IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 17, 2021, accepted September 18, 2021, date of publication September 22, 2021, date of current version October 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3114968

A Hybrid Approach for Semantic Image Annotation

ARDA SEZEN[1], CIGDEM TURHAN[2], AND GOKHAN SENGUL[3]
[1]Department of Software Engineering, OSTIM Technical University, 06374 Ankara, Turkey
[2]Department of Software Engineering, Atilim University, 06830 Ankara, Turkey
[3]Department of Computer Engineering, Atilim University, 06830 Ankara, Turkey

Corresponding author: Arda Sezen (arda.sezen@ostimteknik.edu.tr)

ABSTRACT In this study, a framework that generates natural language descriptions of images within a controlled environment is proposed. Previous work on neural networks mostly focused on choosing the right labels and/or increasing the number of related labels to depict an image. However, creating a textual description of an image is a completely different phenomenon, structurally, syntactically, and semantically. The proposed semantic image annotation framework presents a novel combination of deep learning models and aligned annotation results derived from the instances of the ontology classes to generate sentential descriptions of images. Our hybrid approach benefits from the unique combination of deep learning and semantic web technologies. We detect objects from unlabeled sports images using a deep learning model based on a residual network and a feature pyramid network, with the focal loss technique to obtain predictions with high probability. The proposed framework not only produces probabilistically labeled images, but also the contextual results obtained from a knowledge base exploiting the relationship between the objects. The framework's object detection and prediction performances are tested with two datasets where the first one includes individual instances of images containing everyday scenes of common objects and the second custom dataset contains sports images collected from the web. Moreover, a sample image set is created to obtain annotation result data by applying all framework layers. Experimental results show that the framework is effective in this controlled environment and can be used with other applications via web services within the supported sports domain.

INDEX TERMS Semantic image annotation, picture interpretation, ontology.

I. INTRODUCTION

Deep learning is a fascinating subfield of artificial intelligence especially considering the outcomes. It can be used in many fields including image and speech recognition, medical diagnosis, automated trading strategies, learning associations, and fraud detection [1]–[4]. Although the results are impressive, it is obvious that deep learning has serious difficulties in terms of the learning process when compared to human learning. In connection with grasping the separating principles, experiments demonstrate that human subjects obtain more successful results than visual recognition models [5], [6]. On the other hand, these models achieve outstanding accuracy for restricted tasks such as face detection or iris recognition [7], [8]. Most studies in visual recognition have focused on labeling images with a fixed set of categories. This is similar to the unconnected pieces of a puzzle that need to be properly aligned to visualize the whole picture.

Associated objects are the key to generate a description that is longer than a single label. There are significant studies in literature that aim to address the challenges of generating image descriptions [9], [10]. The study by Karpathy and Fei-Fei [6] differs from these models by proposing a model that simultaneously reasons about the contents of images and their natural language representations to generate the descriptions of the image regions. However, this requires a comprehensive training process on large image-sentence datasets, which are usually created with manual annotation showing variety from annotator to annotator. Moreover, the immense computational power required to train a model such as this one is not cost-effective and limits the usage of hardware at hand [11], [12].

In this work, we aim to extract textual descriptions from visual images of selected sports domains (tennis, baseball,

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

and skiing). The primary challenge towards this goal is that datasets often use generic labels for objects. For example, the balls used in different branches of sports (e.g. tennis balls) are labeled generically as sports balls. Secondly, some objects may be intertwined in the same location in the pictures and may cause false prediction results obtained from the fully connected layers of the class subnet. The final challenge is concerned with obtaining the correct annotations which are based on the semantic queries related to the detected objects, their frequency distribution as well as their permutations. Specifically, the contributions of this study are as follows:

- A framework is proposed firstly to identify the generic objects in an image and then to find their specific annotations with the aid of an ontology, i.e., a generically detected ''sports ball'' is distinguished as a ''tennis ball''.
- A sports ontology is developed which provides useful annotations obtained from the relations between various classes and their individuals to depict the images.
- Finally, keyword combination search is performed by examining the image objects and ontology results, returning the relevant image annotations.

Our work distinguishes the objects and their properties, i.e. color and coordinates in an image using artificial intelligence models. At this stage, the detected objects from an image can be generic and no conceptual relation exists among them. The sports ontology is essential in the framework to convert the generic labels into specific labels, and to obtain annotations based on the relationships between the ontology classes. Compared with the previously developed image annotation systems, the proposed system is ontology-driven, scalable, intelligent, and efficient in terms of the training requirements.

The remainder of this paper is organized as follows. Section II reviews the background information on image annotation and its challenges concerning neural networks and ontologies. Section III introduces the layers of the proposed framework. Section IV is dedicated to the experiments and evaluation of the system from the component as well as the whole system perspective. Section V presents discussions followed by the conclusions and future work in Section VI.

II. IMAGE ANNOTATION

A. TYPES OF IMAGE ANNOTATION

Our framework utilizes the bounding boxes approach and anchor boxes as a set of predefined bounding boxes to start the annotation process. This approach is high-level and is adopted to label objects by many previous studies. Various platforms already exist for image labeling tasks. One conspicuous example is Amazon Mechanical Turk (MTurk). Chen *et al.* [13] studied the automatic labeling of 3D bounding boxes (3D Cuboids approach) in the context of autonomous driving. Our work also utilizes the polygons approach for the objects in an image where the bounding boxes cannot be applied to prepare the custom image dataset. Some of the approaches do not treat object-segmentation as a pixel-problem (semantic segmentation is the process of associating every single pixel in an entire image with a tag) instead, they cast it as a polygon prediction task [14], [15].

B. CHALLENGES OF IMAGE ANNOTATION

Post-production of an image can be counted as one of the major issues in this field since the required information is applicable only during production time. Generic annotation is another challenge since without a clear aim, it is ineffective in annotating images. With the addition of automatic annotation, this phenomenon is known in the literature as the semantic gap. A semantic gap refers to the difference between the high-level content descriptions required by the applications and the low-level features provided by the image analysis tools [16].

C. NEURAL NETWORKS IN IMAGE ANNOTATION

The literature consists of many examples for the textual representation of images with neural networks. In [6], Karpathy and Fei-Fei present an alignment model based on a combination of Convolutional Neural Networks (CNNs) over image regions and bidirectional Recurrent Neural Networks over sentences. Socher *et al.* [17] propose a DT-RNN model that uses dependency trees to embed sentences into a vector space to retrieve images. Kiros *et al.* [18] focus on an encoder-decoder pipeline to unify joint image-text embedding models with multimodal neural language models. Finally, Krishna *et al.* [19] collect dense annotations of objects, attributes, and relationships within each image for models that need to understand the interactions and relationships between the objects in an image.

D. ONTOLOGIES IN IMAGE ANNOTATION

The image annotation domain also includes studies presenting an ontology-based solution. Gu *et al.* [20], Bannour and Hudelot [21], and Baier *et al.* [22] present automatic approaches together with ontology usage. Im and Park [23] propose a semi-automatic approach for image annotation using semantic relationships between image tags. In their study, Franzoni *et al.* [24] focus on the retrieval aspect of images via context-based semantic similarity.

Also in [25] and [26], the researchers present techniques for both annotation and retrieval processing of an image.

The literature provides many solutions using neural networks and ontologies separately. On the other hand, the combination of both is rather scarce in the image annotation domain. The polytomous technologies behind these fields and the unique ways of integrating them are the root causes of why novel solution alternatives can still be produced today. The proposed framework is one of these solutions which will be detailed in the following section.

III. FRAMEWORK OF THE SYSTEM

In this section, the framework layers are introduced.

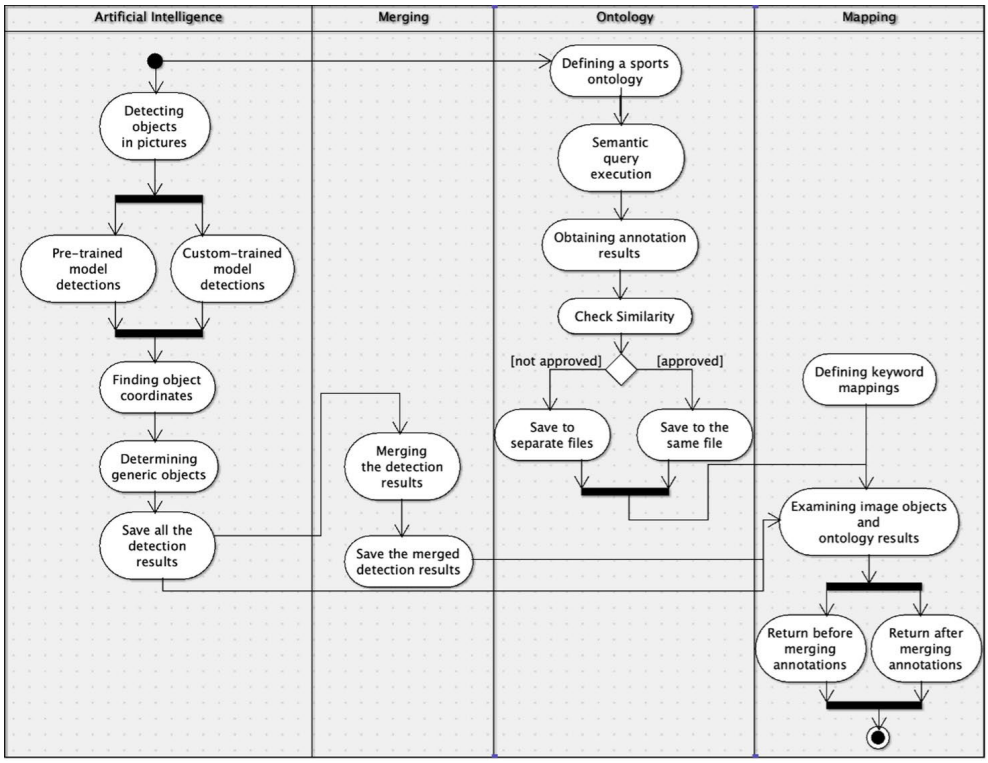

FIGURE 1. The activity diagram of the framework.

A. OVERVIEW

The multi-tier framework consists of the following activities as illustrated in Fig. 1. AI layer is placed at the bottom of the framework to detect the objects in pictures using the pre-trained model and custom-trained models. Both models are RetinaNet using residual network (ResNet-50) and feature pyramid network (FPN) as a backbone but they differ in terms of the training process and the datasets used in the training. Also in this layer, the framework can determine which of the detected objects are generic and return the coordinates of all detected objects.

The merging layer is responsible for merging the detected objects from the prediction results of both models by comparing the coordinate and probability values. This way, we can compare the annotation results obtained from the sports ontology from two different perspectives. One perspective adopts the usage of detected generic objects to specialize them and retrieve annotation results in the ontology layer. The other benefits from the training process with some specialized classes and as a result adopts the usage of some specific object labels within the semantic queries to understand the variance between the annotation results, if any. Finally, the detected objects of an input image and the ontology results are examined at the top layer (mapping layer) for the sole purpose of obtaining the correct annotations of an image. The input/output parameters to and from the models are handled with a RESTful service as illustrated in Fig. 2.

B. DETECTING OBJECTS IN PICTURES

The prediction process of our framework is performed using RetinaNet which is a more advanced object detector compared to the other object detectors mentioned in the literature. It was originally proposed by Facebook AI Research (FAIR), and today there are several versions available which differ based on the backbone usage and model layer operations.

CNN's have already been in use for a long time to extract features with the convolution layers, and to execute the process of striding a small kernel over a target array, obtaining the sum of element-wise multiplication between the kernel and a subset of equal size of the target array at that location [27], [28]. These layers need to be placed on top of each other to learn complex features, such as a nose, ear, eye, etc., without knowing what they really are. They just learn to detect a feature by processing many samples.

Therefore, achieving deeper networks has become a must to overcome complex computer vision tasks. However, the computation also becomes complex when layers are added on top of layers creating a network with many layers [29], [30]. To reduce computational complexity, we adopted a 50-layer deep model (ResNet-50) in the RetinaNet backbone which uses a residual block instead of plain layers as the main base element. This allows the flow of information from the initial layers to the final layers using shortcut connections to skip some layers, and to perform identity mapping.

Artificial neural networks are trained on training sets with a set of weights. During the training process, these weights

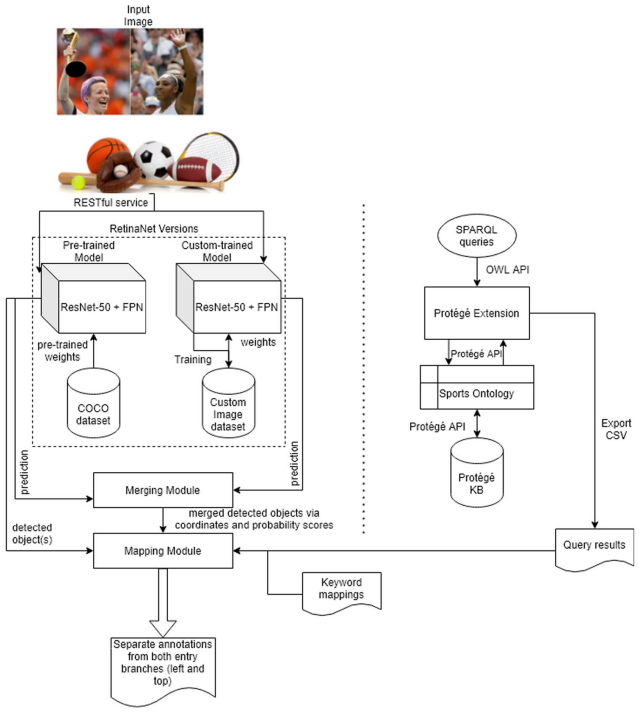

FIGURE 2. The proposed framework architecture.

are optimized and used as one of the basic components of artificial neurons before predicting the final value. In the case of the pre-trained model, we used already adjusted weights obtained from the training process of the 2014 version of the COCO dataset using training and testing images with respective object labels. We also need to see how well the custom-trained model trained itself through forward and backward propagation until reaching some minima for error value. The errors can be found via forward propagation through the activation function:

$$o^k := f(M^k + W^{k-2,k}.o^{k-2}) \tag{1}$$

where, o^k are the outputs of neurons in layer k, f is the activation function for layer k, $W^{k-1,k}$ is the weight matrix for neurons between layer k – 1 and k, and

$$M^k = W^{k-1,k}.o^{k-1} + p^k \tag{2}$$

Backpropagation learning is required to update new values of weights by using an extension to the stochastic gradient descent algorithm for both the normal and the skip paths. In this structure, the residual block notation can be expressed as follows:

$$\begin{bmatrix} A \times A, & C_1 \\ B \times B, & C_2 \end{bmatrix} \times \mathrm{N} \tag{3}$$

where $A \times A$ and $B \times B$ specify the size of the kernel used in that layer. In their study of deep residual learning for image recognition, He *et al.* [31] call them filters. C_1 and C_2 refer to the number of channels in that convolutional layer, and N is the number of times this block is repeated for that residual layer. The ResNet model used in this study is distinguished

TABLE 1. The 50-layer architecture of ResNet.

Layer	Residual Block
Conv2_x	$\begin{bmatrix} 1\text{x}1, & 64 \\ 3\text{x}3, & 64 \\ 1\text{x}1, & 256 \end{bmatrix}$ x 3
Conv3_x	$\begin{bmatrix} 1\text{x}1, & 128 \\ 3\text{x}3, & 128 \\ 1\text{x}1, & 512 \end{bmatrix}$ x 4
Conv4_x	$\begin{bmatrix} 1\text{x}1, & 256 \\ 3\text{x}3, & 256 \\ 1\text{x}1, & 1024 \end{bmatrix}$ x 6
Conv5_x	$\begin{bmatrix} 1\text{x}1, & 512 \\ 3\text{x}3, & 512 \\ 1\text{x}1, & 2048 \end{bmatrix}$ x 3

from the other ResNet models, such as 34-layer ResNet, within the layers below, see Table 1.

To overcome the degradation problem which usually appears while trying to achieve deeper artificial neural networks, preventing the accuracy to get saturated very quickly is essential. Therefore, we adopted a residual learning approach in the backbone to let every few stacked layers fit a residual mapping instead of an underlying mapping. The main difference between the residual blocks and plain blocks is the usage of skip connections and performing identity mapping as illustrated in Fig. 3.

Also, the ResNet-50 architecture differentiates from the layer depth when compared to the less layered ResNet architectures. Each ResNet block is two layers deep in the 34-layer ResNet, but in the ResNet-50, each ResNet block is three-layer deep as shown in Table 1.

In terms of identity mappings, the easier approach is driving the weights of the multiple non-linear layers towards zero instead of finding the identity mappings from the stack of non-linear layers due to the usage of the non-linear CNN layers stack. At the same time, 1×1 convolutions are applied more than any other type of convolutions to make the residual function and identity as the same dimension.

The positions in an image where object detectors are applied are the main distinctions between various object detectors. Two types exist based on where the classifier is applied. If it is applied to a sparse set of candidate object locations, the prediction results can be achieved faster. On the other hand, if it is applied over a regular, dense sampling of possible object locations, higher accuracy can be obtained.

The extreme foreground-background class imbalance during the training of dense detectors is stated as the root cause behind the performance results in [32]. A new type of loss called focal loss was proposed by the same researchers to reshape the standard cross-entropy loss by adding a factor to down-weigh the loss assigned to well-classified instances. In this way, the vast number of easy negatives which cause the detector to be overwhelmed during the training process can be prevented. Our study benefits from this approach with the RetinaNet. It is a unified network composed of multiple networks. The Feature Pyramid Network (FPN) from [33] is used for computing a convolutional feature map over an entire input image in the RetinaNet, see Fig. 4.

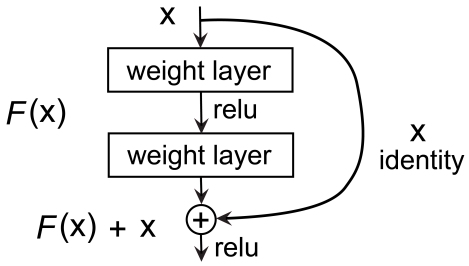

FIGURE 3. A building block of residual learning [31].

The framework architecture in this study uses the FPN on top of a ResNet-50 architecture for the following reasons:

- Generating a multi-scale feature pyramid.
- Attaching two subnetworks: one for classifying anchor boxes.
- And one for regression from anchor boxes to ground-truth object boxes.

The anchor boxes are used as a set of predefined bounding boxes for a resized input image with three aspect ratios 1:1, 1:2 and 2:1. At each spatial position, the classification subnet predicts the probability of the object presence. The parameters of this subnet are shared with all pyramid levels. Over the feature map of a given pyramid level, four 3×3 convolution layers with filters are applied with ReLU activation following each one.

Also per spatial location, sigmoid activations are attached to the output of the binary predictions at the end. The subnet design for box regression to regress the offset from each anchor box to a nearby ground-truth object is identical to the classification subnet except the termination condition. Both have similar structures but different parameters.

Finally, the combined architecture of the model in the framework for object detection and classification has 206 layers including the optimization operations on some layers with a total of 36569662 parameters.

As mentioned before, the bounding boxes approach is utilized together with many different techniques and methods, such as anchor boxes in this study, see Fig. 5. These are highly effective techniques when it comes to surrounding objects in images along with a major disadvantage. If the surrounded object does not fill the area inside the boundaries completely, the box includes various samples of other parts of the image which makes it difficult for our framework to detect the color of the object. Therefore, we proportionally shrink the boxes up to 80% after the detection process and focus on the object's center region by computing an additional region for determining the object color. To work on this region, an image color space function is coded within the framework. This function receives hue, saturation, and value as parameters and returns the object color by determining threshold values. Also, a mean region matrix for RGB (Red, Green, Blue) is coded to obtain the RGB provision of the object color.

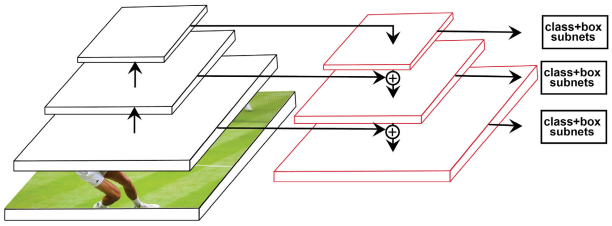

FIGURE 4. The RetinaNet architecture; ResNet (left), FPN (right) [32].

At the end of the AI layer of the framework, the results such as the detected objects' labels, their coordinates in the input image, and their color values in various forms (BGR, RGB, HSV) together with the color label computed from the additional region is exported as a comma-separated values file and stored on a server under a folder automatically created based on the detection response time.

The two results of a sample image obtained from the pre-trained model and the custom-trained model are given in Fig. 6.

C. THE MERGING LAYER

The merging module in this layer is designed and implemented to unify the pre-trained model predictions and custom-trained model predictions based on the probability values and objects' box points (coordinates). In the experiments, we need to observe the variance in the annotation results obtained from the sports ontology with both the pre-trained model's detected objects, and the detected objects after the merging process that uses the custom-trained model results as a supplementary component.

To compare the detected objects of the models according to the generic object list, additional filters are created. With them, the difference between the object detection results and the generic object list is computed, the arguments that should change are extracted, and the data in the latest file is retrieved. The filtered results from both models provide a collection type result that is needed to compare each item by matching object coordinates and probability values. The execution outputs of this step for the sample in Fig. 7 are as follows:

- The pre-trained model's result: {index:0, object: tennis racket, probability:97.863, box point [324, 5, 468, 428]}, {index:1, object: person, probability:99.752, box point:[31, 20, 387, 641]}.
- The custom-trained model's result: {index:0, object: tennis player, probability:92.541, box point:[26, 12, 439, 639]}, {index:1, object: women's suit, probability:64.881, box point: [50, 96, 386, 602]}.
- And after the merging process, the result: {index:0, object: tennis player, probability:92.541, box point:[26, 12, 439, 639]}, {index:1, object: tennis racket, probability:97.863, box point[324, 5, 468, 428]}, {index:2, object: women's suit, probability:64.881, box point: [50, 96, 386, 602]}.

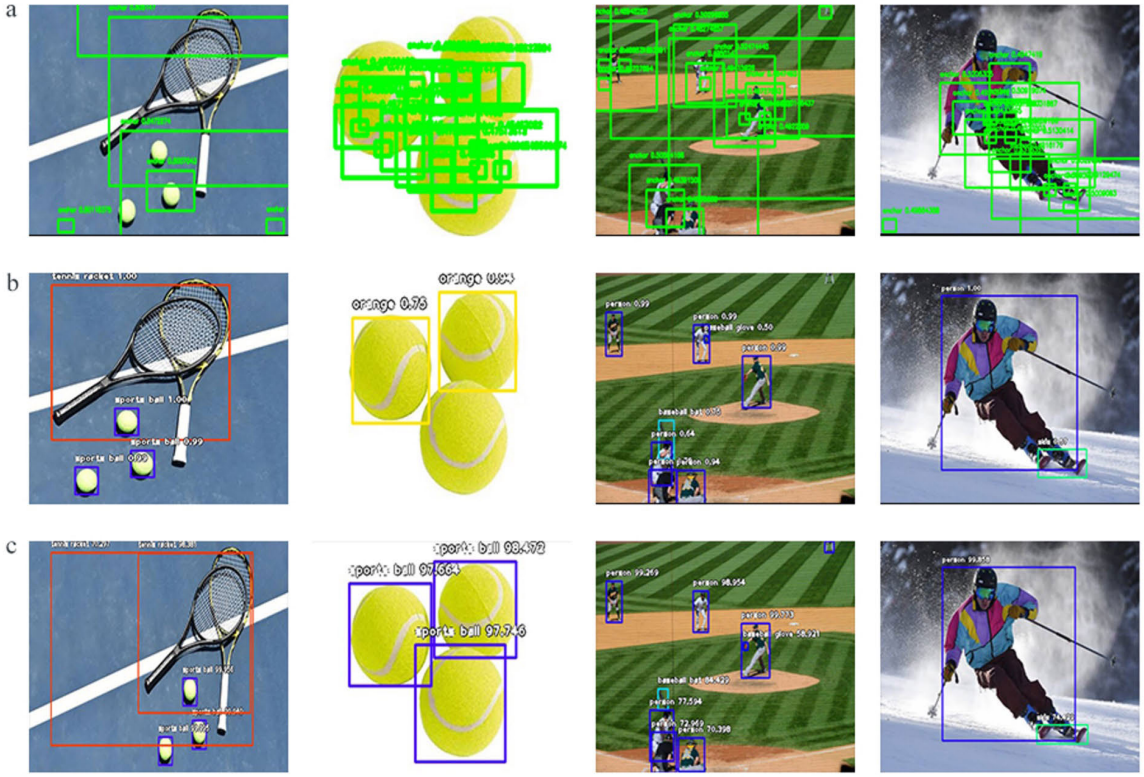

FIGURE 5. (a) Samples with anchor boxes; (b) YOLOv3 detections; (c) RetinaNet detections.

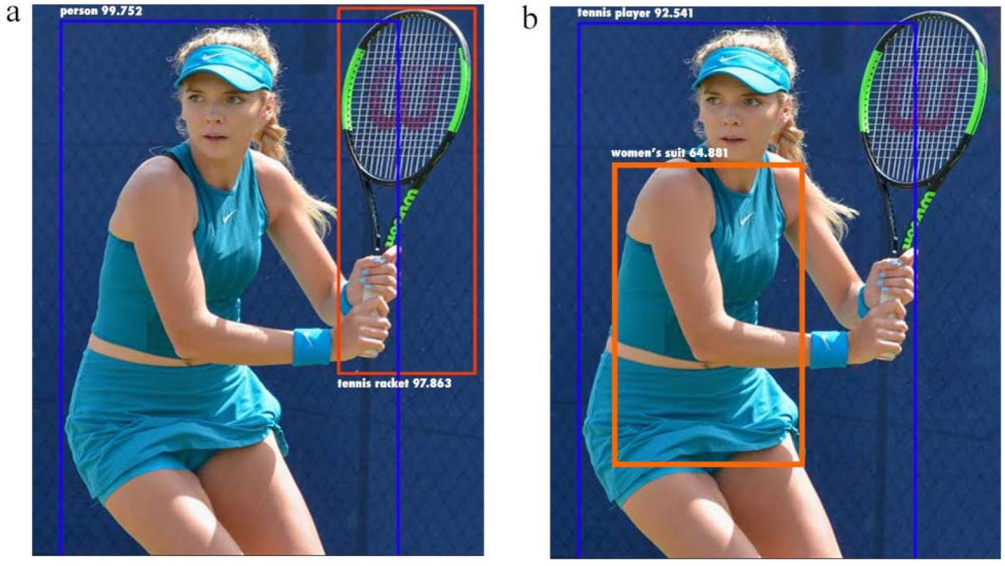

FIGURE 6. (a) The pre-trained model detections; (b) the custom-trained model detections.

More specialized classes can be achieved with the merging process for the detected objects from both models.

The merged detection results obtained after the merging process for the sample image are illustrated in Fig. 7.

D. ONTOLOGY DEVELOPMENT

To obtain accurate and scalable image annotation, the ontology layer was added to the framework. Ontologies contain rich relationships amongst terms and in the framework, the sports ontology exploits these relationships to consider different objects and their joint probability. This layer provides an infrastructure to reach additional sub-classes for three target sports classes and their relations among the detected objects of an image.

Web Ontology Language (OWL) is used together with Protégé, an open-source ontology editor, to build the sports ontology, and HermiT reasoner to determine the consistency

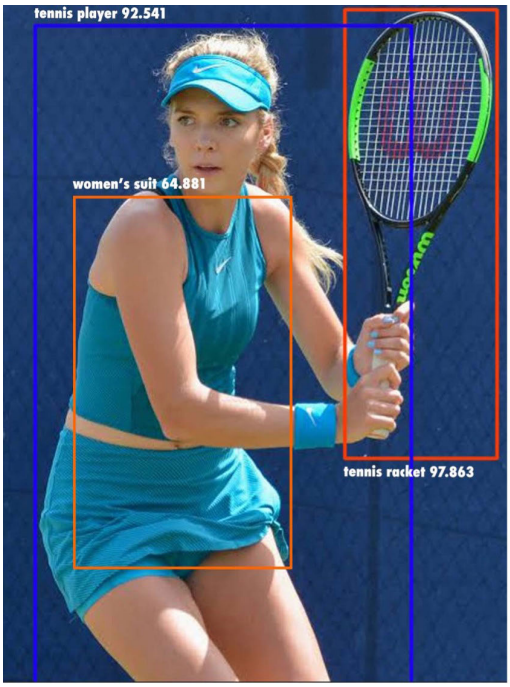

FIGURE 7. After the merging process results.

of the ontology and identify the subsumption relationships between the classes. Overall, the sports ontology includes 268 axioms, 61 classes, 13 object properties, and 40 individuals. These are the core ontological components where classes define an aggregation of things, individuals are instances of classes, and properties link classes/individuals.

The advantages of using the sports ontology in our framework are as follows:

- A common understanding of the structure among the detected objects is provided.
- Domain knowledge reusability is gained with well-defined concepts and their relationships.
- Change in the domain knowledge can easily be reflected in the assumptions.
- Separation of issues between the domain and operational knowledge is accommodated.
- The classification results can be distributed within a multi-layer depth structure.
- Sentential descriptions of images can be achieved.

The ontology editor provides a tiered architecture. Views such as tabs, widgets and menus feed from an object-oriented model. The Protégé SPARQL Query extension is utilized to write and execute semantic queries.

The semantic queries used to obtain results from the sports ontology are classified under three types, see Table 2.

When defining concepts, sameness and uniqueness are explicitly defined using "Equivalent to" and "Disjoint with". Steps required for the sports ontology development process are explained below:

1) A total of 61 classes are defined, where the classifications are distributed within 5-layers depth, see Fig. 8.

TABLE 2. Types of semantic queries.

Type	Relation	Case Sample
Queries based on the relationship between object and color	One-to-many	Sports ball and white
		Sports ball and green
		Sports ball and yellow or green
Queries based on the relationship between the objects	Many-to-many	Person and sports equipment
		Tennis racket and tennis court
		Green ball and tennis court
Queries based on the relationship between object and place	Many-to-one	White ball and baseball field
		Baseball bat and baseball field
		Baseball glove and baseball field

2) 13 object properties are implemented in the ontology such as the examples shown in Fig. 9.
3) Class individuals, instances of classes are declared and can be retrieved with the following query, see Table 3.
4) Object properties are used to bind class individuals, in other words, they help to draw inferences among the different concepts by defining the relationship between the concepts, see Fig. 10.
5) Triples are produced which formally represent the "Subject Verb Object" structure.

Queries are crucial in discovering the semantic relations between the detected objects in the framework. SPARQL queries can be used to obtain results from the ontologies. Such an example can be seen in Table 4 and 5, which presents an example query and its answer, respectively.

Using SPARQL queries, individuals' object property assertion annotations can be reached and exported under csv files to be stored in the memory.

The pre-trained model produces some generic labels which eventually become classification classes. The Person label is such an instance of a generic label which will later be determined specifically as a baseball batter.

E. ANNOTATION SIMILARITY CONTROL

Depicting a picture even with the results obtained from the queries was not sufficient at this point. In some of the relations, especially many-to-many ones, queries produce inaccurate annotations in the same exported file. Baseball player type is an example representing such a situation. The types of equipment, baseball gloves and baseball are both used by two types of baseball players, which are baseball pitcher and baseball catcher. Therefore, the file contains all the semantic annotations belonging to both, see Table 6.

To ensure that the most related sentences are grouped together and irrelevant sentences are eliminated, the Jaccard coefficient and the Cosine similarity are used. Considering words and their orders in sentences is the backbone to calculate sentence similarity.

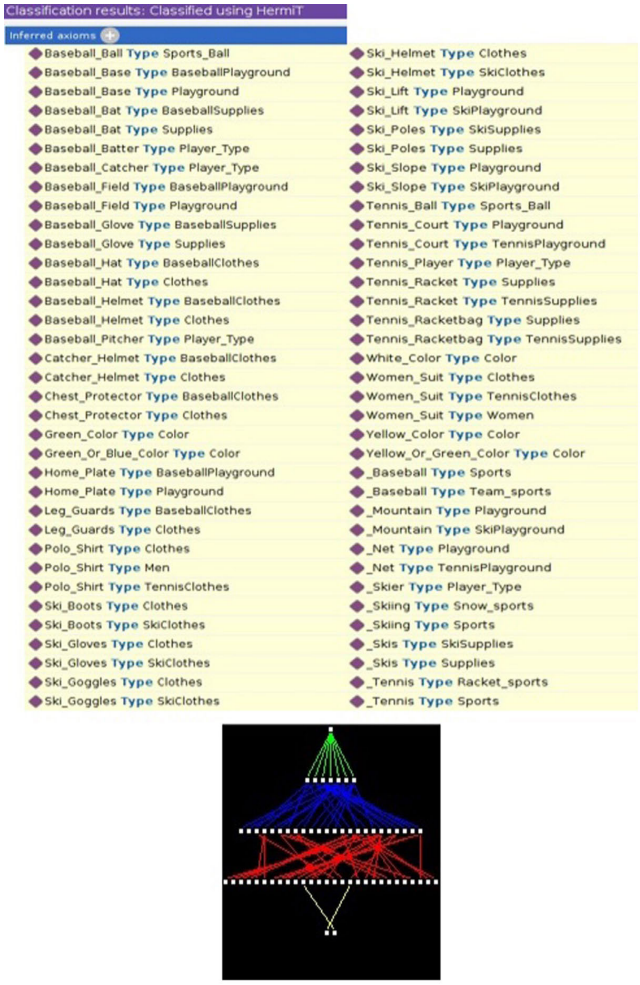

FIGURE 8. The classification results and the hierarchical representation of class dependencies among layers.

We use the two most mentioned statistical calculations (sentence similarity based on a word set and sentence similarity based on the vector) in our study for creating subgroups from the ontology results. The following formulas for sentence similarity based on a word set (Jaccard similarity) and sentence similarity based on the vector (Cosine similarity) are given, respectively [34]:

$$Jaccard\left(c_x, c_y\right) = \frac{|k\left(c_x\right) \cap k\left(c_y\right)|}{|k\left(c_x\right) \cup k\left(c_y\right)|} \tag{4}$$

where, $c_x c_y$ are sentences, $k\left(c_x\right) k\left(c_y\right)$ are word sets of sentences.

$$Cosine\left(c_x, c_y\right) = \frac{\sum_{n=1}^{i+j} W_{xn} W_{yn}}{\sqrt{\sum_{n=1}^{i+j} W_{xn}^2} \sqrt{\sum_{n=1}^{i+j} W_{yn}^2}} \tag{5}$$

where, $W_{xn} W_{yn}$ represent weights assigned to $k\left(c_x\right) k\left(c_y\right)$, and the initial weight of words is 1. If a word occurs more times in one sentence, the weight of the word is accumulated.

FIGURE 9. The usage of an object property.

TABLE 3. Query example to obtain individuals of all the classes.

No	Content
1	PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2	SELECT ?individual ?baseclass
3	WHERE {
4	?individualrdfs:subClassOf ?baseclass
5	}

F. THE MAPPING LAYER

The aim of this module which is developed in Python is to infer from the detected objects and the ontology results. As illustrated in the proposed framework architecture (Fig. 2), the module requires a pre-constructed keyword mapping and query results as well as the detected objects obtained from both pre-merging and post-merging, to produce separate annotations as outputs. The keyword mapping file stores the detected objects and their corresponding annotation results based on the following criteria:

- Frequency distributions of the detected objects in exported results of the semantic queries.
- A keyword can be a single detected object or can be a combination of multiple objects.
- When the keywords and the result files obtained from the sports ontology are matched, each detected object in a keyword is counted separately.
- The cumulative count is considered.
- The file with the highest frequency of a keyword is prioritized.
- The other associated result files are sorted according to the frequency score when linking the result files and the keywords.

The ''term'' is used for each element of a keyword and term frequency (TF) measures how frequently a term occurs in a file:

$$TF\left(x\right) = \frac{Number\ of\ times \times\ appears\ in\ a\ file}{\sum_{i=1}^{n} term_i} \tag{6}$$

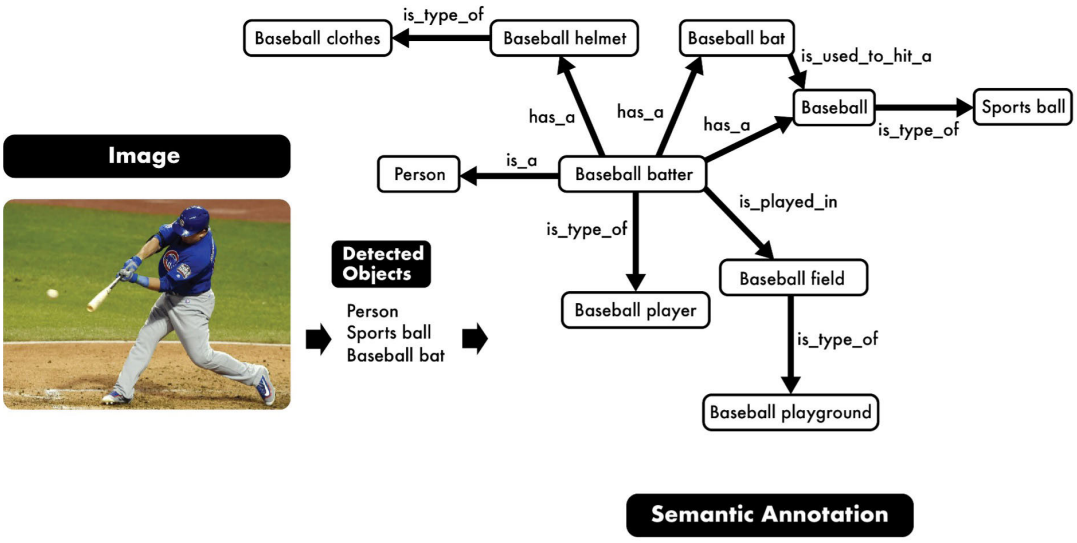

FIGURE 10. Overview of the ontology layer.

TABLE 4. The query for the question "Who plays in baseball field?".

No	Content
1	PREFIX owl: <http://www.w3.org/2002/07/owl#>
2	SELECT ?playertype ?baseballplayground
3	WHERE {
4	?playertyperdf:typeowl:plays_in ?baseballplayground .
5	}

TABLE 5. The answers and its semantic annotation result.

No	Answer	Annotation
1	Baseball_Batter	Baseball batter plays in baseball field.
2	Baseball_Catcher	Baseball catcher plays in baseball field.
3	Baseball_Pitcher	Baseball pitcher plays in baseball field.

One of the challenges here is one cannot know the order of the detected objects before the prediction process. Therefore, an ordering function is implemented to find the permutations of the detected object set and match them with the corresponding combination results. This way, the keyword combination file is not required to store every ordered alternative.

Assume that after the prediction process, the detected objects set from merging are {tennisracket, tennisplayer, women'ssuit}. The mapping module starts to generate alternative permutations for 3 (x) by 3 (y) which are; {tennisracket, women'ssuit, tennisplayer}, {tennisplayer, tennisracket, women'ssuit}, {tennisplayer, women'ssuit, tennisracket}, {women'ssuit, tennisracket, tennisplayer}, and {women'ssuit, tennisplayer, tennisracket}. In the following step, the mapper matches them with the corresponding combination result which is {tennisplayer, tennisracket, women'ssuit} for this case.

TABLE 6. An example of exception cases.

No	Object	Annotation
1	Baseball_Pitcher	Baseball pitcher throws baseball.
2	Baseball_Pitcher	Baseball pitcher plays baseball.
3	Baseball_Catcher	Baseball catcher catches baseball.
4	Baseball_Catcher	Baseball catcher plays baseball.
5	Baseball_Pitcher	Baseball pitcher uses baseball glove.
6	Baseball_Catcher	Baseball catcher uses baseball glove.
7	Baseball_Pitcher	Baseball pitcher wears baseball hat.
8	Baseball_Catcher	Baseball catcher wears catcher's helmet.
9	Baseball_Catcher	Baseball catcher wears chest protector.
10	Baseball_Catcher	Baseball catcher wears leg guards.

The number of the detected objects set elements is denoted by x, and y is the parametric number to determine variations (the initial value of y is equal to x but after each recursion, its numerical value decreases by 1).

The process steps are summarized in Fig. 11.

IV. EXPERIMENTS

A. THE SAMPLE SELECTION ALGORITHM AND THE TESTING STRATEGY

This study aims to produce generalizable knowledge by making statistical inferences about the population. The research takes place in a controlled and constructed setting to draw firm conclusions about cause and effect.

The pseudo-code of the sample selection algorithm and the search terms are given in Fig. 12 and Table 7.

Validation testing with a black-box testing technique, decision table, is adopted as the testing strategy. The steps of decision table testing are as follows [35]:

1) Analyse the given test inputs and list out conditions.
2) Calculate the number of possible combinations.
3) Fill columns of the decision table with combinations.

TABLE 7. The search terms.

Sub-domain	Search Term
Tennis	Tennis racket
	Tennis ball
	Tennis player
	Tennis court
	Tennis match
Skiing	Ski equipment
	Ski poles
	Ski
	Competitive skiing
Baseball	Baseball bat
	Baseball glove
	Baseball
	Baseball player
	Baseball field
	Baseball match

4) Find out cases.
5) Obtain the expected result.

For this study, the decision table structure for validation testing consists of the following components, see Table 8.

Input:
- Expected objects: object names.
- Detected objects: object names.
- Objects in result: object names.
- Match: yes, no.

Output:
- Action: action 1 (correct count + 1), action 2 (false count + 1).

Also, the same technique is used for the validation of the annotation results obtained from the pre-merging and the post-merging annotations.

B. DATASETS AND ONTOLOGY

The two RetinaNet models used for object detection and prediction adjust themselves via different training processes. The pre-trained model benefits from the adjusted weights obtained using the COCO 2014 dataset [36]. The test split of this dataset contains only images with no annotations. Therefore, the validation split is essential for the final adjusted weights. Another issue that is confusing about this dataset is the number of labeling text files and the number of image files in val2014. There are 40504 image files in the validation folder but only 40137 text files for labels of the validation images. Basically, the numbers do not match. The validation can only be done with the images that have respective labeling text files. The training folder of the COCO 2014 dataset contains 82783 images and 80 object categories. Many of these categories are related to the sports domain which are eventually used by our framework [37]. For the experiments, we created another supplementary dataset with more specific sports object categories.

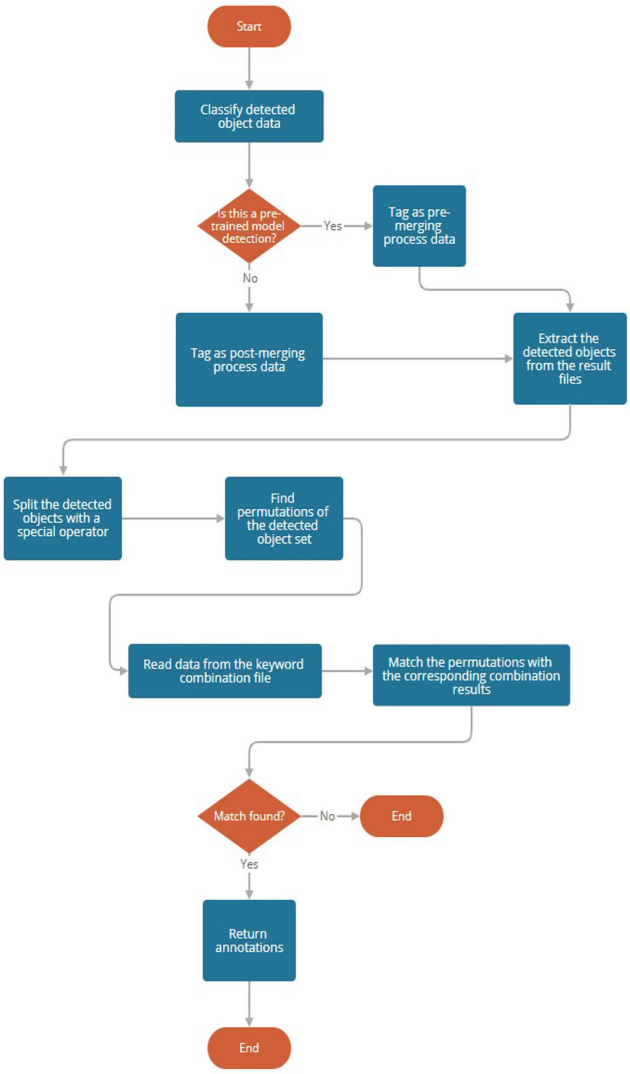

FIGURE 11. The mapping process steps.

The purpose of this second dataset is to compare and understand the variance in the annotation results with specific objects aligned after (post-merging annotations) and before the merging process (pre-merging annotations). In total, our custom sports image dataset [38] includes 1200 sports pictures obtained from the internet and 2120 objects were drawn plus tagged for 1200 images using VoTT software [39], see Fig. 13.

The training process of the custom model was conducted on 80% of the custom sports dataset (960 images) and the remaining 20% was used for validation (240 images). The distribution of classes is given in Table 9.

The sports ontology [40] was used in the experiments together with these datasets.

C. TRAINING CHALLENGE

When training models with custom datasets that usually consist of significantly fewer images and classes when compared

```
Input: n experiments (numbered as 1, 2, .., n), h = hyperparameters,
P = set of population images, S = sample set
Output: a subset of semantic annotations

SELECT random sampling as the probability sampling method
FOR i in SearchTermCombinations
        perform google image search and record top page results
ENDFOR
P ← top page results
randomlist = []
FOR j in range(1,11)
        img = generate random integer between (1,100)
        randomlist ← append generated number
ENDFOR
S ← randomlist
// change h to observe the difference in the object detection accuracy
FOR each image in S
        csv ← detect objects, their colors, coordinates from both AI models
        csv ← merge csv files
        keyword ← generate detected object permutations
        IF keyword in SemanticQueries
                mapper ← map pre-merging and post-merging semantic annotations separately
        ENDIF
        RETURN mapper results (natural language descriptions)
ENDFOR
```

FIGURE 12. The pseudo-code of the sample selection and annotation.

to the datasets suitable for generic usage, excessive learning over some classes almost always occurs.

The main reason we use RetinaNet in our framework is to prevent this scenario (easy negatives) as much as possible with the aid of focal loss. Since one of the models is trained on the custom image dataset, further investigation reveals that the validation loss is lower than the training loss, see Fig. 14.

There are three main reasons for this result. First, dropout was applied, a regularization technique to generalize better to the data outside the testing set. Secondly, the training loss was measured during each epoch, while the validation loss was measured after each epoch. In other words, the place of the training loss should be shifted 1/2 an epoch to the left. The third reason is the test set size.

D. IMAGE AND OBJECT PREDICTION MATCH EVALUATION

There are two versions of the prediction model. One uses pre-trained weights on the COCO dataset (the pre-trained model), and the other uses the weights obtained from the training process with the custom image dataset (the custom-trained model).

The undetected and generic objects are the root cause of why we propose the framework as a hybrid solution. Both models failed to detect some of the expected objects in images. The interactions between the expected objects in the samples and the undetected objects from the perspective of the pre-trained model are visualized below, see Fig. 15.

The ten sample images and their detection results were recorded using the structure of Table 8, and the results are summarized in Table 10 and 11. The success percentage (SP) is calculated according to the following formula:

$$SP = \frac{\sum_{i=0}^{n} true\ detected\ objects_i}{\sum_{i=1}^{n} expected\ objects_i} \times 100 \qquad (7)$$

Several correlation coefficients are used to understand various correlation types between the variables in this study. The types are as follows:

- The Pearson's correlation coefficient (r)

TABLE 8. The decision table structure for validation testing.

Expected Objects	Detected Objects	Match	Action
From a human perspective	From pre-trained AI model	Yes	Action 1
		Or	Or
		No	Action 2
	From custom-trained AI model	Yes	Action 1
		Or	Or
		No	Action 2

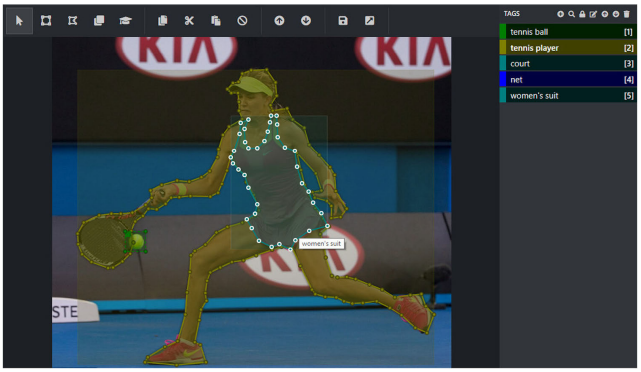

FIGURE 13. An instance of object drawing and tagging.

- The Spearman's rank correlation coefficient (ρ)
- The Kendall rank correlation coefficient (τ)
- Cramér's V (φc) coefficient

The correlation analysis with different coefficients is given in Fig. 16.

Pearson's (r) is used to measure the linear correlation between variable pairs and ranges between -1 and $+1$, which indicates a total negative linear correlation and total positive linear correlation, respectively.

Zero value is used to represent no linear correlation. The highest positive linear correlation belongs to the expected object number (ExpectedObj) and the custom-trained model's undetected object number (CTMUndetect). The second highest belongs to the pre-trained model's generic detected object number (PTM_GDetect) and its true detected object number (PTM_TDetect). The third highest positive linear correlation belongs to the pre-trained model's undetected object number (PTMUndetect) and the custom-trained model's undetected object number. In terms of the negative linear correlation, the highest one belongs to the pre-trained model's undetected object number and the custom-trained model's true detected object number (CTM_TDetect).

Spearman's (ρ) is used to observe the monotonic correlation between variable pairs. In terms of catching nonlinear monotonic correlations, it is better than Pearson's r. It ranges between -1 and $+1$ where -1 indicates a total negative monotonic correlation, 0 indicates no monotonic correlation, and 1 indicates a total positive monotonic correlation. We observe that the negative monotonic correlation between the pre-trained model's undetected object number and the custom-trained model's true detected object number narrows

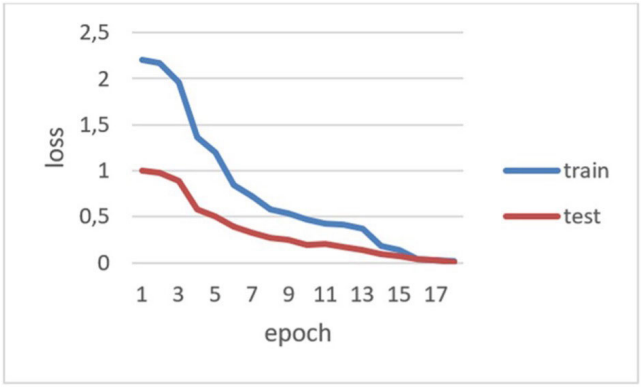

FIGURE 14. The plot of the custom-trained model loss on the training and validation datasets.

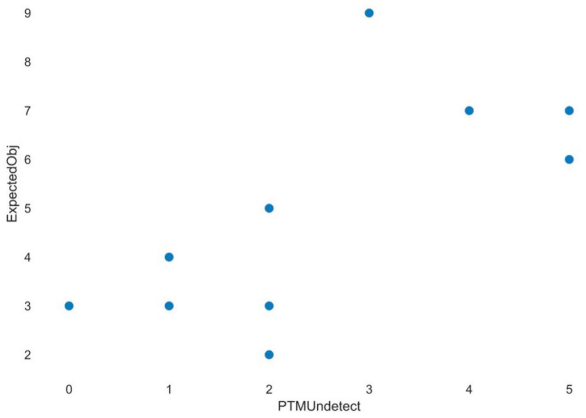

FIGURE 15. The interaction diagram between the expected objects and the pre-trained model's undetected objects.

by 0.125 when compared to Pearson's r. Also, the positive monotonic correlation between the pre-trained model's undetected object number and the custom model's undetected object number has decreased when compared with the positive linear correlation value between these variables.

In Kendall's (τ), τ is given by the number of concordant pairs minus the discordant pairs divided by the total number of pairs. This rank correlation coefficient measures the ordinal association between variable pairs. -1 indicates total negative correlation, 0 indicates no correlation, and 1 indicates total positive correlation. The results of this rank coefficient support the finding of the other rank correlation coefficient (Spearman's) with insignificant margin changes.

For Cramér's V, the bias-corrected measure (proposed by Bergsma) is used in this study to measure the association for nominal random variables.

It ranges from 0 (indicates independence) to 1 (indicates perfect association). Here, the strongest association belongs to two variable pairs which are PTM_TDetect-CTM_TDetect pair and CTMUndetect-PTM_FDetect (the pre-trained model's false detected object number) pair.

In the three out of four correlation analyses, an invalid coefficient occurred for the custom-trained model's generic detected object number variable. This is caused by the class structure of the custom image dataset; it contains only specifically labeled objects. Therefore, the custom-trained model did not detect any generic objects, and this is an expected outcome for this variable.

TABLE 9. Distribution of the class instances.

Class Headings	Total Instance Number	Train Instance Number	Test Instance Number
tennis ball	204	163	41
women's suit	126	101	25
court	73	58	15
net	124	99	25
tennis player	238	190	48
baseball field	252	202	50
baseball batter	208	166	42
baseball	346	277	69
baseball pitcher	279	223	56
baseball catcher	270	216	54

E. IMAGE AND SENTENCE MATCH EVALUATION

The proposed framework produces annotation results for an image from two different stages. One annotation result file is generated using the detected objects that belong to the pre-merging stage.

The other result file is generated using the detected objects obtained after the merging process.

The annotation results for the samples are summarized below, see Table 12 and Table 13.

The percentage of finding the expected objects in sentences increased more where the annotation results were obtained successfully. The number of false sentences found in the annotation results is reduced after the merging process. On the other hand, the number of true sentences found in the annotation results increased except for sample #8. Also, correct annotation results were achieved after the merging process for samples #2 and #6 whereas it was not possible to reach any annotation result for these samples before the merging process.

The correlation analysis with different coefficients is given in Fig. 17.

According to Pearson's (r) the highest positive linear correlations are between the true sentence number obtained from the pre-merging stage annotation results (PreMP_TS) and the false sentence number obtained from the post-merging stage annotation results. Also, The PostMP_FS variable has the second highest positive linear correlation with the expected object number (ExpectedObj) variable.

The third highest positive linear correlation belongs to the false sentence number obtained from the pre-merging stage annotation results and the true sentence number obtained from the post-merging stage annotation results. The highest negative linear correlation is between the true sentence number obtained from the post-merging stage annotation results and the false sentence number obtained from the post-merging stage annotation results. This outcome proves and highlights the increment of the true annotation results

TABLE 10. Object detection results and success percentage of the pre-trained model.

Sample No	Expected Object Number	True Detected Object Number	False Detected Object Number	Generic Detected Object Number	Undetected Object Number	Success Percentage (SP)
1	3	1	0	2	0	33.33%
2	2	0	0	0	2	0.0%
3	3	1	0	1	1	33.33%
4	9	2	0	4	3	22.22%
5	5	1	2	2	2	20.0%
6	6	0	0	1	5	0.0%
7	4	1	0	2	1	25.0%
8	7	1	0	1	5	14.29%
9	7	1	0	2	4	14.29%
10	3	0	0	1	2	0.0%

TABLE 11. Object detection results and success percentage of the custom-trained model.

Sample No	Expected Object Number	True Detected Object Number	False Detected Object Number	Generic Detected Object Number	Undetected Object Number	Success Percentage (SP)
1	3	2	2	0	1	66.67%
2	2	1	0	0	1	50.0%
3	3	2	0	0	1	66.67%
4	9	3	0	0	6	33.33%
5	5	2	1	0	3	40.0%
6	6	1	0	0	5	16.67%
7	4	2	0	0	2	50.0%
8	7	0	1	0	6	0.0%
9	7	2	1	0	5	28.57%
10	3	2	1	0	1	66.67%

obtained by the framework while the false annotation results decreased.

The positive monotonic correlation value produced by Spearman's (ρ) changes (increases) between the PostMP_FS variable and the ExpectedObj variable when compared to Pearson's (r). On the other hand, the value decreases between the PreMP_TS and the PostMP_FS variables for the same correlation type.

Kendall's (τ) results support the findings of Spearman's (ρ) with decreased values. To obtain the association, Cramér's V (φc) coefficient was observed again and the strongest association for nominal random variables is found to be between the true sentence number obtained from the post-merging stage annotation results and the true sentence number obtained from the pre-merging stage annotation results (PreMP_TS). From the recorded data and analysis, our framework produces promising annotation results both from the pre-merging stage and from the post-merging stage. At the same time, it is observed that the generic objects decreased to zero when the annotation results were examined. As a result, an invalid coefficient is detected when analyzing the correlation between the variable pairs that include the generic object variable, such as the generic object number in sentences obtained from the pre-merging stage annotation results and the post-merging stage annotation results.

V. DISCUSSION

A. GENERAL AND SAMPLE-BASED EVALUATION

From a broader perspective, the results prove the correlation between the number of detected objects and annotations by comparing their means. There is a positive correlation between the number of true predicted objects with a mean of 0.31 and correct annotation results with a mean of 0.75. Annotation accuracy is increased with the aid of some specified objects after the merging process. The pre-merging stage annotation results achieve significant success in some samples too. The lowest percentage of increment is 14.28% belonging to sample #9, the highest percentage of increment is 75.0% which belongs to sample #7. In the post-merging stage annotations, the lowest percentage increment is 22.23% which belongs to sample #4, the highest percentage increment is 50.0% belonging to sample #2, #6, and #7. In addition to these, there was no generic object found before and after the merging process annotation results. The annotation results belonging to the pre-merging stage and the post-merging stage differs in many samples, such as sample #3, #4, and #10.

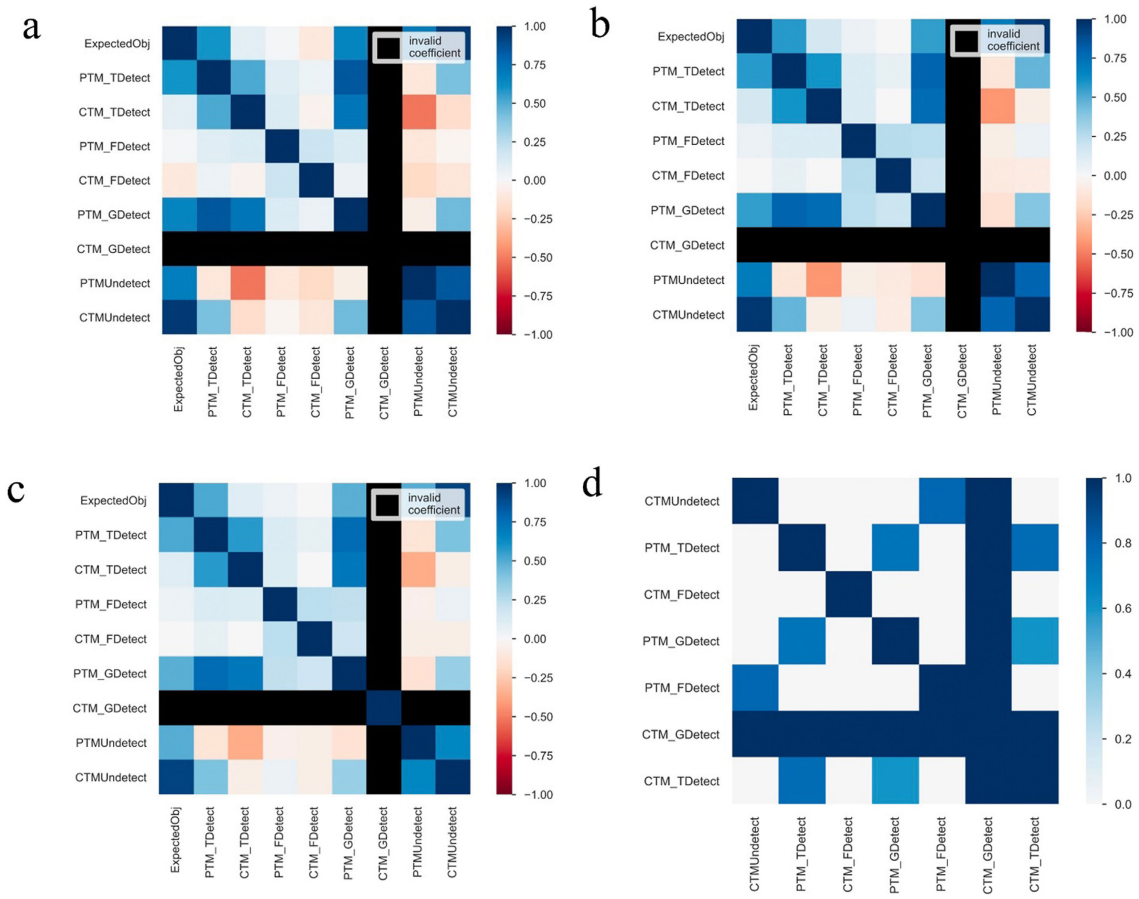

FIGURE 16. Correlation analysis of the variables from Table 10 and Table 11; (a) Pearson's (r); (b) Spearman's (ρ); (c) Kendall's (τ); (d) Cramér's V.

Most of the studies reviewed reveal that the concept of image annotation is perceived in a wide variety of ways. These studies especially include proposals for choosing the right labels and/or increasing the number of related labels. With the help of the sports ontology, which defines the semantics between objects and their annotations, the framework achieved much better results, reaching the accurate sentences (not just multiple accurate tags) to depict the given picture in a controlled environment.

From the sample level, the custom-trained model achieves 33.34%, more than the pre-trained model when it comes to image and object prediction match evaluation. The obtained annotation results include correct sentences that have all the expected objects, and as a result, the expected objects can be found 100.0% in the annotation results even with 2 generic objects (sports ball and person) detected and predicted by the pre-trained model and even with 2 false objects (baseball pitcher and baseball batter) predictions by the custom-trained model. The results obtained for some of the sample images with their explanations will be described in the following paragraphs.

The two classification labels ''baseball field'' and ''home plate'' do not exist in the pre-trained model's classification labels. Therefore, the model did not detect any objects for sample #2. On the other hand, the custom-trained model detected and predicted one (baseball field) of the two expected objects. However, the annotation results of the post-merging stage include the missing object in the sentences. The relations between the classes in the ontology enable this outcome. As a result, the expected objects can be found 100.0% in the annotation results.

In sample #4, a different case occurs. Although the annotations obtained from both stages (pre-merging and post-merging) are correct, the percentage of finding the expected objects in sentences is 55.56%. The root cause behind this result is that there is no distinction between an object and its plural form in the sports ontology. Therefore, the mapper could not find any combination of all the detected objects and retrieved the nearest combination results.

There was an unexpected generic object (person) recorded from the post-merging stage of sample #5. Just by detecting from a half-portion of a body (legs), the pre-trained model predicts a person in the image within a different coordinate than the expected one. A key error was raised in sample #6. The AI model only detects and predicts one object (person) for this image. To obtain semantic annotation results,

TABLE 12. The pre-merging stage annotation statistics of the samples.

Sample No	True SentenceNumber	False SentenceNumber	ExpectedObject Number	Generic Object Number in Sentences	% of Finding Expected Objects in Sentences
1	3	0	3	0	100.0%
2	-	-	2	-	-
3	1	0	3	0	66.67%
4	1	0	9	0	22.22%
5	4	0	5	0	80.0%
6	-	-	6	-	-
7	4	6	4	0	100.0%
8	6	0	7	0	85.71%
9	1	1	7	0	28.57%
10	1	0	3	0	33.33%

TABLE 13. The post-merging stage annotation statistics of the samples.

Sample No	True Sentence Number	False SentenceNumber	ExpectedObject Number	Generic Object Number in Sentences	% of Finding Expected Objects in Sentences
1	3	0	3	0	100.0%
2	2	0	2	0	100.0%
3	3	0	3	0	100.0%
4	3	0	9	0	55.56%
5	4	0	5	0	80.0%
6	3	0	6	0	66.67%
7	4	0	4	0	100.0%
8	0	3	7	-	-
9	1	1	7	0	28.57%
10	1	0	3	0	66.67%

a generic object needs to be compared with at least one more object predicted from an image.

Sample #7 presents the exception case given in Table 6. The similarity checks clear the ambiguity between the object classes that use the same objects. Our framework successfully separates the baseball pitcher from the baseball catcher and retrieves the annotation results for the Baseball_Pitcher and the other object classes within the picture.

One of the samples (sample #8) highlights the overfitting situation for the skier case. Generally, the skier's knees are bent in the images. The size and distribution of the custom image dataset can cause the custom-trained model to learn some of the training classes excessively. In this case, the AI model predicts the skier as a baseball catcher (their knees are also bent in images).

Finally, a different situation was encountered in sample #9. Some of the data frames used for the colors were returned missing the value marker. Sample #9 differs from the other visuals; it is a painting instead of a photo. The ranges between some color values are too narrow in terms of floating-point. This can easily cause such an unexpected situation. In the following sub-sections, the proposed framework will be compared with other systems from an ontological multi-layer system perspective, as well as from the perspective of systems that utilize artificial intelligence methods.

B. COMPARISON WITH OTHER ONTOLOGICAL MULTI-LAYER SYSTEMS

In [20], the researchers propose a solution by combining the decision tree (DT) method and an ontology to exploit the benefits of object-based image analysis in the domain of geographic images. DT has a serious advantage in obtaining high accuracy with the training data. Nevertheless, a major drawback occurs with unseen data which can easily cause terrible results. They compare classification results with and without the ontology. The outcome is a 1.63% increment over classification with the ontology usage when compared with the usage of DT alone. The proposed framework's minimum success percentage increment value over the sample set outperforms their increment and our framework also includes formal sentential descriptions which are not generated in [20].

The study by Bannour and Hudelot [21] proposes the automatic building of multimedia ontologies for the sole purpose of identifying and formalizing the semantic relationships between the concepts with a different dataset, Pascal VOC'2009. From the image classification perspective, their

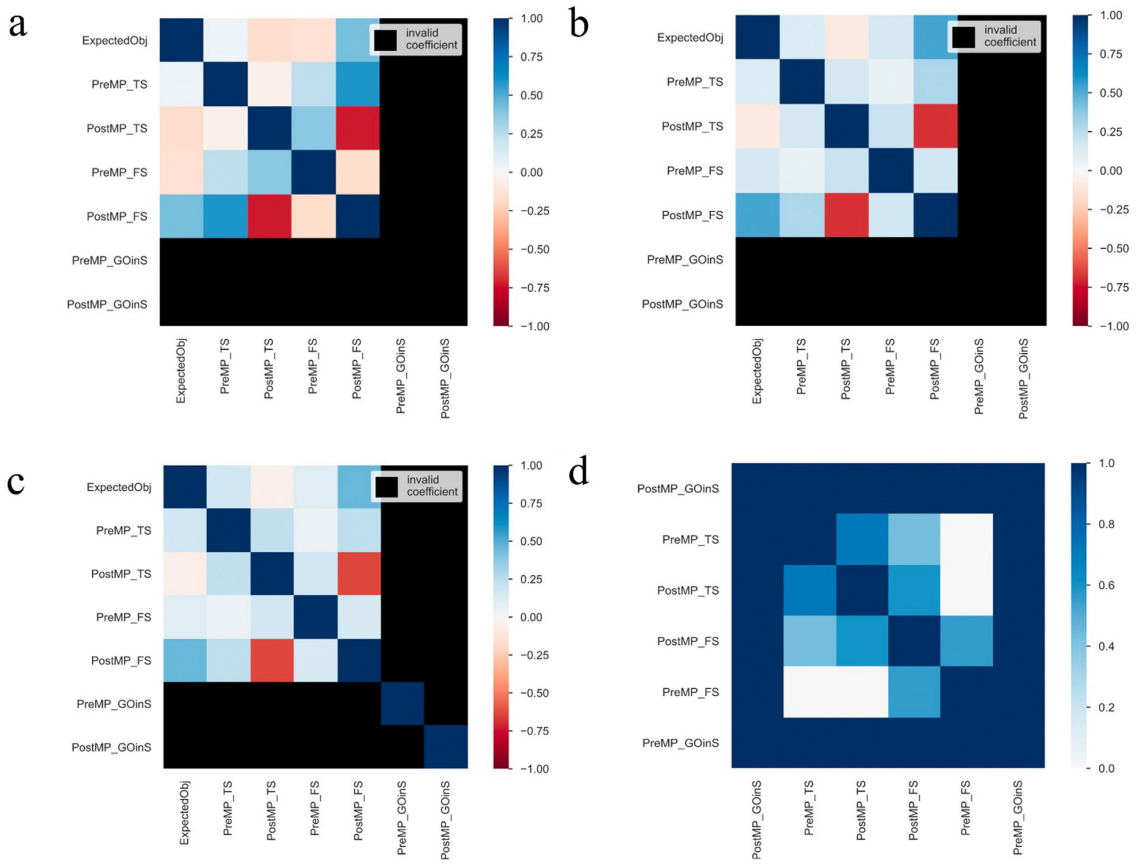

FIGURE 17. Correlation analysis of the variables from Table 12 and Table 13; (a) Pearson's (r); (b) Spearman's (ρ); (c) Kendall's (τ); (d) Cramér's V.

results are above average. However, it would not be correct to compare the performance results obtained on two different datasets. Even though their results are promising for the correct alignment of tags for ontologies and images, they do not generate sentences that depict an image. Another research presented in [25] focuses on investigating the retrieval process of similar images considering their annotations and skips the annotation process completely. Another major difference is their study uses a dissimilarity measure for term comparison instead of the similarity calculations for sentence comparison used in this study (Jaccard and Cosine similarity). In a systematic review study [41], image retrieval concept in the domain of remote sensing imagery is investigated under important categories such as fusion-oriented, geo-localization and disaster rescue. It also offers solutions to image retrieval which are applicable to other fields for matching different image types as in [42].

In terms of semantic segmentation, literature provides some proposed solutions based on the combination of various techniques. The studies presented in [43] and [44] stand out in this regard; [43] focuses on high-level inference and fills the gap with knowledge-guided ontological reasoning where a deep semantic segmentation network (DSSN) fails. The method proposed in [44] is beneficial when massive quantities of training data is a concern where the testing data resembles the training data. Both studies help to improve the robustness of semantic segmentation.

C. COMPARISON WITH SYSTEMS BASED MAINLY ON ARTIFICIAL INTELLIGENCE

As mentioned before, most of the studies in the image annotation domain focus on choosing the right labels or increasing the number of related labels to depict an image. One such instance is presented in [45]. To obtain a relevant label set, the researchers follow label correlations that include symbiotic and semantic relationships. Another research by Zhang *et al.* [46] investigates tag refinement to improve the annotation performance. A useful approach, a joint CNN and RNN framework is proposed in [47] for capturing long-term dependencies. Yet, it does not contain a solution that generates text automatically for image annotation. References [48] proposes the use of label correlation to improve the discriminating power of the classifiers. In terms of obtaining descriptive sentences, it does not exhibit a structure. Although all of these studies produce promising results in their respective fields, they differ from this work in terms of the goal of obtaining complete textual

descriptions from a picture. Only [6] differs from the other artificial intelligence-based systems mentioned here by generating natural language descriptions of images similar to this study. The comparison of the overall annotation results reveals that the accuracy obtained from the proposed framework outperforms the previous studies within the controlled environment.

D. LIMITATIONS

Although we achieve encouraging results, the framework is subject to ontology domain restrictions. The sports ontology only provides annotations based on the defined classes and their individuals. For the object color detection part, concentrating on the center of an object achieves the best results from well-proportioned shapes. Tennis rackets are one of the many examples of such a condition, they are composed of three parts which are head, shaft, and handle. The combination of these parts does not contain every pixel in a bounding box because of its shape. Along with the tennis racket, there are a lot of background image components in a bounding box. Lastly, in the custom-trained model using ResNet-50 and FPN, we faced overfitting due to the incompetent number of dataset elements.

VI. CONCLUSION AND FUTURE WORK

A novel framework is proposed that translates generic objects into specific objects and generates natural language descriptions of images based on deep learning techniques and ontologies. Several studies are found to be highly focused on methods/techniques. Yet in terms of the most effective method, a consensus has not been reached. It is for this reason that a variety of methods, techniques, tools, and processes coexist in most of the examined studies. Similarly, the proposed framework combines two neural network models with identical layers, different hyperparameters, and different training approaches. Also, the framework includes a merging module to unify the two models' predicted objects based on probability values and coordinates for preventing incorrect annotation results and obtain more sentences if possible. In addition to these, a unique sports ontology is developed containing expected object classes, their relations, and annotations. Lastly, a mapping module on top of the other modules generates keywords from the predicted objects obtained before and after the merging process separately to match with the relevant ontology results. The evaluation of the system's performance on both classification and sentence similarity shows that the proposed framework achieves promising results in terms of sentence similarity within the controlled environment.

For future work, the findings of this study can be utilized to support web image annotation projects, to construct image-description datasets automatically, and for the automatic generation of various ontologies. Interdisciplinary approaches such as this study should also be considered in developing new solutions.

REFERENCES

[1] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2018, doi: 10.1109/ACCESS.2017.2788044.

[2] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.

[3] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, and M. Gao, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018, doi: 10.1109/ACCESS.2018.2836950.

[4] S. Zeadally, E. Adi, Z. Baig, and I. A. Khan, "Harnessing artificial intelligence capabilities to improve cybersecurity," *IEEE Access*, vol. 8, pp. 23817–23837, 2020, doi: 10.1109/ACCESS.2020.2968045.

[5] F. Fleuret, T. Li, C. Dubout, E. K. Wampler, S. Yantis, and D. Geman, "Comparing machines and humans on a visual categorization test," *Proc. Nat. Acad. Sci. USA*, vol. 108, no. 43, pp. 17621–17625, Oct. 2011, doi: 10.1073/pnas.1109168108.

[6] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 664–676, Apr. 2017, doi: 10.1109/TPAMI.2016.2598339.

[7] C. C. Loy, D. Lin, W. Ouyang, Y. Xiong, S. Yang, and Q. Huang, "WIDER face and pedestrian challenge 2018: Methods and results," 2019, *arXiv:1902.06854*. [Online]. Available: http://arxiv.org/abs/1902.06854

[8] K. Nguyen, C. Fookes, A. Ross, and S. Sridharan, "Iris recognition with off-the-shelf CNN features: A deep learning perspective," *IEEE Access*, vol. 6, pp. 18848–18855, 2018, doi: 10.1109/ACCESS.2017.2784352.

[9] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, and A. C. Berg, "BabyTalk: Understanding and generating simple image descriptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2891–2903, Dec. 2013, doi: 10.1109/TPAMI.2012.162.

[10] A. Farhadi, M. Hejrati, M. A. Sadeghi, and P. Young, "Every picture tells a story: Generating sentences from images," in *Proc. ECCV*, Berlin, Germany, 2010, pp. 15–29.

[11] A. L'Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, "Machine learning with big data: Challenges and approaches," *IEEE Access*, vol. 5, pp. 7776–7797, 2017, doi: 10.1109/ACCESS.2017.2696365.

[12] A. Akusok, K.-M. Bjork, Y. Miche, and A. Lendasse, "High-performance extreme learning machines: A complete toolbox for big data applications," *IEEE Access*, vol. 3, pp. 1011–1025, 2015, doi: 10.1109/ACCESS.2015.2450498.

[13] L.-C. Chen, S. Fidler, and R. Urtasun, "Beat the MTurkers: Automatic image labeling from weak 3D supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 3198–3205.

[14] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-RNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4485–4493.

[15] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with Polygon-RNN++," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 859–868.

[16] R. Troncy, J. van Ossenbruggen, J. Z. Pan, and G. Stamou. *Image Annotation on the Semantic Web*. Accessed: Aug. 14, 2007. [Online]. Available: https://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/

[17] R. Socher, Q. V. L. A. Karpathy, C. D. Manning, and A. Y. Ng, "Grounded compositional semantics for finding and describing images with sentences," *Trans. Assoc. Comput. Linguistics*, vol. 2, no. 1, pp. 207–218, 2014, doi: 10.1162/tacl_a_00177.

[18] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," 2014, *arXiv:1411.2539*. [Online]. Available: http://arxiv.org/abs/1411.2539

[19] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, and Y. Kalantidis, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 32–73, 2017.

[20] H. Gu, H. Li, L. Yan, Z. Liu, T. Blaschke, and U. Soergel, "An object-based semantic classification method for high resolution remote sensing imagery using ontology," *Remote Sens.*, vol. 9, no. 4, p. 329, 2017, doi: 10.3390/rs9040329.

[21] H. Bannour and C. Hudelot, "Building and using fuzzy multimedia ontologies for semantic image annotation," *Multimedia Tools Appl.*, vol. 72, no. 3, pp. 2107–2141, May 2014, doi: 10.1007/s11042-013-1491-z.

[22] S. Baier, Y. Ma, and V. Tresp, "Improving visual relationship detection using semantic modeling of scene descriptions," in *Proc. ISWC*, Vienna, Austria, 2017, pp. 53–68.
[23] D. Im and G. Park, "Linked tag: Image annotation using semantic relationships between image tags," *Multimedia Tools Appl.*, vol. 74, no. 7, pp. 2273–2287, 2015, doi: 10.1007/s11042-014-1855-z.
[24] V. Franzoni, A. Milani, S. Pallottelli, C. H. C. Leung, and Y. Li, "Context-based image semantic similarity," in *Proc. 12th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Zhangjiajie, China, Aug. 2015, pp. 1280–1284.
[25] C. Kurtz, A. Depeursinge, S. Napel, C. F. Beaulieu, and D. L. Rubin, "On combining image-based and ontological semantic dissimilarities for medical image retrieval applications," *Med. Image Anal.*, vol. 18, no. 7, pp. 1082–1100, 2014, doi: 10.1016/j.media.2014.06.009.
[26] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, "A multi-view embedding space for modeling internet images, tags, and their semantics," *Int. J. Comput. Vis.*, vol. 106, no. 2, pp. 210–233, 2014, doi: 10.1007/s11263-013-0658-4.
[27] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, and J. Cai, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018, doi: 10.1016/j.patcog.2017.10.013.
[28] M. Lamons, R. Kumar, and A. Nagaraja, "Handwritten digits classification using convnets," in *Python Deep Learning Projects*, 1st ed. Birmingham, U.K.: Packt, 2018, pp. 176–179.
[29] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*. [Online]. Available: http://arxiv.org/abs/1605.07146
[30] G. Wang, "A perspective on deep imaging," *IEEE Access*, vol. 4, pp. 8914–8924, 2016, doi: 10.1109/ACCESS.2016.2624938.
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Las Vegas, NV, USA, 2016, pp. 770–778.
[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.
[33] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[34] L. Zahrotun, "Comparison Jaccard similarity, cosine similarity and combined both of the data clustering with shared nearest neighbor method," *Comput. Eng. Appl. J.*, vol. 5, no. 1, pp. 11–18, Jan. 2016, doi: 10.18495/COMENGAPP.V5I1.160.
[35] S. Nidhra, "Black box and white box testing techniques—A literature review," *Int. J. Embedded Syst. Appl.*, vol. 2, no. 2, pp. 29–50, Jun. 2012, doi: 10.5121/ijesa.2012.2204.
[36] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, Zurich, Switzerland, 2014, pp. 740–755.
[37] Technology Blog. *What Object Categories/Labels Are in COCO Dataset*. Accessed: May 16, 2020. [Online]. Available: https://tech.amikelive.com/node-718/what-object-categories-labels-are-in-coco-dataset/
[38] A. Sezen, C. Turhan, and G. Sengul. *Custom Sports Image Dataset*. Accessed: Dec. 10, 2019. [Online]. Available: https://drive.google.com/drive/folders/1pRwANz89dlE1yPf8C29i1nwisCOBS8XS?usp=sharing
[39] *VoTT: Visual Object Tagging Tool*, Microsoft Corporation, Albuquerque, NM, USA, 2010.
[40] A. Sezen, and C. Turhan. *Sports Ontology V1.7*. Accessed: Apr. 15, 2020. [Online]. Available: https://drive.google.com/file/d/17YQYjE_T0_JtN83K_I6FX49IjkckUEoi/view?usp=sharing
[41] Y. Li, J. Ma, and Y. Zhang, "Image retrieval from remote sensing big data: A survey," *Inf. Fusion*, vol. 67, pp. 94–115, Mar. 2021, doi: 10.1016/j.inffus.2020.10.008.
[42] Y. Li, Y. Zhang, X. Huang, and J. Ma, "Learning source-invariant deep hashing convolutional neural networks for cross-source remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6521–6536, Nov. 2018, doi: 10.1109/TGRS.2018.2839705.
[43] Y. Li, S. Ouyang, and Y. Zhang, "Collaboratively boosting data-driven deep learning and knowledge-guided ontological reasoning for semantic segmentation of remote sensing imagery," 2020, *arXiv:2010.02451*. [Online]. Available: http://arxiv.org/abs/2010.02451
[44] Y. Li, T. Shi, Y. Zhang, W. Chen, Z. Wang, and H. Li, "Learning deep semantic segmentation network under multiple weakly-supervised constraints for cross-domain remote sensing image semantic segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 175, pp. 20–33, May 2021, doi: 10.1016/j.isprsjprs.2021.02.009.
[45] Z. Ning, G. Zhou, Z. Chen, and Q. Li, "Integration of image feature and word relevance: Toward automatic image annotation in cyber-physical-social systems," *IEEE Access*, vol. 6, pp. 44190–44198, 2018, doi: 10.1109/ACCESS.2018.2864332.
[46] P. Zhang, Z. Wei, Y. Li, and C. Zhao, "Automatic image annotation based on multi-auxiliary information," *IEEE Access*, vol. 5, pp. 18402–18411, Sep. 2017, doi: 10.1109/ACCESS.2017.2749252.
[47] A. Hassan and A. Mahmood, "Convolutional recurrent deep learning model for sentence classification," *IEEE Access*, vol. 6, pp. 13949–13957, 2018, doi: 10.1109/ACCESS.2018.2814818.
[48] Z. Xue, J. Du, M. Zuo, G. Li, and Q. Huang, "Label correlation guided deep multi-view image annotation," *IEEE Access*, vol. 7, pp. 134707–134717, Sep. 2019, doi: 10.1109/ACCESS.2019.2941542.

ARDA SEZEN received the Ph.D. degree in software engineering from Atilim University, Ankara, Turkey. He is currently working as an Assistant Professor with the Department of Software Engineering, OSTIM Technical University, Ankara. His research interests include artificial intelligence, image processing, semantic web technologies, and software engineering.

CIGDEM TURHAN received the Ph.D. degree in computer engineering from Middle East Technical University, Ankara. She is currently working as an Assistant Professor with the Department of Software Engineering, Atilim University, Ankara, Turkey. She is the author of a number text books in the area of programming. Her research interests include natural language processing, machine translation, semantic web technologies, and engineering education.

GOKHAN SENGUL was born in Ankara, Turkey, in 1976. He received the B.S. degree in electronic engineering from Ankara University, Ankara, and the M.S. and Ph.D. degrees in electrical and electronics engineering from Hacettepe University, Ankara, in 2002 and 2008, respectively. He is currently an Associate Professor with the Department of Computer Engineering, Atilim University, Ankara. His current research interests include signal and image processing, pattern recognition, artificial intelligence, and biomedical systems.