

FINDING VIPS - A VISUAL IMAGE PERSONS SEARCH USING A CONTENT PROPERTY REASONER AND WEB ONTOLOGY

Edward Kim, Xiaolei Huang, Jeff Heflin

Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 18015
{edk208,xih206}@lehigh.edu, {heflin}@cse.lehigh.edu

ABSTRACT

We present a semantic based search tool, VIPs, i.e. Visual Image Persons Search, on the domain of VIPs, i.e. very important people. Our tool explores the possibilities of content based image search supported by ontological reasoning. Our framework integrates information from both image processing algorithms and semantic knowledge bases to perform interesting queries that would otherwise be impossible. We describe a novel property reasoner that is able to translate low level image features into semantically relevant object properties. Finally, we demonstrate interesting searches supported by our framework on the domain of people, the majority of whom are movie celebrities, using the properties translated by our system as well as existing ontologies available on the web.

Index Terms— Content based image retrieval, ontology, semantic search, scalable vector graphics

1. INTRODUCTION

Effectively searching through a large collection of images is a very difficult problem. Current efficient methods of image search rely on textual cues such as image tags or filenames for indexing and retrieval. However, ideally, one would be able to search through images based upon the content within the image, e.g. content based image retrieval (CBIR). Advances have been made in the area of CBIR; however, there still exists a semantic gap between the low level features (color, texture, shape, etc.) and the high level concepts that are represented by these features [1]. To address this problem, we propose a system, VIPs, that, (1) defines a visual person ontology, (2) translates low level image features to semantic properties and, (3) links image content to existing web ontological data. Using our system on a database of celebrity images, we can perform a variety of interesting semantic image searches. An illustrative example can be seen in Figure 1.

The combination of image processing techniques and semantic technologies has become increasingly popular in recent years. One interesting approach towards the integration of image processing and semantics is the categorization of an image, e.g. as an indoor or outdoor scene, or annotation of

an object, e.g. a tiger, elephant, bird, etc. [2, 3, 4]. Each category would have its own classifier and the system could pick the most likely result. However, this can quickly become intractable in complex real world applications where the number of categories is unbounded. In a different approach, other works have attempted to translate spatial information into semantic data so one can perform spatially based image queries [5, 6]. These works typically rely on accurate segmentations or object annotations. Other methods only use basic semantics (blobs of colored regions) and search based upon their spatial layout [7].

Although these related works on annotation and spatial search have motivated our research, we view the problem from a different perspective. Instead of the traditional use of ontologies to aid the classification of images, we use images to enhance the capabilities of semantic search. We primarily focus on extracting semantically relevant properties or attributes about a region, similar to Farhadi *et al.* [8] and Kumar *et al.* [9]. After extracting these properties, we are able to build an image ontology and populate it with semantic image features. Finally, we link our ontology to extensive resources that exist on the web. To create this link, we observe



Fig. 1. We propose a framework to retrieve pictures using information obtained from image processing and knowledge bases. For example, a possible search might be, “actor from the movie ‘Fight Club’ wearing a bowtie next to his partner” instead of the traditional keyword search, “Brad Pitt and Angelina Jolie”. Image courtesy of Georges Biard.

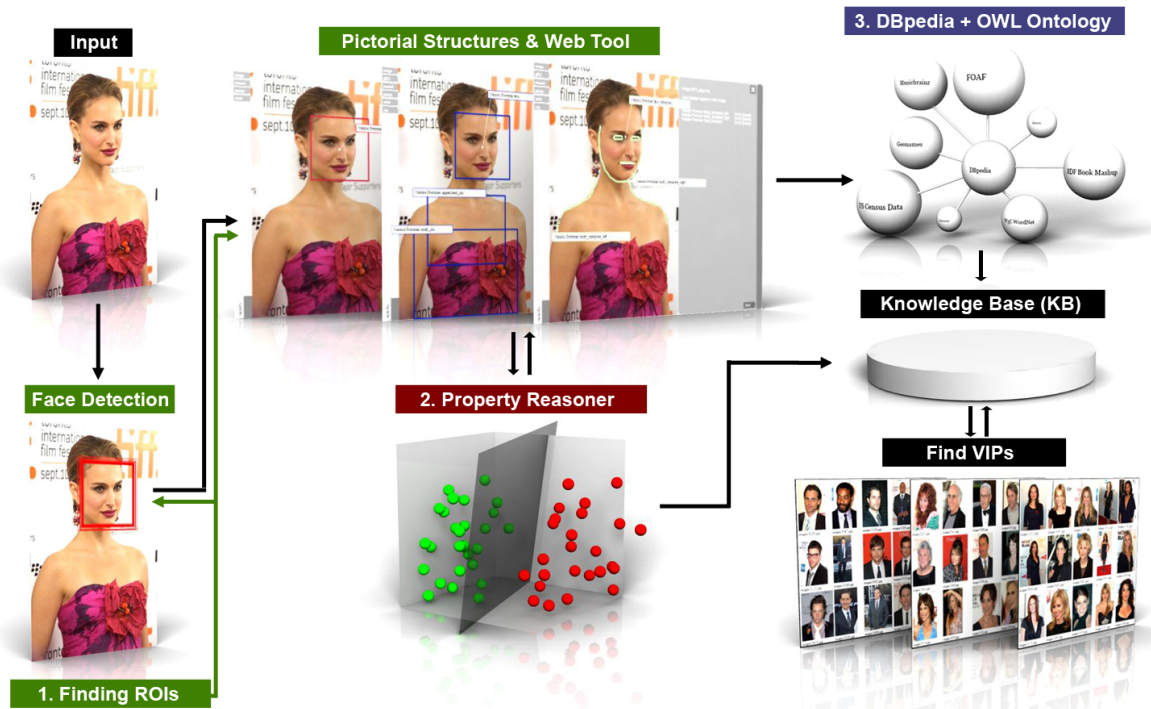


Fig. 2. Overall framework of our VIPs system. In step 1, we find the regions of interest (ROIs) within the image. In step 2, we perform reasoning on the ROIs using our Property Reasoner. In step 3, we map our properties to the DBpedia Ontology. Lastly, we build a knowledge base (KB) from each component and use the KB for VIPs queries. Photograph of Natalie Portman courtesy of makoto2007 from Wikimedia Commons.

in many image collections, there is some tagging information available, whether it be the filename, image tag, or caption text [10]. This tagging information can be used to connect an image to an external ontology. Given this framework, we can perform queries based upon high level relational concepts rather than using traditional keyword matching techniques.

Formally, our contributions are twofold. First, we propose a framework that uses image processing techniques to locate regions of interest within an image and extract semantically relevant descriptions based on these regions. This functionality is defined by a novel property reasoner described in detail within the body of this paper. Second, we build a framework that uses custom and existing ontologies to link semantically rich data to the image regions for users to exploit in their queries. For this process, we utilize a popular semantic language OWL, i.e. Web Ontology Language. OWL provides well defined semantics that enable algorithms to infer logical consequences from facts or axioms. Due to the growth of the Semantic Web, tens of thousands of OWL ontologies are currently available.

2. METHODOLOGY

2.1. VIPs Overall Image Framework

In the first step of our VIPs system, we develop image processing methods to locate regions of interest (ROIs) in an im-

age, e.g. face, torso, etc. In the second step, we propose a property reasoner that specifically translates low level image features extracted from ROIs into semantic concepts, e.g. person is facing to the left, or wearing a tie. These semantic concepts can be related to OWL object and datatype properties. OWL object properties are relations between classes whereas datatype properties are relations between classes and RDF literals or XML schema datatypes. In the third step, we develop a person-centric image ontology that imports existing web ontologies. We choose to work with DBpedia [11], an online ontology that structures information taken from Wikipedia and connects with external ontologies. As of April 2010, DBpedia contained nearly 5 million interlinks to these external datasets. Finally, given our generated knowledge base, we can perform semantic content based image searches within our database of VIPs. In this section, we describe the overall framework in more detail. An illustration of the VIPs system can be seen in Figure 2.

2.2. Finding ROIs

Since we would like to visually describe a person given a picture, we need to find the person in the image, and also localize regions of interest that correspond to that person. Specifically, we look to place bounding boxes for 3 ROIs, the face, neck/upper chest, and torso.

Person finding using Face Detection- To locate the people within an image, we use Haar-like features in conjunction with the Viola-Jones face detector [12] to localize bounding boxes for faces. We run the detector over 20 scales and at each scale we output a vote for the face bounding box. The final bounding box is determined by the maximum voted area over all the scales. This process provides us with the seed location for our next process: pose detection.

Pose determination by Pictorial Structures- The second task is to calculate the general pose of the person so that we can accurately select the 3 ROIs. From the face detection location we initialize a pictorial structure to isolate the exact boundaries and orientations of the face and shoulders. A pictorial structure is a group of parts connected in a deformable configuration [13].

Our pictorial structure can be expressed as a graph, $G = (V, E)$, where the vertices $V = \{v_1, \dots, v_n\}$ are the parts and there is an edge $(v_i, v_j) \in E$ for each pair of connected parts. The locations of the parts are described by l_i , where each location element consists of five degrees of freedom, scale in x direction s_x , scale in y direction s_y , rotation θ , x, and y translation. In an image, the location of all the parts is represented by $L = \{l_1, \dots, l_n\}$. We consider three part templates, the face, left shoulder, and right shoulder, see Figure 3. The best pictorial structure match to an image has the following formulation,

$$L^* = \operatorname{argmin}_L \left(\sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j) + \sum_{v_i \in V} m_i(I_e, l_i) \right) \quad (1)$$

We define an image matching term, $m_i(I_e, l_i)$, that measures how well the templates match the image content. For this term, we use the chamfer distance between our templates to the sobel edges in the image, I_e . Mathematically speaking,

$$m_i(I_e, l_i) = \frac{1}{N_{t_i}} \sum_k \min_{e \in I_e} \|(t_{i,k} - e)\| \quad (2)$$

Where t_i is the point set of the i th part template at location l_i , N represents the total number of points to consider, $t_{i,k}$ is the k th point, and e is a sobel edge point in I_e .

Further, we add a deformation cost $d_{ij}(l_i, l_j)$, that penalizes the model when it deforms from the canonical layout. This deformation term is defined as the following,

$$\begin{aligned} d_{ij}(l_i, l_j) = & w_{ij}^\theta |(\theta_j - \theta_i) - \theta_{ij}| \\ & + w_{ij}^{s_x} |(\log s_{x_j} - \log s_{x_i}) - \log s_{x_{ij}}| \\ & + w_{ij}^{s_y} |(\log s_{y_j} - \log s_{y_i}) - \log s_{y_{ij}}| \\ & + w_{ij}^x |x_{ij} - x_{ji}| \\ & + w_{ij}^y |y_{ij} - y_{ji}| \end{aligned} \quad (3)$$

The first term penalizes the difference between the ideal relative angle of the two parts, θ_{ij} , and the observed relative

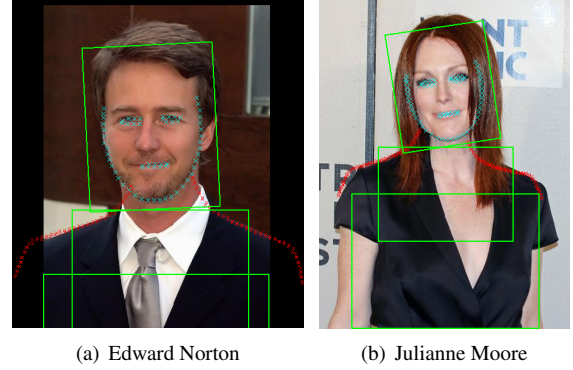


Fig. 3. Registration of pictorial structures on different images. The cyan points belong to our face template, the red points correspond to the shoulder templates, and the green boxes are the regions of interest computed from the pictorial structures. Photographs courtesy of [14].

angle. The second and third term penalize the difference between the ideal relative size and the observed relative size. And the fourth and fifth term penalize the distances in the x and y directions between the observed joint positions of the two parts. In our case, x_{ij} is the x position of the joint that connects part i to j in i 's frame of reference. We set the scale and rotation weights, $w_{ij}^{s_x}, w_{ij}^{s_y}, w_{ij}^\theta = 1$, and the translational weights to, $w_{ij}^x = 3 \times s_x, w_{ij}^y = 3 \times s_y$.

The search for the best location of all parts L^* (Eq. 1) is performed by a dynamic programming approach. Since our pictorial structure with three parts is a tree structure, the complexity of the search is $O(m^2n)$, where m is the number of discrete values for each l_i and $n (=3)$ is the number of parts. Because we limit m via our face detection phase, the registration of our face and shoulder templates is computed efficiently in a matter of seconds.

Once we have the part templates matched to the image, we can identify the three ROIs: face, neck/upper chest, and torso. The face ROI is the bounding box of the closed ellipse face template. The neck/upper chest ROI is transformed to fit between the midpoints of the shoulder templates, below the face ROI. The torso ROI is transformed to 80% of the width between the ends of the shoulder templates and translated halfway below the neck/upper chest ROI. We can use these ROIs to train classifiers in an online tool, or we can use these regions to perform property reasoning in the classification phase (Section 2.3).

Training by Web Tools- To train our system, we used a web based tool [15] that utilizes scalable vector graphics (SVG). Since SVG is an XML format for describing two-dimensional graphics, we are able to interact with and render our results in a standard web browser, see Figure 4. We build upon [15] to display the face detection, pictorial structures, and computed regions of interest within the image. Interaction is defined using Javascript, and the defined AJAX functions allow users

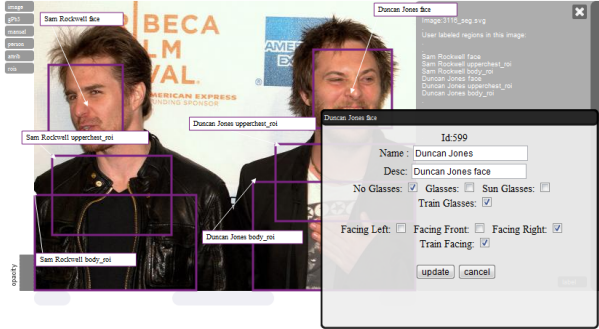


Fig. 4. Web based interface showing regions of interest computed by the pictorial structures. These ROIs are used for training our property reasoner. Photograph courtesy of [14].

to interactively label regions containing glasses/sunglasses, ties/bowties, etc. We can also specify whether or not this region should be included in our training set which will be used to train our property reasoner.

2.3. Property Reasoner

We can now formally describe a person by extracting semantic descriptions from ROIs. First, we define several object and data properties in OWL associated with the location of people in an image. Several spatial properties extracted are,

1. *rightOf/leftOf* - Our face detection stage provides the information necessary for these object properties. Given the centroid position of the face bounding boxes, we can use the relative 'x' pixel positions to determine that person 1 is to the *leftOf* person 2. Here, we can exploit OWL property relations by setting these two properties to be inverse properties, thus automatically inferring that person 2 must be to the *rightOf* person 1. These properties can also be set as transitive, further inferring that if person 2 is to the *leftOf* person 3, then person 1 must also be to the *leftOf* person 3.
2. *nextTo* - We define *nextTo* as being immediately adjacent to another person. If another person is in between two people, they will no longer be classified as *nextTo* one another. Similarly, this description is extracted using 'x' pixel centroid positions from bounding boxes.
3. *isLeftSide/isMiddle/isRightSide* - This is an absolute position property within an image. We split the image into thirds and calculate in which range the centroid position of the face bounding box lies. This object property is then applied to the individual.

For personal descriptions of people, such as what they are wearing or which direction they are facing, we extract information from ROIs corresponding to regions on a person's body. Our object properties are,

1. *isWearing* - Using our neck/upper chest and torso regions, we can train classifiers to detect certain articles of clothing. As an example, we train a classifier to determine if a person *isWearing* a tie, bowtie, or no tie. We use the same method on the face ROI to determine if the person *isWearing* sunglasses, glasses, or no glasses. This is accomplished using a Pyramid of Histograms of Oriented Gradients (PHOG) descriptor [16] with Support Vector Machines (SVM) [17]. For the PHOG descriptor, we extract three pyramid levels and at each level we build a histogram consisting of 8 orientations for a total feature vector length of 680.
2. *isFacing* - Given face ROIs, we can again use the PHOG descriptor with SVM classification to determine if the individual *isFacing* to the left, forward, or right.
3. *hasColor* - In order to name the color of an image region, we use a classification scheme based upon the 140 X11 named colors. For each ROI, we perform a K-means clustering of the perceptually uniform $L^*a^*b^*$ colorspace into 5 clusters. The most dominant cluster is then mapped to an X11 named color by finding the closest match or minimum distance using a Euclidean distance metric.

2.4. Persons Ontology KB

All of the image information extracted from photographs is collected and represented using vocabulary from our person ontology. Our person ontology is built in OWL using classes and object/data properties that incorporate the aforementioned spatial and personal properties extracted by our property reasoner. As an example, we introduce a new color class in OWL that organizes the X11 named colors into structured subclasses. Thus, "LimeGreen", "SpringGreen", and "ForestGreen" all become subclasses of the color "Green". This hierarchy allows a search on "Green" to incorporate various shades of green. Next, we import an external ontology, DBpedia, that exists on the Semantic Web. DBpedia provides a wealth of structured information that we can use for a database of people.

On top of these existing classes and properties, we define several additional class objects to organize our digital image collection. Our digital image collection consists of images, and tag information that name the people in each image.

1. *Image* - The *Image* type class contains the digital image instances in our dataset.
2. *PersonInstance* - This class is used to label people within images from the image tag/caption.

Next, we need to relate pairs of classes to each other. Each instance of the *PersonInstance* class should be related to exactly one *Image* and exactly one instance of the DBpedia *Person* class, although there can be many different *PersonInstances* in any *Image*, and many different *PersonInstances*

for each *Person*. To simplify the notation, two namespaces are defined, *vips* and *dbpedia*, in order to delineate which data instance belongs to which ontology. If we consider an image, *vips:3010.jpg* of type *Image*, which is a picture of *vips:3010_Meg_Ryan* of type *PersonInstance*, we can link resources by the following two pairs of object properties,

1. *inImage/hasPerson* - These object properties relate *vips:PersonInstance(s)* to an *vips:Image*. Thus, *vips:3010_Meg_Ryan* is *inImage vips:3010.jpg*, and *vips:3010.jpg hasPerson vips:3010_Meg_Ryan*.
2. *isPersonInstanceOf/hasPersonInstance* - These properties relate *vips* data to *dbpedia* information. Hence, *vips:3010_Meg_Ryan isPersonInstanceOf dbpedia:Meg_Ryan*, and vice versa with the *hasPersonInstance* property.

Naturally, these properties are defined as OWL inverses. A concrete OWL example is shown in Figure 5.

```
<owl:NamedIndividual rdf:about="&vips;3010_seg.svg">
  <rdf:type rdf:resource="&vips;Image"/>
  <vips:containsPerson rdf:resource="&vips;3010_seg.svg_Meg_Ryan"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="&vips;3010_seg.svg_Meg_Ryan">
  <rdf:type rdf:resource="&vips;PersonInstance"/>
  <vips:isPersonInstanceOf rdf:resource="&dbpedia;Meg_Ryan"/>
  <vips:inImage rdf:resource="&vips;3010_seg.svg"/>
  <vips:isWearing rdf:resource="&vips;noglasses"/>
  <vips:isFacing rdf:resource="&vips:right"/>
</owl:NamedIndividual>
```

Fig. 5. OWL snippet describing an image with our VIPs person ontology and linking this person to the DBpedia resource.

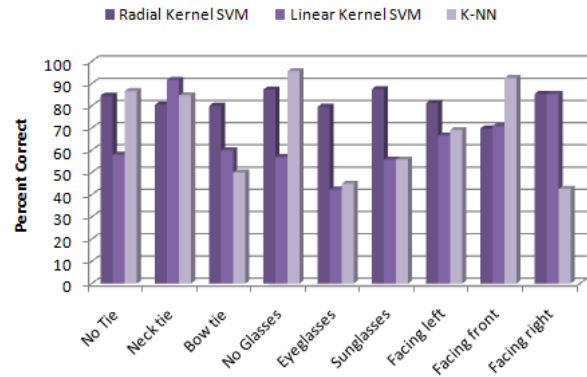
3. EXPERIMENTS AND RESULTS

3.1. Ontology experiments and results

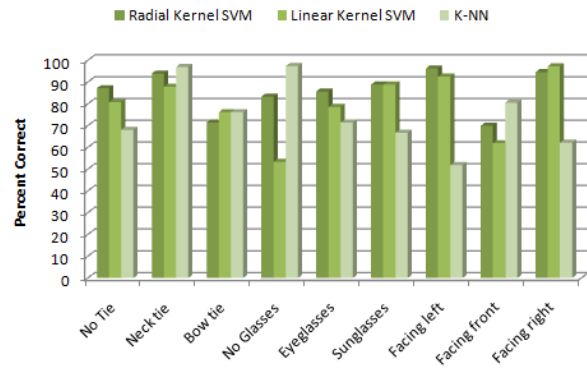
To test our system, we build two datasets of images. The first dataset consists of 2,010 images from the internet movie database (IMDb.com) image galleries. These images are a mix of face photos, half and full body shots, and also group pictures. From the captions on the IMDb webpage, we also collect the names of the people present in the image. Among the 2,010 images, we have 762 distinct celebrities, 16 of which are not able to be automatically mapped through DBpedia, resulting in a 97.9% success rate. The second dataset consists of 556 celebrity images obtained from the Wikimedia Commons gallery courtesy of D. Shankbone [14]. These images are similar to the IMDb dataset, name the people within the image, and are licensed under the Creative Commons Attribution. In this set, we count 437 distinct celebrities, 11 of which are not able to be automatically mapped through DBpedia, a 97.4% success rate. Ontology misses include names of people who do not appear in DBpedia, names that require disambiguation, e.g. Common (the musician), names that correspond to musical bands, or non specific labels such as “spouse”. Due to copyright issues, we will only display image results from this dataset and not the IMDb dataset.

3.2. Finding ROIs and Property Reasoner Accuracy

To describe the celebrities in the photographs, we first need to find the ROIs and then apply our property reasoner. Our face detection method correctly finds the faces within images 92.6% (Wikimedia Commons), 94.6% (IMDb) of the time. Of those correctly detected, our pictorial structures accurately localized the face/neck/body ROIs 93.5% (Wikimedia Commons), 94.3% (IMDb) of the time. Next, we perform a leave-one-out classification experiment to test the accuracy of our property reasoner on the ROIs. We compare the use of several classification methods - radial basis kernel SVM, linear kernel SVM, and a K-NN classifier. For our SVM classifier, we use a 5-fold cross validation method to automatically determine the best parameters. For our K-NN classifier, we use the minimum χ^2 distance between PHOG descriptors. On average, the radial basis kernel SVM works well across all categories, and so we choose this classifier for our system. These results can be seen in Figure 6.



(a) Wikimedia Commons Dataset



(b) IMDb Dataset

Fig. 6. Property reasoner accuracy on extracting semantic descriptions from ROIs. We compare three classification methods, RBF SVM, Linear SVM, and K-NN.



(a) Males wearing neckties born between 1970-1990. Query time is 0.96 sec.



(b) Married people facing to the left. Query time is 1.22 sec.



(c) Actresses wearing “black”, born in USA after Jan. 1, 1960. Query time is 1.02 sec.

Fig. 7. Sample queries and top 15 results from the VIPs system. Queries are composed in SPARQL and displayed in our web tool in a standard web browser. Ontology load time is 3.4 seconds. Photographs courtesy of [14].

3.3. Example Searches

Finally, we demonstrate several interesting queries possible by our VIPs system. We present four different queries that span across image and ontological boundaries.

1. *Query 1* - For our first query in Figure 7(a), we search for males who are wearing neckties and were born between January 1, 1970 and January 1, 1990. In this query, we use the *vips:isWearing vips:necktie* object property with *dbpedia:birthDate* data property.
2. *Query 2* - In our second query in Figure 7(b), we search for married people who are facing to the left. For this query, we use the *vips:isFacing vips:left* object property with *dbpedia:spouse* object property.
3. *Query 3* - In our third query, Figure 7(c), we search for actresses wearing black who were born in the United States after January 1, 1960. Here, we use *vips:hasColor vips:black* object property with *dbpedia:birthDate* and *dbpedia:birthPlace* properties.
4. *Query 4* - Our final query highlights our spatial capabilities. In Figure 8, we search for Actors from the movie “The Cable Guy”, not wearing a tie, to the left of their spouse. This search utilizes the *vips:isLeftOf vips:Person*, *vips:isWearing vips:notie*, *dbpedia:spouse*, and *dbpedia:starring* properties.

We report the ROC curve for each of our queries in Figure 9. From the ROC curve, it can be seen that our top matches very frequently belong to the true positive set. In Query 4, the VIPs system returns as the top two results both images that match the query constraints, thus resulting in perfect accuracy. However, errors may occur in the ontology matching, face detection, or property reasoning stages in our system.

These queries are executed using the ARQ SPARQL query engine in Jena and the performance in seconds is measured on a Dual-Core AMD Opteron processor with 4GB of memory. The SPARQL engine reads our OWL ontology with

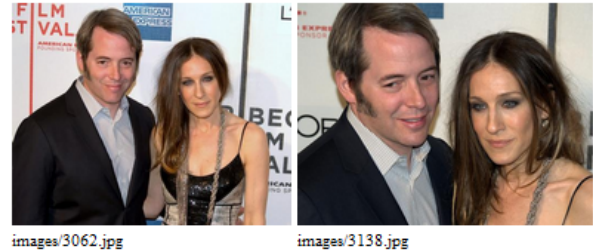


Fig. 8. Actors from the movie “The Cable Guy”, not wearing a tie, to the left of their spouse. Query takes 1.28 secs. Photographs courtesy of [14].

an approximately load time of 3-4 seconds. After the ontology is loaded, executing a query typically takes between 0.8 - 1.2 seconds.

4. CONCLUSION AND FUTURE WORK

In conclusion, we present the VIPs system, a visual image search system that spans image and ontology content. Our novel framework first extracts ROIs from image content and then uses a property reasoner to translate image features into semantic descriptions. We create a new OWL ontology based upon our descriptions and import existing ontologies, specifically, the DBpedia ontology to enhance our search capabilities. Finally, we explore previously impossible searches on a database of people images, ranging from spatial, clothing, color, birth dates, starring, and marital status searches.

In our experiments, we highlight some of the possible complex queries enabled by our system, albeit somewhat unrealistic for the common user. We plan to continue to build upon our system and incorporate more user centric queries and features. As a concrete example, we are utilizing the images tagged in Facebook, and extracting ontological data from user profiles to perform image searches within social

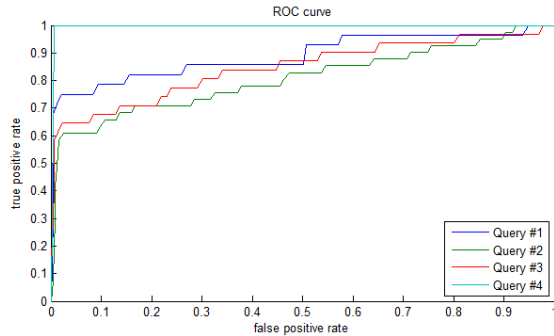


Fig. 9. ROC curves for the four example queries performed on the Wikimedia Commons Dataset.

networks. Additionally, we plan on enhancing our user interface to accommodate natural language queries. We also plan to further evaluate our search capabilities when scaling our database several orders of magnitude.

5. REFERENCES

- [1] Y. Liu, D. Zhang, G. Lu, and W.Y. Ma, “A survey of content-based image retrieval with high-level semantics,” *Pattern Recognition*, vol. 40, pp. 262–282, 2007.
- [2] L. Hollink, G. Schreiber, J. Wielemaker, and B. Wielinga, “Semantic annotation of image collections,” in *Knowledge Capture*, 2003, pp. 41–48.
- [3] G. Carneiro, A.B. Chan, P.J. Moreno, and N. Vasconcelos, “Supervised learning of semantic classes for image annotation and retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 394–410, 2007.
- [4] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V.K. Papatathis, and M.G. Strintzis, “Knowledge-assisted semantic video object detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1210–1224.
- [5] L. Hollink, G. Nguyen, G. Schreiber, J. Wielemaker, B. Wielinga, and M. Worring, “Adding spatial semantics to image annotations,” in *International Workshop on Knowledge Markup and Semantic Annotation*, 2004.
- [6] H.H. Wang, D. Mohamad, and NA Ismail, “Semantic Gap in CBIR: Automatic Objects Spatial relationships Semantic Extraction and Representation,” *International Journal of Image Processing*, vol. 4, no. 3, pp. 192, 2010.
- [7] J.R. Smith and S.F. Chang, “VisualSEEK: a fully automated content-based image query system,” in *ACM Multimedia*, 1997, pp. 87–98.
- [8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, “Describing objects by their attributes,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1778–1785.
- [9] N. Kumar, P. Belhumeur, and S. Nayar, “FaceTracer: A search engine for large collections of images with faces,” in *European Conference on Computer Vision*, 2008, pp. 340–353.
- [10] D. Demner-Fushman, S. Antani, M. Simpson, and G.R. Thoma, “Annotation and retrieval of clinically relevant images,” *International Journal of Medical Informatics*, vol. 78, no. 12, pp. e59–e67, 2009.
- [11] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, “Dbpedia—a crystallization point for the web of data,” *Web Semantics: Science, Services and Agents on the World Wide Web*, 2009.
- [12] P. Viola and M.J. Jones, “Robust real-time face detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [13] P.F. Felzenszwalb and D.P. Huttenlocher, “Pictorial structures for object recognition,” *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.
- [14] David Shankbone, “Creative Commons Photography,” http://en.wikipedia.org/wiki/User:David_Shankbone.
- [15] E. Kim, X. Huang, and G. Tan, “Markup SVG - An Online Content Aware Image Abstraction and Annotation Tool,” *IEEE Transactions on Multimedia*, 2011 (under revision).
- [16] A. Bosch, A. Zisserman, and X. Munoz, “Representing shape with a spatial pyramid kernel,” in *ACM International Conference on Image and Video Retrieval*, 2007, pp. 401–408.
- [17] Chih-Chung Chang and Chih-Jen Lin, “LIBSVM: a library for support vector machines,” 2001, Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.