Learning Beautiful (and ugly) attributes

Luca Marchesotti, Florent Perronnin

Xerox Research Centre Europe
20 Years of Innovation
Motivation
Motivation

\[ p(\text{like}) \]
Motivation

\[ p(\text{like}) \]

[San Pedro, WWW12][Murray, CVPR12] [Murray, BMVC12][Marchesotti, ICCV11] [Oliva, CVPR11][Berg, CVPR11][Nishiyama, CVPR11][Su, MM11][Bhattacharya, MM10] [Orendovici, MM10][Luo, ECCV08][Ke, CVPR06][Datta, ECCV06]
Motivation

Why?

[San Pedro, WWW12] [Murray, CVPR12] [Murray, BMVC12] [Marchesotti, ICCV11] [Oliva, CVPR11] [Berg, CVPR11] [Nishiyama, CVPR11] [Su, MM11] [Bhattacharya, MM10] [Orendovici, MM10] [Luo, ECCV08] [Ke, CVPR06] [Datta, ECCV06]
Limitations
Limitations

Hand-crafted features

[San Pedro, WWW12][Berg, CVPR11]
[Luo, ECCV08][Ke, CVPR06]
[Datta, ECCV06]
Limitations

Hand-crafted features

[San Pedro, WWW12][Berg, CVPR11]
[Luo, ECCV08][Ke, CVPR06]
[Datta, ECCV06]

Rule of 3rd, Color Harmony, etc.
Limitations

Hand-crafted features

[San Pedro, WWW12][Berg, CVPR11]
[Luo, ECCV08][Ke, CVPR06]
[Datta, ECCV06]

Rule of 3rd, Color Harmony, etc.

_similarity non-exhaustive
Limitations

Hand-crafted features

[San Pedro, WWW12][Berg, CVPR11]
[Luo, ECCV08][Ke, CVPR06]
[Datta, ECCV06]

Rule of 3rd, Color Harmony, etc.

 располагаемая

non-exhaustive

 interpretable
Limitations

Hand-crafted features
[San Pedro, WWW12][Berg, CVPR11]
[Luo, ECCV08][Ke, CVPR06]
[Datta, ECCV06]

Generic features
[Marchesotti, ICCV11]
[Murray, CVPR12]
[Oliva, CVPR11]

Rule of 3rd, Color Harmony, etc.

 kém non-exhaustive

😊 interpretable
Limitations

Hand-crafted features

[San Pedro, WWW12][Berg, CVPR11]
[Luo, ECCV08][Ke, CVPR06]
[Datta, ECCV06]

Generic features

[Marchesotti, ICCV11]
[Murray, CVPR12]
[Oliva, CVPR11]

Rule of 3rd, Color Harmony, etc.

😊 non-exhaustive

😊 interprettable

Fisher Vector, BoW with SIFT+color
Limitations

Hand-crafted features
[San Pedro, WWW12][Berg, CVPR11]
[Luo, ECCV08][Ke, CVPR06]
[Datta, ECCV06]

Rule of 3rd, Color Harmony, etc.

non-exhaustive

interpretable

Generic features
[Marchesotti, ICCV11]
[Murray, CVPR12]
[Oliva, CVPR11]

Fisher Vector, BoW with SIFT+color

Non interpretable
Limitations

Hand-crafted features
[San Pedro, WWW12][Berg, CVPR11]
[Luo, ECCV08][Ke, CVPR06]
[Datta, ECCV06]

Generic features
[Marchesotti, ICCV11]
[Murray, CVPR12]
[Oliva, CVPR11]

Rule of 3rd, Color Harmony, etc.

 véhicules non-exhaustive

 véhicules interchangeable

Fisher Vector, BoW with SIFT+color

 véhicules Non interpretable

 véhicules Improved perf.
Objective

Hand-crafted features

[San Pedro, WWW12][Berg, CVPR11]
[Luo, ECCV08][Ke, CVPR06]
[Datta, ECCV06]

- non-exhaustive
- interpretable

Generic features

[Marchesotti, ICCV11]
[Murray, CVPR12]
[Oliva, CVPR11]

Fisher Vector, BoW with SIFT+color

- Non interpretable
- Improved perf.
Objective

Hand-crafted features

[San Pedro, WWW12][Berg, CVPR11]
[Luo, ECCV08][Ke, CVPR06]
[Datta, ECCV06]

Generic features

[Marchesotti, ICCV11]
[Murray, CVPR12]
[Oliva, CVPR11]

Design mid-level features (attributes) which are:

.non-exhaustive
.interpretable

Fisher Vector, BoW with SIFT+color

Non interpretable

Improved perf.
Objective

Hand-crafted features

[San Pedro, WWW12][Berg, CVPR11]
[Luo, ECCV08][Ke, CVPR06]
[Datta, ECCV06]

Design mid-level features (attributes) which are:

- ☺ interpretable
- 😞 non-exhaustive

Generic features

[Marchesotti, ICCV11]
[Murray, CVPR12]
[Oliva, CVPR11]

- Fisher Vector, BoW with SIFT+color
- 😞 Non interpretable
- ☻ Improved perf.
Objective

Hand-crafted features

[San Pedro, WWW12] [Berg, CVPR11]
[Luo, ECCV08] [Ke, CVPR06]
[Datta, ECCV06]

Generic features

[Marchesotti, ICCV11]
[Murray, CVPR12]
[Oliva, CVPR11]

Design mid-level features (attributes) which are:

- ☺ interpretable
- ☺ Improved perf.
- ☹ non-exhaustive

Fisher Vector, BoW with SIFT+color

- ☹ Non interpretable
Contribution
Contribution

AVA: Aesthetic Visual Analysis dataset
Contribution

AVA: Aesthetic Visual Analysis dataset
AVA: Aesthetic Visual Analysis dataset
AVA: Aesthetic Visual Analysis dataset

AVA: Aesthetic Visual Analysis dataset

AVA: Aesthetic Visual Analysis dataset
Outline

Images,
Text,
Attractiveness
Discover Textual Labels

Images, Text, Attractiveness

Natural language text
Discover Textual Labels

Attractiveness scores

Images, Text, Attractiveness

Natural language text
Discover Textual Labels

Images, Text, Attractiveness

Natural language text

Attractiveness scores

unigrams, bigrams
Attractiveness scores

Images, Text, Attractiveness

Natural language text

1. Discover Textual Labels

2. Learn Visual Attributes

unigrams, bigrams
Outline

1. Discover Textual Labels
2. Learn Visual Attributes

Images, Text, Attractiveness

Attractiveness scores

Natural language text
unigrams, bigrams
attributes
Images, Text, Attractiveness

Natural language text

Discover Textual Labels

unigrams, bigrams

Learn Visual Attributes

attributes

OFFLINE

ONLINE

Attractiveness classification, regression, ranking, …

Attractiveness scores

1

2

3

Use!

Outline
Mine labels for visual attributes

Common approaches in the literature:
Mine labels for visual attributes

Common approaches in the literature:

• Use NLP templates [Everingham, BMVC09]
Mine labels for visual attributes

Common approaches in the literature:

- Use NLP templates [Everingham, BMVC09]
- Use Mutual Information [Berg, ECCV10]
Mine labels for visual attributes

Common approaches in the literature:

- Use NLP templates [Everingham ,BMVC09]
- Use Mutual Information [Berg, ECCV10]
- Use Amazon Mechanical Turk [Parrikh, ICCV11]
Mine labels for visual attributes

Common approaches in the literature:

- Use NLP templates [Everingham, BMVC09]
- Use Mutual Information [Berg, ECCV10]
- Use Amazon Mechanical Turk [Parrikh, ICCV11]
- Exploit lexical datasets [Lampert, CVPR09]
Mine labels for visual attributes

Common approaches in the literature:

- Use NLP templates [Everingham, BMVC09]
- Use Mutual Information [Berg, ECCV10]
- Use Amazon Mechanical Turk [Parrikh, ICCV11]
- Exploit lexical datasets [Lampert, CVPR09]

Our requirements:
Mine labels for visual attributes

Common approaches in the literature:

• Use NLP templates [Everingham, BMVC09]
• Use Mutual Information [Berg, ECCV10]
• Use Amazon Mechanical Turk [Parrikh, ICCV11]
• Exploit lexical datasets [Lampert, CVPR09]

Our requirements:
1. Minimum amount of supervision
Mine labels for visual attributes

Common approaches in the literature:

• Use NLP templates [Everingham, BMVC09]
• Use Mutual Information [Berg, ECCV10]
• Use Amazon Mechanical Turk [Parrikh, ICCV11]
• Exploit lexical datasets [Lampert, CVPR09]

Our requirements:

1. Minimum amount of supervision
2. Labels “explaining” aesthetic preference
Mine labels for visual attributes

Common approaches in the literature:

• Use NLP templates [Everingham, BMVC09]
• Use Mutual Information [Berg, ECCV10]
• Use Amazon Mechanical Turk [Parrikh, ICCV11]
• Exploit lexical datasets [Lampert, CVPR09]

Our requirements:

1. Minimum amount of supervision
2. Labels “explaining” aesthetic preference
3. Scalable method capable of producing many labels
Unsupervised mining

**Idea:** consider attributes as $|z| = M$ latent topics shared among textual comments represented using unigrams.
Unsupervised mining

**Idea:** consider attributes as $|z| = M$ latent topics shared among textual comments represented using unigrams.
Unsupervised mining

**Idea:** consider attributes as $|z| = M$ latent topics shared among textual comments represented using unigrams.

Text examples:
- Congrats on the red Margaret. I thought that this beautiful photo would be yours. I can see I am really missing something by not having ever been to Clifton Springs.
- There really is so much to like about this image. Congrats on your ribbon.
- Doing well with the big sticks Margaret, nicely processed, congrats. Sharp and clear nice colors.
- Beautifully composed and full of wonderful textures. I've found myself coming back to this one over, and over,
Unsupervised mining

**Idea:** consider attributes as $|Z| = M$ latent topics shared among textual comments represented using unigrams.
Unsupervised mining

**Idea:** consider attributes as $|z| = M$ latent topics shared among textual comments represented using unigrams.

$$p(w|d) = \sum p(w|z)p(z|d)$$ [Hoffman, 99]
Unsupervised mining

Idea: consider attributes as $|z| = M$ latent topics shared among textual comments represented using unigrams.

$$p(w|d) = \sum p(w|z)p(z|d)$$ [Hoffman, 99]

$Z_8$ : portrait, eyes, face, expression, beautiful, skin, hair, character, portraits, eye, smile, nose, lovely, self, girl, look, wonderful, great, lighting, crop

$Z_{11}$ : DD, beautiful, wow, amazing, congratulations, top, congrats, love, stunning, great, wonderful, excellent, awesome, perfect, fantastic, gorgeous, absolutely, capture

$Z_{28}$ : idea, creative, clever, concept, cool, executed, execution, original, well, great, pencil, job, creativity, thought, top, work, interesting, good

$Z_{35}$ : motion, panning, blur, speed, movement, shutter, moving, blurred, abstract, blurry, pan, stopped, sense, camera, fast, train, slow, background, exposure

$Z_{57}$ : colors, red, colours, green, abstract, color, yellow, orange, beautiful, colour, border, vibrant, complementary, composition, leaf, lovely, love, background, bright, purple
Unsupervised mining

**Idea:** consider attributes as $|z| = M$ latent topics shared among textual comments represented using unigrams.

$$ p(w|d) = \sum p(w|z)p(z|d) \quad [\text{Hoffman, 99}] $$

- $Z_8$: portrait, eyes, face, expression, beautiful, skin, hair, character, portraits, eye, smile, nose, lovely, self, girl, look, wonderful, great, lighting, crop
- $Z_{11}$: DD, beautiful, wow, amazing, congratulations, top, congrats, love, stunning, great, wonderful, excellent, awesome, perfect, fantastic, gorgeous, absolutely, capture
- $Z_{28}$: idea, creative, clever, concept, cool, executed, execution, original, well, great, pencil, job, creativity, top, work, interesting, good
- $Z_{35}$: motion, panning, blur, speed, movement, shutter, moving, blurred, abstract, blurry, pan, stopped, sense, camera, fast, train, slow, background, exposure
- $Z_{57}$: colors, red, colours, green, abstract, color, yellow, orange, beautiful, colour, border, vibrant, complementary, composition, leaf, lovely, love, background, bright, purple

😊 topics are related to visual attractiveness

😢 too vague (i.e. not granular enough)
Supervised mining

**Idea**: leverage scores as a supervising entity to guide the discovery.
Supervised mining

**Idea**: leverage scores as a supervising entity to guide the discovery.
Supervised mining

Idea: leverage scores as a supervising entity to guide the discovery.
Supervised mining

Idea: leverage scores as a supervising entity to guide the discovery.
Supervised mining

Idea: leverage scores as a supervising entity to guide the discovery.

$$\hat{\beta} = \arg\min_{\beta} \| y - X\beta \|^2 + \lambda_1 \| \beta \|_1 + \lambda_2 \| \beta \|^2 \quad \text{[Zou,05]}$$
Idea: leverage scores as a supervising entity to guide the discovery.

\[ \hat{\beta} = \arg \min_{\beta} \| y - X\beta \|^2 + \lambda_1 \| \beta \|_1 + \lambda_2 \| \beta \|^2 \]  

[Zou, 05]
Supervised mining

Idea: leverage scores as a supervising entity to guide the discovery.

\[ \hat{\beta} = \arg \min_{\beta} \| y - X\beta \|^2 + \lambda_1 \| \beta \|_1 + \lambda_2 \| \beta \|^2 \] [Zou,05]

Elastic Net (EN) can produce sparse predictors

It overcomes the problems of Lasso
Supervised mining
Supervised mining

| **UNIGRAMS+** | great (0.4351), like (0.3301), excellent (0.2943), love (0.2911), beautiful (0.2704), done (0.2609), very (0.2515), well (0.2465), shot (0.2228), congratulations (0.2223), perfect (0.2142), congrats (0.2114), wonderful (0.2099), nice (0.1984), wow (0.1942), one (0.1664), top (0.1651), good (0.1639), awesome (0.1636), |
| **UNIGRAMS-** | sorry (-0.2767), focus (-0.2345), blurry (-0.2066), small (-0.1950), not (-0.1947), don (-0.1881), doesn (-0.1651), flash (-0.1326), snapshot (-0.1292), too (-0.1263), grainy (-0.1176), meet (-0.1122), out (-0.1054), try (-0.1041), low (-0.1013), poor (-0.0978), distracting (-0.0724), |
Supervised mining

| UNIGRAMS+ | great (0.4351), like (0.3301), excellent (0.2943), love (0.2911), beautiful (0.2704), done (0.2609), very (0.2515), well (0.2465), shot (0.2228), congratulations (0.2223), perfect (0.2142), congrats (0.2114), wonderful (0.2099), nice (0.1984), wow (0.1942), one (0.1664), top (0.1651), good (0.1639), awesome (0.1636), |
| UNIGRAMS- | sorry (-0.2767), focus (-0.2345), blurry (-0.2066), small (-0.1950), not (-0.1947), don (-0.1881), doesn (-0.1651), flash (-0.1326), snapshot (-0.1292), too (-0.1263), grainy (-0.1176), meet (-0.1122), out (-0.1054), try (-0.1041), low (-0.1013), poor (-0.0978), distracting (-0.0724), |

😊 Attributes are very specific
### Supervised mining

<table>
<thead>
<tr>
<th>Attributes are very specific</th>
<th>They don’t have a polarity!</th>
</tr>
</thead>
<tbody>
<tr>
<td>smile</td>
<td>frown</td>
</tr>
</tbody>
</table>

#### UNIGRAMS+
- great (0.4351)
- like (0.3301)
- excellent (0.2943)
- love (0.2911)
- beautiful (0.2704)
- done (0.2609)
- very (0.2515)
- well (0.2465)
- shot (0.2228)
- congratulations (0.2223)
- perfect (0.2142)
- congrats (0.2114)
- wonderful (0.2099)
- nice (0.1984)
- wow (0.1942)
- one (0.1664)
- top (0.1651)
- good (0.1639)
- awesome (0.1636)

#### UNIGRAMS-
- sorry (-0.2767)
- focus (-0.2345)
- blurry (-0.2066)
- small (-0.1950)
- not (-0.1947)
- don (-0.1881)
- doesn’t (-0.1651)
- flash (-0.1326)
- snapshot (-0.1292)
- too (-0.1263)
- grainy (-0.1176)
- meet (-0.1122)
- out (-0.1054)
- try (-0.1041)
- low (-0.1013)
- poor (-0.0978)
- distracting (-0.0724)
## Supervised mining

### Attributes are very specific

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>great</td>
<td>0.4351</td>
</tr>
<tr>
<td>like</td>
<td>0.3301</td>
</tr>
<tr>
<td>excellent</td>
<td>0.2943</td>
</tr>
<tr>
<td>love</td>
<td>0.2911</td>
</tr>
<tr>
<td>beautiful</td>
<td>0.2704</td>
</tr>
<tr>
<td>done</td>
<td>0.2609</td>
</tr>
<tr>
<td>very</td>
<td>0.2515</td>
</tr>
<tr>
<td>well</td>
<td>0.2465</td>
</tr>
<tr>
<td>shot</td>
<td>0.2228</td>
</tr>
<tr>
<td>congratulations</td>
<td>0.2223</td>
</tr>
<tr>
<td>perfect</td>
<td>0.2142</td>
</tr>
<tr>
<td>congrats</td>
<td>0.2114</td>
</tr>
<tr>
<td>wonderful</td>
<td>0.2099</td>
</tr>
<tr>
<td>nice</td>
<td>0.1984</td>
</tr>
<tr>
<td>wow</td>
<td>0.1942</td>
</tr>
<tr>
<td>one</td>
<td>0.1664</td>
</tr>
<tr>
<td>top</td>
<td>0.1651</td>
</tr>
<tr>
<td>good</td>
<td>0.1639</td>
</tr>
<tr>
<td>awesome</td>
<td>0.1636</td>
</tr>
</tbody>
</table>

### They don’t have a polarity!

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sorry</td>
<td>-0.2767</td>
</tr>
<tr>
<td>focus</td>
<td>-0.2345</td>
</tr>
<tr>
<td>blurry</td>
<td>-0.2066</td>
</tr>
<tr>
<td>small</td>
<td>-0.1950</td>
</tr>
<tr>
<td>not</td>
<td>-0.1947</td>
</tr>
<tr>
<td>don</td>
<td>-0.1881</td>
</tr>
<tr>
<td>doesn't</td>
<td>-0.1651</td>
</tr>
<tr>
<td>flash</td>
<td>-0.1326</td>
</tr>
<tr>
<td>snapshot</td>
<td>-0.1292</td>
</tr>
<tr>
<td>too</td>
<td>-0.1263</td>
</tr>
<tr>
<td>grainy</td>
<td>-0.1176</td>
</tr>
<tr>
<td>meet</td>
<td>-0.1122</td>
</tr>
<tr>
<td>out</td>
<td>-0.1054</td>
</tr>
<tr>
<td>try</td>
<td>-0.1041</td>
</tr>
<tr>
<td>low</td>
<td>-0.1013</td>
</tr>
<tr>
<td>poor</td>
<td>-0.0978</td>
</tr>
<tr>
<td>distracting</td>
<td>-0.0724</td>
</tr>
</tbody>
</table>

---

**Idea**: Use bigrams to capture non-compositional meanings [Pang, 12]
<table>
<thead>
<tr>
<th>Attributes are very specific</th>
<th>They don’t have a polarity!</th>
</tr>
</thead>
</table>

**Unigrams+**
great (0.4351), like (0.3301), excellent (0.2943), love (0.2911), beautiful (0.2704), done (0.2609), very (0.2515), well (0.2465), shot (0.2228), congratulations (0.2223), perfect (0.2142), congrats (0.2114), wonderful (0.2099), nice (0.1984), wow (0.1942), one (0.1664), top (0.1651), good (0.1639), awesome (0.1636),

**Unigrams-**
sorry (-0.2767), focus (-0.2345), blurry (-0.2066), small (-0.1950), not (-0.1947), don (-0.1881), doesn (-0.1651), flash (-0.1326), snapshot (-0.1292), too (-0.1263), grainy (-0.1176), meet (-0.1122), out (-0.1054), try (-0.1041), low (-0.1013), poor (-0.0978), distracting (-0.0724),

**Bigrams+**
well done (0.6198), very nice (0.6073), great shot (0.5706), very good (0.3479), great job (0.3287), your top (0.3262), my favorites (0.3207), top quality (0.3198), great capture (0.3051), lovely composition (0.3014), my top (0.2942), nice shot (0.2360), th placing (0.2330), great lighting (0.2302), great color (0.2245), excellent shot (0.2221), good work (0.2218), well executed (0.2069), great composition (0.2047), my only (0.2032), awesome shot (0.2030),

**Bigrams-**
too small (-0.3447), too blurry (-0.3237), not very (-0.3007), does not (-0.2917), not meet (-0.2697), wrong challenge (-0.2561), better focus (-0.2280), not really (-0.2279), sorry but (-0.2106), really see (-0.2103), poor focus (-0.2068), too out (-0.2055), keep trying (-0.2026), see any (-0.2021), , not sure (-0.2017), too dark (-0.2007), next time (-0.1865), missing something (-0.1862), just don (-0.1857), not seeing (-0.1785), your camera (-0.1775),

**Idea:** Use bigrams to capture non-compositional meanings [Pang,12]
Learning Visual attributes

Images, Text, Attractiveness

Attractiveness scores

Discover Textual Labels

Natural language text

Xerox Research Centre Europe
20 Years of Innovation
Learning Visual attributes

Images, Text, Attractiveness

Attractiveness scores

Natural language text

unigrams, bigrams

Discover Textual Labels
Learning Visual attributes

1. Discover Textual Labels
2. Learn Visual Attributes

Images, Text, Attractiveness

Attractiveness scores

Natural language text

unigrams, bigrams
Learning Visual attributes

1. Discover Textual Labels
2. Learn Visual Attributes

Images, Text, Attractiveness

Attractiveness scores

Natural language text

unigrams, bigrams

attributes
Learning Visual attributes

Hypothesis on potential attributes: which ones should be taken into account?
Learning Visual attributes

Hypothesis on potential attributes: which ones should be taken into account?

Can these labels be learned by a visual categorizer with sufficient accuracy?
Learning Visual attributes

1. Train as many visual classifier as you can, starting from the 3,000 most discriminant labels
Learning Visual attributes

1. Train as many visual classifier as you can, starting from the 3,000 most discriminant labels

=> Use the custom learning pipeline: SIFT and color descriptors + spatial pyramid + Fisher Vector + PQ-compression + SGD with log Loss
Learning Visual attributes

1. Train as many visual classifier as you can, starting from the 3,000 most discriminant labels

=> Use the custom learning pipeline: SIFT and color descriptors + spatial pyramid + Fisher Vector + PQ-compression + SGD with log Loss

2. Re-rank the classifiers based on their performances
Learning Visual attributes

1. Train as many visual classifier as you can, starting from the 3,000 most discriminant labels

   => Use the custom learning pipeline: SIFT and color descriptors + spatial pyramid + Fisher Vector + PQ-compression + SGD with log Loss

2. Re-rank the classifiers based on their performances

   => Use AUC or whatever else measure of performance (e.g. Top-K)
Learning Visual attributes

1. Train as many visual classifier as you can, starting from the 3,000 most discriminant labels

   => Use the custom learning pipeline: SIFT and color descriptors + spatial pyramid + Fisher Vector + PQ-compression + SGD with log Loss

2. Re-rank the classifiers based on their performances

   => Use AUC or whatever else measure of performance (e.g. Top-K)

3. Handpick the most interpretable ones
Learning Visual attributes

1. Train as many visual classifier as you can, starting from the 3,000 most discriminant labels

=> Use the custom learning pipeline: SIFT and color descriptors + spatial pyramid + Fisher Vector + PQ-compression + SGD with log Loss

2. Re-rank the classifiers based on their performances

=> Use AUC or whatever else measure of performance (e.g. Top-K)

3. Handpick the most interpretable ones

=> Note that this is the only manual intervention required in the whole pipeline!
What is beautiful in photography?

- great_colours
- beautiful_scene
- so_cute
- nice_perspective
- great_portrait
- black_background
- absolutely_beautiful
- amazing_capture
- very_create
- great_tones
- gorgeous_image
- those_eyes
- good_lighting
- great_focus
- stunning_image
- works_well
- beautiful_lighting
- tonal_range
and what is ugly..

- soft_focus
- small_too
- distracting_background
- snap_shot
- focus_appears
- very_dark
- bad_focus
- very_grainy
- camera_flash
- too_soft
- too_abstract
- camera_shake
- light_source

- too_harsh
- little_dark
- too_fuzzy
- too_busy
- very_poor
- over_exposed
- too_dark
- washed_out
- very_out
- motion_blur
- too_little
- more_dof
- not_sharp
Qualitative results
Qualitative results

Great Colours
Qualitative results

Great Colours

Nice Perspective
Qualitative results

Great Colours

Nice Perspective

Very sharp
Qualitative results

Great Colours

Nice Perspective

Very sharp

Very cluttered
Qualitative results

Great Colours

Nice Perspective

Very sharp

Very cluttered

Color Cast
Quantitative results
Quantitative results

Ugly attributes can be learned with lower precision than the beautiful ones.
Prediction performances

![Graph showing prediction performances]

- Attributes
- Fisher vector
Note that the Fisher Vector has 131,072 dimensions vs 600 dimensions of the attributes features.
Application – Photo Annotation
Application – Photo Annotation
Application – Photo Annotation

Like

Like

Dislike

Dislike
Application – Photo Annotation
Application – Photo Annotation

great_macro, very.pretty, great_focus, nice_detail, so.cute

great_capture, great_angle, nice_perspective, lovely_photo, nice_detail

more_dof, not_sure, too_busy, motion_blur, blown_out

soft_focus, not_sure, more_light, sharper_focus, more_dof
Conclusions & Future work

For learning high-level quality:

- use large scale datasets (e.g. AVA)
- use generic high-dimensional features
- Learn content-dependent models (if you have enough data)

and also consider:

- textual features
nice_dof
thank you
What generates most consensus?

![Box plot showing the distribution of variance for different mean score ranges for ugly and beautiful attributes.](image)

**Ugly Attributes**

**Beautiful Attributes**