IEEE Content-Based Multimedia Indexing CBMI 2015

Interactive detection of incrementally learned concepts in images with ranking and semantic query interpretation

Klamer Schutte [1], Henri Bouma [1], John Schavemaker [1], Laura Daniele [1], Maya Sappelli [1,3], Gijs Koot [1], Pieter Eendebak [1], George Azzopardi [1,2], Martijn Spitters [1], Maaike de Boer [1,3], Maarten Kruithof [1], Paul Brandt [1,4]

[1] TNO, The Hague, The Netherlands
[2] RUG, Groningen, The Netherlands
[3] Radboud University, Nijmegen, The Netherlands
[4] TU Eindhoven, Eindhoven, The Netherlands

***Abstract*— The number of networked cameras is growing exponentially. Multiple applications in different domains result in an increasing need to search semantically over video sensor data. In this paper, we present the GOOSE demonstrator, which is a real-time general-purpose search engine that allows users to pose natural language queries to retrieve corresponding images. Top-down, this demonstrator interprets queries, which are presented as an intuitive graph to collect user feedback. Bottom-up, the system automatically recognizes and localizes concepts in images and it can incrementally learn novel concepts. A smart ranking combines both and allows effective retrieval of relevant images.**

Keywords—Semantic reasoning, concept detection, incremental learning

I. INTRODUCTION

The number of networked cameras is growing exponentially. Multiple applications in different domains result in an increasing need to search semantically over video sensor data. The GOOSE concept [5][12] aims to allow semantic search in such networked cameras. Typical applications might include domains as diverse as camera surveillance, traffic monitoring as well as social networking.

In contrast to standard internet search engines who are based on text tags found in a fairly static web, the GOOSE concept facilitates search on dynamic scene content as depicted by camera systems. To make this happen image content – i.e. a set of pixels – needs to be matched to words specified by the user in the search query. This challenge – the large difference in abstraction level between the raw data and the user query – can be viewed as an instance of the *semantic gap* [13]. Similar to internet search engines we envision the GOOSE system to allow many users to simultaneously pose different queries. In such a setup, any cost effective solution cannot afford that the number of users U evaluate imagery consisting of P pixels (possibly the subset of pixels needed to accomplish this task) in N different networked cameras, as that would lead to a total computation cost of $O(U \cdot P \cdot N)$. This is the *scalability* challenge. Application areas such as traffic monitoring, crowd control, urban warfare or burglary alarm services, all carry their own vocabulary, concepts of interest and specific assumptions on situational awareness. Domain independence requires flexible incremental learning of novel concepts for new situations.

In this paper, we present the GOOSE demonstrator system which addresses the semantic gap by computing for all incoming images the presence of a set of concepts, related attributes, and relations between the concepts. In the query domain the concepts map to verbs and nouns; their related attributes relate to adjectives; their cardinality relates to determiners; and the relations between to concepts to prepositions. At the same time this addresses the scalability challenge as it allows to calculate a concept-based imagery representation independent of the actual queries. This means we have computational costs in the camera part of the processing of $O(P \cdot N)$, followed by a query part of $O(U)$, which are both coupled by a common (distributed) database similar to the database used with regular internet search. To guarantee independence of application we take an approach that relies on run-time user-assisted incremental learning and online external resource query expansion, as opposed to domain-specific design-time preparation and concept training.

The GOOSE concept differs from earlier work in video tagging [8] and semantic browsing of videos [1][16] as it is aimed at processing live video and incremental learning of novel concepts rather than exploiting static video archives. Earlier ontologies such as LSCOM [9] often are limited to a fixed domain.

The outline of this paper on the GOOSE demonstrator is as follows: methods are described in Section 2, results are shown in Section 3, and conclusions are presented in Section 4.

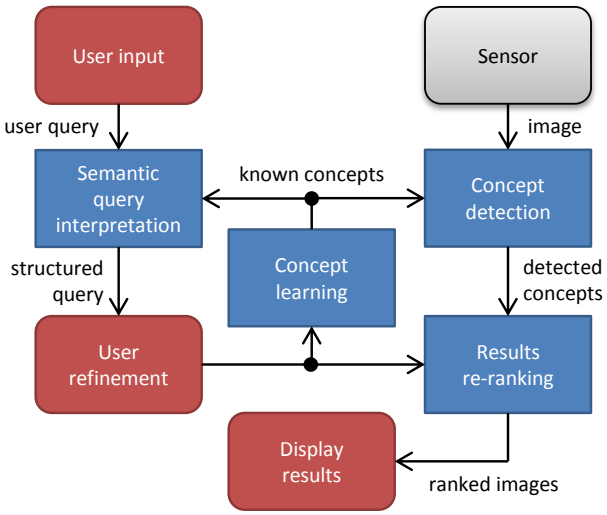

Fig. 1. System overview.

978-1-4673-6870-4/15/$31.00 © 2015 IEEE
http://dx.doi.org/10.1109/CBMI.2015.7153623

II. METHOD

The system consists of the following main components: semantic query interpretation, concept learning, concept detection, and user interaction and results re-ranking (see Fig. 1).

A. Semantic Query Interpretation

The aim of the semantic query interpretation is to transform the user query to a structured query that can either be mapped to known concepts or for which novel concepts become necessary. The semantic query interpretation consists of several modules. First, the user query is lexically analyzed using the Stanford Typed Dependency parser [9] that transforms the natural language query into a directed graph representation. In this graph representation, words of the query are transformed into nodes and grammatical relations into edge labels. A set of rules is applied to the lexical graph to generate a new graph in terms of objects, attributes, actions, scenes and relations – the semantic categories defined in our semantic meta-model. Details on this meta-model can be found in [3]. This semantic graph is matched against the concepts that can be detected by the concept-detection component in Fig. 1. If there is an exact match with known concepts, then the corresponding nodes in the graph are colored in green to indicate that these concepts have been recognized. If there is no exact match, then the semantic graph is expanded [2][4] using ConceptNet, which is a large knowledge base constructed by combining multiple web sources, such as DBpedia, Wiktionary and WordNet [14]. During the expansion, our algorithm matches the unknown concepts against concepts from ConceptNet. If a match is found, these concepts and their corresponding 'IsA' and 'Causes' relations are imported in our graph. If there is still no match, the expansion cycle is repeated a second time, where the results of the first cycle now are used as seed. Concept nodes that correspond to expansions with ConceptNet are colored in orange and concepts that are still unknown to the system after the expansion are colored red. An example of the colored semantic graph for the query 'Find a brown animal in front of a red mercedes' is shown in Fig. 2. This graph is the final output of the semantic query interpretation and corresponds to the *structured query* that is input to the *User refinement* component in Fig. 1.

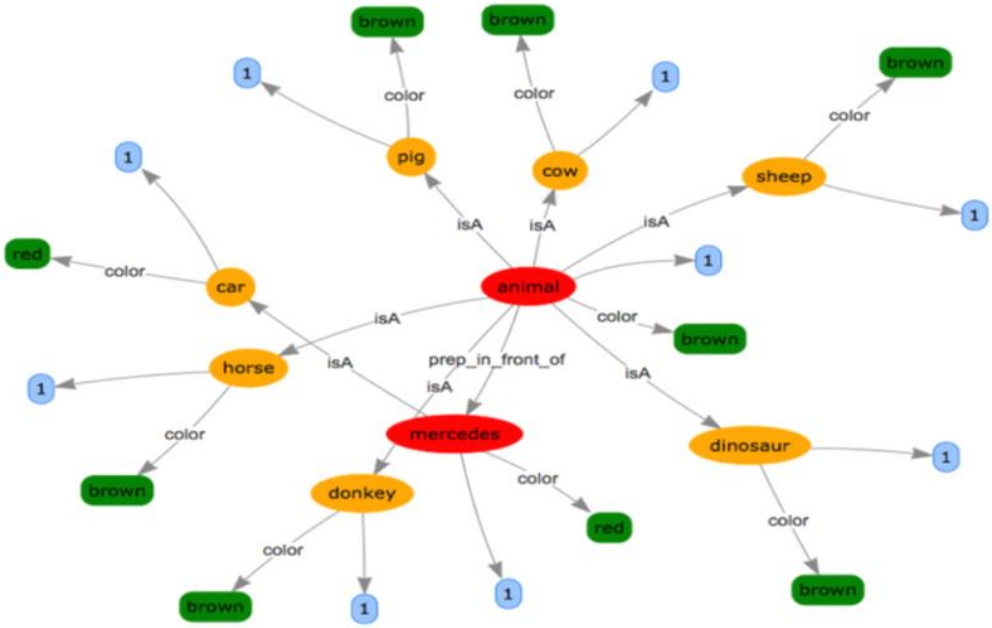

Fig. 2 Semantic expansion of example query.

B. Concept Learning and Concept Detection

Concept detection makes the transition from pixels in the image to concepts that can be interpreted semantically. In each of the images, concepts are recognized and spatially localized in the image. At initialization, there is a collection of known concepts that are pre-trained on average 100 training images per concept, which are available in the TOSO (toy and office-supplies objects) dataset[1]. The concept detection and localization is based on the R-CNN [7]. The system takes an input image, extracts bottom-up region candidates, computes features for each candidate using a large convolutional neural network (CNN) and then classifies each region using class-specific linear support vector machines (SVMs) for the recognition of the concepts. We map the SVM output to a confidence score by a mapping function, where the confidence score predicts the expected concept precision.

When the user is interested in a novel concept, the system can learn these novel concepts incrementally in an online learning mode. Novel concepts will initially have only a limited number of positive training samples (typically 4 to 10) recorded by the user. The user can add images with a bounding box and a concept label annotation for the training of new concepts. Negatives are taken from samples of all previously trained concepts. For a new concept we train a new one-to-the-rest SVM classifier, where the CNN 4096 element output proved sufficient general to support new concept classes. Hard negative mining is applied to address the unbalance in number of positive and negative training images; resulting unbalance leads to biased SVM scores. This bias is corrected by increasing the SVM-weights of the classes with a low number of positive examples. Training time for a novel concept is dominated by calculation of the region candidates within the training images and subsequent application of the CNN to the region candidates. More details about incremental concept recognition and concept learning with few examples are described in [6].

[1] https://www.researchgate.net/publication/275347805_TOSO-dataset. DOI: 10.13140/RG.2.1.1219.0248

C. User Interaction and Results Re-ranking

The GOOSE concept is a user-centric search concept. The user can provide queries in a natural language, which is transformed in a structured query by the query-expansion module. Because natural language is ambiguous by nature and the number of known concepts in the system is limited, the user can refine queries, select the proper expansion or learn novel concepts. Furthermore, the user can influence the ranking of results by selecting good results and deleting bad ones. Based on these choices, the system can learn about the user and the context. This allows the system to handle ambiguities and adjust to subjective aspects. For example, the query "find a black dangerous animal" maybe interpreted differently by different people because the concept of a dangerous animal may differ per user [11].

In order to bootstrap user ranking, a baseline ranking is implemented that fulfills a basic way of ranking search results. In Table I, all dimensions considered when ranking results with respect to the query are listed.

All images containing one of the requested concepts are retrieved unless it contains a concept that is negated. The ranking of these results is then based on a penalty system. The penalties are calculated by comparing detected concepts in an image to requested concepts in a query. A first penalty is given based on the confidence of a detection; a penalty of (1-confidence) is given to create a preference for images with higher confident detections. Second, a penalty is assigned for each concept attribute in the image that is missing or not as requested by the query. Finally, a penalty is given for missing concepts in the images. This includes a penalty for images that have too few instances of a requested concept with a certain cardinality. If an image contains more instances of a concept than requested, the best scoring concept instance is used to determine the score for the image.

TABLE I. RANKING CONSTRUCTS

Construct	Example
Concept	an animal and a mercedes
Attributes	the mercedes should be red
Relationships	the animal should be in front of the mercedes
Cardinality	one animal and two cars
Negation	a car but not a road

D. Integration of the Live Demonstrator

For validation of the generic GOOSE search-engine approach, a demonstrator is built. This demonstrator implements two search modes. The first is a historical search, operating on stored images. The second mode implements (near) real-time search, where queries are active over incoming video streams [15] and the user is alerted on matching imagery.

The database and many of the processes listed in Table II, as well as their communication protocols and the libraries are state-of-the-art mainstream components for building general web-applications. Most processes have been deployed on a single server, except for the GUI process – which runs in the browser – and the concept detection – which can be distributed over a cluster of computers scaling with the amount of connected cameras. For small-scale demonstrations, one computer is sufficient. There were no performance issues in the webserver under the load that we tried, ingesting and persisting up to 10 images per second, while also comparing them to all alerts set by users. It has to be noted though, that this excludes the object detection and learning. Detection time for a single image is a couple of seconds (depending on image size and content). The learning time for each additional concept (with 10 example images) is about 1 minute. This process is typically CPU or GPU-bound, which makes it a good candidate for deployment on a cluster. Current query time, without optimization and 77000 detections in the database takes up to 200 ms including network roundtrips. We expect this to scale well up to millions of entries without needing more than some simple optimizations.

Most communication is done over HTTP, using JSON encoding, with two notable exceptions. First, the alerting / notification feature of the GUI does not fit well with the classical REST model, so instead SocketIO technology is used, as compatibility layer for the emerging WebSocket technology. This allows for active push capabilities from the server and the database communications follows the specific protocol of MongoDB. The concept features are stored as JSON files in the MongoDB server, alternatively the database can be stored using a traditional relational database.

TABLE II. PROCESSES AND TECHNOLOGY

Process	Technology	Responsibility
GUI	AngularJS, Bootstrap, SocketIO	Monitoring and querying.
Webserver	Flask, MongoKit, SocketIO	Manage parser, handle queries and ranking.
Database	MongoDB	Store detections, images, queries, concepts, attrib.
Parser	Java, Stanford parser	Parsing natural language into structured format.
Concept detection	Matlab, R-CNN, C/Mex, python.	Detect and localize concepts in images.
Concept learning	Matlab, libSVM, R-CNN, C/Mex, python	Online learning of additional concepts.
Camera GUI	Python, OpenCV	Image acquisition.

III. RESULTS

A. Demonstrator Capabilities

The GOOSE system is deployed as a so-called table-top demonstrator. This portable set-up allows an interactive query expansion, concept detection and re-ranking of results. As a baseline, the system starts with 36 known concepts that are easy to transport for demonstration purposes, including office supplies, e.g., stapler and computer mouse, and toys, e.g., car, horse, donkey, cow, barbie, airplane, motorcycle and many more. People can put one or multiple of the known objects in front of the camera and the system will be able to recognize them with high mean average precision [6]. The images are

stored in the database and attendees will be able to define complex natural language queries. The system will present the structured query and the ranked resulting images. Furthermore, we can learn novel concepts, such as a conference badge, on-the-fly during the demonstration.

B. *Example User Query and Ranked Results*

The user can type the natural-language query in the input dialog, as shown in Fig. 3.

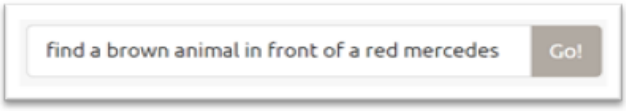

Fig. 3. Demonstrator interface with query example.

The user query is converted to a structured query and the concepts are mapped to known concepts. An example expansion is shown in Fig 2. in section II.A. Nodes are colored green if a direct match is found with known concepts (without expansion), they are orange if an indirect match was possible after expansion in ConceptNet, and they are red if matching was not possible. The cardinality is shown in blue nodes. We experienced that the expansion may include a transition from specific to generic, e.g. mercedes expands to car) or vice versa (e.g. animal expands to donkey, horse, pig etc.). The directional part of the structured query of Fig. 2 therefore relates to the expansion direction, and not an ontological direction (e.g., specialization). The shown structured query contains information about a car with cardinality one and color red, and several examples of animals with cardinality one and color brown. Since attributes depend on their object, each expansion will carry the original object's attribute.

Finally, the resulting images are ranked by their normalized confidence score and presented to the user, as shown in Fig. 4. These images indeed show the presence of at least one red car (expanded from mercedes) and at least one brown horse or sheep (expanded from animal).

IV. Conclusions

In this paper, we presented the GOOSE demonstrator, which is a search engine for images or video that allows users to pose natural language queries to retrieve corresponding images. The system interprets queries using query expansion using external resources, which are presented as an intuitive graph to collect user feedback. The system automatically recognizes and localizes concepts in video using state-of-the-art Deep Learning Convolutional Neural Networks and it can learn novel concepts on-the-fly. A smart re-ranking allows effective retrieval of relevant images.

Acknowledgment

This research was performed in the GOOSE project, which is jointly funded by the MIST research program of the Dutch Ministry of Defense and the AMSN enabling technology program.

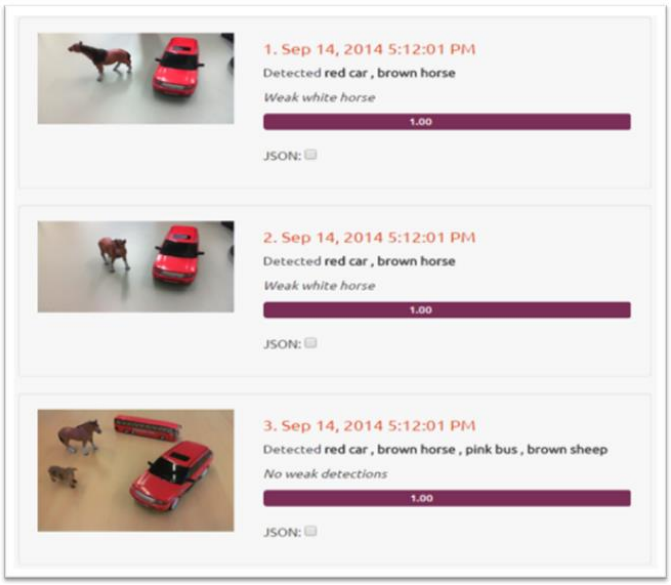

Fig. 4. First three ranked results for the example query.

References

[1] Bertini, M. et al., "Web-based semantic browsing of video collections using multimedia ontologies," ACM MM 2010

[2] Bhogal, J., Macfarlane, A. Smith, P., "A review of ontology based query expansion,", Information Processing&Management, 43(4),866-886,2007

[3] Boer, M. de, Daniele, L., Brandt, P., and Sappelli, M., "Applying semantic reasoning in image retrieval," Proc. ALLDATA 2015: The First International Conference on Big Data, Small Data, Linked Data and Open Data, pp. 69-74. IARIA, 2015.

[4] Boer, M. de, Schutte, K., and Kraaij, W., "Knowledge based query expansion in complex multimedia event detection," submitted MTAP 2015.

[5] Bouma, H., Azzopardi, G., Spitters, M., et al., "TNO at TRECVID 2013: multimedia event detection and instance search," Proc. TRECVID 2013.

[6] Bouma, H. et al., "Incremental concept learning with few training examples and hierarchical classification," Optics and Photonics for Counterterrorism, Crime Fighting and Defence, Proc. SPIE 2015.

[7] Girshik, R., Donahue, J., Darrell, and T., Malik, J., "Rich feature hierarchies for accurate object detection and semantic segmentation," IEEE CVPR 2014.

[8] Larson, M., Soleymani, M., Serdyokov, P., "Automatic tagging and geotagging in Video Collections and Communities,", ACM ICMR 2011

[9] Marneffe, M. de, MacCartney, B., and Manning, C., "Generating typed dependency parses from phrase structure parses," LREC 2006.

[10] Naphade, M. et al., "Large-Scale Concept Ontology for Multimedia," IEEE Multimedia, 13(3), 86-91, 2006.

[11] Schavemaker, J., Spitters, M., Koot, G., and Sappelli, M., "Fast re-ranking of visual search results by example selection", unpublished.

[12] Schutte, K. et al., "GOOSE: semantic search on internet connected sensors," Next-Generation Analyst, Proc. SPIE Vol. 8758 2013.

[13] Smeulders, A. et al., . "Content based image retrieval at the end of the early years," IEEE Trans. PAMI. 22(12), 1349–1380, 2000.

[14] Speer, R., and Havasi, C., "Representing general relational knowledge in ConceptNet 5," LREC, 3679-3686, 2012.

[15] Versloot, C., Kruithof, M., and Schavemaker, J., "StormCV: a scalable, computer-vision platform," unpublished.

[16] Wei, X-Y., Ngo, C-W., "Ontology-Enriched Semantic Space for Video Search,", ACM Int. Conf. on Multimedia 2007