Estimating the Visual Complexity of Images from Textual Descriptions

Speech Lab Talks | September 30, 2022

Eleanor Lin

Outline

1. Goal & motivations
2. Developing a visual complexity metric
3. Predicting visual complexity from text

Goals

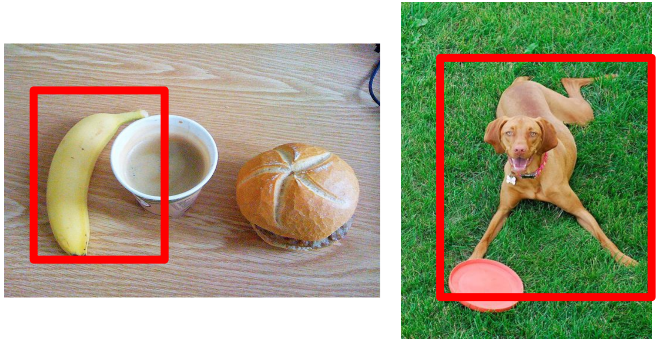
- Develop automated metric for visual complexity
- Identify visually complex images from text descriptions

Intuition: Biases exist in how humans describe images of different complexities

“a **very cluttered** chinese street showing **many** business signs”

Motivation

- CV models struggle on complex images
- Examples
  - visual search
  - caption generation
  - object detection/segmentation



What is visual complexity?

- SAVOIAS dataset: **cluttered background, number/diversity of objects, people, textures, patterns, shapes** (Saraee et al., 2018)
- Other definitions/dimensions:
  - Difficulty to describe image
  - Amount of information contained (image compression ratio)
  - Colorfulness

(. . . and more)

SAVOIAS Dataset (Saraee et al., 2018, p. 5)

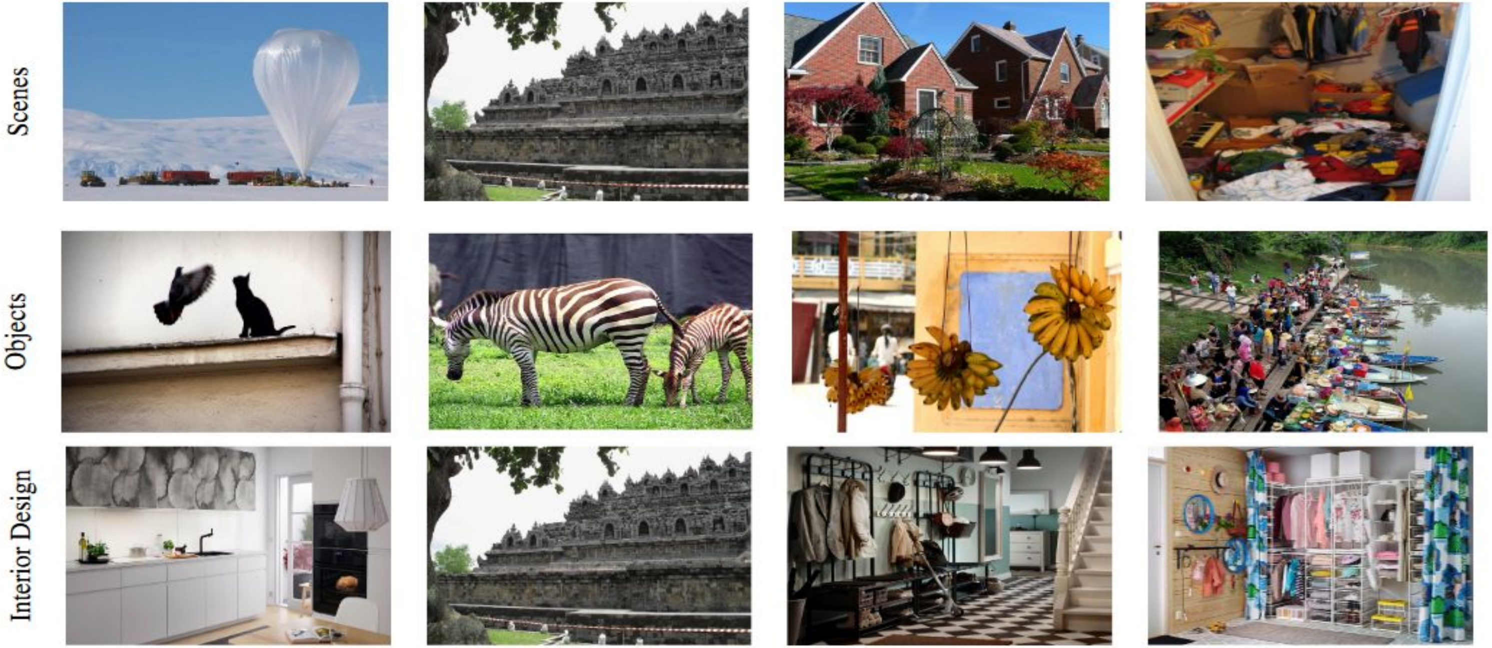


Table 2: Sample images of the SAVOIAS dataset with increased visual complexity from left to right in each row.

Choosing a Visual Complexity Metric

Problems:

- SAVOIAS dataset lacks image captions
- SAVOIAS is small (200 images per most categories)

Approach:

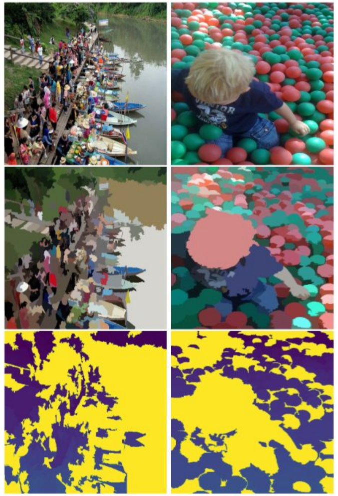
1. Find automated visual complexity metric correlated with SAVOIAS human visual complexity scores
2. Use metric to score complexity of images from **COCO dataset**
3. Train model to identify complex images from captions

COCO Dataset (Lin et al., 2014)

123,287 images (train/val sets), 80 object categories, 11 supercategories

“a store with bunches of bananas hanging from a wire.”
“a man putting something on is desk while food is sitting in the front in boxes.”
“a kitchen with a bunch of food in boxes and bananas hanging from hooks”
“a man working in an outdoor market with various vegetables and fruits.”
“the storefront of a small open produce market.”

Visual complexity metric: Distinct # of regions

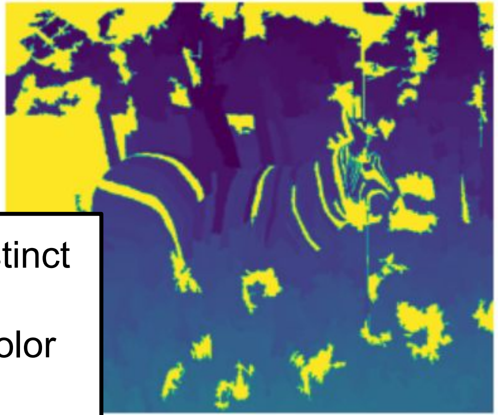


Type	Metric	Scenes	Objects	Interior Design	All
Low-level	Compression ratio (Saraee et al., 2020)	0.30	0.16	0.72	–
Low-level	Feature congestion (Saraee et al., 2020)	0.42	0.30	0.63	–
Low-level	Number of regions (Saraee et al., 2020)	0.57	0.29	0.69	–
High-level	VGG16 Scene Recognition, UAE (Saraee et al., 2020)	0.76	0.67	0.82	–
High-level	VGG16 Object Classification, UAE (Saraee et al., 2020)	0.77	0.64	0.83	–
High-level	**VGG16 Object Classification, SAE from Depth Features** (Saraee et al., 2020)	**0.85**	**0.80**	**0.86**	–
Low-level	Number of regions [Ours] (Comaniciu and Meer, 2002; Jean, 2020)	0.63	0.36	0.71	0.50
Low-level	**Number of distinct regions [Ours]**	**0.73**	**0.55**	**0.81**	**0.62**

Mean-shift segmentation

Filter for distinct regions (compare color and size)

```
Unfiltered regions: 382
Number of distinct regions: 83
```



Training the Models: Classify Complex v. Noncomplex

- Images with top/bottom 10% most/fewest distinct regions
  - label “complex”/”noncomplex”

Task	Split	Image source	# images	# captions
Binary classification	train	MS COCO 2017 train set	22656	113342
Binary classification	val	MS COCO 2017 train set	1000	5001
Binary classification	test	MS COCO 2017 val set	1000	5004
Regression	train	MS COCO 2017 train set	113287	566747
Regression	val	MS COCO 2017 train set	5000	25006
Regression	test	MS COCO 2017 val set	5000	25014

Probability that image is complex: 0.923

“people watching an elephant near some water and a fence”

$\text{BERT}_{\text{BASE}}$

Label: complex

Training the Models: Classify Complex v. Noncomplex (+ regression)

$$P(complex) = p = \sigma(x) = \frac{1}{1+e^{-x}}$$

$$l = y * log(p) + (1-y) * log(1-p)$$

Classification

- **Inputs:** tokenized COCO captions, size = 128
- **Labels:** "complex" or "noncomplex"
- **Output:** probability that input caption describes a complex image
- **Loss:** Binary cross-entropy loss
- λ = 2 ∗ 10−5
- Fine-tune for 4 epochs > choose model with highest accuracy on validation set

Regression

- **Inputs, learning rate, # of epochs**: same as above
- **Labels:** complexity score in (0, 1)
  - **Normalization:** c = tanh(r/80)
- **Output:** Normalized complexity score
- **Loss:** MSE loss

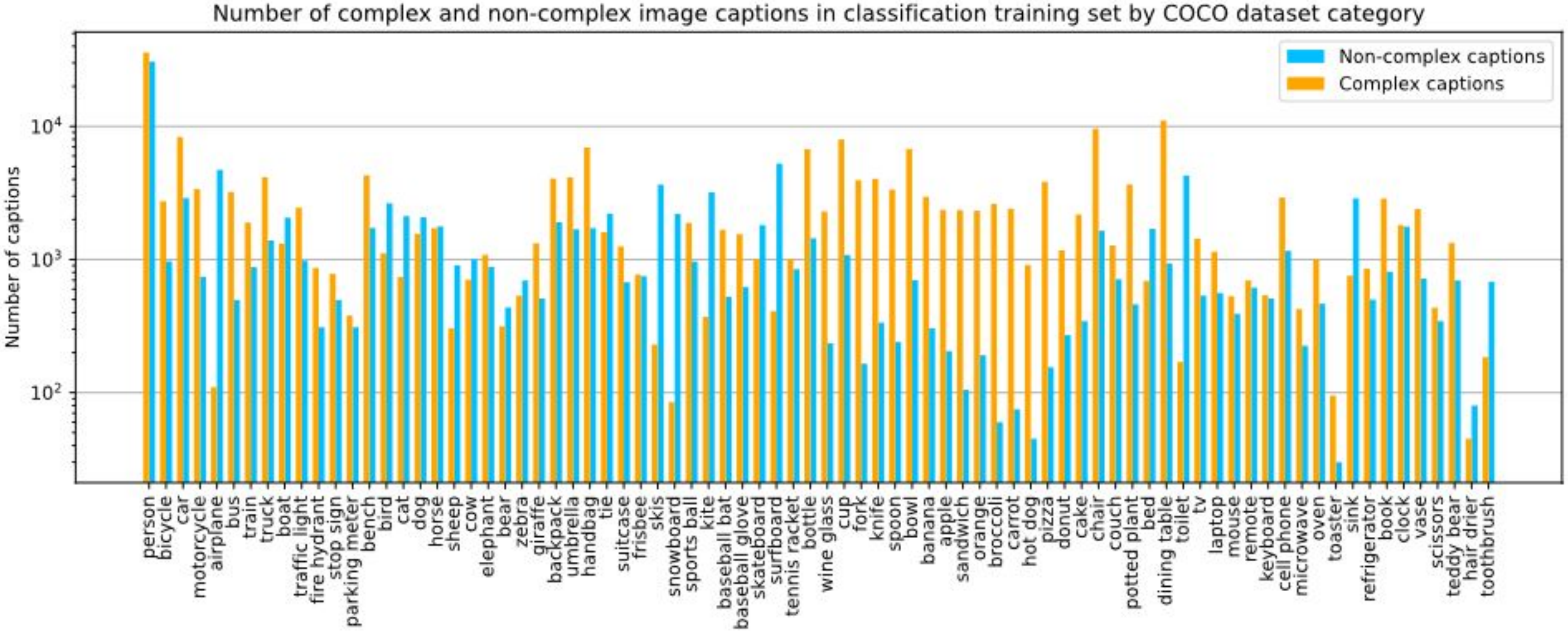
$$l = (x-y)^2$$

Results: What's going on?

Classifier:
83.9% accuracy
BCE Loss = 0.411
(val set)

Regression model:
MSE = 0.03
$r = 0.659$
($p < 0.001$)
(val set)

Problem: Class Imbalance between Complex/Noncomplex



Solutions to Class Imbalance

1. Cross-domain evaluation
2. Transformed captions

What if we fine-tune only on images containing ___ ?

Goal: reduce ability of model to exploit biases in COCO dataset wrt complexity of specific object type images

COCO (super)category	classification set # complex	classification set # noncomplex	regression set # total
person	35,895	30,674	307,365
vehicle	16,808	11,748	131,297
outdoor	8,075	3,673	61,860
animal	8,860	12,163	114,834
accessory	13,200	6,817	84,781
sports	6,466	17,956	111,282
kitchen	15,976	3,137	99,430
food	16,792	1,521	77,820
furniture	18,321	8,785	141,086
electronic	5,282	2,897	62,151
appliance	2,111	3,527	37,632
indoor	7,773	4,821	75,917

all n labels in our training set

$$t_1, t_2, \ldots, t_n \tag{7}$$

the Weighted Random Sampler samples from the set according to probabilities (or weights)

$$p_1, p_2, \ldots, p_n \tag{8}$$

We compute the weights as follows. If $t_i = 1$ (complex) for $1 \leq i \leq n$, then

$$p_i = \frac{1}{n_{\text{complex}}} \tag{9}$$

i.e., the weight for a complex sample is the reciprocal of the number of complex training samples. Similarly, if $t_i = 0$ (noncomplex), then

$$p_i = \frac{1}{n_{\text{noncomplex}}} \tag{10}$$

Results

COCO (super)category of dataset	Best classifier trained on	Validation set accuracy [Baseline accuracy]	Average validation set loss (Cross-entropy)	Average precision
none (full set)	full set	0.839 [0.500]	0.411	0.913
person	full set	0.830 [0.539]	0.432	0.908
vehicle	vehicle	0.821 [0.589]	0.444	0.919
outdoor	**person**	**0.758 [0.687]**	**0.613**	**0.864**
animal	full set	0.802 [0.579]	0.501	0.818
accessory	accessory	0.818 [0.659]	0.531	0.902
sports	full set	0.851 [0.735]	0.370	0.762
kitchen	**electronic**	**0.909 [0.646]**	**0.342**	**0.965**
food	full set	0.923 [0.917]	0.273	0.974
furniture	**indoor**	**0.892 [0.617]**	**0.308**	**0.939**
electronic	electronic	0.811 [0.646]	0.547	0.900
appliance	**indoor**	**0.836 [0.626]**	**0.421**	**0.865**
indoor	indoor	0.827 [0.617]	0.427	0.924
COCO (super)category of dataset	Best regression model trained on	Pearson's r	Average validation set loss (Mean squared error)	Average precision
none (full set)	full set	0.659 ($p < 0.001$)	0.030	0.951
person	full set	0.594 ($p < 0.001$)	0.031	0.946
vehicle	**full set**	**0.016** ($p = 0.238$)	**0.031**	**0.954**
outdoor	full set	0.483 ($p < 0.001$)	0.032	0.939
animal	full set	0.517 ($p < 0.001$)	0.032	0.861
accessory	full set	0.506 ($p < 0.001$)	0.035	0.968
sports	full set	0.603 ($p < 0.001$)	0.030	0.866
kitchen	kitchen	0.520 ($p < 0.001$)	0.027	0.977
food	food	0.500 ($p < 0.001$)	0.029	0.991
furniture	furniture	0.595 ($p < 0.001$)	0.028	0.988
electronic	electronic	0.479 ($p < 0.001$)	0.025	0.978
appliance	full set	0.571 ($p < 0.001$)	0.023	0.896
indoor	full set	0.497 ($p < 0.001$)	0.029	0.961

Transformed captions

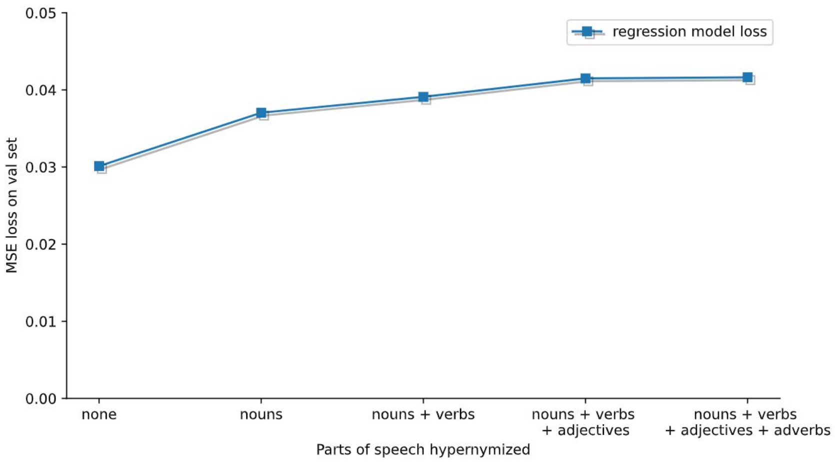
Thus the caption

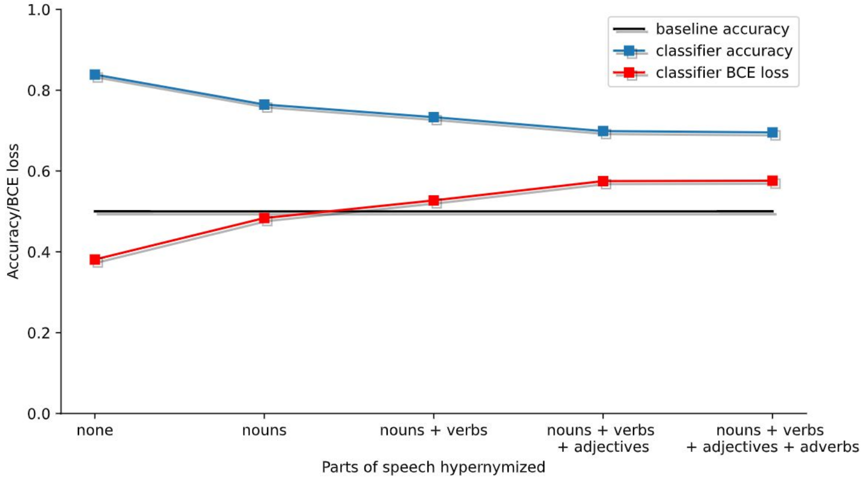
(1) Shelves of stuffed animals of various color and shapes.

becomes

(2) objects of plain objects of plain object and objects.

Word tagged with	Substitute with
NN, NNP	object
NNS, NNPS	objects
VB, VBP	act
VBD, VBN	acted
VBG	acting
VBZ	acts
JJ	plain
JJR, RBR	plainer
JJS, RBS	plainest
RB	plainly





Conclusions

- Visual complexity ~ Description of image
- BERT learns complexity biases in COCO
- Other possible directions:
    - Using different groundtruth visual complexity metric
    - Training on other captioned image datasets
    - Are images predicted complex by text-based model actually more difficult for CV models (caption generators, object detectors, etc.)?
    - Are images with high complexity score (distinct # of regions) actually more difficult for CV models?

Acknowledgements

Thank you to Vicente Ordonez and Ziyan Yang for their mentorship throughout this project.

This work was supported by the Computing Research Association's Distributed Research Experiences for Undergraduates program, which receives funding from the National Science Foundation.

This work was also supported by the National Science Foundation under Grant No. 2221943. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.