پیوست (آموزش و مثال های نشانه گذاری)

نحوه تعامل با نرم افزار

تعریف موجودیت

کافی است از نوار بالا نوع موجودیت را انتخاب کنید و با گرفتن کلیک سمت چپ موس کلمه و یا عبارت مورد نظر را انتخاب کنید. نرم افزار فضاهای خالی (اسپیس) را در نظر نمیگیرد پس نگران فضاهای خالی نباشید.

تعریف گروه برای موجودیت ها

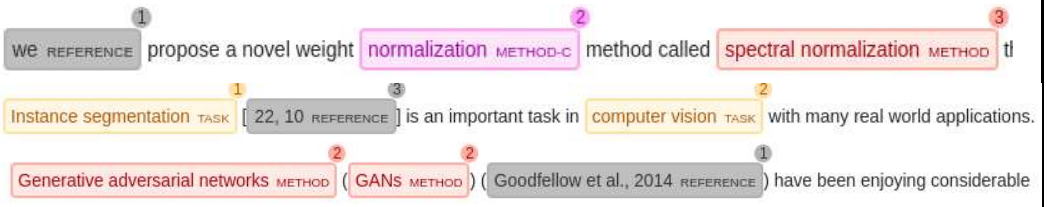
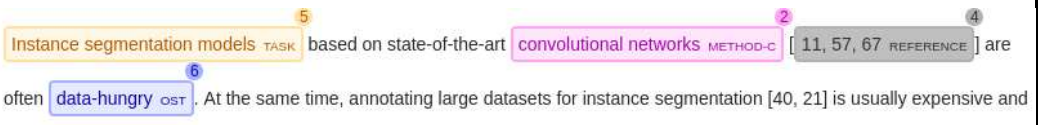
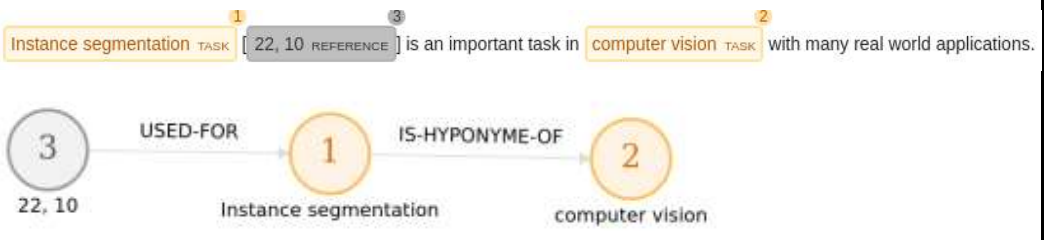
اگر نام موجودیت های انتخاب شده کاملا یکسان باشند نرم افزار به صورت اتوماتیک آن ها در یک گروه قرار می دهد. شماره ای که در بالای هر موجودیت نوشته شده است نشان دهنده ایندکس گروه آن موجودیت می باشد و موجودیت هایی که در یک گروه قرار دارند دارای شماره یکسانی خواهند بود.
برای قرار دادن یک موجودیت در یک گروه کافی است دکمه شیفت را نگه دارید و آن موجودیت را روی موجودیت یا گروه مورد نظر با موس بکشانید و رها کنید. با این کار در یک گروه قرار خواهند گرفت. اما برای خارج کردن یک موجودیت از گروه شما نیاز دارید به بخش گروه ها رفته و با آیکن جدا کننده که در هر گروه مشاهده می کنید موجودیت های از یکدیگر جدا نمایید.

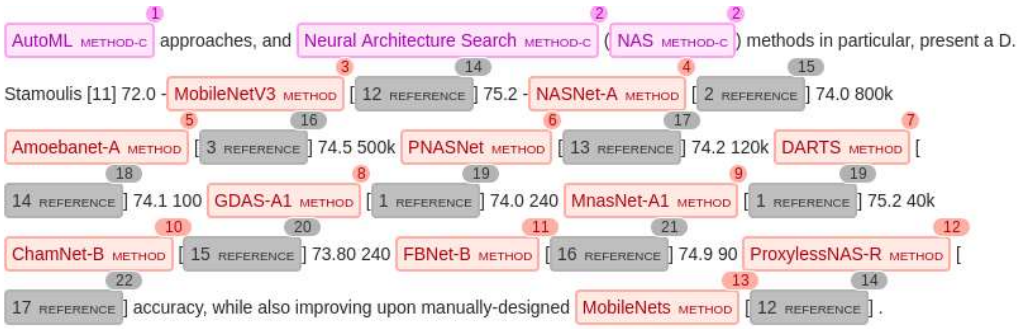
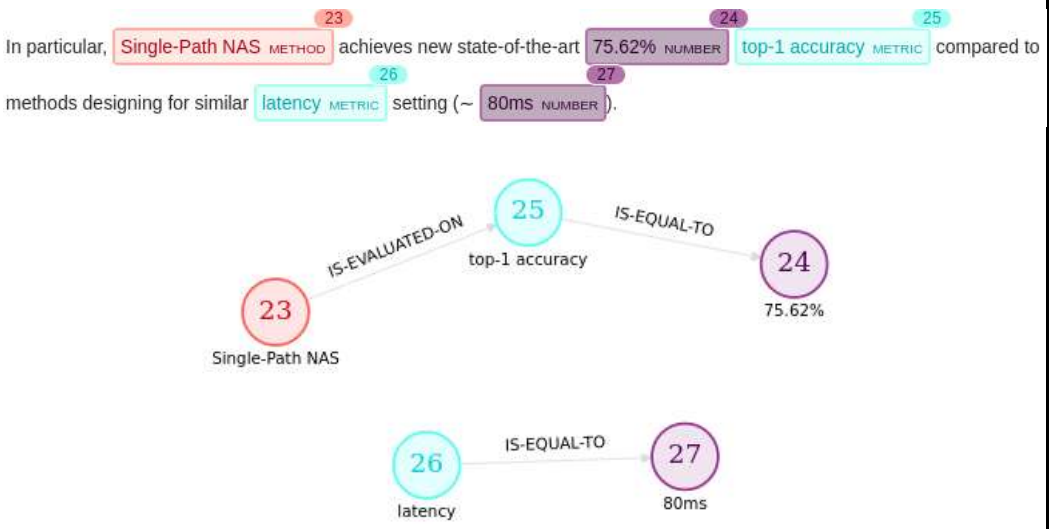
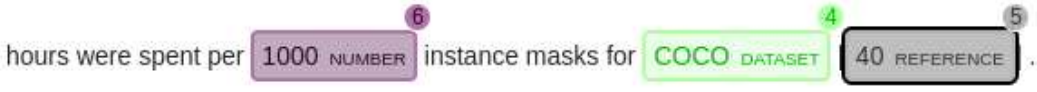
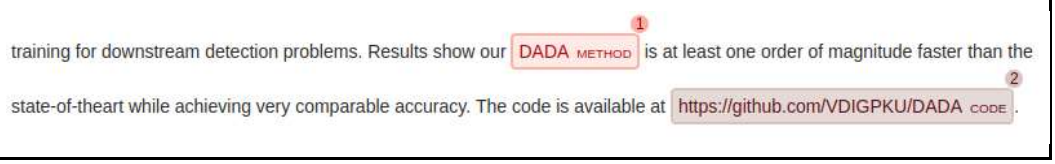
تعریف ارتباط بین گروه ها

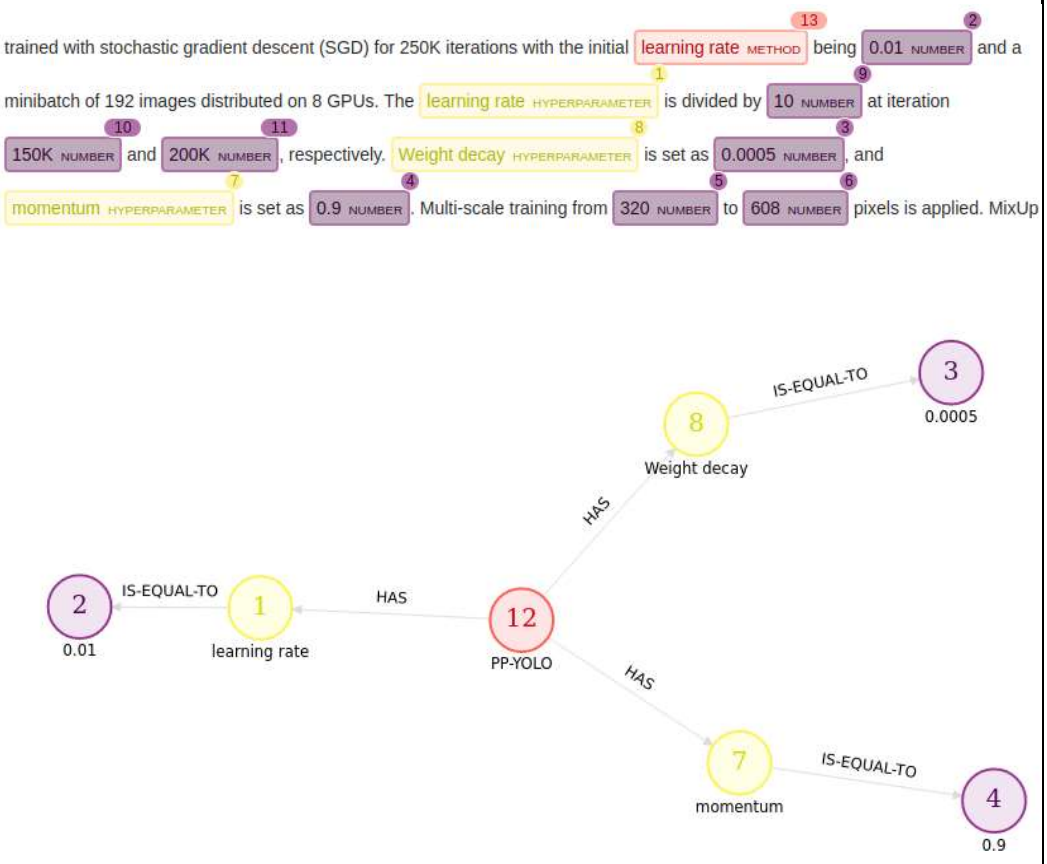
برای ایجاد یک ارتباط تنها کافی است یک موجودیت (در واقع یک گروه) را با موس بکشانید و روی یک موجودیت دیگر (در واقع یک گروه دیگر) رها کنید. با این کار یک پنجره جدید برای شما ظاهر خواهد شد که ارتباطاتی که برای این دو گروه قابل تعریف هستند نشان داده می شود و شما یک ارتباط را انتخاب می کنید. برای تغییر ارتباطات تعریف شده دو راه وجود دارد. یا می توانید به بخش ارتباطات رفته و ارتباط مورد نظر را بیابید و ویرایش نمایید و یا اینکه روی موجودیت مورد کلیک نمایید و پنجره باز شده به شما ارتباطات تعریف شده برای آن موجودیت را به شما نمایش می دهد.

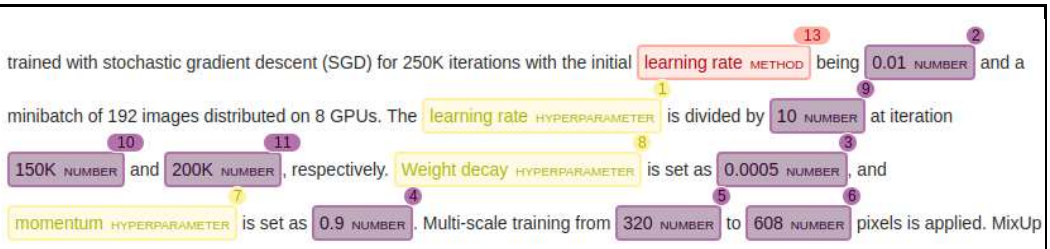
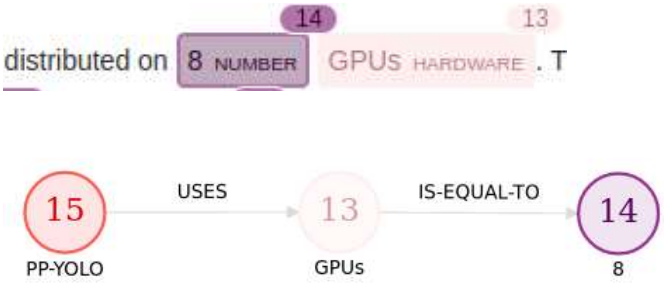
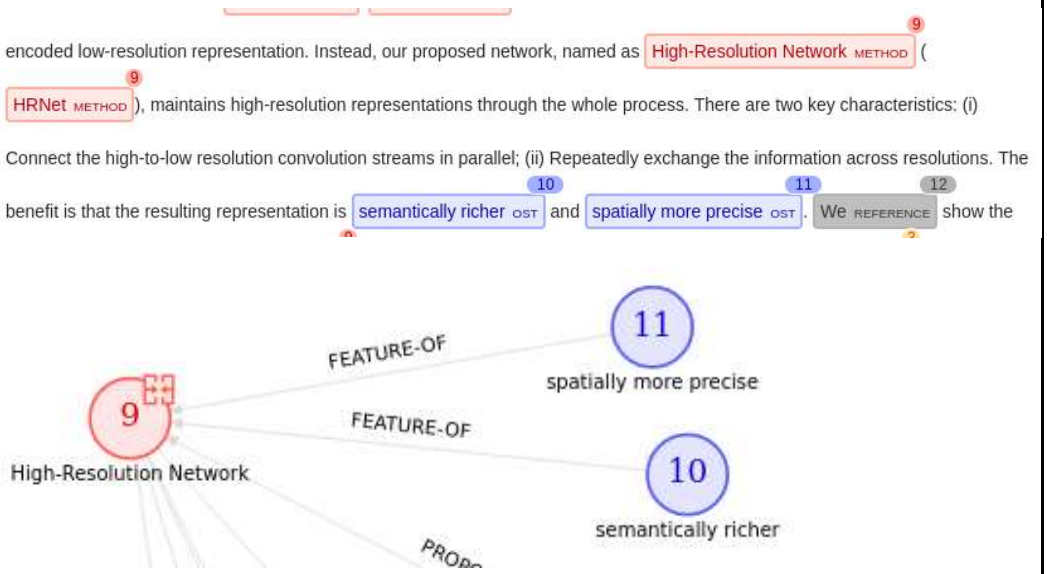
موجودیت ها و ارتباطات قابل تعریف

Entities:

	Rule & Example
۱	**Reference** • We, this paper, this work , … are annotated as Reference • The multiple numbers or references must be annotated as one reference. • The reference could be mentioned by name of author of paper
	
۲	**Method-Class** • The general methods that refer to a class of methods should be annotated as Method_Class (METHOD_C). These methods usually have plural “s” or indefinite article “a or an”.
	
۳	Task, sub-tasks, fields are annotated as **Task**. Usually relation of tasks and fields are defined by HYPONYM
	
۴	The specific method names should be annotated as **Method**. The methods usually have some reference papers

	
۵	**Metrics** are the evaluation criteria that are used for evaluation purposes. Usually methods are EVALUATED ON a specific metric.
	
۶	**Dataset**: Each dataset name should be annotated as a dataset. The general terms like “image datasets” must not be annotated. Usually a method IS_EVALUATED_ON a dataset.
	
۷	**Code**: The code URLs should be annotated as Code. The code usually has a “HAS” relation with method or reference.
	

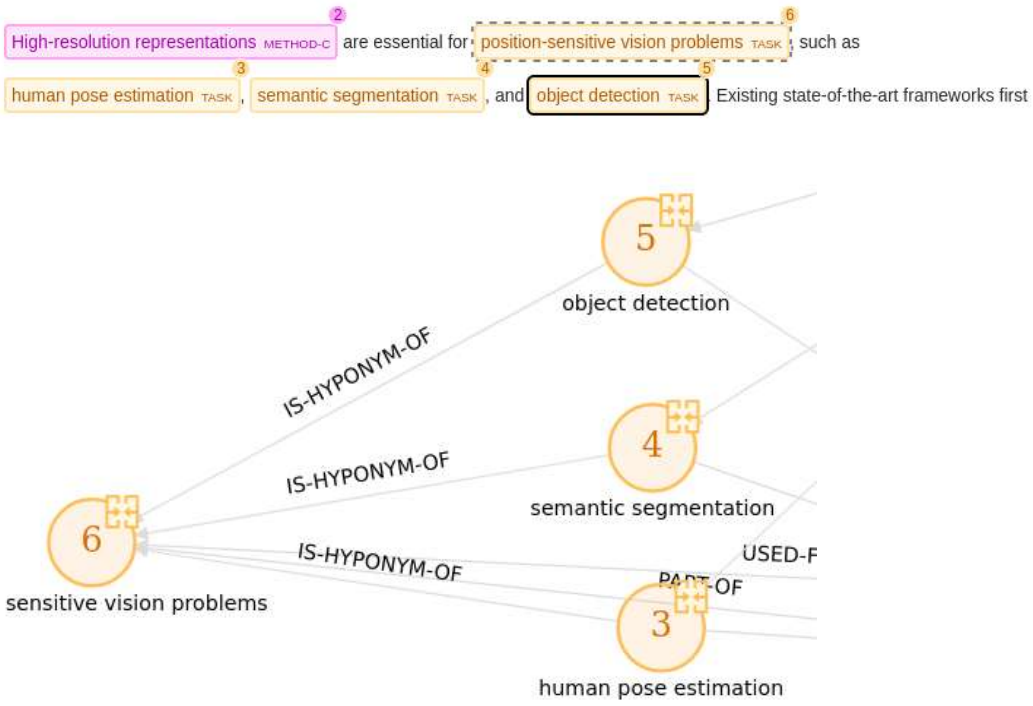
	
ʌ	**HYPER_PARAMETER:** all parameters and hyper parameters of a method should be annotated as this type. Usually hyperparameters should be connected to a method with a “HAS” relationship.
	
٩	**NUMBER:** any number must be annotated as a NUMBER. The unit of each number should be annotated alongside the number.

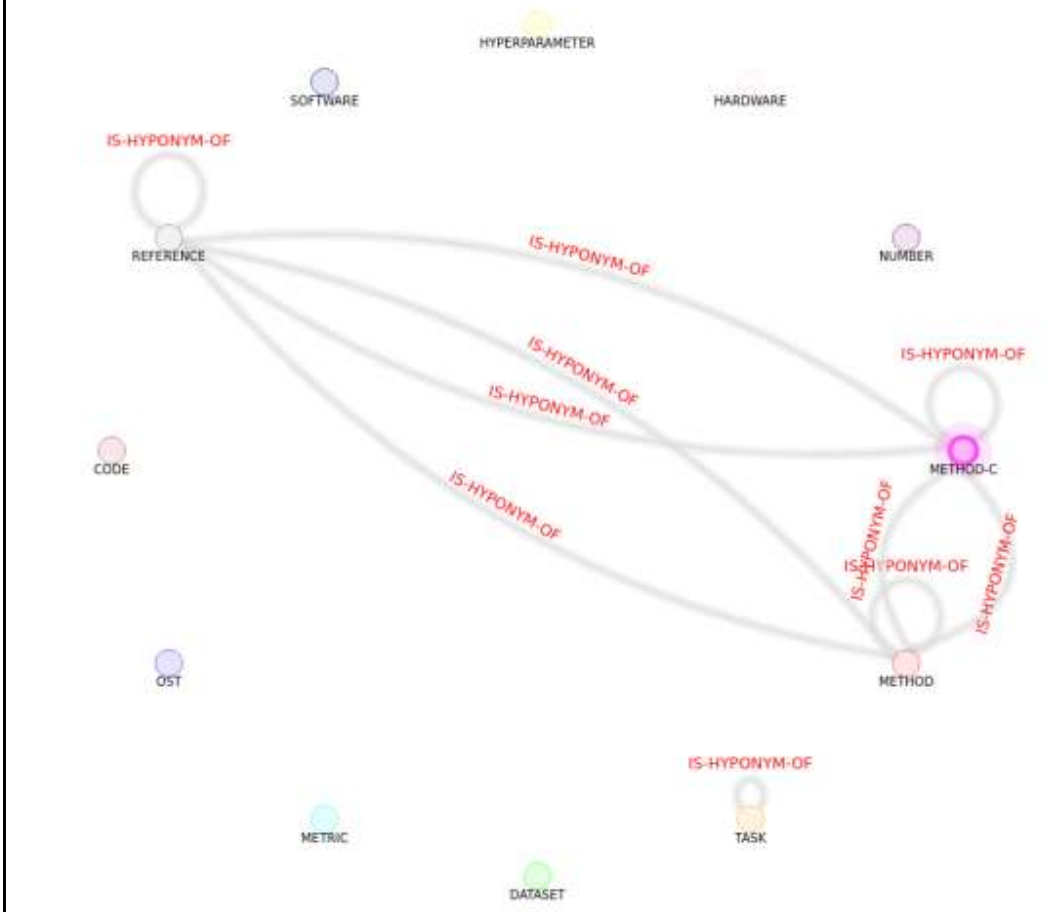
	
۱۰	**HARDWARE:** The hardwares presented in papers. Hardwares are usually USED _FOR a specific method and IS_EQUAL_TO some numbers.
	
۱۱	**FEATURE:** Any features, pros, cons and important aspects of a method or task or reference should be annotated as **FEATURE**
	
۱۲	**OTHER-SCIENTIFIC-TERM (OST):** any other entity that does not fall in any of the above entity groups and it conveys important information should be annotated as OST. These entities could be annotated by all

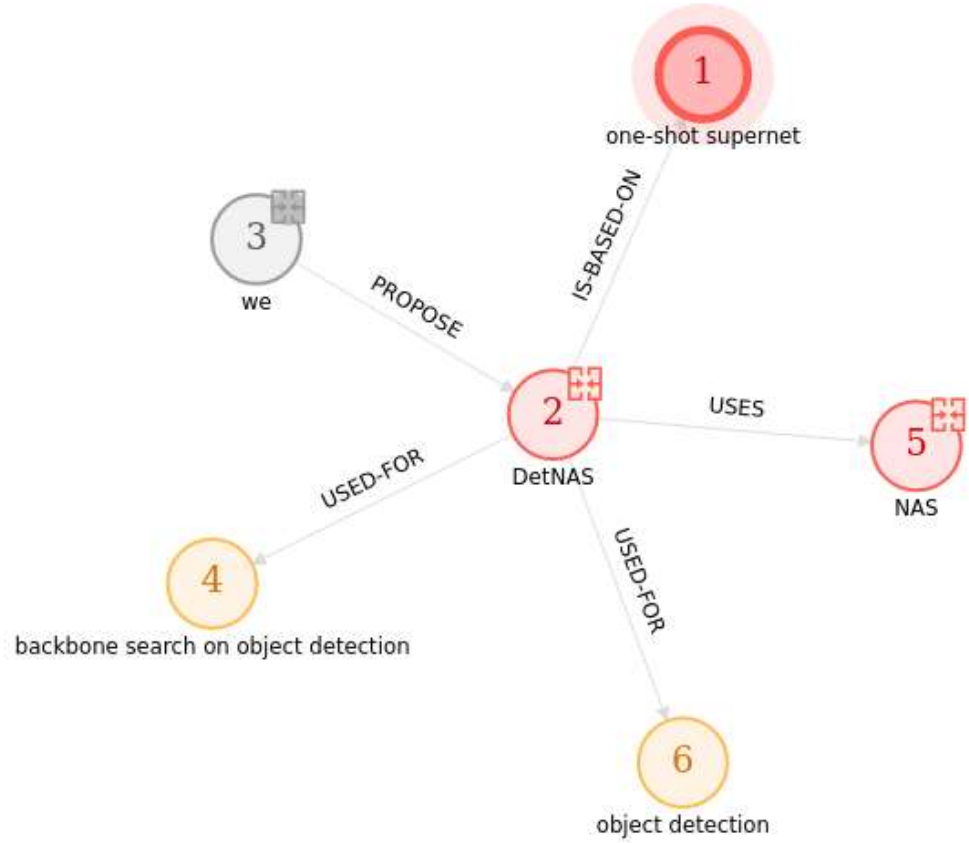
	possible relationships.

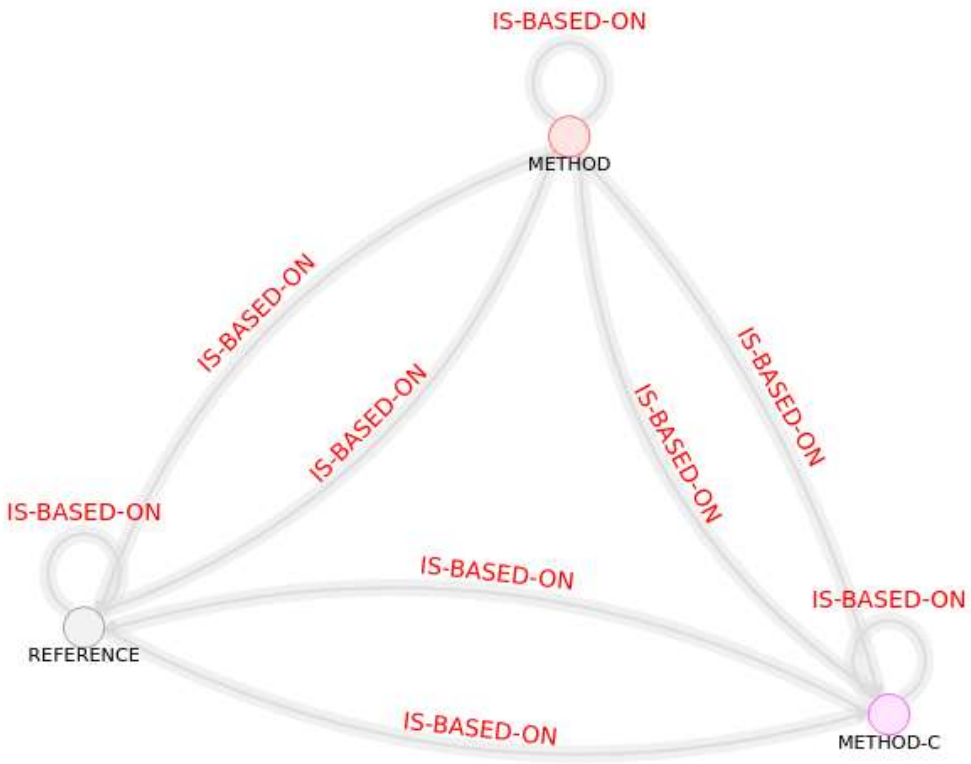
Relations:

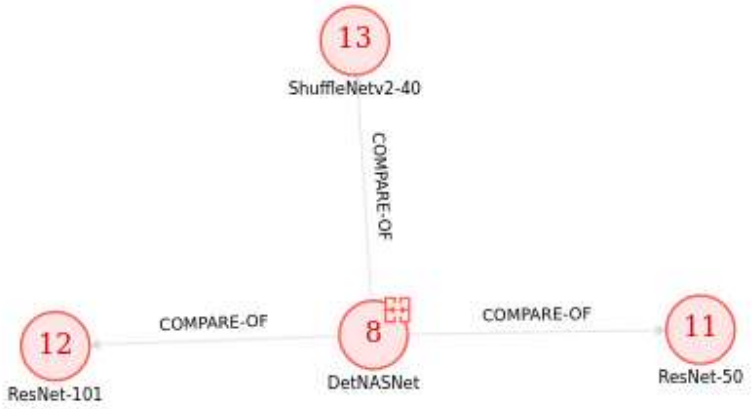
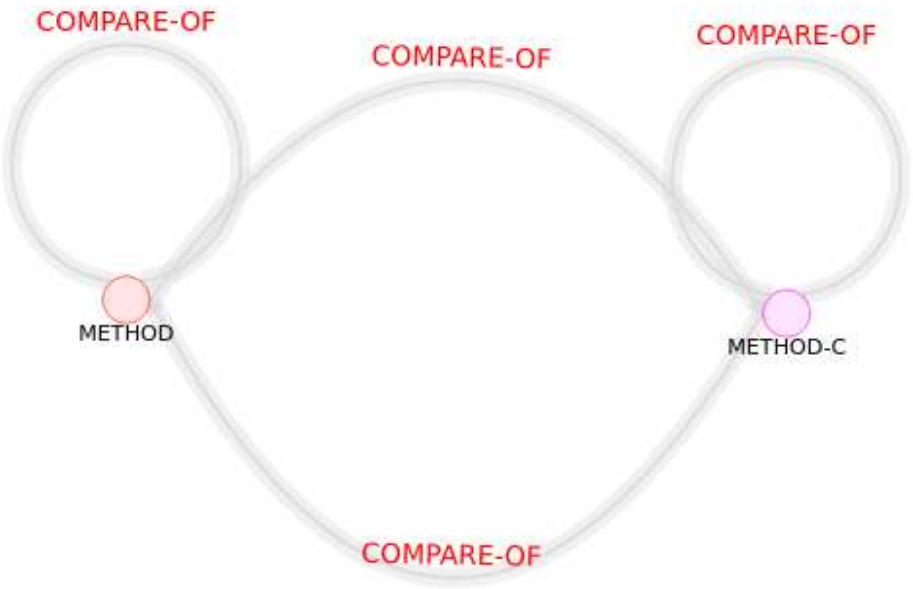
For the relations please see the schema of data in the application. The schema shows you all possible relations that could be defined between each two entities. In the following table all relations of OST entities are removed from schema to simplify the schema. The OST terms could be annotated by all possible relations.

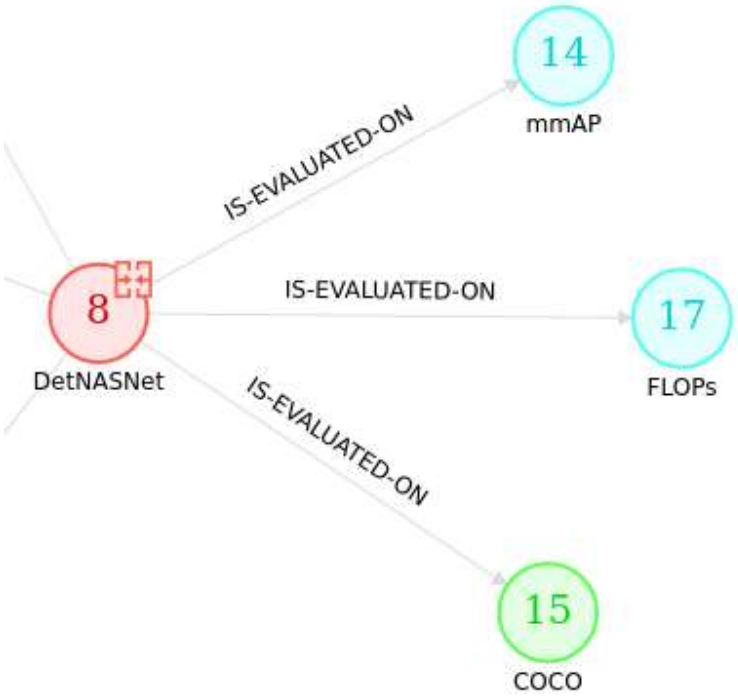
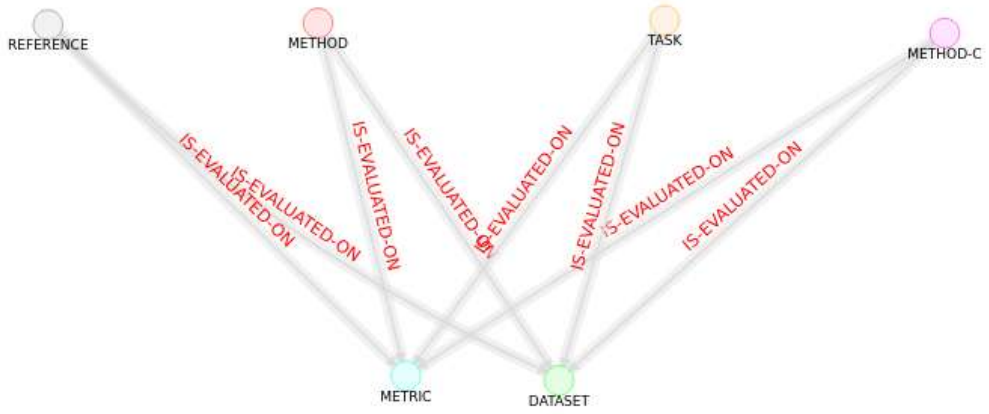
	Relation & Schema & Examples
\	**HYPONYM (Kind of):** Hyponymy shows the relationship between a generic term (hypernym) and a specific instance of it (hyponym). A hyponym is a word or phrase whose semantic field is more specific than its hypernym.
Example	

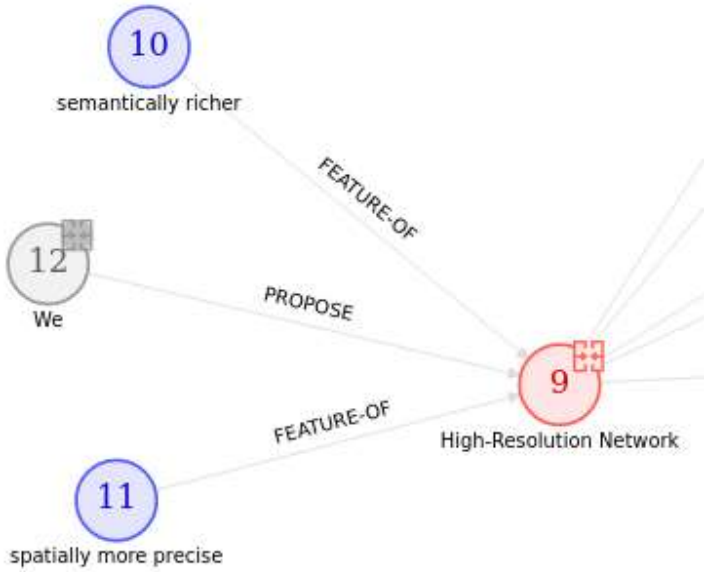
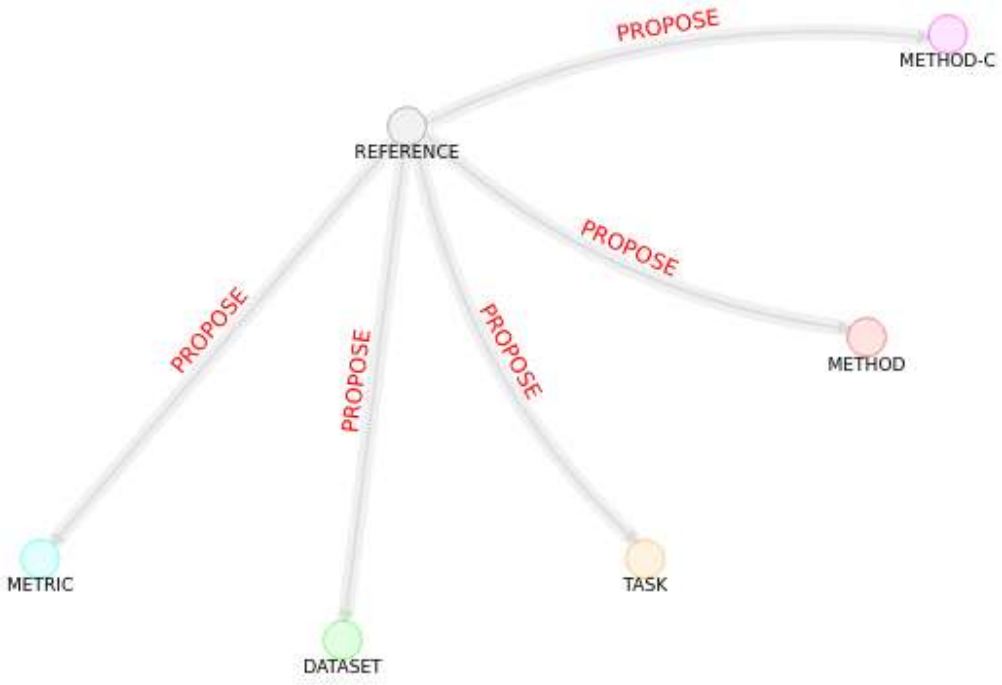
Schema	
۲	**Based-On**: When a (METHOD or METHOD-CLASS or ReFERENCE) is based on another one we annotated this relationship as BASED-ON
Example	Object detectors are usually equipped with backbone networks designed for image classification. It might be sub-optimal because of the gap between the tasks of image classification and object detection. In this work, we REFERENCE 3 present DetNAS METHOD 2 to use Neural Architecture Search METHOD 5 (NAS METHOD 5) for the design of better backbones for object detection TASK 6 . It is non-trivial because detection training typically needs ImageNet DATASET 7 pre-training while NAS METHOD 5 systems require accuracies on the target detection task as supervisory signals. Based on the technique of one-shot supernet METHOD 1 , which contains all possible networks in the search space, we REFERENCE 3 propose a framework for backbone search on object detection TASK 4 . We

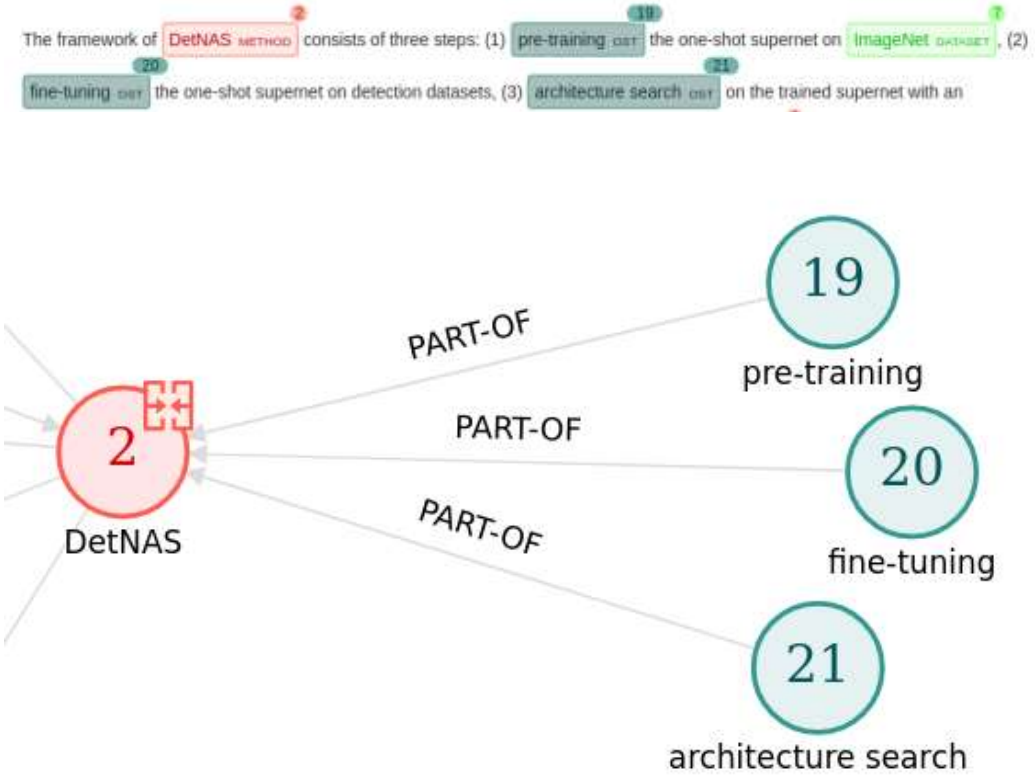

1
one-shot supernet
3
we
PROPOSE
IS-BASED-ON
2
DetNAS
USES
5
NAS
USED-FOR
USED-FOR
4
backbone search on object detection
6
object detection

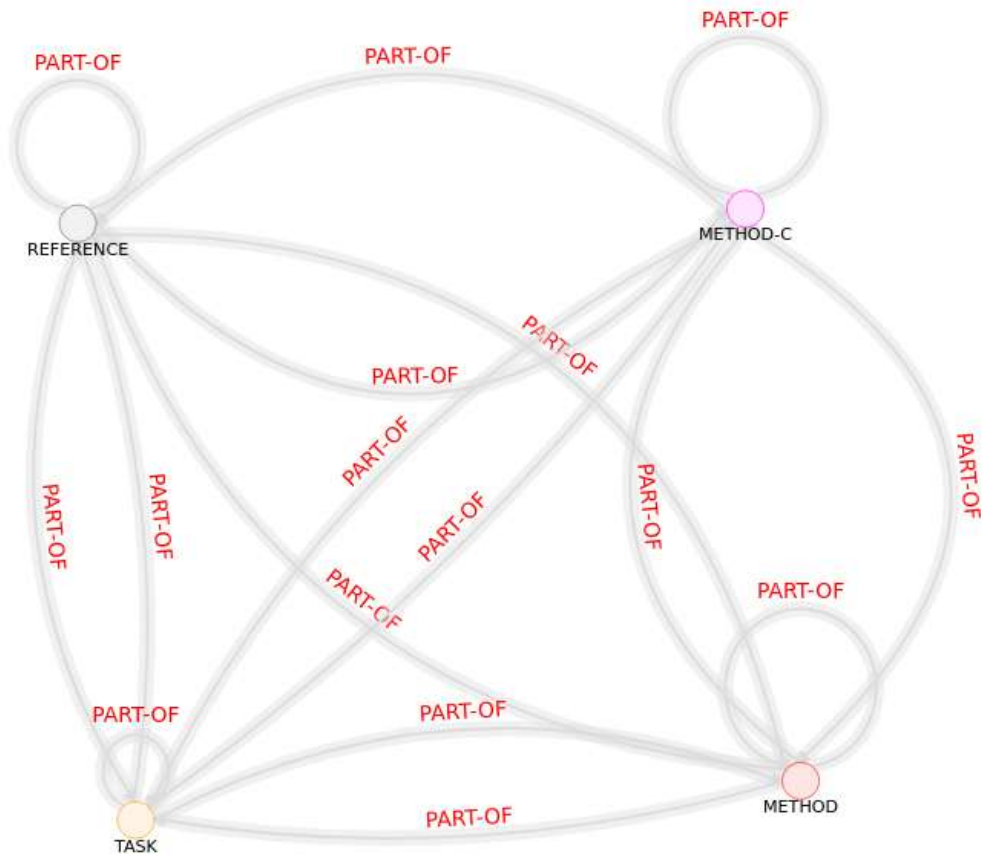
Schema	
۳	**Compare-Of:** When two **METHOD** or **METHOD-C** are comparing with each other then we annotate relation of COMPARE-OF for that two entities.
Example	Our main result architecture, DetNASNet METHOD 8, is searched on FPN in the large search space. The architecture of DetNASNet METHOD 8 is depicted in the supplementary material. We search on FPN because it is a mainstream two-stage detector that has been used in other vision tasks, e.g., instance segmentation TASK 9 and skeleton detection TASK 10 [9] . Table We list three hand-crafted networks for comparisons, including ResNet-50 METHOD 11, ResNet-101 METHOD 12 and ShuffleNetv2-40 METHOD 13.

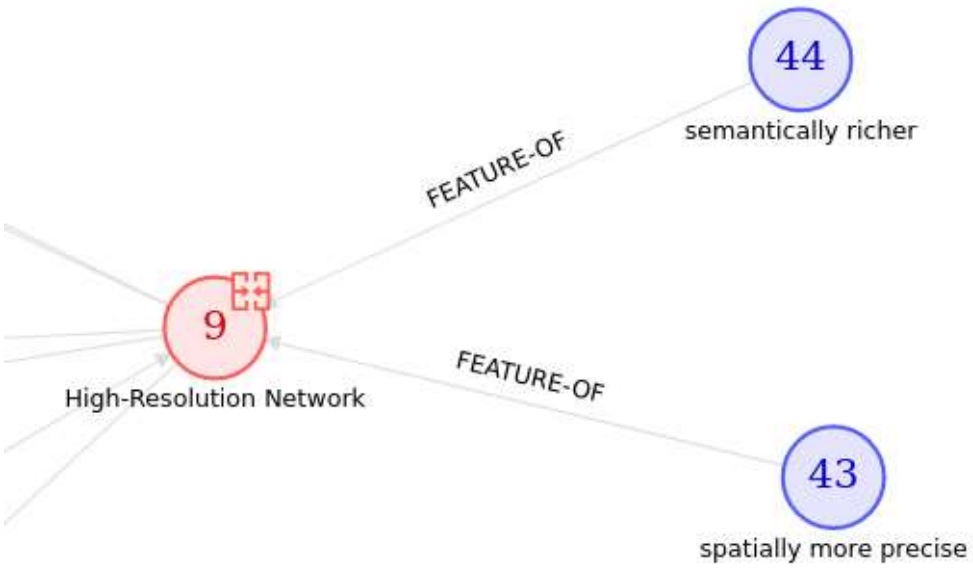
	
Schema	
۴	**Evaluated-ON:** Every **METHOD**, **METHOD-C**, **TASK** or **REFERENCE** could be evaluated on a **METRIC** or **DATASET**
Example	The framework of DetNAS METHOD 2 consists of three steps: (1) pre-training the one-shot supernet on ImageNet DATASET 7, (2) fine-tuning the one-shot supernet on detection datasets, (3) architecture search on the trained supernet with an evolutionary algorithm (EA). In experiments, the main result backbone network, DetNASNet METHOD 8, with much fewer FLOPs METRIC 17, achieves 2.9% NUMBER 18 better mmAP METRIC 14 than ResNet-50 on COCO DATASET 15 with the FPN detector METHOD 16. Its enlarged version,

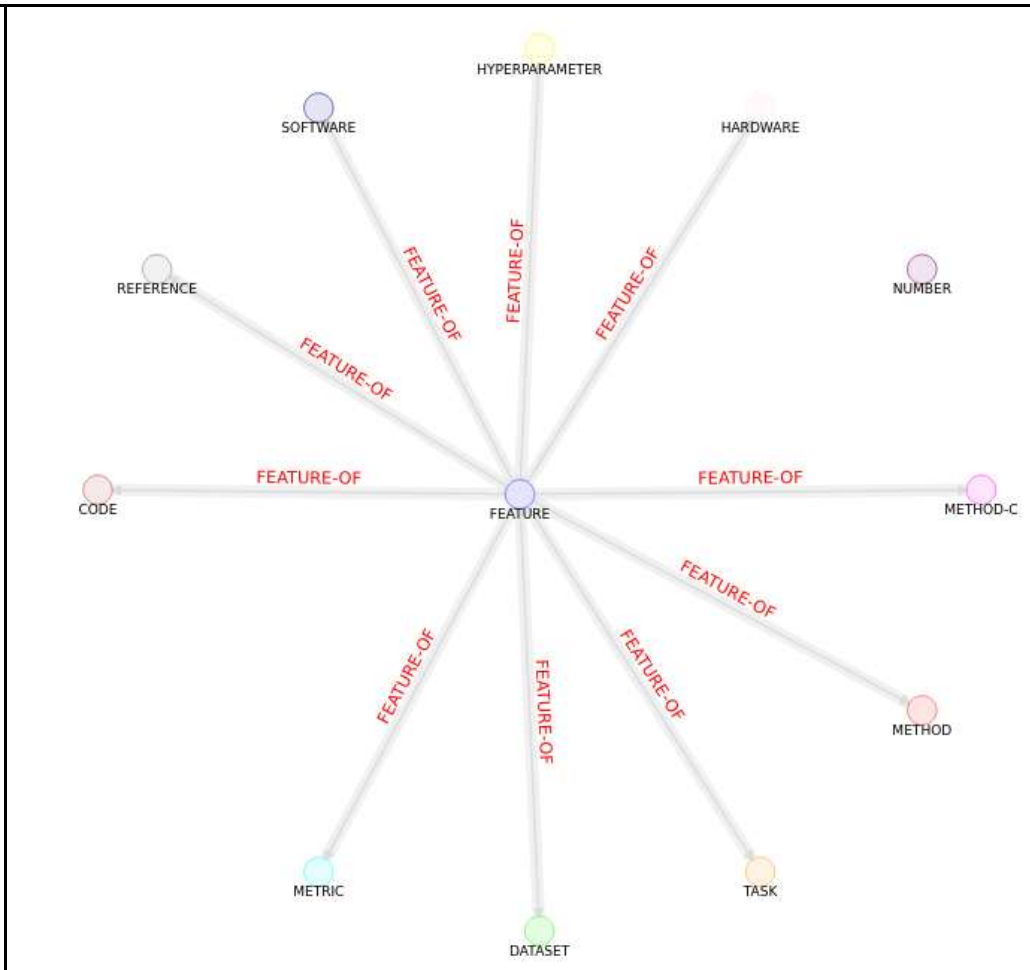
	
Schema	
∆	**PROPOSE:** If a REFERENCE propose a METHOD, DATASET, METRIC or TASK we have to annotate it as **PROPOSE** relationship
Example	benefit is that the resulting representation is semantically richer OST and spatially more precise OST . We REFERENCE show the superiority of the proposed HRNet METHOD in a wide range of applications, including human pose estimation TASK ,

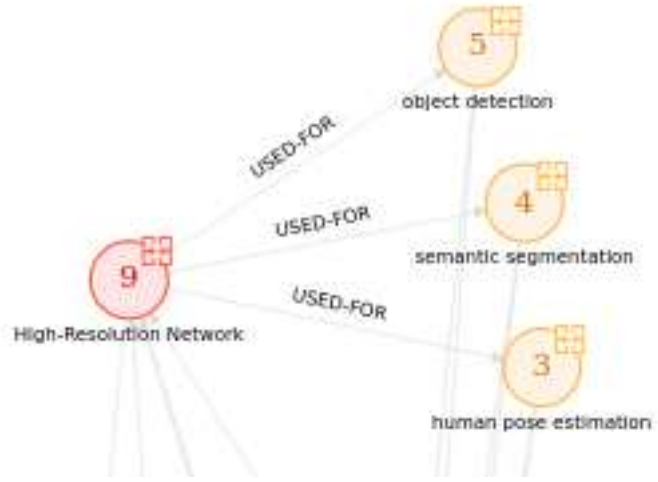
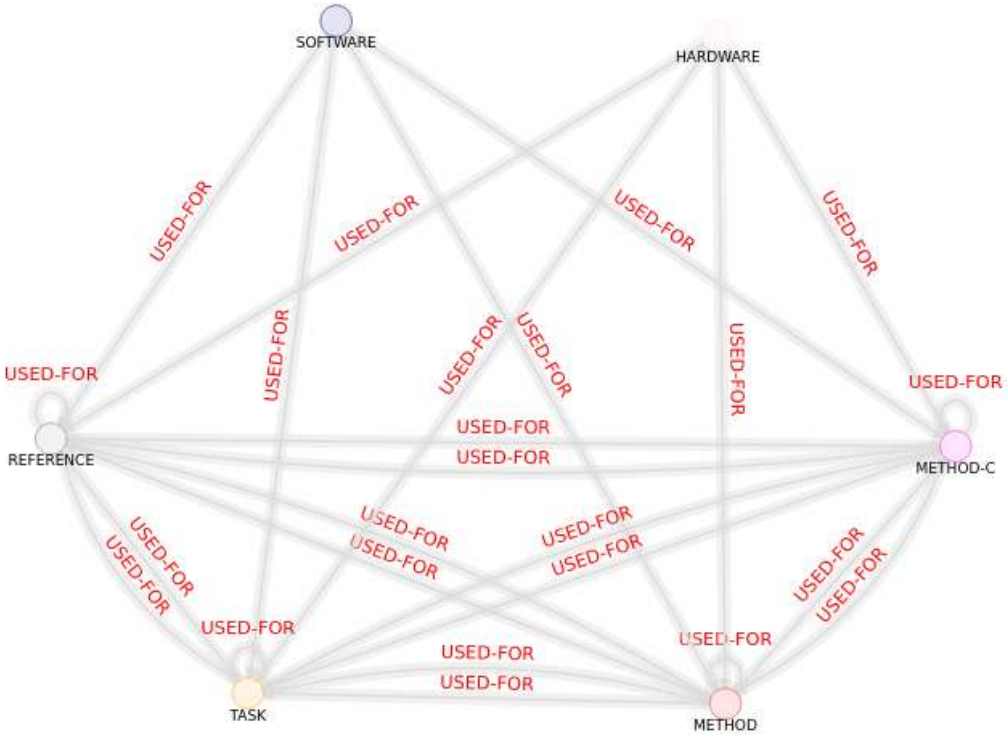
	
Schema	
۶	**PART-OF:** every METHOD, METHOD-C, REFERENCE, TASK and OST could be part of each other. For example when a method consists of several other methods then those methods are part of the main

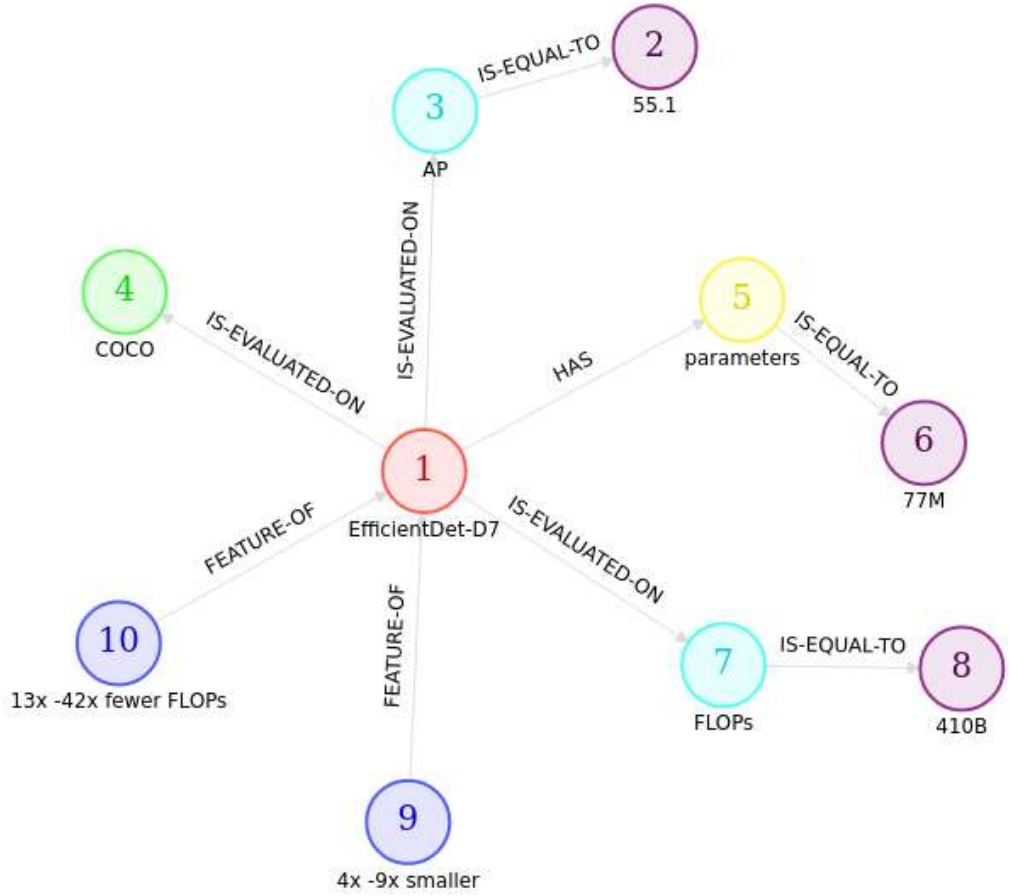
	method.
Example	

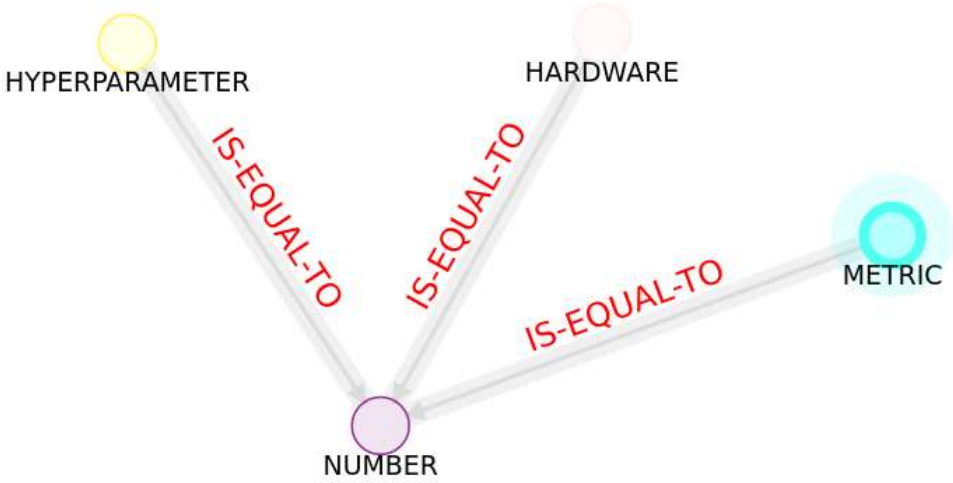
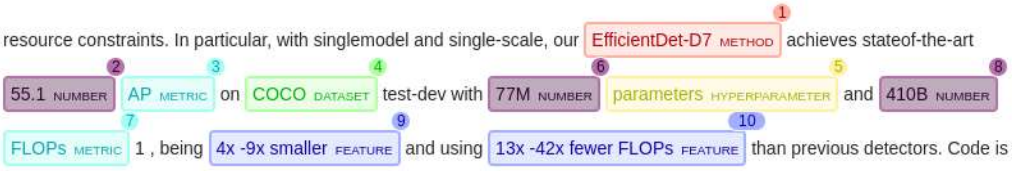
Schema	
V	**FEATURE-OF** The FEATURE entities are connected to all entities by relation of FEATURE-OF.
Example	encoded low-resolution representation. Instead, our proposed network, named as High-Resolution Network METHOD (HRNet METHOD), maintains high-resolution representations through the whole process. There are two key characteristics: (i) Connect the high-to-low resolution convolution streams in parallel; (ii) Repeatedly exchange the information across resolutions. The benefit is that the resulting representation is semantically richer FEATURE and spatially more precise FEATURE . We REFERENCE

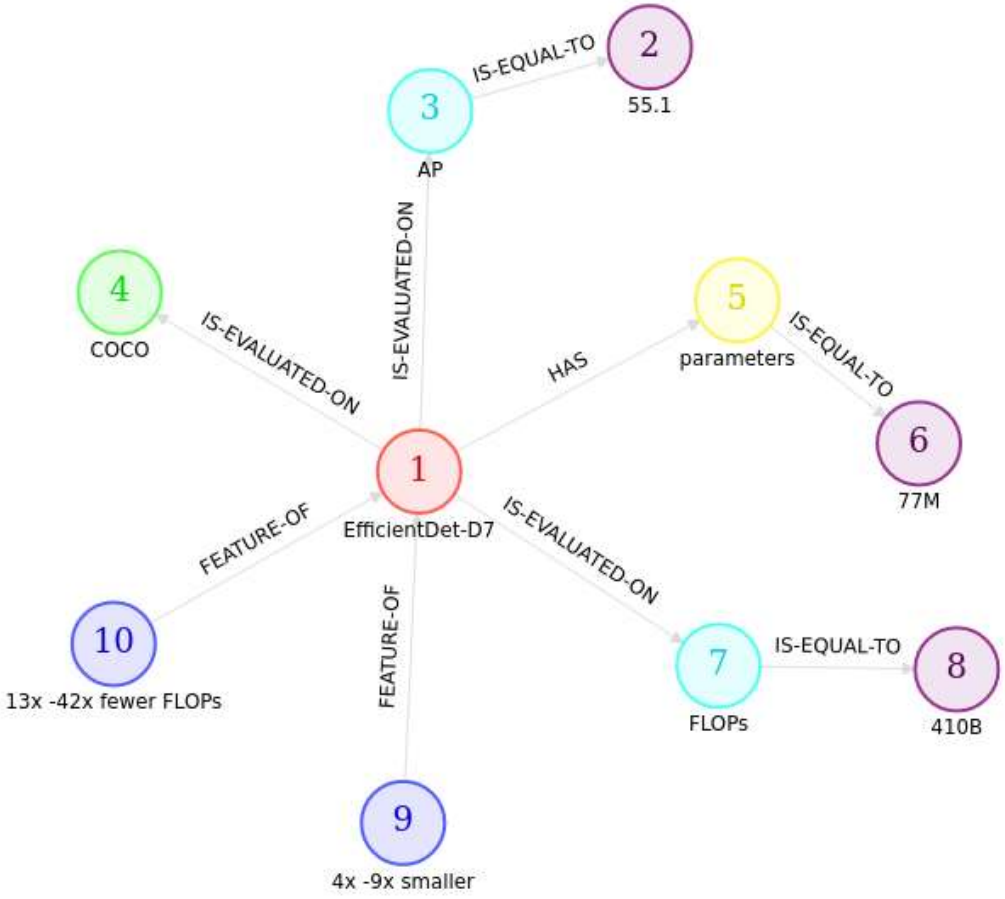

44
semantically richer
FEATURE-OF
9
High-Resolution Network
FEATURE-OF
43
spatially more precise

Schema	
^	**USED_FOR (Invert of USES)** When an entity uses another one we use this relationship. For entity types please see the schema of this relation.
Example	benefit is that the resulting representation is semantically richer OST and spatially more precise OST. We REFERENCE show the superiority of the proposed HRNet METHOD in a wide range of applications, including human pose estimation TASK, semantic segmentation TASK, and object detection TASK, suggesting that the HRNet METHOD is a stronger backbone for computer vision TASK problems. All the codes are available at https://github.com/HRNet CODE.

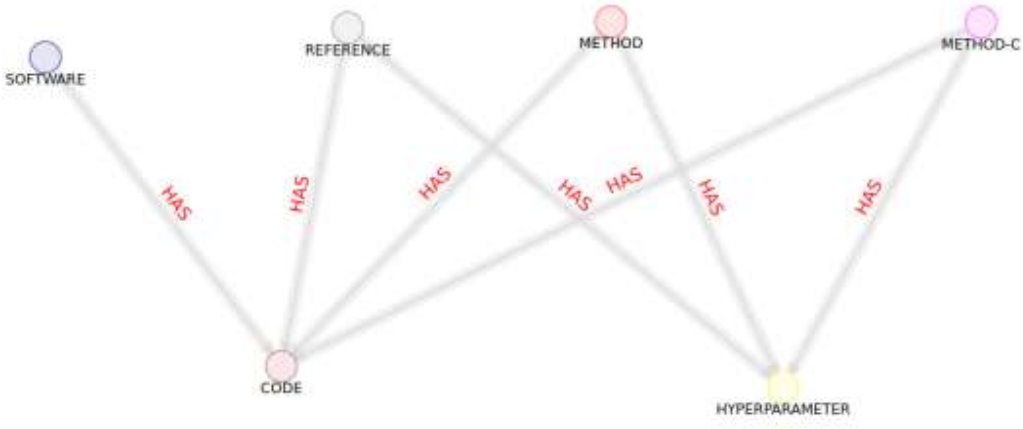
	
Schema	
۹	**IS-EQUAL-TO** This relation is used for numbers to assign to a HYPERPARAMETER, METRIC or HARDWARE

Example	resource constraints. In particular, with singlemodel and single-scale, our 1 EfficientDet-D7 METHOD achieves stateof-the-art 2 55.1 NUMBER 3 AP METRIC on 4 COCO DATASET test-dev with 6 77M NUMBER 5 parameters HYPERPARAMETER and 8 410B NUMBER 7 FLOPs METRIC 1 , being 9 4x -9x smaller FEATURE and using 10 13x -42x fewer FLOPs FEATURE than previous detectors. Code is 

Schema	
۱۰	**HAS** This relation is used to connect METHOD, METHOD-C, REFERENCE or SOFTWARE to CODE and HYPERPARAMETERS. (please see the schema)
Example	


2
55.1
IS-EQUAL-TO
3
AP
IS-EVALUATED-ON
4
COCO
IS-EVALUATED-ON
5
parameters
HAS
IS-EQUAL-TO
6
77M
1
EfficientDet-D7
FEATURE-OF
10
13x -42x fewer FLOPs
IS-EVALUATED-ON
FEATURE-OF
7
FLOPs
IS-EQUAL-TO
8
410B
9
4x -9x smaller

Schema


SOFTWARE
REFERENCE
METHOD
METHOD-C
HAS
HAS
HAS
HAS
HAS
HAS
HAS
CODE
HYPERPARAMETER