Mansoura Journal For Computers and Information Sciences

Relevant Image Ranking Based on Transfer RetinaNet Learning

Hoda El-Batrawy[1], Ahmed Atwan[2], Hassan Soliman[3], and Mohammed Elmogy[4]
Information Technology Department, Faculty of Computers and Information, Mansoura University, Mansoura, Egypt
[1]hodacis@gmail.com, [2]atwan_2@yahoo.com, [3]hhsoliman2015@gmail.com, and [4]melmogy@mans.edu.eg

ABSTRACT

Computer vision and deep learning are intrinsic applications in the machine learning field that become smarter day by day. The significant challenge in deep learning tasks converges on extracting the deepest and the most semantic features of the image. So, the promotion of deep learning techniques has enormous leverage in the retrieval of deep image ranking. This research has tackled an essential task of relevant deep image ranking based on learning from the RetinaNet superior detection technique. RetinaNet learning technique is employed to learn deep semantic features embeddings from the imaging dataset. Transfer of learning is a powerful scheme that proposes hyper-parameterization of the RetinaNet network for relevant image ranking. It transfers RetinaNet detector learning (weights) for deep relevant image ranking systems. Thus, we achieved the best accuracy. Our experimental results manifest that our deep learning procedure enhances the retrieval results efficiently and accurately and focuses on inhibiting the learning time of a deep relevant ranking task. As compared with other state-of-the-art object detectors, the RetinaNet detector accomplished more than a 97% mean average precision (MAP). This results in outperformed tested work. These superior results pretend the effective impact of our proposed procedure learning that drives the more efficient and relevant result of the deep ranking task.

Keywords

RetinaNet Deep Learning, Feature Pyramids Network Extractor, Focal Loss for class imbalance, Transfer Of Learning, Triplet Embedding Sampling, Ranking Loss.

1. INTRODUCTION

Relevance image ranking is considered a significant application in a computer vision field. The ranking process is considered as widely challenging due to soft in-class differences and large out-class differences with variance sub-categories. The problem of deep image ranking concentrates on returning relevance images that belong to the same fundamental category. Because of the significant progress of deep learning, the deep convolutional neural networks (CNNs) have been concluded in a wide range of image detection task [1,2]. Recently, it generates marvelous results of learning deep semantic representation of images for deep relevance image ranking. It was shown that CNN is also practical, especially in image retrieval procedure, which receives a lot of concern nowadays. Deal an image as target and source by computing the similarity from a target image. How to generate features that are respective on semantic appearance playing an essential role in relevance image ranking criteria. With surpassing progress in modern computing technologies, machine learning, especially deep learning, has become the most thrilling technology [3,4]. This phenomenal success made the machine learning algorithm crowded with more experience and efficiency [5] to solve multiple problems like deep relevance image ranking problems. A lot of procedures were proposed to solve learning for the ranking problem such as features embeddings detection, image classification using features embedding and images ranking with the rise to similarity vision of semantic data that widely ranged.

Deep CNNs are a potent category of deep learning applications responsible for enormous progress in computer vision fields [6]. Deep learning employs multi-layered deep CNNs to obtain the best results of accuracy at various machine learning fields like object detection fields that are implemented to be able to detect the distinct object in images. Then, this template can be reused to detect any object else.

Ranking relevant images is considered a machine learning problem that employs several functions based on the similarity of image features embeddings to build a deep relevant image ranking model. The first learning strategy is a features learning, which extracts features embeddings for given source images. The second learning strategy is a similarity learning between the features embeddings. The third learning strategy is a ranking that is based on computing embeddings loss. The images feature extraction task considers the significant task in relevant ranking images. There are several techniques for extracting image features. The old techniques were using hand-crafted procedures like Gabor filters [7], scale-invariant feature transform (SIFT) [8], and histograms of oriented gradients (HoG) [9], and then learned image similarity metric on top of these generated features. The strength of the hand-crafted features has been based on the efficiency of these procedures [10]. Modern feature extraction techniques use deep CNNs to extract semantic features and then learn a similarity metric based on these features embedding. Modern feature extraction techniques accomplish superior performance than the state-of-the-art hand-crafted features techniques.

In this work, a deep relevant image ranking model is proposed, which leverages modern feature extraction learning techniques. Then, transfer of learning procedure [11,12] is employed to transfer feature learning for improvement retrieval of relevant image ranking. Transfer learning leverages relevant knowledge from previous learning experiences to improve generalization. RetinaNet [13] is a

modern learning technique focusing on "hard" features of images, which make detection techniques more robust and efficient helping to improve prediction accuracy. The experimental evaluation shows that the transfer of RetinaNet learning (weights) using transfer learning technique achieves a massive improvement for deep relevant image ranking model. It led to the best result rather than previous old feature extraction techniques like state-of-the-art hand-crafted features-based modality and fine-grained image retrieval modality based CNNs like AlexNet [14].

This rest of the paper is organized as follows. We present related work of deep relevant image ranking in Section 2. In section 3, we discuss in detail the methodology of the deep relevant ranking technique based on the RetinaNet learning technique and transfer learning technique. Experiments and analysis are specified in Section 4. The conclusions are summarized in Section 5. Finally, In section 6, we define acknowledgment. References are proposed in section 7.

2. RELATED WORK

This section discusses the current related work on image ranking that is based on deep learning techniques. The idea of a pre-trained image ranking using a deep learning module is proposed to generate the high-level features that have been extracted in the deeper layers (last fully connected) of deep convolutional neural networks. This Semantic information has been utilized to achieve relevant retrieval and increase the accuracy of the image ranking. Chechik et al. [15] online algorithm for scalable image similarity learning (OASIS) trained the neural network by "hand-crafted" features to captures both semantic and visual aspects of image similarity, which learn similarity ranking procedure on top of the hand-crafted features. OASIS performs well in a wide range of scales rather than learned directly from the image pixels. Wang et al. [16] proposed learning fine-grained image similarity. They employed the first deep CNN AlexNet to retrieve relevant images from large data training according to the target image using a triplet loss ranking procedure. Fine-grained deep ranking modality accomplish better performance than category-level image similarity models hand-crafted approach. Krizhevsky et al. [17] trained a large deep convolutional neural network to classify the 1.2 million high-resolution images of the ImageNet LSVRC-2010 contest into the 1000 various classes. The deep CNN is capable of achieving record-breaking results on a highly challenging dataset.

Girshick et al. [18] proposed region-convolutional neural networks (R-CNN). He utilized external procedure selective search (SS) [19] to generate candidate regions through CNNs Alex-Net. These generated candidate detection regions forward to fully connected layers to generate feature vectors with fixed length from each region. Classification and regression training is done by a support vector machine (SVM) [20,21]. Merging various image features of low-level with high-level from detection schema, it is a larger relative improvement in detection precision. He et al. [22] proposed spatial pyramid pooling networks procedure (SPPNets) that uses end-to-end convolution training to speed up the R-CNN technique. Convolutional feature maps are generated for the given input image and then forward-propagate the region proposals extracted using SS at the feature map level. Classification and regression training replaced with end-to-end CNNs training. Girshick [23] Presented a Fast R-CNN that fed-forward images in a sequence through CNN. These progress drive to minify computational processes and model costs. Ren et al. [24] introduce a Faster R-CNNs that is significantly faster in training and testing sessions than other previous R-CNNs. Faster R-CNNs replace the external approach in a recent work that generates detection regions of low-level image features by one separately deep CNNs called region proposal network (RPN) [25]. RPN is learned to pool high-level image features that obtained a great improvement of accuracy that has ever been released.

Although, the detection process based on the region proposal has great leverage on improving the accuracy of the object detection process. It impacts on the speed of the whole detection model because extracting these region proposals that contain objects takes more time-consuming. Therefore, Redmon et al. [26] introduced a You only look once that achieved more than 1000x faster than R-CNN and 100x faster than Fast R-CNN. YOLO handles whole images through CNNs that divide the image into chunks of C x C using convolution layers extracting feature maps. However, YOLO is faster than previous models and minifies background mistakes by taking into consideration the objects in the background. It has a spatial constraint that limits the number of nearby objects that our model can predict because each grid cell only predicts two boxes and can only have one class. Liu et al. [27] introduced a single-shot multibox detector (SSD) that is deeper than the previous modalities. It demands two deep networks like Faster R-CNNs. SSD added additional ancillary convolution layers called multi-scaled layers at the end of the truncated base convolution layer instead of the fully connected layer in Faster R-CNNs. Changeable size of feature maps on these scaled layers from high to low lead to detect the different sizes of objects. These generated feature maps congregated from all convolution layers and passed to the prediction layer. Then, they convolved with one convolutional filter of size 3×3 to predict the bounding boxes around the object and probability score. One-single stage detector procedures SSD faces a large class-imbalance throughout the training. Class imbalance is considered an enormous problem that occurred due to generated bounding boxes around objects [28]. SSD detector generated (eg.,10-100k) candidate bounding boxes per image, but only a few bounding boxes implicate objects. Redmon et al. [29] proposed an improved YOLO model named YOLOv2 conceived to make YOLO Better, Faster and Stronger. YOLOv2 proposed training on higher resolution and introduced the passthrough layer, which is responsible for predicting bounding boxes in the image using an anchor box.

YOLOv2 predicts more than a thousand boxes per image in reverse, YOLO only predicts 98 boxes per image, which made YOLOv2 better. In YOLOv2, the author sought to make YOLO faster by developing a base network that used Darknet-19 that was responsible for the feature extraction process. The author also presents a stronger YOLO called YOLO9000, a real-time system that is combining COCO's detection dataset [30] with ImageNet's classification dataset [31]. YOLO9000 detects more than 9000 objects categories by learning new deeper categories from the classification-labeled data to expand the number of categories that the model can detect. YOLOv2 achieved a significant improvement compared with YOLO. However, it still needs embedded computing devices with high processing memory for the object detection process. And it also struggles to localize small-size objects precisely. Rui Li et al. [32] propose improvements to YOLOv2. Improved YOLOv2 replaces the standard convolution layer with a depth-wise separable convolutions layer as in [33] to enhance the calculation speed of the YOLOv2 to some extent. It also replaces the feature fusion method with feature pyramid network [34] that generate multi-scale feature maps to enhance the object detection task to be able to detect different size objects. Although Improved YOLOv2 enhanced the precision of the detection process depth-wise separable convolution method makes the model weaken on object presentation. And, the feature pyramid network is more complex to concatenate the feature map.

The limitations of the image detection techniques have been evaluated and discussed in [35,36]. These limitations made the researchers improve detection techniques of deep learning to achieve the best results of the detection learning task. This study introduces the superior dep learning technique to achieve the best image ranking retrieval result. The ranking result dependens on the detection technique used. So, we used the RetinaNet detection technique, which employs a feature pyramid network technique to detect objects on a different scale and detect all small objects in the image. We need to achieve the best relevant ranking result according to the target image. The current technique, like SSD or YOLO, suffers from an extreme class imbalance. These detectors appreciate roughly between ten to a hundred thousand candidate locations. Most of these candidate boxes do not contain an object. RetinaNet detector technique improved cross-entropy loss called focal loss. Focal loss is a beautiful idea to scale the cross-entropy loss. It helps to reduce the relative loss for well-classified examples (easy example) and putting more focus on misclassified examples (hard examples). The main contributions of this paper include the following. (1) The deepest detection method, called RetinanNet [13] is learned to generate the most semantic and deepest feature embeddings. It is a highly effective system that can handle the class imbalance problem in one stage detector method and exceed the second network RPNs problem in the two-stage detector method. (2) The deep ranking model learns the similarity of pairs features embeddings based on the output of RetinaNet learning to achieve the best retrieval. (3) Transfer of Learning technique is utilized to transfer feature embeddings of RetinaNet detector learning (weights) to deep image ranking systems. It saves the time consumed in learning for ranking and achieves the relevant ranking result according to the target.

3. METHODOLOGY

The proposed system is constructed with three techniques, which aggregate all experience of deep learning to construct a robust and efficient deep relevance image ranking system. The first technique proposes an efficient learning approach called RetinaNet, which pre-processes images based on feature pyramid network (FPN) [34]. FPNs allow us to detect both low-level features (visual features) and high-level features (semantic features) of the image helping to generate the good and most significant features embeddings of images. The second technique is a deep ranking technique. It learns a triplet sampling approach to drive similar feature embeddings with a small distance for pairs of similar images and drive different feature embeddings with a large distance for pairs of different images. Then, it proposes the ranking approach for computing ranking loss of triplets. The third technique is the transfer learning technique that transfers the output of the first technique to be used for the second technique for speeding up the training and achieving high performance in a deep image ranking model. It transfers the learning of RetinaNet detection (weights) to improve the ranking result. So, the model retrieves the relevant images to the target image in the top and the irrelevant images appear in the bottom achieving the best ranking result. The model overview of the proposed system is demonstrated in Figure 1. These techniques details are further explained in the following subsection.

3.1 RetinaNet Deep Learning

RetinaNet considers the most significant object detector modality. It utilizes FPN as one unified network constructed on the finest level of feedforward of ResNet [37] architecture. FPN is a convolution network that utilizes multiple convolution layers for generating deep semantic feature maps. It consists of one network with two fully connected networks called subnetworks for detection, classification, and regression processes. FPN specified features in different scales to generate multi-scale feature maps. Object classification process utilizing a convolution class subnetwork to classify FPN output. Bounding box regression process utilizing a convolution box regressor subnetwork to regress the bounding box around the object using feature maps extracted from FPN. The experimental result of RetinaNet architecture affirms that it is the best modality to obtain a lower prediction time and achieve a higher accuracy.

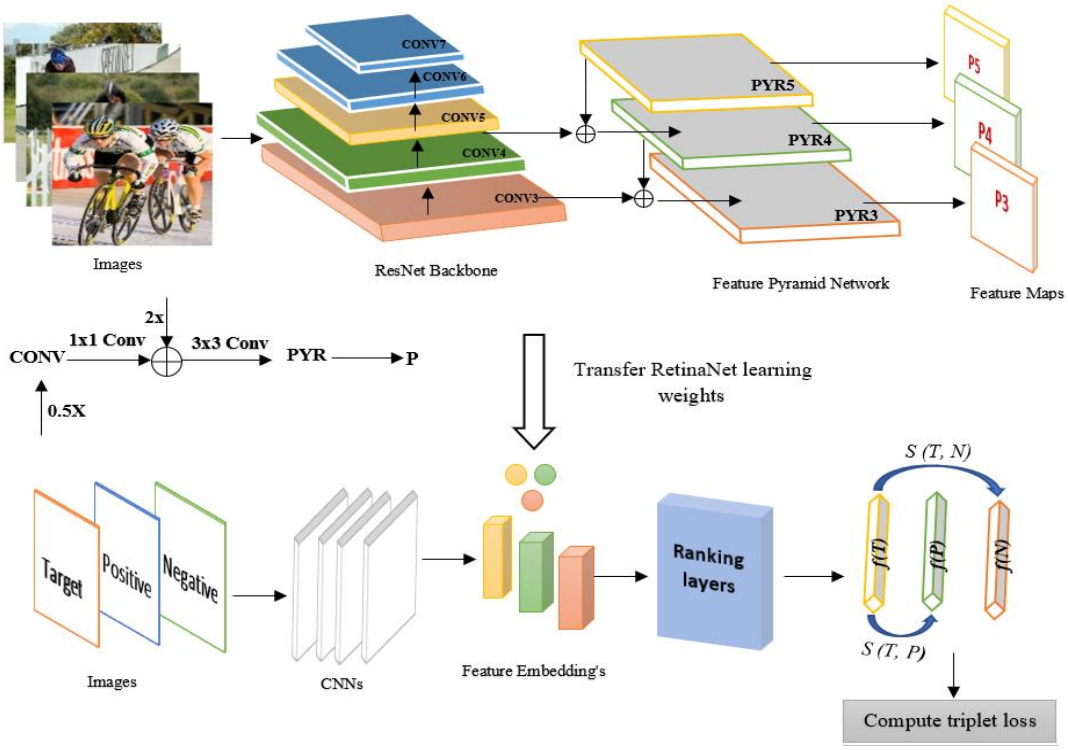

Fig 1: The architecture of the deep relevant image ranking technique based on transfer RetinaNet learning using transfer learning technique.

3.1.1 Feature Pyramid Network Extractor (FPN)

FPN is a feature extractor that enables a model to detect objects across vastly various scales [34]. Detecting small objects in image efficiently and correctly is a challenging task that RetinaNet [13] architecture is attempting to do using FPN. It constructs in the top of the backbone network (ResNet architecture) of RetinaNet architecture, which integrates deeper robust features with deeper weak features from intrinsic multi-scale feature maps of deep CNNs through lateral connections. FPN constitutes two pathways, bottom-up pathway and top-down pathway.

3.1.1.1 *Bottom-Up Pathway*

The First pathway serves to extract features of objects from high-resolution input using CNNs. Input image feedforward to ResNet backbone network generating deep semantic feature maps. Each layer in the bottom-up pathway is stacked with convolutional layers (Conv), rectified linear units (ReLU), and pooling layers (Pool) to generate feature maps from conv3 to conv7. With multiple high-level feature maps, generating the deep robust features value for each layer increases.

3.1.1.2 *Top-Down Pathway and Lateral Connections*

The second pathway is an FPN constructed on the top of the ResNet backbone network as a feature extractor to generate final feature maps utilized in prediction layers. It constructs multi-scale layers with high-resolution from the last layer in the bottom-up pathway by merging generated feature maps from ResNet with the same size of corresponding feature maps generated from FPN. They obtaining the most powerful semantic features in multi-scale feature maps. The last generated feature map layer C5 in ReseNet forward to the first layer in the top-down pathway of FPN called pyr5. The 3x3 convolutional kernel applies to pyr5, generating the first feature map p5 that will propose for prediction layers. The spatial resolution of reconstructed feature maps is up-sampled with factor X2 using the nearest neighbor up-sampling approach. Every feature map generated from the bottom-up pathway scaled using convolution kernel 1X1 to reduce dimensions of the feature map channel. Then, the lateral connection is used to merge these upsampled pyramid feature maps with corresponding downsampled feature maps in the backbone separately, constructing a top-down pathway. On each merged map, a 3X3 convolution kernel is applied for generating all final feature maps p3, p4, p5 to predict superior locations of the object. This results in stronger feature maps containing both information about high-level features leading to better object detecting results.

3.1.2 Anchor boxes

The anchors are a set of boxes that utilize to detect the presence of objects from the set of k classes in the detection area. Anchors are generated according to the dimensional (NXM) of FPN feature maps [38]. Feature maps pyramids are set of various scales. So, every feature map level is assigned to one scale. For each pixel cell of FPN maps, define anchor bounding boxes with different sizes and different aspect ratios A=NXM. An anchor bounding boxes and a ground truth box matching according to their intersection-over-union (IOU). The study used the IOU threshold of 0.5. IOU is appreciated to assign a positive label to the anchor bounding box that serves the highest IOU with ground-truth box and drive negative label when IOU is lower than a threshold. So, when their intersection-over-union is more significant than 0.5. An anchor box is matched to a ground-truth box. Anchor box is considered background and has no matching ground-truth when its IOU with any ground-truth box is below 0.4. it applied for all ground-truth boxes and then returned the matched ground-truth according to measuring the similarity.

3.1.3 The class subnetwork

It is a fully CNNs applied to every feature map generated from the FPN level, which predicts the probability of object existence for every B anchor and K object classes at each spatial location. Anchor boxes define as B=9 boxes, each having different sizes and aspect ratios and covering an area in the input feature map. Each anchor box is responsible for detecting the existence of objects from K classes in the area that it covers. This class subnetwork involves four 3x3 convolution layers with C filters for each generated map from pyramid network-level training and followed by rectified linear unit (ReLU) [39] activations functions and sigmoid activations functions. The output feature map shape would be (W, H, KB), where W and H are proportional to the width and height of the input feature map, K and B are the numbers of object class and anchor box.

3.1.4 The box regressor subnetwork

It is designed for regressing the anchors offset to nearby ground-truth objects, which utilize the anchor box to appreciate object location. The box regressor subnetwork design is almost conformable to the class subnetwork. It applied the convolution operation on every FPN feature map output parallel to the classification subnet. Except, the final convolution layer is a 3x3 convolutional layer applied with 4B filters. Therefore, the output feature map shape would be (W, H, 4B).

3.1.5 The focal loss for class imbalance

The focal loss (FL) is a procedure focussing on handling the class imbalance problem of object detection techniques. It considers the machine learning and optimization function that has been proposed for classification problems. The class imbalance problem [40] makes the training inefficient and impacts on the detection accuracy. During the training, FPN compares the candidate bounding boxes with the ground truth to classify the positive and negative examples. Most locations have easy negative examples (background classes), which have a higher classification than hard positive example foreground classes), which have a few classified locations, thus class imbalance occurred. To overcome the class imbalance problem, the loss of the output of the classification subnet is introduced and applied to all the anchors in each generated feature image. Total focal loss is generated from the summation of the focal loss over all the anchors. Focal loss scales the cross-entropy loss by enhancing the cross-entropy (CE) function. This loss is CE loss adjusting a CE that leverages the loss of easy and hard examples. So, during training, loss of easy example decreases while increasing hard example loss expressed as Equation 1,2. FL emphasizes hard positive misclassified examples that are rising during training. Thus, it prevents detectors from being accumulated by ignoring easy negative examples during training. It is expressed as Equation 3.

$$P_t = \begin{cases} P & if \ \ y = 1 \\ -p & otherwise \end{cases} \quad (1)$$

$$CE(P_t) = \propto log(P_t) \quad (2)$$

$$F_L(P_t) = -\propto (1 - P_t)^{\gamma} log(P_t) \quad (3)$$

Where $p_t \in [0, 1]$ is the probability that the model predicts positive class with ground-truth $\gamma > 0$, CE loss is dependent on the value of P_t. When P_t is larger, the weight is smaller. When P_t is smaller, the weight is large. Alpha ($\propto_t$) and (γ) gamma are two hyper-parameters introduced for moderating the weights between easy and hard examples. Alpha is a balancing parameter $\alpha \in [0, 1]$ for class 1 and $1-\alpha$ for class -1. γ is a focusing positive scale parameter. The value of the gamma effect of the modulating factor. If the value of $\gamma = 0$, the focal loss is equivalent to categorical cross-entropy, and as γ is increased, the modulating factor is increased.

Algorithm 1 describes the details of the Non-Maximum Suppression algorithm to evaluate the RetinaNet detector technique. NMS filters the proposals based on some criteria using Intersection Over Union (IOU) of the selected proposal with every other proposal. It removes the proposal with IOU smaller than a threshold.

Algorithm (1): Non-Maximum Suppression

Input : D_B ⟶ List of detection bounding box
S_C ⟶ List of detection bounding box score
D_{Bnms} ⟶ Final proposal list
Th_{NMS} ⟶ NMS Threshold

Output : D_{Bnms} ⟶ List of filtered proposals
S_{cnms} ⟶ List of filtered proposals score

```
1.  begin
2.    D_B = { d_b1, d_b2 ..... d_bx) ,  S_C = { s_c1, s_c2 ..... s_cz) ,  Th_NMS = 0.5
3.    D_Bnms <-- ∅
4.    S_cnms <-- ∅
5.    while D_B ≠ ∅ : do
6.      for d_by ∈ D_B do :
7.        F_B <-- d_by
8.        F_s <-- s_cy
9.        D_Bnms = D_Bnms ∪ F_B
10.       S_cnms = S_cnms ∪ F_s
11.       D_B = D_B − F_B
12.       for d_bz ∈ D_B do :
13.         If IOU (F_B, d_bz ) ≥ Th_NMS Then
14.           If F_s > s_cz Then
15.             D_B <-- D_B − d_bz
16.             S_C <-- S_C − S_Cz
17.
18.           End
19.         End
20.       End
21.     End
22.   End
23.   return D_Bnms , S_cnms
24. End
```

3.2 Transfer of Learning

Transfer of learning [41] is a significant challenge in many applications in the field of machine learning. Transfer of learning is the most significant progress procedure that makes all deep network requirements far more accessible than before. This study helps us to obtain a better feature object representation from the migration of prior knowledge from the trained models for deep relevant image ranking. So, it is a powerful tool that makes the deep ranking target systems more reliable and robust. Feature embeddings are transferred using the

transfer learning technique from the trained model RetinaNet (weights of all layers of convolutional features and FPN layers) to adapt the deep ranking model. The experiment results show that the use of the transfer of learning from RetinaNet source trained tasks has positive leverage on efficiency and speediness of learning for the deep ranking target task. Feature embedding generated from RetinaNet learning is the deepest and the most semantic for ranking schema than training ranking models with normalized CNNs like AlexNet. It helps in achieving the best image retrieval and precision of the relevant image ranking model.

3.3 Deep Relevant Image Ranking

Recent deep learning models have a significant impact on computer vision applications. The success of deep relevant ranking relies on learning techniques that are used to detect features embeddings from images. A deep relevant image ranking model receives features embeddings (weights) from transferring RetinaNet learning to conserve the time that waste in training ranking model with detection technique and enhancement ranking process with using the deep detection techniques. It aims to measure the ranking loss between features embeddings of images. It makes the dissimilar images far from the result forced to retrieve the best retrieval of relevant image ranking results according to the goal. It consists of two approaches, the triplet sampling approach and the ranking approach.

3.3.1 Triplet embeddings sampling

Triplets are a significant procedure of the deep image ranking model. Triplets procedure assign similar feature embedding for images in the same class and assign dissimilar feature embeddings for images in different classes. Positive image class, sample from the same class of target image. Negative image class, sample randomly from any class except the target image class. For example, when the training is for the similarity between a set of classes. If our target image is a blue bus. Then, its positive image might be a red bus, and its negative image will be a train or any image from any class of dataset except the target class. The training dataset in this study contains almost 20 thousand images. The number of all possible triplets in this dataset is approximately $(20x10^3)3 = 8x10^{12}$. Robust deep ranking model dependent on the efficient triplet sampling method of generating triplets from a large amount of training data.

3.3.2 Ranking loss

The ranking loss considers a significant distance measured approach in deep relevant image ranking models. The ranking layer appends to the end of the network that utilized the weight transferred from RetinaNet learning. This weight parsing to the network to generate feature embedding of triplets and then compute the distance of pairs of embeddings. The network generates similar embeddings with distance near to zero for the images in the same class, and dissimilar embeddings with the distance tending towards one for the images from various classes. It is responsible for computing similarity distance between all pairs of feature embeddings as at Equation 4,5 and then measure the triplet ranking loss [42] as at Equation 6. Euclidean distance [43] metric considers the most effective procedure that has been utilized to measure image similarity between pairs for relevant ranking.

$$ED\big(f(T), f(p)\big) = \sqrt{(T1 - p1)2 + (T2 - P2)2 + \cdots\cdots + (Tn - Pn)2} \quad (4)$$

$$ED\big(f(T), f(N)\big) = \sqrt{(T1 - N1)2 + (T2 - N2)2 + \cdots\cdots + (Tn - Nn)2} \quad (5)$$

$$S\,(T, P) = ED\big(f(T), f(p)\big) \quad (6)$$

$$S\,(T, N) = ED\big(f(T), f(N)\big) \quad (7)$$

$$Tl\;\;(T, P, N) = \max\{\,0,\;\; g + ED\big(f(T), f(P)\big) - ED(f(T), f(N))\} \quad (8)$$

$$where\ ED\ (f(T), (f(P) <\ ED(f(T), (f(N)\ ) \quad (9)$$

$$\forall\ (T, P, N)\ such\ that\ S(T, P) > S(T, N) \quad (10)$$

where *T*, *P*, *N* target, positive and negative images. *'TL'* is the triplet loss. '*g*' is a gap parameter that adjusts the gap between the distances of the two pair's image: *(T, P)* and *(T, N)*. 'ED' is the Euclidean distance between the two Euclidean points *f(T)*, *f(P)*, and *f(N)*. '*f*' is the feature embedding function that assigns the image to a vector. '*S*' is the similarity distance between the two images.

Algorithm 2 describes the details of the proposed deep ranking method based on transfer RetinaNet learning. We used the triplet sampling approach to generate target, positive, and negative image. Then we used the learning of RetinaNet detection technique (weights) to generate feature embeddings of all triplet images. And then, we computed the focal loss of the triplet.

Algorithm (2): Deep Relevance Image Ranking

Input : Target image
Positive Image
Negative Image
weights

Output : Triplet Ranking loss

```
begin
  for T ∈ [ T_1, T_2, T_3 .... T_n ]: do
    for P ∈ [ P_1, P_2, T_3 ... P_m ] : do
      for N ∈ [ N_1, N_2, N_3 ... N_k ] : do
        fT_1 = ComputeFeatureEmbedding 1(T_1);
        fP_1 = ComputeFeatureEmbedding 2(P_1);
        fN_1 = ComputeFeatureEmbedding 3(N_1);
        ….;
        fT_n = ComputeFeatureEmbedding n(T_n);
        fP_m = ComputeFeatureEmbedding m(P_m);
        fN_k = ComputeFeatureEmbedding k(N_k);
        for Sim_TP = ED(TP): do
          for Sim_TN = ED(TN): do
            ED(TP)_1 = ComputeEuclideanDistance1(D(fT_1), (fP_1));
            ED(TN)_1 = ComputeEuclideanDistance2(D(fT_1), (fN_1));
            …..;
            ED(TP)_l = ComputeEuclideanDistance l(D(fT_n), (fP_m);
            ED(TN)_z = ComputeEuclideanDistance z(D(fT_n), (fN_k);
            If ED(TP)_l < ED(TN)_z then
              Sim_TP > Sim_TN
              for Tl = [Tl_1, Tl_2 .... Tl_k] : do
                Tl_1 = compute loss 1(ED(TP)_1, ED(TN)_1)
                Tl_2 = compute loss 2(ED(TP)_2, ED(TN)_2)
                ….;
                Tl_k = compute loss 3(ED(TP)_l, ED(TN)_z)
                If Tl near to zero then
                  T , P is more similar
                  T , N is dissimilar
                End
              End
              return Tl
            End
          End
        End
      End
    End
  End
End
```

4. EXPERIMENTAL AND ANALYSIS

4.1 Experimental Data

In our experiments, two sets of training data are used to train our deep learning model and to learn the similarity metric for deep image ranking. The first training dataset is PASCAL VOC 2007-2012 training and validation dataset [44]. It has around 1k images in each of 20 categories. In total, there are about 20k training and validation images. This dataset is employed to learn image semantically and visually. It is used to learn the "RetinaNet" learning model and employ the parameters and features of training data for transfer learning to lean similarity metric for relevance image ranking modality. The test set of the PASCAL VOC 2007 test dataset is used to evaluate the performance of the detection technique and deep ranking technique.

The second training dataset is COCO 2017 training and validation dataset [30]. It has 118K/5k training/validation images with 90 categories. This dataset is employed to learn image semantically and visually. It is used to learn the "RetinaNet" learning model and

employs the parameters and features of training data for transfer learning to lean similarity metric for relevance image ranking modality. The validation set of the COCO 2017 Val dataset is used to evaluate the performance of the detection technique and deep ranking technique.

4.2 Detection Performance

In this subsection, the experimental results are summarized as follows. All experiments were implemented based on the deep learning framework Keras and executed on a PC with an Intel single Core i7. Our training and testing were running on GPU NVIDIA GeForce GTX 1050, with 4GB memory using weight decay of 0.0001 and a momentum of 0.9. The mean average precision (MAP) of RetinaNet is reported by using the IOU threshold set as 0.5.

4.2.1 VOC2007-2012 Dataset

The first training dataset is VOC 2007-2012 Train and Validation dataset. We train the RetinaNet models for 90 epochs for 108 hours using ResNet50. Show the training result in (Table 1). At first, this study investigates the impact of using FPNs of the top-down and bottom-up pathways on detection accuracy. The experiments are conducted using the depth of the network of 50 layers. As listed in Table 1, to further validate the advantage of the proposed RetinaNet, the accuracy is compared between different methods. This study manifests that the proposed RetinaNet detection obtains a higher accuracy for all categorize of the dataset. Testing is applied to PASCAL VOC 2007- testset (see Table 2). RetinaNet proved that it could localize the large objects and recognize the small objects accurately. In (Figure 3) shows the output of RetinaNet detection using the PASCAL VOC 2007-2012 dataset. Our method detects objects of multi-scales using a feature pyramids network. A score threshold of 0.5 is used to display these images.

Table 1. The learning result of our RetinaNet detector technique and current detection technique.

Detection procedure	Input size (pixels)	Train	MAP
Faster RCNN [24]	600 X 800	VOC 2007+2012	88.26
SSD [27]	300 X 300	VOC 2007+2012	88.32
RetinaNet [13]	800 X 800	VOC 2007+2012	97.2

Table 2. The comparison between our detection technique (RetinaNet) and current techniques in the same dataset.

Classes	Dataset	Method					
		Fast RCNN [23]	**Faster RCNN** [24]	**SSD** [27]	**YOLOv2** [29]	**Improved YOLOv2** [32]	**RetinaNet** [13]
aero	VOC 07-12	77.0	76.5	75.5	87.9	88.2	**89.0**
Bike		78.1	79.0	80.2	87.5	87.3	84.5
Bird		69.3	70.9	72.3	78.2	79.3	**79.7**
Boat		59.4	65.5	66.3	61.5	61.3	**70.2**
Bottle		38.3	52.1	47.6	57.9	61.0	53.6
Bus		81.6	83.1	83.0	84.9	85.2	83.7
Car		78.6	84.7	84.2	82.9	81.3	**85.8**
Cat		86.7	86.4	86.1	90.6	90.0	**94.4**
Chair		42.8	52.0	54.7	54.9	58.8	**60.2**
Cow		78.6	81.9	78.3	83.6	83.4	**84.4**
Table		68.9	65.7	73.9	66.5	68.3	67.7
Dog		84.7	84.8	84.5	90.0	89.3	90.8
Horse		82.0	84.6	85.3	85.2	84.6	**90.3**
Mobike		76.6	77.5	82.6	85.8	85.5	**85.9**
Person		69.9	76.7	76.2	82.9	82.7	**84.8**
Plant		31.8	38.8	48.6	54.2	54.0	**60.3**
Sheep		70.1	73.6	73.9	78.9	80.2	77.4
Sofa		74.8	73.9	76.0	65.2	66.3	75.4
Train		80.4	83.0	83.4	87.3	87.6	**90.1**
Tv		70.4	72.6	74.0	69.8	71.8	**81.1**
MAP		**70.0**	**73.2**	**74.3**	76.8	77.3	**79.5**

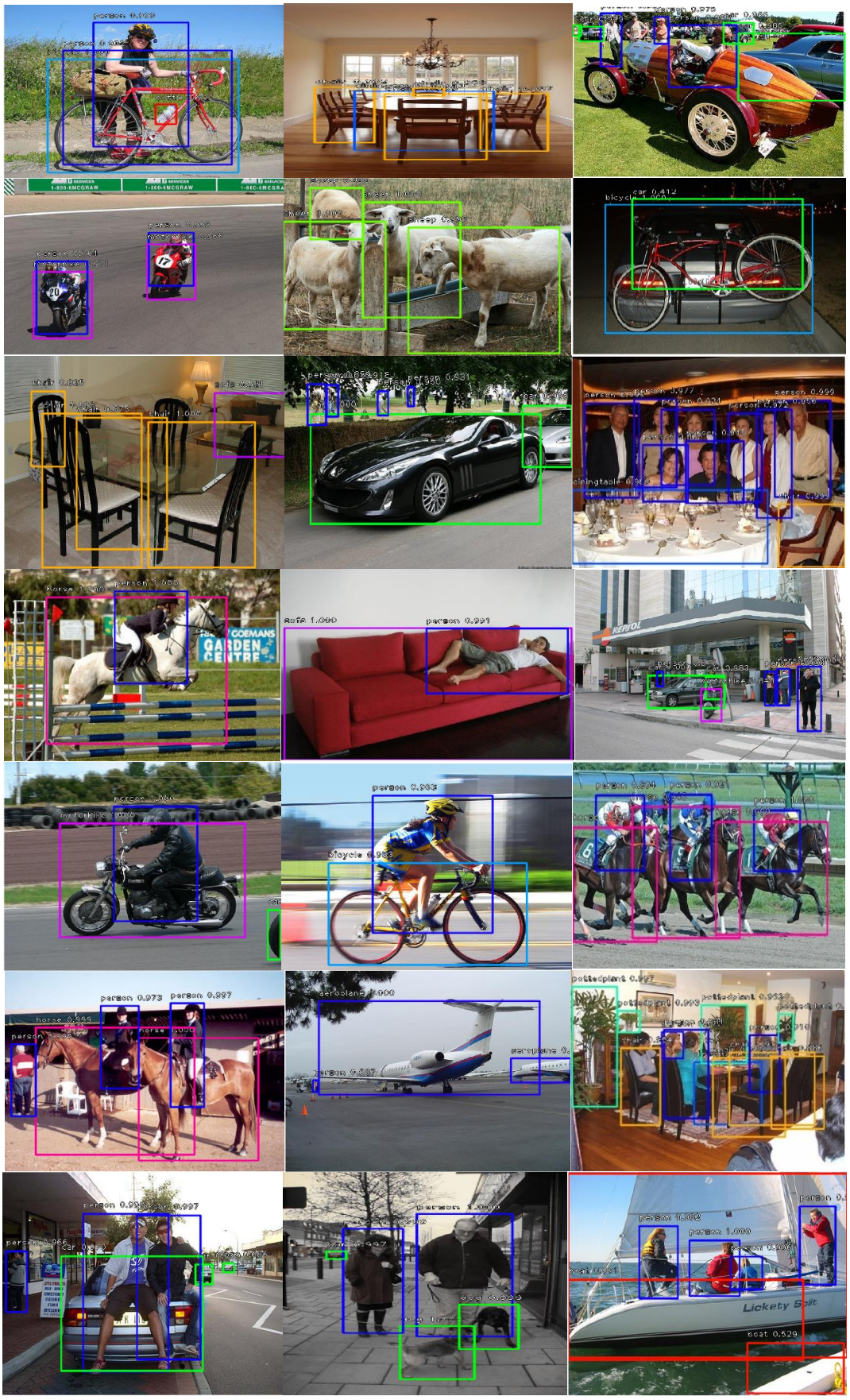

Fig 2: Examples of object detection result on the PASCAL VOC 2007 test set using the RetinaNet learning model.

4.2.2 COCO 2017 Dataset

The second training dataset is COCO 2017 Train and Validation dataset. We train the RetinaNet models for 40 epochs for 86 hours using ResNet50. Show the testing result in (Table 3). The experiments are conducted using the depth of the network of 50 and a different scale. In (Figure 4) shows the output of RetinaNet detection using the COCO dataset. Our method detects objects of multi-scales using a feature pyramids network. A score threshold of 0.5, 0.4 is used to display these images

Table 3. Average Precision (AP) result of RetinaNet object detector learning on COCO 2017 validation dataset

Depth	scale	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
50	224	26.0	40.9	27.2	10.8	28.7	38.0
50	800	33.1	50.4	35.3	14.2	36.3	49.6

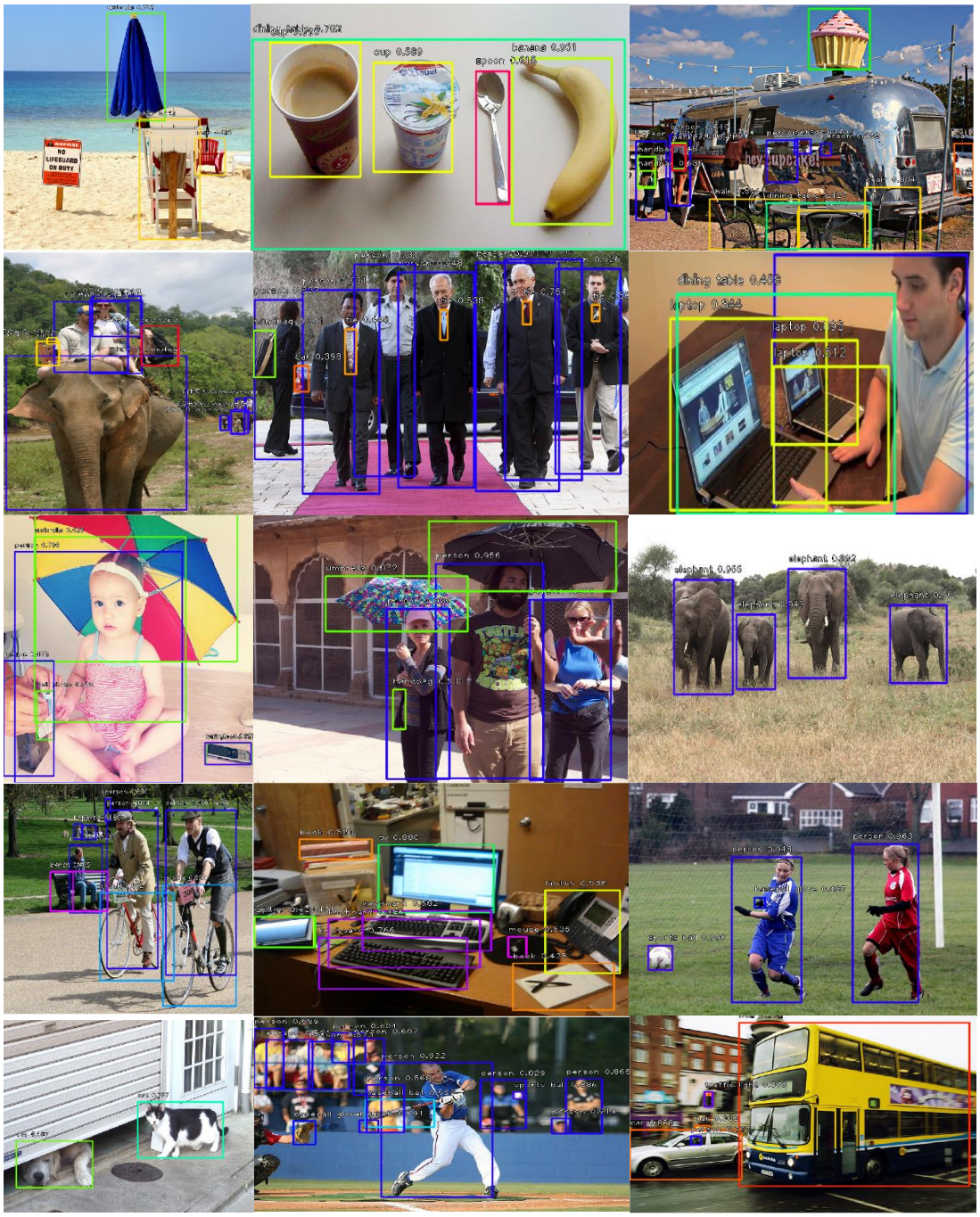

Fig 3: Examples of object detection result on the COCO 2017 test set using the RetinaNet learning model.

4.3 Deep Ranking Performance

In this subsection, this study designs an extensive set of experiments for evaluating the performance of a deep relevance ranking model. All experiments were implemented based on the deep ranking framework Keras and executed on a PC with an Intel single Core i7. Our training and testing were running on GPU NVIDIA GeForce GTX 1050, with 4GB memory.

At first, this study examines how the deep CNN (RetinaNet) model performed for deep relevance image ranking tasks on the same dataset that was used to train the learning model. Then this study investigates the impact of using the triplet approach and ranking approach on ranking accuracy. Experiments are conducted using the depth of a network of 50 layers. As listed in (Table 3). to further validate the advantage of the proposed deep relevance ranking, This study compares the accuracy between different methods. It manifests that the proposed deep relevance ranking model obtains a higher accuracy. In (Figure 4). show the top 5 images and bottom five images results of relevance deep ranking model based on RetinaNet learning for six target images. Each row is Target. Some example of test images is selected and find its five top-most similar images (nearest neighbors), and bottom five similar images using our learned embeddings from RetinaNet architecture.

Table 4. Accuracy for our deep relevant image ranking technique and current technique of ranking.

Procedure	Accuracy
OASIS [16]	82.5
ConvNet [17]	82.8
Deep Ranking [15]	85.7
Deep Ranking based on transfer RetinaNet detector learning	86.8

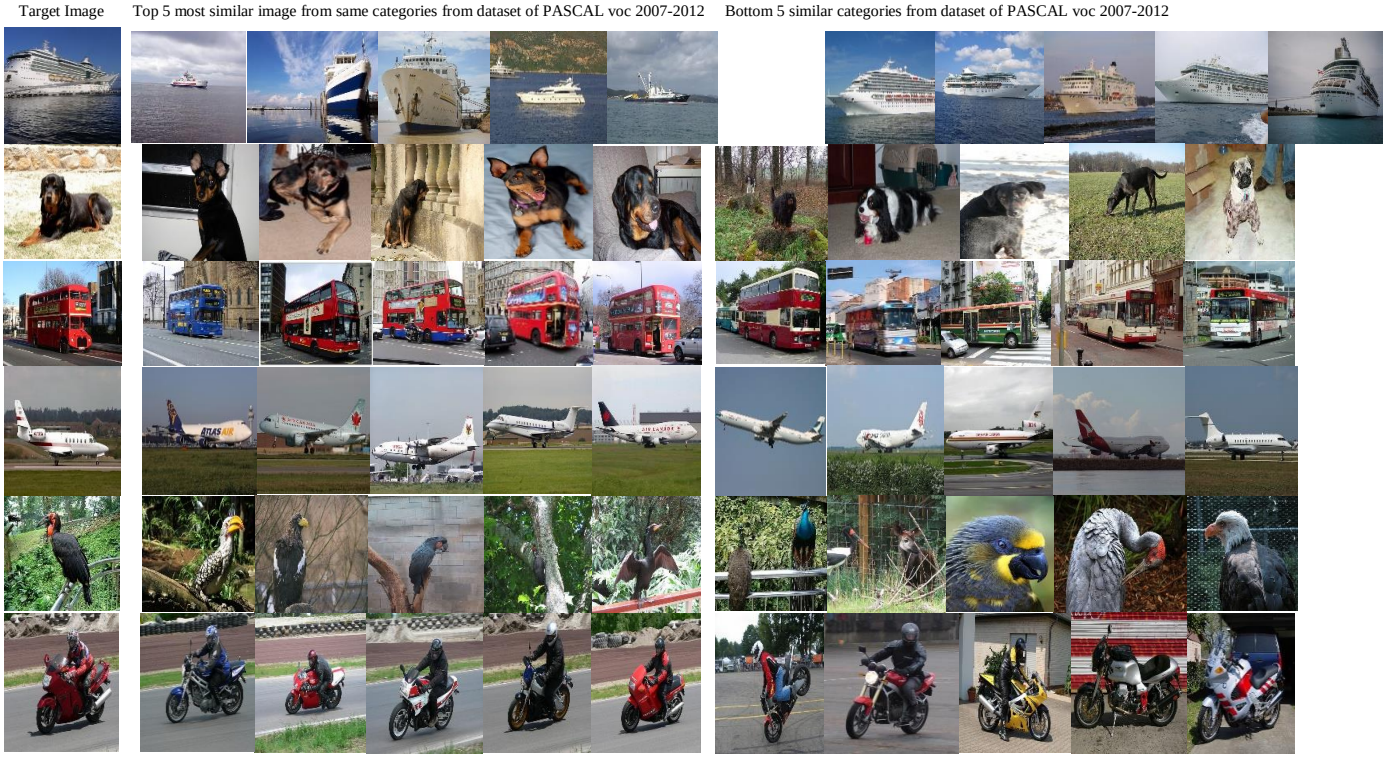

Fig 4: Ranking results for six target images.

4.4 Evaluation Criteria

In the deep learning field, the most common metrics to evaluate object detection method and ranking relevancy method are accuracy (MAP), precision, and recall. Accuracy is the ratio of correctly retrieving relevant images to total image and can be calculated according to Equation 11. Precision measures the ability of the model to retrieve the only image that is relevant and can be calculated according to Equation 12. Recall measuring the ability of the model to retrieve all the images that are relevant and can be calculated according to Equation 13.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Example} \quad (11)$$

$$Precision = \frac{True\ Positives}{Total\ Predicted\ Positive} \quad (12)$$

$$Recall = \frac{True\ Positives}{Total\ Actual\ Positive} \quad (13)$$

The performance of our proposed model is evaluated before transfer RetinaNet learning and after transfer RetinaNet learning, as shown in (Table 4). and (Figure 5). It can be observed that the accuracy of the proposed model after transfer RetinaNet detection learning is higher than the accuracy before transfer RetinaNet detection learning. The rise in ranking accuracy after transfer learning drive to retrieve the best relevant images according to the target image.

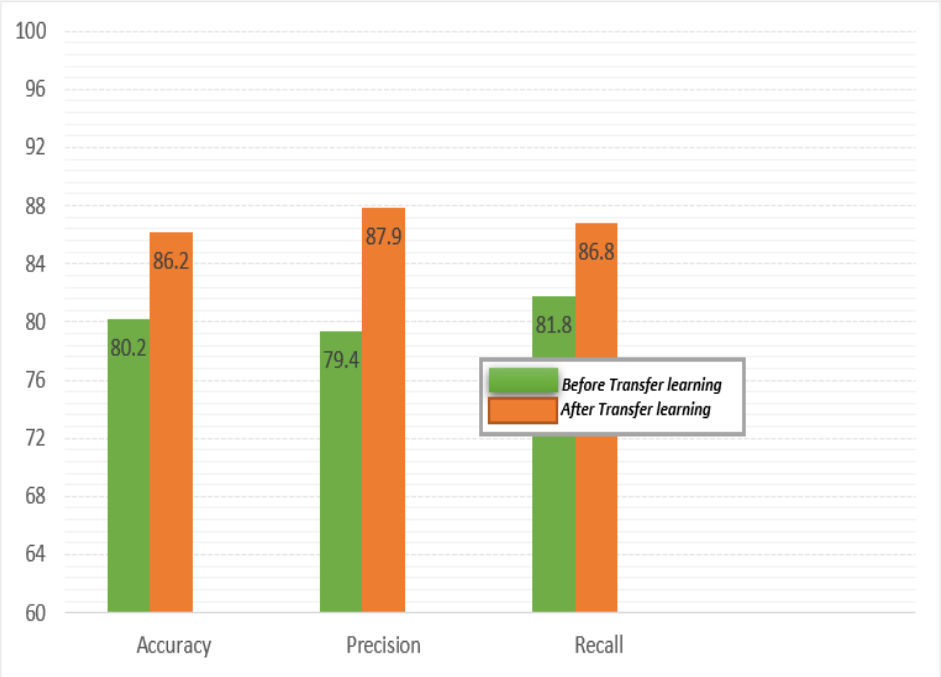

Fig 5: The test results of deep image ranking before and after transfer RetinaNet learning.

Table 4. The result of Evaluation Metric of deep relevant image ranking model before and after transfer RetinaNet learning.

Method	Evaluation Metric		
	Precision	Recall	Accuracy
Proposed Deep Ranking before Transfer RetinaNet Learning	80.2	79.4	81.8
Proposed Deep Ranking after Transfer RetinaNet Learning	86.2	87.9	86.8

6. CONCLUSION

Deep Learning is discovered to be used as a new tool that helps in solving the limitation of traditional machine learning techniques. It is employed to analyze the massive amount of data image. The recent advances in deep learning and machine learning technology are moving toward making computer vision processes more efficient and accurate. Advanced deep learning techniques are revealed to do well at extracting the deepest features of the large-scale data image. The generated features by deep learning techniques can be stimulated to act as a knowledge for other target domains like information retrieval. This paper proposed a system that intends to retrieve the best relevant image according to the target image. This study defeats the limitation of the current deep image ranking technique. It is prompted by the most powerful object detection technique that analyzes the massive volume of images data and extracts hard examples using FPN

feature extractor. The proposed system is a deep relevant image ranking model based on transfer RetinaNet learning. This study trains RetinaNet on PASCAL VOC 2007-2012 trainval image dataset based on feature pyramids network technique that focuses on small objects to extract the deepest learned features embeddings (weights). The transfer learning schema is utilized to transfer feature embeddings of RetinaNet learning (weights) for achieving the best relevant retrieval. Deep relevance ranking architecture introduces the triplet network procedure to learn useful representation explicitly. It adds a ranking layer that utilizes euclidean distance as a similarity metric on pairs of feature embeddings of RetinanNet learning. Then, it computes triplet loss for retrieving the most relevant images similar to Target's image. The RetinaNet method generated the best features embeddings needed in retrieval schema. It beat the current problem of detection methods. It uses the feature pyramid network for feature extraction, unlike traditional machine learning techniques, which use hand-crafted features technique. It is based on one stage detector for detection and classification, unlike a faster RCNN technique based on two-stage detector techniques making detection slow. It beats the Class imbalance problem using an anchor box, unlike the SSD technique that generates a few bounding boxes implicating objects. The results showed that the ranking precision when utilizing RetinaNet learning is the best than the previous ranking model based on the weak machine learning technique. In future work, we will investigate more advanced ways to enhance the learning of the RetinaNet detector. We will use a huge dataset for more in-depth empirical studies to give more insights for further enhancing the result of deep image ranking retrieval.

7. REFERENCES

[1] Du, X., Cai, Y., Wang, S., and Zhang, L. 2016. Overview of deep learning. In 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, 159-164.

[2] Shrestha, A., and Mahmood, A. 2019. A.: Review of Deep Learning Algorithms and Architectures. J. IEEE Access. 22 (April.2019), 53040-53065.

[3] Louridas, A., and Ebert, C. 2016. Machine Learning. J. IEEE Software. 24 (August.2016), 110–115.

[4] Angra, S., and Ahuja, S. 2017. Machine learning and its applications: International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), 57-60.

[5] Singh, A., Thakur, N., and Sharma, A. 2016. A Review of Supervised Machine Learning Algorithms. In 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 1310-1315.

[6] Aloysius, N. and Geetha, M. 2017. A Review on Deep Convolutional Neural Networks. In International Conference on Communication and Signal Processing (ICCSP), 588-592.

[7] Barbu, T. 2009. Content–based Image Retrieval using Gabor Filtering. In 20th International Workshop on Database and Expert Systems Application, 236-240.

[8] Lowe, D.G. 1999. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, 1150–1157.

[9] Dalal, N., and Triggs, B. 2005. Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 886–893.

[10] Nanni, L., Ghidoni, S., and Brahnam, S. 2017. Hand-crafted vs. non-handcrafted features for computer vision classification. J. Pattern. Recognition. (November.2017), 158-172.

[11] Pan, S.J., Yang, Q. 2010. A Survey on Transfer Learning. J. Knowl and Data. Eng. 10 (October.2010), 1345-1359.

[12] Shaha. M., and Pawar, M. 2018. Transfer Learning for Image Classification. In 2nd International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, 656-660.

[13] Lin, T.Y., Goyal, P., Girshick, R., He, K., and Dollar, P. 2017. Focal loss for dense object detection. In IEEE International Conference on Computer Vision, 2980–2988.

[14] Sun, J., Cai, X., Sun, F., and Zhang, J. 2016. Scene image classification method based on Alex-Net model In 3rd International Conference Informative and Cybernetics for Computational Social Systems. Jinzhou, 363-367.

[15] Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., James, P., Chen, P., and Wu, Y. 2014. Learning Fine-grained Image Similarity with Deep Ranking. In proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, 1386-1393.

[16] Chechik, G., Sharma, V., Shalit, U., and Bengio, S. 2010. Large scale online learning of image similarity through ranking. J. Mach. Learn. Res. 3 (January.2010), 1109–1135.

[17] Krizhevsky, A., Sutskever, I., and Hinton, G.E. 2017. ImageNet classification with deep convolutional neural networks. J. Commun. ACM. (June.2017) , 84-90.

[18] Girshick, R., Donahue, J., Darrell, T., and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, 580–587.

[19] Uijlings, J.R.R., Sande, K.E.A., and Smeulders, A.W.M. 2013. Selective Search for Object Recognition. J. Computer. Vision. (September.2013), 154–171.

[20] Mavroforakis, M.E., and Theodoridis, S. Support Vector Machine (SVM) classification through geometry. 2005. In 13th European Signal Processing Conference, Antalya, 1-4.

[21] Sun, X., Liu, L., Wang, H., Song, W., and Lu, J. 2015. Image Classification via Support Vector Machine. In 4th International Conference on Computer Science and Network Technology (ICCSNT), 485-489.

[22] He, k., Zhang, x., Ren, s., and Sun, j. 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. In European Conference on Computer Vision, 346-361.

[23] Girshick, R. 2015. Fast rcnn. In IEEE International Conference on Computer Vision (ICCV), 1440–1448.

[24] Shaoqing, R., Kaiming, H., Ross, G., Jian, S. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. J. Pat. Anal and Mach. Intel. (June.2015), 1137-1149.

[25] Bappy, J.H., and Roy-Chowdhury, A.K. 2016. CNN Based Region Proposal for Efficient Object Detection. IEEE International Conference on Image Processing (ICIP) , Phoenix, 3652-3662.

[26] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 779-788.

[27] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., and Berg, A.C. 2016. Ssd: Single shot multibox detector. In European Conference on Computer Vision, 21–37.

[28] Buda, M., Maki, A., and Mazurowski, M.A. 2018. A systematic study of the class imbalance problem in convolutional neural networks. J. Neu. Net. 13 (October.2018), 249-259.

[29] Redmon, J., and Farhadi, A. 2017. YOLO9000: Better, Faster, Stronger. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), USA, 6517-6525.

[30] Lin, T-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll' ar, P., and Zitnick, C.L. 2014. Microsoft coco: Common objects in context. In European Conference on Computer Vision, 740–755. Springer

[31] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. International J. Compu.Vision (IJCV).

[32] Li, R., and Jun, Yang. 2018. Improved YOLOv2 Object Detection Model. IEEE 6th International Conference on Multimedia Computing and Systems (ICMCS). 8 (Nov.2018)

[33] Kaiser, L., Gomez, A.N., and Chollet, F. 2017. Depthwise Separable Convolutions for Neural Machine Translation. arXiv preprint arXiv:1706.03059.

[34] Lin, S.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. 2017. Feature pyramid networks for object detection. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2117–2125.

[35] Qiu-Zhao, H., Zheng, P., Xu, T., and Wu, X. 2019. Object Detection With Deep Learning: A Review. J. IEEE Transactions. Neu.Net and Learn.Sys. 28 (January.2019), 3212-3232

[36] Du, J. 2018. Understanding of Object Detection Based on CNN Family and YOLO. J. Phys. 25 (February.2018).

[37] He, K., Zhang, X., Ren, S., and Sun, J. 2016. Deep Residual Learning for Image Recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.

[38] Zhong, Y., Wang, J., Peng, J., and Zhang, L. 2018. Anchor Box Optimization for Object Detection. J. ArXiv. 2 (Dec.2018).

[39] Bircanoğlu, C., and Arıca, N. A comparison of activation functions in artificial neural networks. 2018. In 26th Conference of Signal Processing and Communications Applications, Izmir, 1-4.

[40] Oksuz, K., Cam, B.C., Kalkan, S., and Akbas, E. 2019. Imbalance Problems in Object Detection: AReview. J. Compu. Vision. 31 (Aug.2019).

[41] Shaha, M., Pawar, M. 2018. Transfer Learning for Image Classification. In Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 656 -660.

[42] Ge, W., Huang, W., Dong, D., and Scott, M.R. 2018. Deep Metric Learning with Hierarchical Triplet Loss. In European Conference on Computer Vision, 272-288.

[43] Khosla, G., Rajpal, N., and Singh, J. 2015. Evaluation of Euclidean and Manhanttan Metrics in Content Based Image Retrieval System. In 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 12-18.

[44] Everingham, M., Gool, L.V., Williams, C.K.L., Winn, J., and Zisserman, A. 2010. The PASCAL Visual Object Classes (VOC) Challenge. J. Compu. Vision. (June.2010), 303–338.