A Visual Image Persons Search using Image Content and Web Ontologies

Edward Kim, Xiaolei Huang, and Jeff Heflin
Motivate the Problem

• Performing an image search...

• Google Image Search for “barack obama next to smiling michelle obama”
Motivate the Problem

- Performing an image search...
- Google Image Search for “44th president next to smiling first lady”
Motivate the Problem

- Performing an image search...

- Google Image Search for “43\textsuperscript{th} president next to smiling first lady”
Motivate the Problem

- Performing an image search...
- Google Image Search for “people wearing ties”
Motivate the Problem

- Performing an image search...
  - “males wearing ties born between 1970 and 1990”
Motivate the Problem

1. Ridiculous query...
2. Unfair, can’t expect good results...
3. Based on user filenames, tags...
4. Google image wasn’t built that way...

• But why?
  • “males wearing ties born between 1970 and 1990”

Can’t Image Processing & Machine learning techniques classify these?
Aren’t there social networks, wikipedia data that knows this?
Outline

• Motivate the Problem
• Introduce the VIPS system
• Methodology & Framework
  – Finding ROIs (regions of interest)
  – Classification in these regions via Property Reasoner
  – Building and linking to KBs (knowledge bases)
• Experiments and Results
  – Queries on several databases of celebrities
• Conclusion and Future Work
Introduction

• VIPS system
  – Visual Image Persons Search
  – Very Important People
• Our framework incorporates
  – Current image processing and computer vision techniques to extract semantically relevant properties
  – Build an ontology that is able to utilize external knowledge bases for complex queries

Query: “actor who starred in the movie ‘Fight Club’ wearing a bowtie next to his partner”
Methodology & Framework

1. Finding ROIs
2. Property Reasoner
3. DBpedia + OWL Ontology

Pictorial Structures & Web Tool

Image courtesy of makoto2007 from Wikimedia Commons
Finding ROIs

• End Goal: Localize 3 regions
  – Face region
  – Upper neck/chest area
  – Rest of the upper body

• Face detection using the Viola-Jones [1] face detector
  – Run over 20 different scales
  – Maximum voted area over all scales

Image courtesy of makoto2007 from Wikimedia Commons
Pose determination by Pictorial Structures [2]

• Simple face/left shoulder/right shoulder template

• Expressed as a graph, where
  – The vertices (v) are the parts
  – The edges (e) connect the parts

• If each part has a location (l) with 5 degrees of freedom (x,y translation, x,y scale, and rotation), then the best positions of the part is represented by,

\[ L^* = \arg\min_L \left( \sum_{(v_i,v_j) \in E} d_{ij}(l_i,l_j) + \sum_{v_i \in V} m_i(l_i) \right) \]
Pose determination by Pictorial Structures [2]

\[ L^* = \arg\min_L \left( \sum_{(v_i,v_j) \in E} d_{ij}(l_i, l_j) + \sum_{v_i \in V} m_i(I_e, l_i) \right) \]

• Where “m” represents the image matching term...

\[ m_i(I_e, l_i) = \frac{1}{N_{t_{l_i}}} \sum_k \min_{e \in I_e} \| (t_{l_i,k} - e) \| \]

Image courtesy of David Shankbone, “Creative Commons Photograpy”
Pose determination by Pictorial Structures [2]

\[ L^* = \arg \min_L \left( \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j) + \sum_{v_i \in V} m_i(I_e, l_i) \right) \]

- And “d” penalizes the model when it deforms from the canonical

\[ d_{ij}(l_i, l_j) = w_{ij}^\theta |(\theta_j - \theta_i) - \theta_{ij}| \]
\[ + w_{ij}^{sx} |(\log s_{xj} - \log s_{xi}) - \log s_{xij}| \]
\[ + w_{ij}^{sy} |(\log s_{yj} - \log s_{yi}) - \log s_{yij}| \]
\[ + w_{ij}^x |x_{ij} - x_{ji}| \]
\[ + w_{ij}^y |y_{ij} - y_{ji}| \]

Image courtesy of David Shankbone, “Creative Commons Photograpy”
Pose determination by Pictorial Structures [2]

• Solved via dynamic programming technique

• Isolate our 3 ROI bounding boxes
  – Face – Bounding box of face template
  – Neck/Upper chest – Midpoints of the shoulder templates, below the face
  – Body – 80% width of the shoulder templates, halfway below the upper chest

Image courtesy of David Shankbone, “Creative Commons Photograpy”
Face Bounding Boxes by ASM

\[ x \approx \bar{x} + Pb \]

\[ C = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T \]

\[ b = P^{-1}(x - \bar{x}) \]

Face points = mean face + Eigenvectors of Cov*parameters

Covariance Matrix

Shape parameters, Translation, scale, rotation
Semantic Web and Ontologies

- Ontology – “explicit and formal specification of a conceptualization” – R.Studer

![Diagram showing relationships between Animal, Carnivore, Herbivore, Lion, Antelope, Plant]

Diagram shows:
- Animal is-a Carnivore
- Animal is-a Herbivore
- Carnivore eats Lion
- Herbivore eats Antelope
- Carnivore is-a Plant
Person Ontology & Property Reasoner

• Formally describe an image using our custom ontology
  – Spatial Properties – use x,y position
    • rightOf/leftOf
    • nextTo
    • isLeftSide/isMiddle/isRightSide
    • isWearing
    • isFacing
    • hasColor
  – Ex. No tie, neck tie, bowtie
  – Ex. No glasses, eye glasses, sun glasses

Person Ontology Link to DBpedia

- DBpedia is an online ontology that structures information from Wikipedia
1. Background

Wikipedia has grown into one of the central knowledge sources of mankind and is maintained by thousands of contributors. Wikipedia articles consist mostly of free text, but also contain different types of structured information, such as infobox templates, categorisation information, images, geo-coordinates, and links to external Web pages. For instance, the figure below shows the source code and the visualisation of an infobox template containing structured information about the town of Innsbruck.

```json
{{Infobox Town AT |
  name = Innsbruck |
  image_coa = InnsbruckWappen.png |
  image_map = Karte-tirol-1.png |
  state = [[Tyrol]] |
  regbez = [[Statutory city]] |
  population = 117,342 |
  population_as_of = 2008 |
  pop_dens = 1,119 |
  area = 104.91 |
  elevation = 574 |
  lat_deg = 47 |
  lat_min = 16 |
  lat_dir = N |
  lon_deg = 11 |
  lon_min = 23 |
  lon_dir = E |
  postal_code = 6010-6080 |
  area_code = 0512 |
  licence = i |
  mayor = Hilde Zach |
  website = [http://innsbruck.at] |
}}
```

The DBpedia project extracts various kinds of structured information from Wikipedia editions in 37 languages and combines this information into
Person Ontology Link to DBpedia

- DBpedia is an online ontology that structures information from Wikipedia.

- We can link our ontology to DBpedia using OWL (Web Ontology Language)
  - `isPersonInstanceOf/hasPersonInstance`
  - `vips:3010_Meg_Ryan isPersonInstanceOf dbpedia:Meg_Ryan`

- Image to Instance link
  - `inImage/hasPerson`
  - `vips:3010_Meg_Ryan inImage vips:3010.jpg`
Experiments and Results

• Build 2 datasets on celebrity images
  – IMDb image gallery – 2,010 images (762)
  – Creative Commons image gallery – 556 images (437)
    • http://en.wikipedia.org/wiki/User:David_Shankbone
  – Each database identifies the person within the image
• Build a web interface using SVG/Javascript
  – Training classifiers
  – Visualization of results
Experiments and Results

• DBpedia matches
  – IMDb – 16 out of 762 missed – 97.9% success
  – Shankbone – 11 out of 437 missed - 97.4% success
  – Ex. Failures – “spouse”, “common”, music bands, etc.

• Viola-Jones face detection
  – IMDb – 94.6% accuracy
  – Shankbone – 92.6% accuracy
Experiments and Results

- Property Reasoner –
  - Tested on 3 classifiers
    - K-NN
    - Linear Kernel SVM
    - Radial Kernel SVM
Experiments and Results

- Query 1 – “males who are wearing neckties born between Jan 1, 1970 and Jan 1, 1990”
- Translated to SPARQL and run on the ARQ SPARQL query engine in Jena (dual core AMD 3.2 Opteron)
Experiments and Results

• Query 1 – “males who are wearing neckties born between Jan 1, 1970 and Jan 1, 1990”

PREFIX ont: <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#>

SELECT DISTINCT ?y
WHERE { ?x <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#isWearing> ont:necktie . 
  ?x <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#isPersonInstanceOf> ?person . 
  FILTER(?p <= "1990-01-01"^^<http://www.w3.org/2001/XMLSchema#date>) . 
}
Experiments and Results

• Query 2 – “married people facing to the left”

PREFIX ont: <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#>

SELECT DISTINCT ?y
WHERE {
  ?x <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#isFacing> ont:lei .
  ?x <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#isPersonInstanceOf> ?person .
}
Experiments and Results

• Query 3 – “actresses wearing black, born after January 1, 1960”

PREFIX ont: <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#>

SELECT ?y
WHERE {
  ?x <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#hasColor> ont:black .
  ?x <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#isPersonInstanceOf> ?person .
  FILTER(?p >= "1960-01-01"^^<http://www.w3.org/2001/XMLSchema#date>) .
}
Experiments and Results

• Query 4 – “actors not wearing a tie, starring in the moving “The Cable Guy” to the left of their spouse.”

PREFIX ont: <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#>
PREFIX resource: <http://dbpedia.org/resource/>

SELECT ?y
WHERE {
  ?actor <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#hasPersonInstance> ?person .
}
Experiments and Results

• Query 5 – “Grumpy old men.”

PREFIX ont: <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#>

SELECT ?y
WHERE {
  ?x <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#isSmiling> ont:notsmiling .
  ?x <http://idealab.cse.lehigh.edu/svg/icme2/vips.owl#isPersonInstanceOf> ?person .
  FILTER(?p <= "1960-01-01"^^<http://www.w3.org/2001/XMLSchema#date> ) .
}
Conclusion

• We proposed a system that spans across both image processing and semantic web technology
• Localize ROIs and extract semantic content
• Build a custom person ontology and link to existing knowledge bases on the web

• Future work
  – Utilize social networks for ontology information
  – Build a more “friendly” search method
Thank you!

• Questions?