CNN [1] adopted more efficient network, a region proposal network, to create region proposals

based on Fast R-CNN [28]. In real world case, the detection speed is still not very fast, because the RoI pooling layer cannot share parameters inside every convolution layer in the Faster R-CNN. Compared to previous region-based detectors, e.g., Fast/Faster R-CNN [1], R-FCN [29] employs a fully convolutional network to shorten the time spending on training and testing , for it can share parameters through convolution in the RoI. R-FCN accelerated the detection speed compared with Faster R-CNN. [29] used hand-engineered features with Faster R-CNN to improve model perfection.

2.2.2 One-state object detection: The one-stage detector, a faster and clearer object detector but weaker in accuracy, predicts object classes and bounding boxes without generating a region proposal. Overfeat [30] which is considered earliest object detector has integrated classification and localization all into one convolutional network. YOLO [4], a real time framework based on CNN, is introduced by R. Joseph et al. YOLO is faster and relatively lower accurate framework, for it uses a single network in whole process without proposing bonding boxes. Later on, researchers improved YOLO and released YOLO v2 [31], YOLO v3[32], YOLO v4 [33] and YOLO v5 [34] which improves detector on time spending and precision. Predicting objects with anchor boxes is implemented by the improved version of YOLO without using a fully connected layer which is changed by convolutional layers. However, increasing detection speed is not sufficient for YOLO, because it still hard to detect too small and crowded objects in real world case. SSD [5] is proposed to mitigate the limitation of YOLO. Compared to earliest frameworks, the main advantages of SSD is that it adopted Fig. 1. (c) like structure, a multi-resolution technique, which fuses different scale features to improve detection performance. In conventional SSD, extracting information related to small objects depend on only one layer, e.g., Conv4-3 layer. However, the feature extraction in SSD is not enough for detecting small objects, thus we still need to improve for detecting crowded and small objects. RetinaNet [6] which is also one of the best frameworks uses focal loss, a new type of cross entropy loss, to addresses class imbalance problem. In recent years, researchers have also introduced anchor free object detectors, which find out the key points without using anchor boxes in an image. Anchor free framework is proposed by Law and Deng called Cornernet [35].

2.3. Feature fusion

As mentioned in the before, the extracted features in shallow layer mainly contain features with higher resolution, while the extracted features in deep layer contain semantic rich information and larger receptive fields. Feature pyramid representation is introduced and widely used for solving scale variation problem. Researchers proposed Skip connection [36] and hyper-column [37] methods to combine different scale features to get more useful features maps. Object detection include both classification and vector regression, but most of the detectors uses backbones which designed for image classification task. To mitigate the dissimilarity between detector and image classification, many methods [5] with the lightweight property is proposed to extract more representative features to get higher accuracy. PANet [38] introduces an extra bottom-up pathway to accelerate training and reduces the extent between different level features. A similar bottom-up pathway structure is introduced by BiFPN [39], which uses scale-wise level reweighting to fuse different level features efficiently. In recent years, [40-41] adopts the Neural Architecture Search (NAS) algorithm to extract useful features in more efficient way.

2.4. Low rank matrix recovery

Recovering low dimensional features from given data is used for many research areas including machine learning and data analysing. The Low Rank Matrix Recovery (LRMR) is extended to various

fields and its applicability has been tested by researchers. Output sensor data which is one of the main fields using LRMR technique to get more useful features is crucial task in the signal processing [49] [50]. Hankel matrix [51] [52] can also be represented by low dimensional features to simplify our tasks in real word application. a low dimensional features which has significant rule in graphical theory [53] [54] is used to constitute adjacency matrix. Further, low dimensional representations are predominant in computer vison and object detection. [55] [56] applied low dimensional features to get SVD. [57] [58] applied low dimensional features to get covariance matrix. [59] [60] [61] applied low dimensional features to obtain useful information. [62] [63] [64] [65] [66] further interpreted main advantage of low dimensional features in different research areas.

A low dimensional component $L \in \mathbb{R}^{m\times n}$ can be formulated as the following forms [67]:

$$\min_{L,E} \ rank(L)+\lambda\|E\|_p \ \ s.t. \ \ P_\Omega(Y) = P_\Omega(L+E), \tag{1}$$

where $\|\cdot\|_p$ is general term. The p represents any number, such as $\|\cdot\|_F^2$ is applied for extracting noise [68], l_0 is applied for handing outlier [56]. $Y \in \mathbb{R}^{m\times n}$ which contains error matrix $E \in \mathbb{R}^{m\times n}$ is original dat. We can control noisy levels by using non-negative $\lambda > 0$, which is typically set to 0.001.

We cannot directly solve equation 1 due to its np-hardness [69]. There are two kind of methods used for solving equation 1. One of methods is Matrix Factorization (MF) [70] [71]. This kind of method can be applied when ground truth rank is available. The main principle of this method is that any observation matrix can be represented by inner product of submatrices. If our submatrices consist of two matrices, one matrix represents original data and others is rearrests noisy data or error. Further MF method [72] [73] has some advantages when the GT rank which is not available in the real world application is available.

Data usually is corrupted by various reason, and we do not know intrinsic rank *r* without interpreting the data. If data did not contain any noise, we can get low rank component from the following problem:

$$\min_{U,V} \|P_\Omega(Y-UV)\|_F^2, \tag{2}$$

where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{r\times n}$.

Papers [74] [75] [76] studied Nuclear-Norm Minimization (NNM), which is required to calculate singular values of the matrix. MF method is not best option when GT rank is not available. In this situation, we can use NNM to solve our problem to get a low dimensional component of the original data. We can formulate same problem via NNM method as follows:

$$\min_{L} \lambda \ \|L\|_* + \frac{1}{2}\|P_\Omega(Y-L)\|_F^2. \tag{3}$$

where λ is positive parameter which often set to smaller number.

Unfortunately, most of the papers only considered that the low rank component consists of small noise. Papers [68] [77] [78] were considered original data with some noisy levels to solve low rank components. As mentioned, both NNM and MF methods have own advantages and disadvantages. NNM mainly is used for unknown rank problem and MF mainly is used for rank available problem. In large scale problem, the applicability of NNF is largely reduced by the necessary to calculate singular values in every process. Other mixing methods [79] [80] [81] combined various methods to increase efficiency of LRMR. Fusing different layers increases applicability of the algorithm. We can combine extra layers to increase useful information in object detection task. Non-negativity allows us to combine different

models [82] [83].

To increase low dimensional features and relative useful information, we adopted Tikhonov regularization and l_1 norm. Further, we combined two methods to increase applicability and efficiency of our algorithm, which handles both outlier and other noises. We proposed efficient solver for LRMR which is not required to use SVD for every loop. In most of the situation, we do not know GT rank of the data which can affect final performance of the model. We also solved this problem in efficient manner using Theorem 1. Unlike existing mixture algorithms, we not only increased applicability but also reduced time demanding in various tasks to get better low dimensional component.

3. Methodology

The whole architecture of MF-SSD in Fig. 2 consists of three parts including backbone which adopts pre-trained VGG-16, feature fusion module (FFM) and detection head. We first introduce the FFM which includes Connection Module (CM) which are blue and green rectangular boxes in Fig. 2, Two-branch Residual Dilated Convolution Module (TRDCM) which is orange rectangular boxes in Fig. 2 and Two-branch Residual Dilated Add Convolution Module (TRDACM) which is purple rectangular box in Fig. 2.

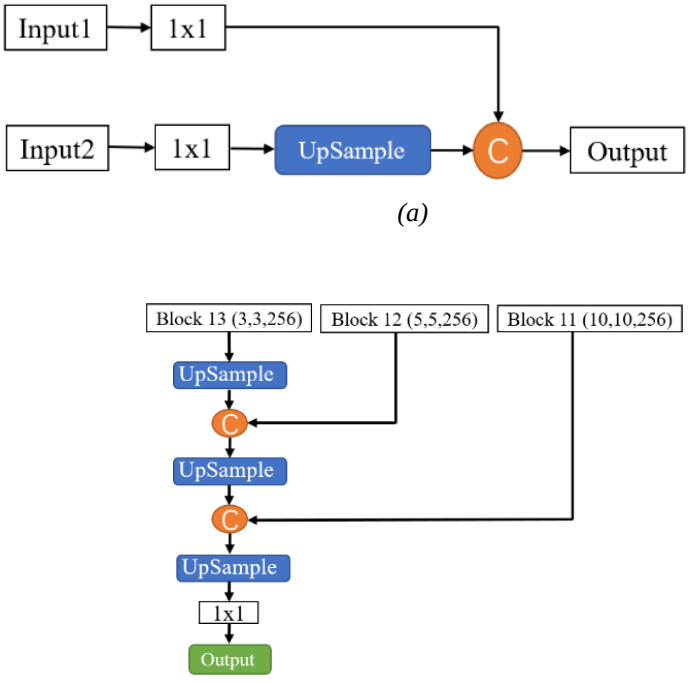
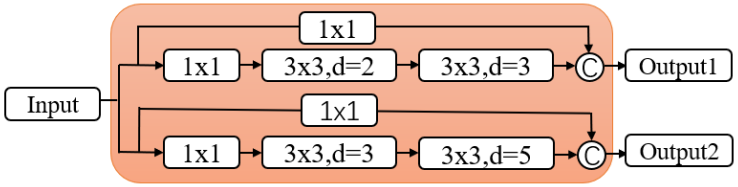


Fig. 3. *Detailed structure of connection modules: (a) Concats high level feature map with shallow layer feature map using concatenation approach with bilinear interpolation; (b) Concats feature maps without adjusting channel number using concatenation approach with bilinear interpolation.*

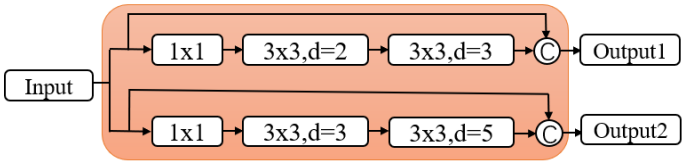
3.1. Connection Module (CM)

There are two types of Connection Module (CM) which are shown in Fig. 3. We first introduce the CM module in Fig. 3. (a) or orange rectangular box in Fig. 2. Suppose input 1 (N, C_1, H_1, W_1) is the

lower-level feature map and input $2(N, C_2, H_2, W_2)$ represents higher level feature map; where N means batch size, C means channel number, H and W represents feature map size. First of all, we use convolution layer with kernel size 1x1 to adjust two input features into same channel number. Typically, the higher-level feature contains lower resolution compared to lower-level feature. We use up-sampling operation with bilinear interpolation to adjust higher-level feature map size (H_2, W_2). Noting that up-sampling operation did not change channel number. Deconvolution also uses for similar purpose besides up-sampling operation. After conducting numerous experiments to test negative effects of both up sampling and deconvolution, we decided to use up-sampling operation in the proposed MF-SSD. After up sampling operation we obtain same feature map size $(H_1 = H_2, W_1 = W_2)$. Finally, we fused two inputs using concatenation attention approach [37], and obtained final output $(N$, C, H, W). After conducting several experiments to investigate various attention approaches including concatenation, element-wise summation [13] and element-wise product [14], concatenation approach is selected as our attention approach. We splice feature maps using concatenation approach into channel dimension without increasing feature map size. As shown in Fig. 3. (b) (It is green rectangular box in Fig. 2), the CM is different from previous CM in Fig. 3. (a). The dissimilarity is that we did not use the convolution layer with kernel size 1x1 before up sampling operation with bilinear interpolation, because three features we concatenated are already same number of channels. We only use convolution layer with kernel size 1x1 before the final output.



(a)



(b)

Fig. 4. *Detailed structure of two-branch residual dilated convolution module (TRDCM): (a) Residual connection without convolution operation; (b) Residual connection with convolution operation with kernel size 1x1.*

3.2. Two-branch Residual Dilated Convolution Module (TRDCM)

As mentioned in the before, the main reason of the lack of sufficient features for detecting small objects is that the high-resolution features are reduced or unobtainable in the deeper layers. In conventional SSD, the extracted features in the shallower layers contain a small amount of semantic information, which is the main problem causes insufficient features for small object detection. Therefore, we introduce Two-branch Residual Dilated Convolution Module (TRDCM), where we can expand kernel size with original weights using a dilated convolution [17] which samples sparsely at different locations and increases receptive field with same computational cost. Unlike previous feature fusion methods, our

method fuses feature from deeper layers to shallower layers to increase semantic information, and we can still use the high-resolution features in the deeper layers. Insufficient information for small objects in traditional SSD is only obtained from Conv4-3 layer. To enhance insufficient features, we added TRDCM which receives features from Conv4-3 layer. As shown in Fig. 4, the TRDCM consists of two branches with residual connection. We introduced two versions for TRDCM. The first one in Fig. 4. (a) consists of residual connection without convolution, and the second one in Fig. 4. (b) consists of residual connection with 1x1 convolution layer. In the TRDCM, we use 1x1, 3x3 (dilated ratio is 2) and 3x3 (dilated ratio is 3) convolutions with residual connection for first branch, and 1x1, 3x3 (dilated ratio is 3) and 3x3 (dilated ratio is 5) convolutions with residual connection for second branch. We have conducted several experiments to verify the dilation ratio setting is optimal. More results about two versions of the TRDCM are listed in the ablation study section.

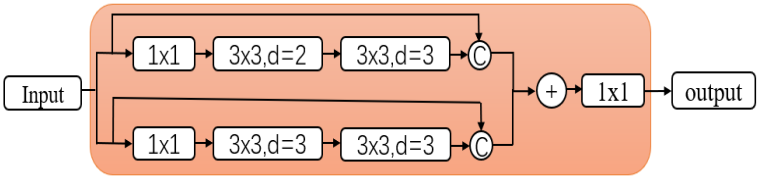


Fig. 5. *Detailed structure of two-branch residual dilated add convolution module (TRDACM).*

3.3. Two-branch Residual Dilated Add Convolution Module (TRDACM)

As shown in Fig. 5, we also introduced Two-branch Residual Dilated Add Convolution Module (TRDACM) which are purple rectangular boxes in Fig. 2 to enhance contextual information. In the TRDACM, we also used two branches with dilated convolution and residual connection to combine of them, which not only increases receptive field, but also reduce gradient vanishing problem. Firstly, two branch go through 1x1 convolution to adjust the channel number, and then two of branches go through two 3x3 dilated convolution with different dilated ratios to capture contextual information. After conducting several experiments, we choose dilatated ratio (d=2 and d=3) for first branch, and dilated ratio (d=3 and d=3) for second branch. Both branches use residual connection without convolution for better optimization. Finally, we use concatenation attention approach to combine two branches. After concatenation, final output goes through 1x1 convolution to integrate contextual information. More results about the TRDACM module are highlighted in ablation study section.

3.4. Low rank matrix recovering method

3.4.1 Problem formulation: As mentioned in the previous section, NNM has the need to calculate SVD in the whole process. We cannot apply directly this method to big data. The theorem 1 gives us a chance to solve large scale problem.

Theorem 1: For any matrix $L \in \mathbb{R}^{m \times n}$, the following relationship holds [68]:

$$\|L\|_* = \min_{U,V} \frac{1}{2}\|U\|_F^2 + \frac{1}{2}\|V\|_F^2 \qquad s.t. \qquad L = UV.$$

After applying theorem 1 to combine NNM and MF, we obtained following Tikhonov regularisation [84] problem which increases useful features for recovering low dimensional component form noisy data.

$$\min_{U,V} \ \frac{\lambda}{2}\|U\|_F^2 + \frac{\lambda}{2}\|V\|_F^2 + \frac{1}{2}\|P_\Omega(Y - UV)\|_F^2. \tag{4}$$

After generalizing equation 4, we obtained generalised Tikhonov regularization which extends applicability for recovering low dimensional component problem from noisy data [85].

$$\min_{U,V} \ \frac{\lambda}{2}\|\Gamma_U U\|_F^2 + \frac{\lambda}{2}\|V\Gamma_V\|_F^2 + \frac{1}{2}\|P_\Omega(Y - UV)\|_F^2$$
$$s.t. \quad {z_U}^T \Gamma_U z_U > 0 \ \ and \ \ z_V{}^T \Gamma_V z_V > 0, \forall \ z_U \in R^m \backslash \{0\} \ \ and \ \ z_V \in R^n \backslash \{0\}, \tag{5}$$

where Γ_U and Γ_V represents positive definite matrices; the equation 5 has some advantages compared to previous problem formulation for ill-conditioned problem.

In real world case, we can combine different terms to increase efficiency of getting low dimensional features. To increase sparsity levels of original data, L_1-norm is one of the main regularization strategy.

$$\min_{U,V} \ \frac{\lambda}{2}\|\Gamma_U U\|_F^2 + \frac{\lambda}{2}\|V\Gamma_V\|_F^2 + \frac{1}{2}\|P_\Omega(Y - UV)\|_F^2 + \beta_U\|U\|_1 + \beta_V\|V\|_1$$
$$s.t. \quad {z_U}^T \Gamma_U z_U > 0 \ \ and \ \ z_V{}^T \Gamma_V z_V > 0, \ \forall \ z_U \in R^m \backslash \{0\} \ \ and \ \ z_V \in R^n \backslash \{0\}, \tag{6}$$

where $\beta_U \geq 0$ and $\beta_V \geq 0$ are used to better control of specific levels of sparsity.

The observation data not only corrupted by small noises but also corrupted various type of noises. The weight matrix $(W \ \ and \ \ \bar{W}) \in [0,1]^{m \times n}$ is used to better handle various noises. Besides weight matrix, we also applied maximum entropy terms, which is useful to know whether original data is corrupted by random noise or other types of noises. After applying both weight matrix and entropy which is defined by $-\sum_{i=1}^{k} p_i log p_i$ with $\sum_{i=1}^{k} p_i = 1$, we obtained following problem formulation:

$$\min_{U,V,W} \ \frac{\lambda}{2}\|\Gamma_U U\|_F^2 + \frac{\lambda}{2}\|V\Gamma_V\|_F^2 + \frac{1}{2}\|W \odot (Y - UV)\|_F^2$$
$$+\beta_U\|U\|_1 + \beta_V\|V\|_1 + \gamma\Sigma_{ij}(w_{ij} log w_{ij} + \bar{w}_{ij} log \bar{w}_{ij})$$
$$s.t. \quad W + \bar{W} = \mathbf{1}; \ \ W \ \ and \ \ \bar{W} \ \in [0,1]^{m \times n}; \quad {z_U}^T \Gamma_U z_U > 0 \ \ and \ \ z_V{}^T \Gamma_V z_V > 0;$$
$$\forall \ z_U \in R^m \backslash \{0\} \ \ and \ \ z_V \in R^n \backslash \{0\}, \tag{7}$$

where $\gamma \geq 0$ represents a nonnegative term and $\mathbf{1}$ represents an all-one matrix. The all-one matrix is same size with W. In this equation, we used W as a weight matrix to change P_Ω. Because it is difficult to calculate the binary valued matrix. Therefore, we can get final results by using derivatives.

3.4.2 *Solving the Problem*: We can solve equation 7 by using ALM [86] [87] which divides main problem into subproblems. Other methods such as ADMM [88] also is used to solve same problem. The ALM like approaches are mainly applicable for separable problem [89] [90]. Our final formulation is not separable, and we need to convert our problem into applicable form.

To solve our problem by using ADMM, we added constraint term $L = UV$ to our final equation 7 and obtained following problem formulation:

$$\min_{U,V,W,L} \ \frac{\lambda}{2}\|\Gamma_U U\|_F^2 + \frac{\lambda}{2}\|V\Gamma_V\|_F^2 + \frac{1}{2}\|W \odot (Y - L)\|_F^2 + \beta_U\|U\|_1$$
$$+\beta_V\|V\|_1 + \gamma\Sigma_{ij}(w_{ij} log w_{ij} + \bar{w}_{ij} log \bar{w}_{ij})$$
$$s.t. \quad L = UV, \ W + \bar{W} = \mathbf{1}; \ \ W \ \ and \ \ \bar{W} \ \in [0,1]^{m \times n}; \ z_U \Gamma_U z_U{}^T > 0 \ \ and \ \ z_V \Gamma_V z_V{}^T > 0;$$
$$\forall \ z_U \in R^m \backslash \{0\} \ \ and \ \ z_V \in R^n \backslash \{0\} \quad . \tag{8}$$

After applying Lagrangian multiplier to above equation, we obtained following forms:

$$\mathcal{L}_{\{ \ W+\bar{W}=1; W,\bar{W} \ \in [0,1]^{m \times n}\}}(U, V, L, W, M) :=$$

$$\frac{\lambda}{2}\|\Gamma_U U\|_F^2 + \frac{\lambda}{2}\|V\Gamma_V\|_F^2 + \frac{1}{2}\|W\odot(Y-L)\|_F^2 + \beta_U\|U\|_1 + \beta_V\|V\|_1$$

$$+\gamma\Sigma_{ij}(w_{ij}logw_{ij} + \bar{w}_{ij}log\bar{w}_{ij}) \quad + \frac{\alpha}{2}\|L-UV\|_F^2 + \langle M, L-UV\rangle, \tag{9}$$

where $\alpha > 0$ is a positive term, and M is matrix multiplier with same size of original data. We updated weight matrix and Tikhonov matrix by using same method which introduced later in this section. We take equation 8 into two sub-equations including: the problem related to weight matrix and the problem related to other parameters.

$$V^{(t+1)} \leftarrow \left(\lambda\Gamma_V^T\Gamma_V + \alpha^{(t)}U^{(t)T}U^{(t)}\right)^{-1}(\alpha U^{(t)T}L^{(t)} + U^{(t)T}M^{(t)} - \frac{\beta}{2})$$

$$U^{(t+1)} \leftarrow [(\alpha^{(t)}L^{(t)}V^{(t+1)T} - \frac{\beta}{2} + M^{(t)}V^{(t+1)T})\ \left(\lambda\Gamma_U\Gamma_U^T + \alpha^{(t)}V^{(t+1)}V^{(t+1)T}\right)^{-1}]$$

$$L^{(t+1)} \leftarrow \frac{W^{(t)}\odot Y + \alpha^{(t)}U^{(t+1)}V^{(t+1)} - M^{(t)}}{W^{(t)} + \alpha^{(t)}I}. \tag{10}$$

We take a derivative for equation 9 to get derivatives of $U, V,$ and L matrices. The final results are shown in equation 10. We updated L in elementwise method and solved U and V as a matrix. We updated weight matrix by using same process and obtained the solution of the weight matrix.

$$\min_{W,\bar{W}} \frac{1}{2}\left\|W\odot\left(Y - L^{(t+1)}\right)\right\|_F^2 \quad + \gamma\Sigma_{ij}(w_{ij}logw_{ij} + \bar{w}_{ij}log\bar{w}_{ij})$$
$$s.t. W + \bar{W} = \mathbf{1};\ \ W\ \ and\ \ \bar{W}\ \ \in [0,1]^{m\times n}\ . \tag{11}$$

We can solve above equation without any changes. Therefore, we get the following results:

$$Q\left(w_{ij}, \bar{w}_{ij}, \lambda_{ij}\right) \coloneqq w_{ij}[Y - L^{(t+1)}]_{ij}^2 + \ \gamma\left(w_{ij}logw_{ij} + \quad \bar{w}_{ij}log\bar{w}_{ij}\right) + \lambda_{ij}\left(w_{ij} + \bar{w}_{ij} - 1\right), \tag{12}$$

$$w_{ij}^{(t+1)} \leftarrow \frac{1}{1+\exp\{[\left(Y - L^{(t+1)}\right)_{ij}^2/2]/\gamma\}}\ , \tag{13}$$

4. Loss Function and Metrics

4.1. Loss Function

Our loss function in equation 14 includes two parts: the localization loss (loc) (this is also known as vector regression) and classification loss (clc).

$$L(x, c, l, g) = \frac{1}{M}(L_{clc}(x, c) + \gamma L_{loc}(x, l, g)) \tag{14}$$

where *M* represents the whole number of boxes and γ represents a regularization term which controls balances of the loss. We used Smooth *L1* loss as a localization loss which predicts parameters between the ground truth box (g) and the predicted box (l). In vector regression, we are very hard to find the correct location of the object. Therefore, we introduce to offsets for the default boxes (d) with the center (cx, cy), width (w) and height (h).

$$L_{loc}(x, l, g) = \sum_{i\in Pos}^{N} \ \sum_{m\in\{cx,cy,w,h\}} \ x_{ij}^k S_{L1}(l_i^m - \hat{g}_j^m) \tag{15}$$

where:

$$\hat{g}_j^{cx} = (\frac{g_j^{cx} - d_i^{cx}}{d_i^w} - \mu_x)/\sigma_x\ ,\quad \hat{g}_j^{cy} = (\frac{g_j^{cy} - d_i^{cy}}{d_i^h} - \mu_y)/\sigma_y$$

$$\hat{g}_j^w = (log\frac{g_j^w}{d_i^w} - \mu_w)/\sigma_w \ , \quad \hat{g}_j^h = (log\frac{g_j^h}{d_i^h} - \mu_h)/\sigma_h \tag{16}$$

$$S_{L1} = \begin{cases} 0.5x^2 & |x| \leq 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{17}$$

We set $\mu_x = \mu_y = \mu_w = \mu_h = 0.001$, $\sigma_x = \mu_y = 0.1$ and $\sigma_w = \sigma_h = 0.2$. In equation 15, We used softmax loss as our classification loss over multiple classes confidences (c).

$$L_{clc}(x, c) = -\sum_{i\in Pos}^{N} \quad x_{ij}^{\mathrm{p}} lo\ g\left(\hat{c}_i^p\right) - \sum_{i\in Neg} \quad (\hat{c}_i^0) \tag{18}$$

where: $\hat{c}_i^p = \exp\left(c_i^p\right) / \sum_p \quad exp\ (c_i^p)$.

4.2. Metrics

Frames Per Second (FPS), Recall and Precision are used for evaluating the performance of the frameworks. To apply above metrics, we need to find out the value of Intersection over Union (IoU), where corresponding area is calculated by two factors including the predicted bounding boxes from our model and ground truth boxes.

Precision and Recall are calculated through under equations, respectively. Precision represents how many objects are correctly predicted, while Recall represents how many objects correctly predicted respect to the ground truth. We called True Positive if the IoU is more than a threshold (typically set to 0.5). We called False Positive if IoU is less than a threshold; and we called False Negative if detector fails to detect.

$$IoU = \frac{\text{Area of overlap}}{\text{Area of union}}$$

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{All\ observations}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{All\ Ground\ Truth}$$

In our experiment, we also tested mAP values derived from mean of Average Precision (AP) of whole classes.

5. Experiment results

5.1. Datasets

In order to compare our MF-SSD framework with the other object detector fairly, we evaluated the MF-SSD on three datasets including: MS COCO [19], PASCAL VOC2007 and PASCAL VOC2012 [20]. Three datasets are introduced in Table 1. Different from VOC dataset, the MS COCO mainly contains various natural pictures, common objects with complicated background and multiple objects with small size; the evaluation process on this dataset is more complex compared to other dataset. The mAP and FPS are used to assess performance results for the MF-SSD and other compared algorithms.

Table 1: Construction of datasets.

Dataset	VOC2007	VOC2012	MSCOCO
Train set(images)	2502	5018	118k
Validation (images)	2512	5824	5k
Test set (images)	4953	10993	41k
Categories	20	20	80
Website	VOC:host.robots.ox.ac.uk/pascal/VOC/ COCO:cocodataset.org/		

5.2. Implementation

In all experiments, we have adopted same data augmentation strategy for SSD [5] and experiments are tested under Pytorch framework [42]. Firstly, our proposed MF-SSD is trained by the combined training set of PASCAL VOC2007 and PASCAL VOC2012. Testing process includes three datasets and we will introduce all of them in the next section.

Two step is required to train the MF-SSD. In first step, freeze train, we set requires grad setting to false for top 28 layers. In second step, unfreeze train, we did not set requires grad setting for all layers. We adopted adam optimizer and set batch size, learning rate and weight decay to 32, 0.00045 and 0.00045 for freeze train, respectively. We also adopted adam optimizer for unfreeze train and set batch size, learning rate and weight decay to 16, 0.00011 and 0.00045 respectively. We have used learning scheduler with step size 1 and gamma 0.94 for both freeze train and unfreeze train. We totally used 50 epochs for freeze train and 100 epochs for unfreeze train. As for MSCOCO, the train set is used to train our model and the evaluation server is used to test our model. We used same training strategy. We used same training strategy for MSCOCO. We adopted adam optimizer and set batch size, learning rate and weight decay to 32, 0.0002 and 0.0002 for freeze train, respectively. For unfreeze train, same optimizer is used and we set learning rate and weight decay to 0.00011, 0.0002 and 0.0002, respectively. We totally used 80 epochs and 100 epochs for freeze train and unfreeze train, respectively.

5.3. Experiment on PASCAL VOC2007 dataset

The comparison results between the MF-SSD and other popular detectors are shown in Table 2. All of the detectors are trained by combined training set of VOC2007 and VOC2012 and evaluated on VOC2007 test set. The proposed MF-SSD300 has achieved 79.8% mAP and 20.6 FPS detection speed for low resolution image with input size 300×300, while the proposed MF-SSD512 with input size 512×512 has achieved 81.5% mAP and 17.5 FPS on PASCAL VOC2007. As shown in Table 2, the performance result of our algorithm is better than two stage object detectors including Faster RCNN, ION, MR-CNN and R-FCN. The R-FCN still did not achieve high performance result, even if it uses Resnet101 as backbone and high-resolution image with input size 600×1000.

The proposed MF-SSD300 outperforms SSD300 by 2.6% mAP, FSSD300 by 1.0% mAP, DSOD300 by 2.1% mAP, FA-SSD by 1.7% mAP and DF-SSD by 0.9% mAP for same input size 300×300. The proposed MF-SSD512 with input size 512×512 has achieved 81.5% mAP, which is higher than most of the detector including one or two stage frameworks. The DSSD513 has achieved 81.5% mAP which is same result to MF-SSD512 but 3 times slower that MF-SSD512. This is because DSSD513 uses ResNet101 as backbone which is stronger network compared with VGG-16. Additionally, DSSD513 method creates 43688 anchor boxes which takes more time to reduce redundant boxes. Compared to DSSD513, our model adapts 24564 anchor boxes.

Table 2 Comparison results between different object detectors on PASCAL VOC2007 dataset.

Method	Data	Backbone	Input size	Boxes	FPS	mAP(%)
FastRCNN[28]	07+12	VGG-16	~600×1000	~2000	0.5	70.0
FasterRCNN[1]	07+12	VGG-16	~600×1000	300	7	73.2
FasterRCNN[1]	07+12	ResNet-101	~600×1000	300	2.4	76.4
ION[43]	07+12	VGG-16	~600×1000	4000	1.25	76.5
MR-CNN[44]	07+12	VGG-16	~600×1000	250	0.03	78.2
R-FCN[2]	07+12	ResNet-50	~600×1000	300	11	77.4
R-FCN[2]	07+12	ResNet-101	~600×1000	300	9	79.5
YOLO[4]	07+12	GoogleNet	448×448	98	98	63.4
YOLOv2[31]	07+12	Darknet-19	352×352	-	81	73.7
SSD300[5]	07+12	VGG-16	300×300	8732	46	77.2
SSD512[5]	07+12	VGG-16	512×512	24564	19	78.5
DSOD300[46]	07+12	DS/64-192-48-1	300×300	8732	17.4	77.7
DSSD321[14]	07+12	ResNet-101	321×321	17080	9.5	78.6
DSSD513[14]	07+12	ResNet-101	513×513	43688	5.5	81.5
DFPR300[45]	07+12	VGG-16	300×300	-	39.5	79.6
DFPR512[45]	07+12	VGG-16	512×512	-	-	81.1
FSSD300[15]	07+12	VGG-16	300×300	8732	65	78.8
FSSD512[15]	07+12	VGG-16	512×512	24564	35	80.9
FA-SSD [47]	07+12	VGG-16	300×300	-	30	78.1
DF-SSD [48]	07+12	DenseNet-S-32–1	300×300	-	11.6	78.9
MF-SSD300	07+12	VGG-16	300x300	8732	20.6	79.8
MF-SSD512	07+12	VGG-16	512x512	24564	17.5	81.5

5.4. Experiment on PASCAL VOC2012 dataset

In second experiment, VOC12 with same experiment setting is used to evaluate different algorithms which are compared with MF-SSD. The proposed MF-SSD300 achieved 77.7% mAP and 80.1% mAP with input size 300×300 and 512×512, respectively. We can see from Table 3 that our framework achieved higher mAP score compared to other detectors. Especially, the MF-SSD512 achieved higher average precision on several categories including Bird, Cow, Car, Person, Plant and Sheep. The proposed MF-SSD achieved higher AP score, which higher than conventional SSD, on many categories, especially for categories with small size such as bird, bottle etc.

Table 3 Comparison results between different object detectors on PASCAL VOC2012 dataset.

Method	mAP(%)	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Person	Plant	Sheep	Sofa	Train	Tv
FasterRCNN[1]	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	80.7	48.1	77.3	66.5	84.7	65.6
R-FCN[2]	77.6	86.9	83.4	81.5	63.8	**62.4**	81.6	81.1	93.1	58.0	83.8	60.8	92.7	86.0	84.4	59.0	80.8	68.6	86.1	72.9
YOLO[4]	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	63.5	28.9	52.2	54.8	73.9	50.8
YOLOv2[31]	73.4	86.3	82.0	74.8	59.2	51.8	79.8	76.5	90.6	52.1	78.2	58.5	89.3	82.5	81.3	49.1	77.2	62.4	83.8	68.7
SSD300[5]	75.8	88.1	82.9	74.4	61.9	47.6	82.7	78.8	91.5	58.1	80.0	64.1	89.4	85.7	82.6	50.2	79.8	73.6	86.6	72.1
SSD512[5]	78.5	90.0	85.3	77.7	64.3	58.5	**85.1**	84.3	92.6	61.3	83.4	65.1	89.9	88.5	85.5	54.4	82.4	70.7	87.1	75.6
DSSD321[14]	76.3	87.3	83.3	75.4	64.6	46.8	82.7	76.5	92.9	59.5	78.3	64.3	91.5	86.6	82.1	53.3	79.6	**75.7**	85.2	73.9
DSSD513[14]	80.0	**92.1**	86.6	80.3	**68.7**	58.2	84.3	85.0	**94.6**	63.3	85.9	65.6	**93.0**	88.5	86.4	57.4	85.2	73.4	87.8	76.8
DSOD300[46]	76.3	89.4	85.3	72.9	62.7	49.5	83.6	80.6	92.1	60.8	77.9	65.6	88.9	85.5	84.6	51.1	77.7	72.3	86.0	72.2
DFPR300[45]	77.5	89.5	85.0	77.7	64.3	54.6	81.6	80.0	91.6	60.0	82.5	64.7	89.9	85.4	84.1	53.2	81.0	74.2	87.9	75.9
DFPR512[45]	80.0	89.6	**87.4**	80.9	68.3	61.0	83.5	83.9	92.4	**63.8**	85.9	63.9	89.9	**89.2**	86.2	56.3	84.4	75.5	**89.7**	**78.5**
MF-SSD300	77.7	87.9	85.5	75.5	63.7	51.8	83.8	80.3	91.5	61.4	83.6	**66.0**	91.2	87.8	84.3	53.1	83.1	73.6	88.4	73.9
MF-SSD512	**80.1**	91.8	87.0	**81.6**	67.6	60.2	85.0	**85.6**	94.0	63.1	**86.4**	65.2	92.1	88.8	**87.3**	**58.8**	**86.9**	70.2	88.2	76.7

Table 4 Comparison results between different object detectors on MSCOCO dataset.

Method	Backbone	Dataset	Avg. Precision, IoU:			Avg. Precision, Area:			Avg. Recall, #Dets:			Avg. Recall, Area:		
			0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
Fast R-CNN [28]	VGG-16	train	19.7	35.9	–	–	–	–	–	–	–	–	–	–
Fast R-CNN [28]	ResNet-101	train	20.5	39.9	19.4	4.1	20.0	35.8	21.3	29.5	30.1	7.3	32.1	52.0
Faster R-CNN [1]	VGG-16	trainval	21.9	42.7	–	–	–	–	–	–	–	–	–	–
YOLOV2[31]	Darknet-19	train135k	21.6	44.0	19.2	5.0	22.4	35.5	20.7	31.6	33.3	9.8	36.5	54.4
YOLOV3-320[32]	Darknet-53	train2017	28.2	51.5	29.7	11.9	30.6	43.4	–	–	–	–	–	–
YOLOV3-608[32]	Darknet-53	train2017	33.0	57.9	34.4	18.3	35.4	41.9	–	–	–	–	–	–
R-FCN[2]	ResNet-101	trainval	29.2	51.5	–	10.3	32.4	43.3	–	–	–	–	–	–
ION [43]	VGG-16	train	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
SSD300 [5]	VGG-16	trainval135k	23.2	41.2	23.4	5.3	23.2	39.6	22.5	33.2	35.3	9.6	37.6	56.5
SSD512[5]	VGG-16	trainval135k	28.8	48.5	30.3	10.9	31.8	43.5	26.1	39.5	42.0	16.5	46.6	60.8
FSSD300 [15]	VGG-16	trainval135k	27.1	47.7	27.8	8.7	29.2	42.2	24.2	37.4	40.0	15.9	44.2	58.6
FSSD512[15]	VGG-16	trainval135k	31.8	52.8	33.5	14.2	35.1	45.0	27.6	42.4	45.0	22.3	**49.9**	62.0
DSSD321 [14]	ResNet-101	trainval135k	28.0	46.1	29.2	7.4	28.1	47.6	25.5	37.1	39.4	12.7	42.0	62.6
DSSD513[14]	ResNet-101	trainval135k	33.0	33.2	35.2	13.0	35.4	**51.1**	**28.9**	43.5	46.2	21.8	49.1	**66.4**
DSOD [46]	DS/64-192-48–1	trainval	29.3	47.3	30.6	9.4	31.5	47.0	27.3	40.7	43.0	16.7	47.1	65.0
MF-SSD300	VGG-16	train2017	29.7	49.8	30.6	12.1	31.9	43.8	26.0	39.5	41.6	17.3	46.5	59.3
MF-SSD512	VGG-16	train2017	**34.1**	**58.5**	**35.5**	**20.2**	**35.9**	46.0	28.4	**44.0**	**46.5**	**26.2**	49.8	65.9

5.5. Experiment on MSCOCO dataset

In third experiment is conducted on MSCOCO which is used with same experiment setting to

evaluate different algorithms with MF-SSD. The results highlighted on Table 4 and the MF-SSD300 achieved 29.7% mAP with input size 300×300 which can be seen higher score for VGG backbone in detector. When we extended input size to 512, the MF-SSD512 got 34.1% mAP. Specifically, the MF-SSD512 has achieved 34.1% on the primary metric $AP_{0.5:0.95}$, 58.5% on the strict metric $AP_{0.5}$ and 35.5% on the strict metric $AP_{0.75}$ respectively. In addition, the proposed MF-SSD has achieved higher score with 20.2% AP on small objects which detectors uses more efficient FFM to detect more precisely. It clearly showed that the outstanding for extracting features for smaller objects of our feature fusion module, which fuses different level features to increases useful contextual information for object detection. The DSSD513 has achieved 33.0% mAP which is higher than other compared methods. The DSSD513 also achieved good performance, especially for large objects.

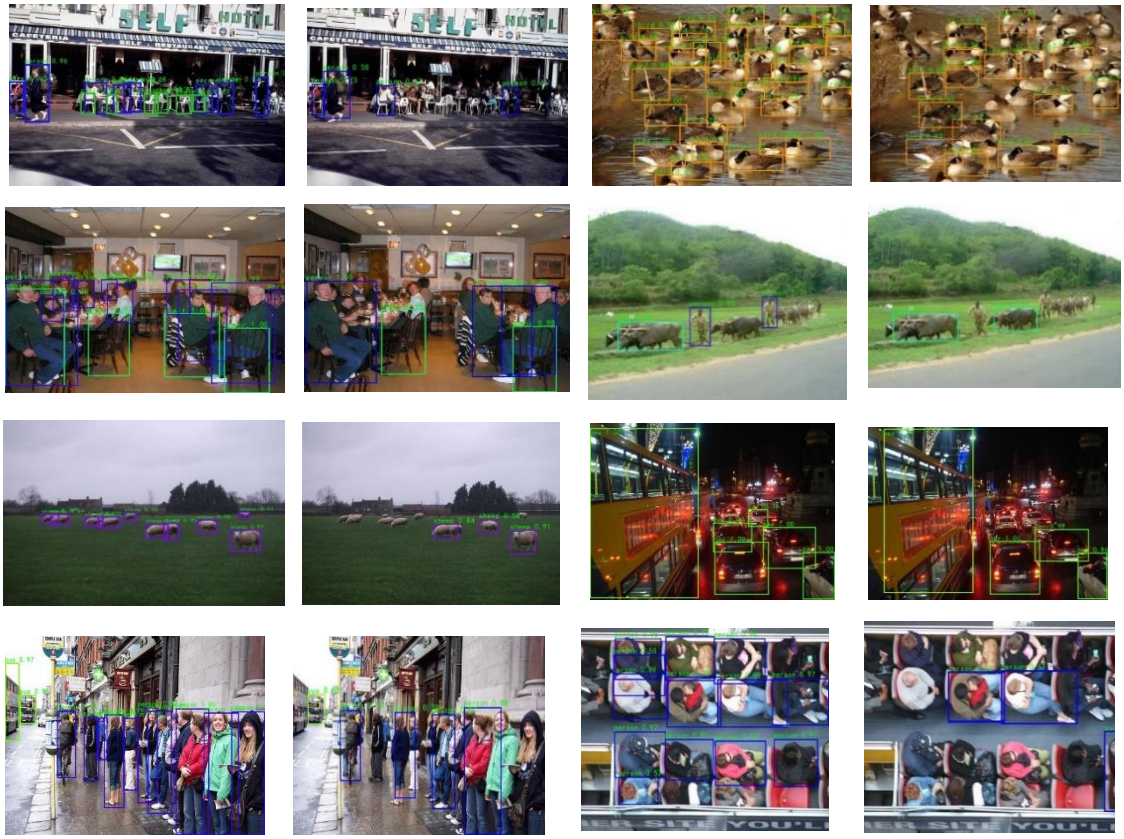


Fig. 6. *Quantitative results between the proposed MF-SSD and conventional SSD on PASCAL VOC2007 test set. The first and third columns represents for MF-SSD; second and forth columns represents for conventional SSD.*

5.6. Quantitative Result

As shown in Fig. 6, we randomly chose 8 images from VOC2007 test set and visualize them with IoU scores higher than 0.7. The final results of the conventional SSD are shown and highlighted in second forth columns. The corresponding results for same picture are highlighted with boxes in the first and third columns. Our MF-SSD detected more objects from all picture we have tested compared to conventional SSD300 with same inputs. For better visualization, we use bonding boxes with same colour for same category and detection score for each category are shown above the bonding boxes. Our

proposed feature fusion module works efficiently in terms of fusing contextual and boundary features, which proves efficacy of our proposed framework.

5.7. Experiment with LRMR method

In this section, we have tested the performance of our LRMR by using two datasets. For all compared algorithms, we have performed experiments on same PC. In our experiments for LRMR testing, we did not use GPU. In the object detection part, a PC with GPU is used for all algorithms on VOC. Our algorithm is run by random parameters settings in the first loop. From second loop, we updated all parameters by using following solution which are given in the above section. The whole process of updating parameters is same.

In our experiment, we selected two kinds of methods to compare the extracting performance of low dimensional component tasks. Some methods we selected in this section are MF based algorithms, and others are NNM based algorithms. We followed authors recommendation of each algorithm in terms of parameters settings. This is the list of compared algorithms: PCP [56], RegL1[63], ROUTE [67], Active [93], AIS-Impute [94], IRNN [95], PRMF [96], and Unify [97].

Table 5: Comparison results between different algorithms in terms of Time, MAE and RMSE.

$r = 5 \mid r(gt) = 5$	$m = n = 500$			$m = n = 800$			$m = n = 1000$			$m = n = 1500$		
$s = 0.5$	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)
ROUTE	0.0237	0.0184	3.6	0.0188	0.0145	10.1	0.0165	0.0128	15.5	0.0136	0.0106	35.3
Our LRMR	**0.0216**	**0.0167**	**2.6**	**0.0172**	**0.0133**	**6.3**	**0.0151**	**0.0117**	**9.7**	**0.0124**	**0.0096**	**22.0**
$r = 10 \mid r(gt) = 5$	$m = n = 500$			$m = n = 800$			$m = n = 1000$			$m = n = 1500$		
$s = 0.5$	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)
ROUTE	0.0382	0.0297	4.0	0.0303	0.0236	9.8	0.0268	0.0209	16.0	0.0222	0.0173	35.2
Our LRMR	**0.0312**	**0.0243**	**2.6**	**0.0258**	**0.0201**	**6.3**	**0.0231**	**0.0181**	**9.7**	**0.0191**	**0.0149**	**22.0**
$r = 20 \mid r(gt) = 20$	$m = n = 500$			$m = n = 800$			$m = n = 1000$			$m = n = 1500$		
$s = 0.5$	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)
ROUTE	0.0480	0.0379	3.8	0.0377	0.0298	9.1	0.0335	0.0265	14.4	0.0273	0.0216	33.4
Our LRMR	**0.0445**	**0.0361**	**2.7**	**0.0351**	**0.0278**	**6.4**	**0.0310**	**0.0245**	**9.9**	**0.0251**	**0.0199**	**22.0**
$r = 40 \mid r(gt) = 20$	$m = n = 500$			$m = n = 800$			$m = n = 1000$			$m = n = 1500$		
$s = 0.5$	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)
ROUTE	0.0852	0.0618	4.2	0.0601	0.0474	10.3	0.0535	0.0423	17.5	0.0437	0.0346	37.7
Our LRMR	**0.0684**	**0.0534**	**2.7**	**0.0541**	**0.0426**	**6.6**	**0.0482**	**0.0380**	**10.8**	**0.0393**	**0.0311**	**23.0**
$r = 40 \mid r(gt) = 40$	$m = n = 500$			$m = n = 800$			$m = n = 1000$			$m = n = 1500$		
$s = 0.5$	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)	RMSE	MAE	Time(s)
ROUTE	0.0718	**0.0565**	4.1	0.0545	0.0432	10.5	0.0483	0.0433	14.9	0.0390	0.0310	35.4
Our LRMR	**0.0715**	**0.0565**	**2.7**	**0.0544**	**0.0430**	**6.8**	**0.0475**	**0.0376**	**10.1**	**0.0367**	**0.0291**	**23.1**

5.7.1 *Synthetic Data*: In this section, we made a synthetic data on which we added all types of noises. We followed following steps for generating a synthetic data. In the first step, we generated a matrix Y which is typically high dimensional data. This matrix can be decomposed into many sub-matrices. In our case, we only selected two sub-matrices to get our data. The second step is corrupting data part. We mixed various noises to obtain high level noisy data. We deliberately added some noisy data which consists of two parts including a uniform distribution over $[-25,25]$ and Gaussian part $\mathcal{N}(0, \sigma^2)$. We can change the outlier ratio (s) from 0% to 70% in our experiment. We selected root mean square error (RMSE) as a main metric. We also adopted mean absolute error (MAE) when two compared algorithm get same results. Each result for all algorithm obtained by averaging 30 times runs.

After having generated synthetic data, we prepared to compare all algorithms. We first tested whether our algorithm has a better tolerance for data with higher outlier. We prepared a data with size (1500*1500). All algorithms estimated our prepared data only for rank 10 estimation. In the beginning, the error ratio is lower than 10%, which is beneficial to better performance. As shown in Fig. 7, most of

the algorithms got better results when the error ratio is low. But some algorithms including RegL1, IRNN and Active did not get better performance for data with lower error ratio. When error ratio has changed from 10% to 40%, we can see a small difference. When error ratio has changed from 40% to 70%, we can clearly see a big difference. Our LRMR got better performance compared to other algorithms including ROUTE, Unify, PRMF, and PCP in terms of rank 10 estimation. There are only two algorithms including our LRMR and ROUTE, which can performance data when error ratio more than 60%. In the next experiment, we further compare these two algorithms in terms of time.

In this experiment, we can change the rank, size and outlier ratio of the data. We first set matrix size, rank, and GT rank to 500, 5 and 5 respectively. In the second time, we set matrix size, rank and GT rank to 800, 5 and 5 respectively. In the third time, we set matrix size, rank and GT rank to 1500, 5 and 5 respectively. We have only changed rank and GT rank for other group of experiments for same data size. As shown in Table 5, two algorithms got relatively similar results when data size is small. We can clearly see our algorithm got better results when data size is high dimensional. In terms of time and other two metrics, our LRMR performance better than compared algorithm. If data is high dimensional and the rank of the data is lower than GT rank, we need to spend more time on this data.

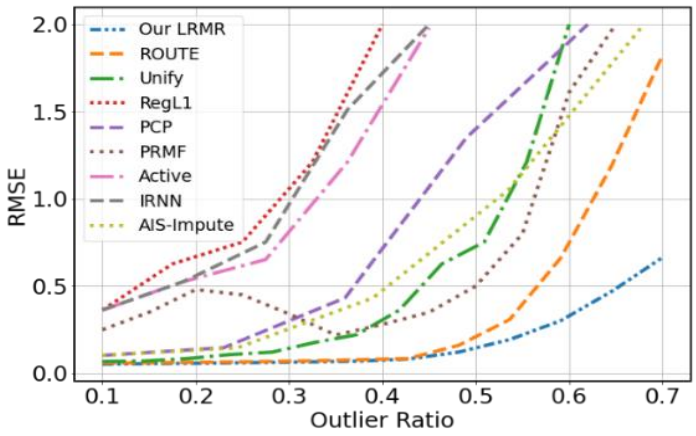


Fig. 7. *Comparison results between different algorithms in terms of RMSE and error ratio s.*

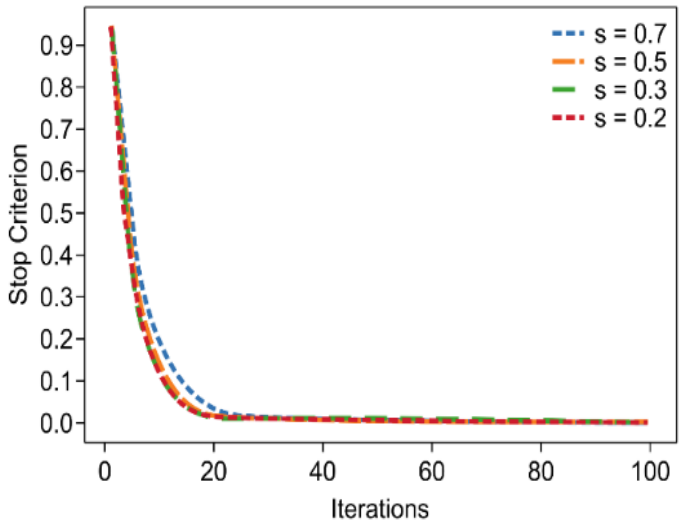


Fig. 8. *Testing Convergence speed on various error ratios.*

Convergence speed is one of the main factors to evaluate better algorithm. In this experiment, we further tested our algorithm in terms of convergence speed. Firstly, we prepared a data with size (1500*1500). For two algorithms, we estimated rank 15 estimation and error ratio has changed 20% to 70%. We can clearly see from Fig. 8 that our LRMR can converge quickly within 30 to 50 iterations. When the error ratio is 70%, our LRMR can also converge within 50 iterations.

In this experiment, we prepared a data with size (800*800), and we can change rank estimation from 20 to 60 and error ratios from 10% to 70% at the same time for our LRMR and OUTE. We can see from Fig. 9 that two algorithms got similar results when error ratios were lower than 40%. When error ratios were larger than 40%, our LRMR got better results compared to OUTE.

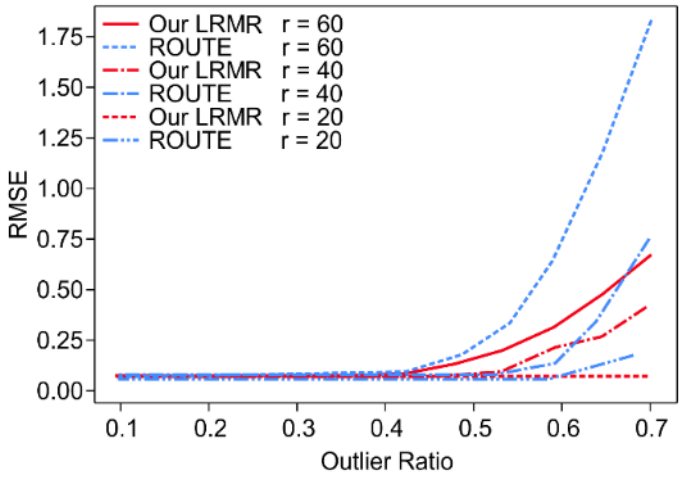


Fig. 9. *Performance comparison on different rank and error ratios.*

Table 6: Comparison results on the DiLiGenT dataset.

	Buddha	Goblet	Reading	Cow	Harvest	Ball	Cat	Pot1	Bear	Pot2	AVE. ERR	Rank
ROUTR	12.01	6.96	8.81	7.41	16.04	3.38	4.1	10	9.98	10.05	8.874	2
RegL1	14.94	11.31	12.62	8.2	25.4	5.44	9.72	8.99	12.82	11.21	12.065	9
AIS-Impute	8.66	10.12	13.06	6.3	24.6	6.26	5.65	7.91	6.51	15.33	10.44	4
PCP	10.43	11.55	12.13	13.3	22.6	4.54	6.73	12.23	11.55	12.55	11.761	7
IRNN	12.71	10.53	11.01	13.8	26.66	5.52	6.11	12.11	7.12	13.22	11.879	8
Unify	10.55	9.72	15.62	16.7	24.5	4.82	6.59	11.29	6.31	10.06	11.616	6
PRMF	11.2	9.88	13.5	13.2	19.9	5.25	6.32	11.56	8.88	12.56	11.225	5
Active	8.6	8.75	13.55	8.01	16.2	5.33	5.44	12.12	10.02	15.55	10.357	3
Our LRMR	8.21	6.73	7.39	5.58	14.3	3.46	3.5	9.1	5.95	9.52	7.374	1

5.7.2 Real Data: In this section, we evaluated our LRMR in Photometric stereo which is mainly used for evaluating the performance of extracting low rank component [98]. The DiLiGenT benchmark [99] which consists of different complex objects for photometric stereo is used to evaluate our algorithm efficacy. The mean angular error (MAE) is used as a metric in this experiment. GT-normal maps and 96 images with different angle are available for each object. We have chosen four algorithms which got better results in the previous experiment to evaluate performance on this real dataset. The difference between two experiments (synthetic data and real data) is that we cannot change ground truth rank in this experiment.

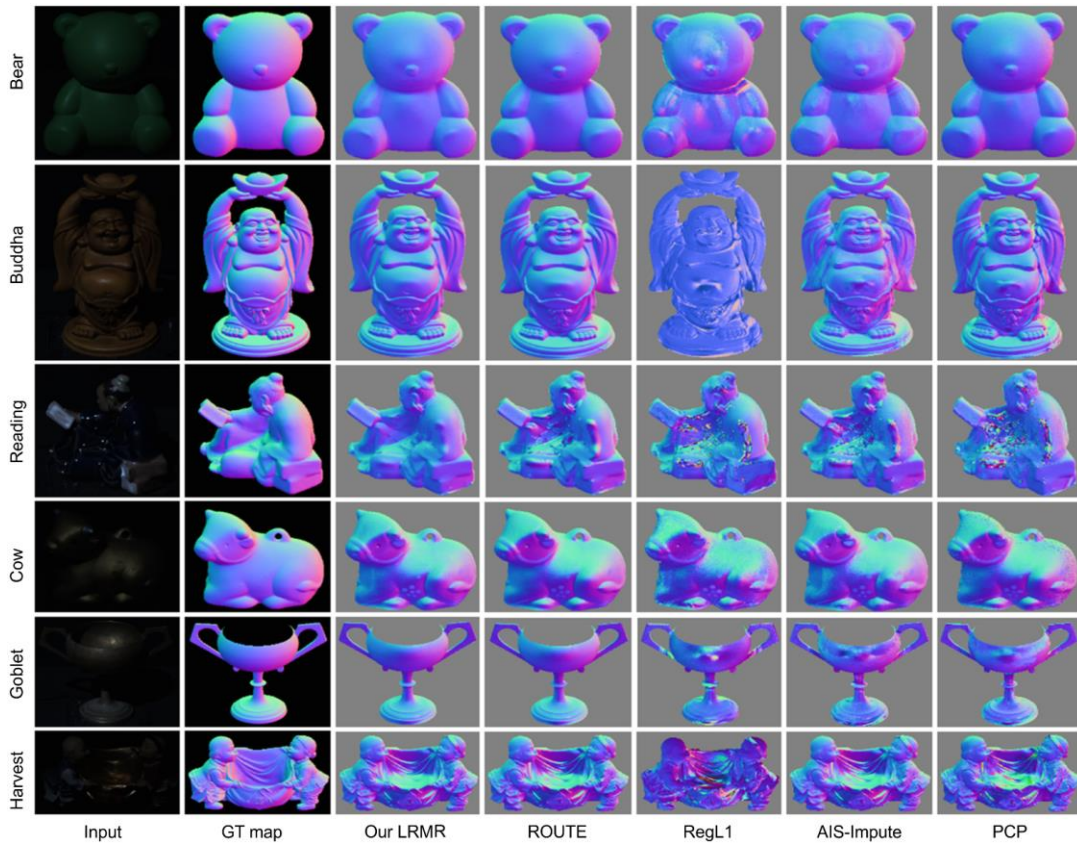


Fig. 10. *Qualitative results between different algorithms on the DiLiGentT.*

We selected algorithms which got better results in the previous section to further test the performance of extracting low rank component. As shown in Fig. 10, we have selected six objects from the dataset, which consists of various materials with different angle and lighting condition. In this experiment, Mean Angular Error (MAE) is used as a metric. As shown in Table 6, our algorithm got lower MAE compared to other algorithms. For the case of Buddha and Harvest, some algorithms cannot perform well. Because it is hart to perform those materials with high error ratios. In Table 6, we give all the experiment results for all algorithms we selected in this section. we can clearly see that our LRMR can perform complex materials better. The reason is that we used weight matrix and entropy term to maximize useful information.

6. Ablation Study

6.1. Feature Fusion Layers

As shown in Table 7, we fuse different layers to test how different layers affect model accuracy. In this experiment, the proposed MF-SSD with input size 300×300 is trained on the combination set of VOC2007/2012, and the MF-SSD is evaluated on VOC 2007 test set. In the first group, we only fused different layers in VGG backbone to test the MF-SSD. In the second group, we fused different layers from both VGG backbone and Extra Layers. In conventional SSD, only one layer, e.g., Conv4-3 layer, is used to extract features for small objects, thus there are no sufficient features to detect more objects. It can be seen from the Table 7, when we fuse both VGG backbone (mainly contain boundary information)

and Extra layers (mainly contain semantic information) the model accuracy is increased. Specially, when we fused Conv 4-3, Conv 5 and Con7 layers in VGG backbone and Block 11, Block 12 and Block 13 layers in Extra layers, our proposed model has achieved 79.8% mAP.

6.2. Number of branches

As shown in Table 8, We evaluated how different N_b (the number of branches in the TRDACM) can affect the model performance. The proposed MF-SSD with input size 300×300 is used, and the experiment conducted on PASCAL VOC dataset. In this experiment, we change the branches of the TRDACM to test model performance. N_2 means that two branch with different dilated ratios $d_1 \in \{2,3\}$ and $d_2 \in \{2,5\}$. N_3 means that three branches with different dilated ratios $d_1 \in \{2,3\}$, $d_2 \in \{2,4\}$ and $d_3 \in \{2,5\}$. N_4 means that four branches with different dilated ratios $d_1 \in \{2,3\}$, $d_2 \in \{2,4\}$, $d_3 \in \{2,4\}$ and $d_4 \in \{2,5\}$. It is worth to noting that the number of dilated ratios represents from top branch to down branch. We achieved best result with N_2. Therefore, we named residual dilated add module as TRDACM.

6.3. Model simplification task

As shown in Table 9, we try to evaluate the effectiveness of the TRDCM and TRDACM modules on conventional SSD. This experiment conducted on combination set of the VOC2007/2012 train set and tested on VOC 2007 test set. The conventional SSD without any modules has achieved 72.2% mAP. The SSD model with TRDACM has achieved 78.1% mAP and 33 FPS. The SSD with TRDACM has achieved a certain enhancement compared with original SSD, but still lack of sufficient information to reach higher accuracy. In first experiment, we only add the TRDACM module and directly use Conv4-3 layer to replace the TRDCM module. In second experiment, we added both modules on SSD at the same time. When we added combination of the TRDACM and TRDCM(a) which is shown in Fig. 3. (a) on SSD, the model has achieved 78.5% mAP and 28.1 FPS. The combination of the TRDACM and TRDCM(b) which is shown in Fig. 3. (b) on SSD has achieved 78.8% mAP and 27.8 FPS.

As shown in Table 7, the model with both TRDCM and TRDACM modules has achieved higher performance, but relatively lower speed. We improved the detection accuracy of the original SSD using our proposed feature fusion module which embeds the TRDCM and TRDACM modules. The experimental result clearly shows that the efficiency of the proposed feature fusion module.

Table 7 Detection speed of MF-SSD with different Blocks on VOC dataset.

VGG Backbone	Extra Layer	mAP(%)
Conv4-3, Conv5, Conv6	–	77.1
Conv4-3, Conv5, Conv6	Block11, Block12, Block13	79.1
Conv4-3, Conv5, Conv7	–	77.5
Conv4-3, Conv5, Conv7	Block12, Block13	78.9
Conv4-3, Conv5, Conv7	Block11, Block12, Block13	79.8
Conv4-3, Conv5, Conv7	Block12, Block13, Block14	79.6
Conv4-3, Conv5, Conv7	Block11, Block12, Block13, Block14	79.5
Conv4-3, Conv6, Conv7	–	77.2
Conv4-3, Conv6, Conv7	Block12, Block13	78.1
Conv4-3, Conv6, Conv7	Block12, Block13, Block14	78.5
Conv4-3, Conv6, Conv7	Block11, Block12, Block13	78.9

Table 8 Detection speed of MF-SSD on different branches of TRDACM module. N_b represents the number of branches.

Number of branches (N_b)	Data	mAP (%)	FPS
N_2	VOC 07/12	79.8	20.6
N_3	VOC 07/12	79.7	20.1
N_4	VOC 07/12	79.4	19.8

Table 9 Simplification task using different modules.

Model	Data	mAP (%)	FPS
SSD	VOC 07/12	77.2	46
SSD+TRDACM	VOC 07/12	78.1	33
SSD+TRDCM(a)+TRDACM	VOC 07/12	78.5	28.1
SSD+TRDCM(b)+TRDACM	VOC 07/12	78.8	27.8
MF-SSD300 (ours)	VOC 07/12	79.8	20.6
MF-SSD512 (ours)	VOC 07/12	81.5	17.5

7. Conclusion

In this work, we proposed a new detector, the MF-SSD, which consists of two newly designed modules to improve detection ability. Whole framework consists of three parts including: 1. First part is backbone layer which is mainly used for extracting large and boundary elements such as line, angle etc. 2. Second part is multi-path feature fusion layer which contains various modules to fuse more information. 3. Third part is detection head which contains classification and regression layers to get final results. We also designed the TRDCM and TRDACM modules, which not only enlarges the receptive field without losing spatial resolution, but also improves contextual information without extra computational cost. Those two modules designed based on different convolutional layers with different filter size such as kernel size from 1 to 5. The efficacy of our MF-SSD is tested multiple datasets to prove our statements. The detection speed of objects including small size or large size two times faster than conventional SSD and the efficacy of detection in terms of small objects is higher than conventional SSD. The current work focuses on VGG as backbone. We will further improve MF-SSD with same feature fusion module using other powerful backbones in our future work. The main disadvantage of VGG is that different layer features cannot be fused at the same time. Because VGG stacks all layers only one direction. We will solve this problem in the future work to further improve MF-SSD.

8. References

[1] Ren, S., He, K., Girshick, R., Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems 2015,91–99.

[2] Li, Y., He, K., Sun, J., et al. R-fcn: Object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems 2016, 379–387.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, Computer Vision – ECCV 2014 2014, Volume 8691.

[4] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 779–788.

[5] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. Ssd: Single shot multibox

detector. InEuropean conference on computer vision 2016, pp. 21-37.
[6] Lin, T.Y., Goyal, P., Girshick, R., He, K., Doll´ar, P. Focal loss for dense object detection. InProceedings of the IEEE international conference on computer vision 2017, pp. 2980-2988.
[7] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).
[8] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, 770–778.
[9] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. International Journal of Computer Vision ,2015, 211–252.
[10] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, 1–9.
[11] Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition 2017, pp. 4700-4708.
[12] Adelson EH, Anderson CH, Bergen JR, Burt PJ, Ogden JM. Pyramid methods in image processing. RCA engineer 1984 Nov 1;29(6):33-41.
[13] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition 2017, pp. 2117-2125.
[14] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. DSSD : Deconvolutional single shot detector. CoRR, abs/1701.06659, 2017.
[15] Z. Li, F. Zhou, Fssd: Feature fusion single shot multibox detector, 2017, arXiv preprint arXiv:1712.00960.
[16] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollr. Learning to refine object segments. In ECCV 2016.
[17] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In ICLR 2016.
[18] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. DetNet: Design backbone for object detection. In ECCV 2018.
[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision 2014, pages 740–755.
[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International Journal of Computer Vision 2010, 588(2):303–338.
[21] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition 2005.
[22] Lowe, D.G. Distinctive image features from scale-invariant keypoints. International journal of computer vision 2004, 91–110.
[23] Girshick, R.B., Felzenszwalb, P.F., McAllester, D. Discriminatively trained deformable part models, release 5. 2012.
[24] Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W. Selective search for object recognition. International journal of computer vision 2013, 154–171.
[25] Zitnick, C.L., Doll´ar, P.: Edge boxes: Locating object proposals from edges. In: European Conference on Computer Vision, Springer 2014, 391–405.
[26] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems 2015, 28.
[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR 2014.
[28] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision 2015, pp. 1440–1448
[29] H. Zheng, J. Chen, L. Chen, Y. Li, Z. Yan, Feature enhancement for multi-scale object detection, Neural Process.Lett 2020, 1-13.
[30] Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229. 2013 Dec 21.
[31] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp.7263–7271.

[32] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. arXiv:1804.02767, 2018.
[33] Bochkovskiy A, Wang CY, Liao HY. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. 2020 Apr 23.
[34] A G. Jocher, et al., ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support, https://doi.org/10.5281/zenodo.5563715, Oct. 2021.
[35] Law H, Deng J. CornerNet: Detecting Objects as Paired Keypoints[J]. 2018.
[36] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition 2015, pp. 3431–3440.
[37] B. Hariharan, P. Arbel´aez, R. Girshick, J. Malik. Hypercolumns for object segmentation and finegrained localization, in: Proceedings of the IEEE conference on computer vision and pattern recognition 2015, pp. 447–456.
[38] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition 2018, pp. 8759–8768.
[39] M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, pp. 10781–10790.
[40] B. Chen, G. Ghiasi, H. Liu, T.-Y. Lin, D. Kalenichenko, H. Adam, Q. V. Le, Mnasfpn: Learning latency-aware pyramid architecture for object detection on mobile devices, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, pp. 13607–13616.
[41] Wang N, Gao Y, Chen H, Wang P, Tian Z, Shen C, Zhang Y. NAS-FCOS: Fast neural architecture search for object detection. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, pp. 11943-11951.
[42] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A. Automatic differentiation in pytorch 2017.
[43] S. Bell, C. Lawrence Zitnick, K. Bala, R. Girshick, Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 2874–2883
[44] S. Gidaris, N. Komodakis, Object detection via a multi-region and semantic segmentation-aware cnn model, in: Proceedings of the IEEE International Conference on Computer Vision 2015, pp. 1134–1142.
[45] T. Kong, F. Sun, C. Tan, H. Liu, W. Huang, Deep feature pyramid reconfiguration for object detection, in: Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 169–185.
[46] Z. Shen, Z. Liu, J. Li, et al. Dsod: Learning deeply supervised object detectors from scratch, in: Proceedings of the IEEE International Conference on Computer Vision 2017, pp. 1919–1927.
[47] J. -S. Lim, M. Astrid, H. -J. Yoon and S. -I. Lee, "Small Object Detection using Context and Attention," 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2021, pp. 181-186, doi: 10.1109/ICAIIC51459.2021.9415217.
[48] S. Zhai, D. Shang, S. Wang and S. Dong, "DF-SSD: An Improved SSD Object Detection Algorithm Based on DenseNet and Feature Fusion," in IEEE Access, vol. 8, pp. 24344-24357, 2020.
[49] Liu C, Wei D, Xiang J, Ren F, Huang L, Lang J, Tian G, Li Y, Yang J. An improved anticancer drug-response prediction based on an ensemble method integrating matrix completion and ridge regression. Mol Ther Nucl Acids. 2020 Sep 4;21:676-86.
[50] Ahmed A, Romberg J. Compressive multiplexing of correlated signals. IEEE Trans Inf Theory. 2014 Oct 31;61(1):479-98.
[51] Liu Q, Shang C, Huang D. Efficient low-order system identification from low-quality step response data with rank-constrained optimization. Control Eng Pract. 2021 Feb 1;107:104671.
[52] Gibanica M, Abrahamsson TJ, McKelvey T. State-space system identification with physically motivated residual states and throughput rank constraint. Mech Syst Signal Process. 2020 Aug 1;142:106579.
[53] Li X, Wang Z. Trees with extremal spectral radius of weighted adjacency matrices among trees weighted by degree-based indices. Linear Algebra Appl. 2021 Jul 1;620:61-75.
[54] Su Y, Bai X, Li W, Jing P, Zhang J, Liu J. Graph regularized low-rank tensor representation for feature selection. J Vis Commun Image Representation. 2018 Oct 1;56:234-44.
[55] Tsagkarakis N, Markopoulos PP, Sklivanitis G, Pados DA. L1-norm principal-component analysis of complex data. IEEE Trans Signal Process. 2018 Mar 30;66(12):3256-67.
[56] Candès EJ, Li X, Ma Y, Wright J. Robust principal component analysis?. J ACM. 2011 Jun 9;58(3):1-37.
[57] Chi Y, Lu YM, Chen Y. Nonconvex optimization meets low-rank matrix factorization: An overview.

IEEE Trans Signal Process. 2019 Aug 23;67(20):5239-69.
[58] Chen Y, Chi Y, Goldsmith AJ. Exact and stable covariance estimation from quadratic sampling via convex programming. IEEE Trans Inf Theory. 2015 May 4;61(7):4034-59.
[59] Zhang Q, Wang H. Collaborative filtering with generalized Laplacian constraint via overlapping decomposition. IJCAI, 2016 Jan 1; 2329-2335.
[60] Wang H, Li Y, Cen Y, He Z. Multi-matrices low-rank decomposition with structural smoothness for image denoising. IEEE Trans Circuits Syst Video Technol. 2019 Jan 4;30(2):349-61.
[61] Zhang F, Yang G, Xue JH. Hyperspectral image denoising based on low-rank coefficients and orthonormal dictionary. Signal Process. 2020 Dec 1;177:107738.
[62] Basri R, Jacobs D, Kemelmacher I. Photometric stereo with general, unknown lighting. Int J Comput Vis. 2007 May;72(3):239-57.
[63] Zheng Y, Liu G, Sugimoto S, Yan S, Okutomi M. Practical low-rank matrix approximation under robust l 1-norm. 2012 IEEE Conf Comput Vis Pattern Recognit. 2012 Jun 16; 1410-1417.
[64] Otazo R, Candes E, Sodickson DK. Low‐rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components. Mag Reson Med. 2015 Mar;73(3):1125-36.
[65] Wang Y, Wu L, Lin X, Gao J. Multiview spectral clustering via structured low-rank matrix factorization. IEEE Trans Neural Netw Learn Syst. 2018 Jan 4;29(10):4833-43.
[66] Liu G, Lin Z, Yan S, Sun J, Yu Y, Ma Y. Robust recovery of subspace structures by low-rank representation. IEEE Trans Pattern Anal Mach Intell. 2012 Apr 10;35(1):171-84.
[67] Guo X, Lin Z. Low-rank matrix recovery via robust outlier estimation. IEEE Trans Image Process. 2018 Jul 12;27(11):5316-27.
[68] Candes EJ, Plan Y. Matrix completion with noise. Proc IEEE. 2010 Apr 26;98(6):925-36.
[69] Chi Y, Lu YM, Chen Y. Nonconvex optimization meets low-rank matrix factorization: An overview. IEEE Trans Signal Process. 2019 Aug 23;67(20):5239-69.
[70] Shang F, Liu Y, Tong H, Cheng J, Cheng H. Robust bilinear factorization with missing and grossly corrupted observations. Inf Sci. 2015 Jun 20;307:53-72.
[71] Lakshminarayanan B, Bouchard G, Archambeau C. Robust Bayesian matrix factorisation. Proc AISTATS. 2011 Jun 14; 425-433
[72] Fan H, Li J, Yuan Q, Liu X, Ng M. Hyperspectral image denoising with bilinear low rank matrix factorization. Signal Process. 2019 Oct 1;163:132-52.
[73] Deville Y. From separability/identifiability properties of bilinear and linear-quadratic mixture matrix factorization to factorization algorithms. Digit Signal Process. 2019 Apr 1;87:21-33.
[74] Mazumder R, Hastie T, Tibshirani R. Spectral regularization algorithms for learning large incomplete matrices. J Mach Learn Res. 2010 Aug 1;11:2287-322.
[75] Candès EJ, Tao T. The power of convex relaxation: Near-optimal matrix completion. IEEE Trans Inf Theory. 2010 Apr 19;56(5):2053-80.
[76] Candès EJ, Recht B. Exact matrix completion via convex optimization. Found Comput Math. 2009 Dec;9(6):717-72.
[77] Recht B. A simpler approach to matrix completion. J Mach Learn Res . 2011 Dec 1;12(12).
[78] Koltchinskii V, Lounici K, Tsybakov AB. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. Annals Stat. 2011 Oct;39(5):2302-29.
[79] Niknazar H, Nasrabadi AM, Shamsollahi MB. A new blind source separation approach based on dynamical similarity and its application on epileptic seizure prediction. Signal Process. 2021 Jun 1;183:108045.
[80] Babatas E, Erdogan AT. Time and frequency based sparse bounded component analysis algorithms for convolutive mixtures. Signal Process. 2020 Aug 1;173:107590.
[81] Peng C, Zhang Z, Kang Z, Chen C, Cheng Q. Nonnegative matrix factorization with local similarity learning. Inf Sci. 2021 Jul 1;562:325-46.
[82] Wei W, Feiyu C, Yongxin G, Sheng H, Xiaohong Z, Dan Y. Discriminative deep semi-nonnegative matrix factorization network with similarity maximization for unsupervised feature learning. Pattern Recognit Lett.,2021,149:157-163.
[83] Peng X, Xu D, Chen D. Robust distribution-based nonnegative matrix factorizations for dimensionality reduction. Inf Sci. 2021 Apr 1;552:244-60.
[84] Djennadi S, Shawagfeh N, Arqub OA. A fractional Tikhonov regularization method for an inverse backward and source problems in the time-space fractional diffusion equations. Chaos, Solitons & Fractals. 2021 Sep 1;150:111127.
[85] Zhang J, Qi H, Jiang D, He M, Ren Y, Su M, Cai X. Acoustic tomography of two dimensional velocity field by using meshless radial basis function and modified Tikhonov regularization method. Measurement. 2021 Apr 1;175:109107.

[86] Galvan G, Lapucci M. On the convergence of inexact augmented Lagrangian methods for problems with convex constraints. Op Res Lett. 2019 May 1;47(3):185-9.
[87] Dostál Z, Vlach O. An accelerated augmented Lagrangian algorithm with adaptive orthogonalization strategy for bound and equality constrained quadratic programming and its application to large-scale contact problems of elasticity. J Comput Appl Math. 2021 Oct 1;394:113565.
[88] Liu T, Sun M, Liu Y, Hu D, Ma Y, Ma L, Feng N. ADMM based low-rank and sparse matrix recovery method for sparse photoacoustic microscopy. Biomed Signal Process Control. 2019 Jul 1;52:14-22.
[89] Bai J, Ma Y, Sun H, Zhang M. Iteration complexity analysis of a partial LQP-based alternating direction method of multipliers. Appl Numer Math. 2021 Jul 1;165:500-18.
[90] Yang J, Yuan X. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. Math Comput. 2013;82(281):301-29.
[91] Dreves A, Facchinei F, Kanzow C, Sagratella S. On the solution of the KKT conditions of generalized Nash equilibrium problems. SIAM Journal on Optimization. 2011 Jul 1;21(3):1082-108.
[92] Fazel, Maryam. "Matrix rank minimization with applications." PhD diss., PhD thesis, Stanford University, 2002.
[93] Hsieh CJ, Olsen P. Nuclear norm minimization via active subspace selection. Int Conf Mach Learn. 2014 Jan 27; 575-583.
[94] Yao Q, Kwok JT. Accelerated inexact soft-impute for fast large-scale matrix completion. Proc. 24th Int Joint Conf Artif Intell. 2015 Jun 27; 4002-4008.
[95] Lu C, Tang J, Yan S, Lin Z. Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm. IEEE Trans Image Process. 2015 Dec 22;25(2):829-39.
[96] Wang N, Yao T, Wang J, Yeung DY. A probabilistic approach to robust matrix factorization. Proc ECCV. 2012 Oct 7; 126-139.
[97] Cabral R, De la Torre F, Costeira JP, Bernardino A. Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. Proc. ICCV. 2013; 2488-2495.
[98] Woodham RJ. Photometric method for determining surface orientation from multiple images. Opt Eng. 1980 Feb;19(1):191139.
[99] Shi B, Wu Z, Mo Z, Duan D, Yeung SK, Tan P. A benchmark dataset and evaluation for non-Lambertian and uncalibrated photometric stereo. Proc IEEE Conf Comput Vis Pattern Recognit. 2016; 3707-3716.