Text-Based Prediction of Visual Complexity

Eleanor Lin
Columbia University
e.lin2@columbia.edu

Abstract

Visual complexity is of interest across cognitive science, computer science, advertising, web design, and other areas, due to the increased difficulty both computers and humans encounter in processing complex visuals. Intuitively, one might expect biases in how complex visuals are described: e.g., using adjectives like "busy" or "cluttered." We explore the relationship between linguistic and visual complexity by asking if it is possible to predict an image's visual complexity based on its textual description alone. Our text-based approach contrasts with the majority of past work on analyzing visual complexity, which focuses on the images themselves, rather than their descriptions. We introduce a new automated complexity metric, number of distinct regions per image, which serves as an effective predictor of human judgments of visual complexity. Using this metric to identify complex and noncomplex images from the Microsoft COCO Dataset, we fine-tune BERT$_{\text{BASE}}$ to predict visual complexity from image captions. We find that the model is able to predict visual complexity with a high degree of accuracy, and appears to rely on vocabulary and sentence structure for its predictions. Our work suggests a relationship between visual complexity of images and linguistic complexity of image descriptions, which may be leveraged to better identify complex images and improve algorithms for tasks that involve both vision and language, such as automatic caption generation.

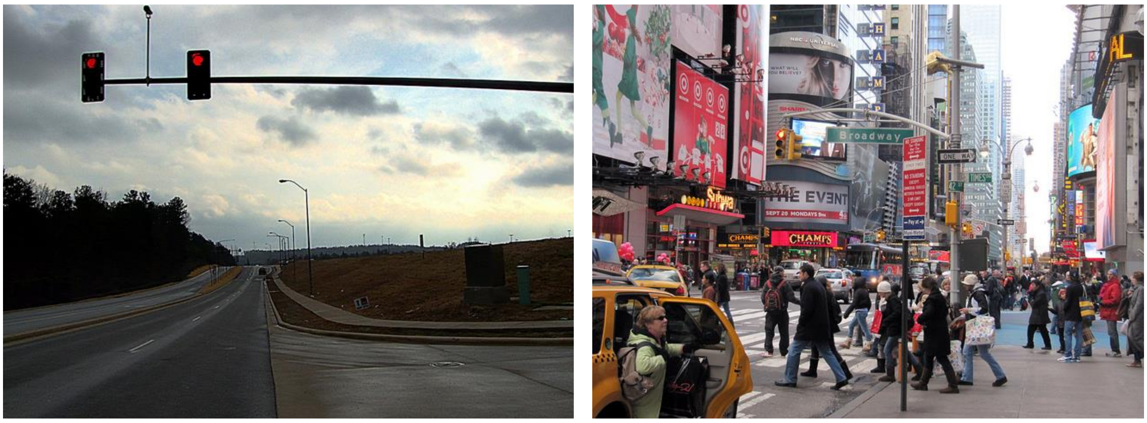
1 Introduction

Visual complexity has been variously defined as the amount of information in, number of elements in, and difficulty of describing an image (Saraee et al., 2018, 2020). The latter definition suggests the link between visual complexity and linguistic complexity which we explore in this work, through the task of text-based prediction of visual complexity.

The ability to identify visually complex images is of interest due to the particular challenge that such images present in computer vision tasks such as object detection and segmentation. Furthermore, tasks that involve both visual and linguistic components, such as visual question answering and caption generation, would benefit from deeper investigation of the relationship between visual and linguistic complexity (Saraee et al., 2018, 2020).

Our text-based visual complexity prediction task is motivated by the observation that image descriptions often contain visual complexity cues, as shown in Figure 1. The image on the left is less complex than the image on the right, in terms of factors such as number and diversity of objects, shapes, and colors. The descriptions of the less complex image include adjectives such as "empty," hinting at the relative scarcity of objects, whereas the descriptions of the more complex image include adjectives such as "many" and "busy," indicating numerosity of objects. We hypothesize that image descriptions contain enough visual complexity cues for a model such as BERT to predict an image's visual complexity from its description alone.

Past attempts to quantify visual complexity have included low-level metrics, such as the number of regions resulting from segmenting an image (Saraee et al., 2018, 2020). We expand upon this approach in order to determine if it is possible to predict an image's visual complexity from its textual description. First, we define a new visual complexity metric designed to correspond to intuitive human judgements of visual complexity, the number of distinct regions in the mean-shift segmentation of an image (§4). We show that our metric outperforms other low-level metrics previously tested on the Objects, Scenes, and Interior Design categories of the SAVOIAS visual complexity dataset (see §§2 and 3). Next, we use our metric to groundtruth the complexity of images from the Microsoft COCO Dataset (Lin et al., 2014). Finally, we fine-tune BERT$_{\text{BASE}}$ sequence classification and regression models to predict image complexity from COCO



"This highway is **empty** this early on the morning,“
"A traffic light suspended over a **rural** road.“
"A red traffic light at an **empty** intersection“
"An intersection with a stoplight on a roadway that has **no vehicles** traveling on it.“
"View down a two lane road at a red stop signal."

"**Dozens of individuals** all crossing the street on Broadway.“
"a **bunch of people** cross a city street“
"There are **many people** walking on the **busy** downtown street.“
"**Many people** walking and crossing a downtown city street.“
"**Many people** crossing the street in a **busy** city."

Figure 1: Examples of visually complex (left) and a noncomplex (right) images from the Microsoft COCO Dataset (Lin et al., 2014), along with their captions. Visual complexity cues in the captions are **boldfaced**: e.g., plural nouns and certain adjectives ("busy," "empty"). Some complexity cues, such as the fact that a "rural road" is likely to be less busy than a "city street," can only be recognized using commonsense knowledge, which machine learning models may not have access to. The last caption describing the noncomplex image is ambiguous as to whether the scene described is visually complex, underscoring the difficulty of text-based prediction of visual complexity,

captions (§§5-7.3).

We find that text-based BERT classification and regression models are able to predict the complexity of COCO images from their captions alone with a high degree of accuracy. However, BERT seems to learn complexity biases in the COCO Dataset, i.e., that the presence of certain object types in an image is associated with its complexity (§6). Through cross-domain evaluation and caption hypernymization experiments designed to remove such biases, we find evidence that BERT also relies on sentence structure in order to make complexity predictions (§§7 and 7.3). Our work suggests that visual and linguistic complexity are related across images and image descriptions in the COCO Dataset.

2 Related work

In section 2.1, we survey previous work on modelling human perception of visual complexity. In section 2.2, we discuss Jas and Parikh (2015)'s study of image specificity, which parallels our study in relating image content to image descriptions.

2.1 Modelling human perception of visual complexity

Prior work has shown that there exist an array of visual cues correlated with human judgements of complexity. Oliva et al. (2004) found that people intuitively relied on several key factors to judge the complexity of indoor scenes, including the numerosity, colorfulness, and arrangement of objects in a scene.

Rosenholtz et al. (2007, p. 3) define "clutter" as "the state in which excess items, or their representation or organization, lead to a degradation of performance at some task." Notably, they describe clutter as related to complexity, and discuss the difficulty of defining clutter due to the difficulty of defining what an "item" is across scales. Rosenholtz et al. (2007) introduce a visual clutter measure based on the variability of color, orientation, and luminance in a visualization. Their metric successfully predicts the difficulty for humans to search visualizations for target symbols, corresponding to the idea that visually cluttered images are more difficult to search.

Saraee et al. (2020) introduce the Unsupervised and Supervised Activation Energy (UAE and SAE) methods for measuring visual complexity. They demonstrate that both UAE and SAE are moderately to strongly correlated with human judgements of complexity across the diverse types of images in their SAVOIAS visual complexity dataset (see §3). UAE and SAE are computed from feature maps produced by the intermediate convolutional layers of deep neural networks. These neural nets are pre-trained on object or scene classification tasks.

The UAE method averages the values in the feature map from a layer over all receptive fields and channels to produce an image's complexity score. SAE refines UAE by using a learned, weighted average of activations. By taking activations from the intermediate layers, UAE and SAE balance the contributions of both low-level features (e.g., edges), extracted in early convolutional layers, and high-level features (e.g., objects), formed in deep layers, toward visual complexity.

Building upon the findings of Oliva et al. (2004), Rosenholtz et al. (2007), and Saraee et al. (2018, 2020), our visual complexity metric (number of distinct mean-shift-segmented regions) is designed to serve as an effective proxy for human complexity perception by estimating the diversity and numerosity of objects in an image. We decide not to rely on complexity metrics derived from pre-trained models, such as UAE or SAE, due to the inherent biases and necessarily limited vocabulary of such models (see §4 for further discussion). Instead, we run the mean-shift segmentation algorithm to generate regions of homogeneous color in an image, followed by filtering those regions for unique color and size.

In contrast to Oliva et al.'s focus exclusively on indoor scenes, and Rosenholtz et al. (2007)'s focus on artificially constructed visualizations, we test our metric for robust performance on real-world images from the Scenes, Objects, and Interior Design categories of the SAVOIAS Dataset (Saraee et al., 2018, 2020), as discussed in §§3 and 4. The development of a novel low-level visual complexity metric is motivated by Saraee et al. (2018, 2020)'s demonstration that existing low-level complexity metrics perform poorly on SAVOIAS.

2.2 Image specificity

Jas and Parikh (2015) introduce the concept of "image specificity," the tendency for multiple people to describe an image in similar words, for application to text-based image retrieval. They find that image specificity is associated with the image memorability and the presence of "important objects," i.e., the objects in an image that draw human attention and are likely to be included in an image description if present. Image specificity is also related to image content in general, as shown by the success of regression models in predicting specificity from semantic features. However, length of image descriptions alone could not predict image specificity.

Jas and Parikh's concept of image specificity is related to definitions of visual complexity that focus on the difficulty of describing complex images. Perhaps difficulty of describing complex images would translate into more variability in image descriptions, and lower image specificity. Jas and Parikh's findings about the relationship between image specificity and image content are also relevant to our own study of the relationship between linguistic complexity of image descriptions and image content. Finally, the finding that image description length alone was not predictive of image specificity suggests that predicting complexity from image descriptions, too, may be a non-trivial task.

3 Data

We use the SAVOIAS visual complexity dataset to develop our groundtruth complexity metric (§§3.1, 4), and the Microsoft COCO Dataset to train BERT for text-based prediction of visual complexity (§§3.2, 5).

3.1 The SAVOIAS visual complexity dataset

The SAVOIAS dataset addresses the problem of analysing visual complexity across diverse image types by providing human-rated visual complexity scores for images from seven categories: Scenes (200 images), Advertisements (200 images), Visualization and Infographics (200 images), Objects (200 images), Interior Design (100 images), Art (420 images), and Suprematism (100 images) (Saraee et al., 2018, 2020). We use only the Scenes, Objects, and Interior Design images, which Saraee et al. randomly sampled from Zhou et al. (2017), Lin et al. (2014), and IKEA.

Each SAVOIAS image is annotated with a visual complexity score from 0 (least complex) to 100 (most complex). Scores were generated by asking raters to compare two images at a time and choose which one was more complex. The resulting relative complexity scores were converted to absolute complexity scores via the Bradley-Terry method and matrix completion.

Saraee et al. (2018, 2020) instructed raters that visual complexity could be judged by image attributes such as number and type of objects, people, textures, patterns, and shapes present, and clutteredness of the background. Raters were also asked to use intuition to break ties between images that appeared to be approximately equally complex. Thus, the SAVOIAS visual complexity scores can be considered the result of a combination of raters' natural intuitions about visual complexity and attention to the visual cues listed in the instructions for the task.

3.2 The Microsoft COCO dataset

The Microsoft Common Objects in Context (MS COCO) Dataset was introduced by Lin et al. (2014) with the goal of improving scene understanding. COCO consists of 328,000 images annotated with pixel-level segmentation masks for 91 common object categories. Images were collected by querying Flickr with pairs of object category names as keywords, in order to yield more "non-iconic" images picturing objects in their real-world contexts.

The characteristics of iconic and non-iconic images in MS COCO, as described by Lin et al., overlap with those of noncomplex and complex images, respectively. For instance, presence of multiple object types, occlusion of objects, and clutter are characteristics of both non-iconic and visually complex images. In contrast, iconic images share many characteristics with visually noncomplex images, such as presenting objects in isolation, against a plain background, or featuring scenes devoid of people. The mix of iconic and non-iconic images in COCO makes it an appropriate choice for our own study of visual complexity, which requires examples of both complex and noncomplex images.

The Microsoft COCO Caption datasets (MS COCO c5 and MS COCO c40) provide human-generated descriptions for MS COCO images, with the goal of aiding research in automatic caption generation (Chen et al., 2015). In this work, we use MS COCO c5, which provides 5 independently generated captions per training, validation, and test set image. Captions for MS COCO c5 were collected via AMT. Captioners were instructed to focus on describing "important" parts of the images; avoid starting with "There is," making hypothetical statements about the events or people pictured, or naming people; and write in sentences of at least 8 words.

4 Visual complexity: definition

In this section, we introduce our novel number of distinct regions visual complexity metric. We use our metric to label MS-COCO images for training our text-based regression and classification models (§5).

We developed and evaluated our automated visual complexity metric based on prior work on the SAVOIAS dataset by Saraee et al. (2018, 2020). Saraee et al. (2020) introduced the Unsupervised and Supervised Activation Energy metrics, which rely on extracting activations from pre-trained object classification and scene recognition models (see §2.1). However, rather than rely on high-level complexity scores derived from pre-trained models, due to our concerns about the inherent biases of such models, we choose to focus on lower level metrics based on image processing techniques.

We are especially concerned about the accumulation of biases given the design of our text-based complexity prediction task. Recall that we first score Microsoft COCO images' visual complexity, then use these groundtruth scores to fine-tune pre-trained text-based classification and regression models. Using a complexity metric output by one model as the input labels to our text-based models could transmit and amplify biases acquired by the complexity model from training on a specific dataset.

In addition to their own UAE and SAE metrics, Saraee et al. (2020) evaluated five existing low-level automated visual complexity metrics by computing their correlations with the SAVOIAS groundtruth complexity scores. Because Scenes, Objects, and Interior Design images are most representative of the real-world photographs we are interested in analysing complexity for, we focused on developing a low-level metric that would perform well on these categories.

Of the low-level complexity metrics tested by Saraee et al. (2020), number of regions per image has the highest correlation for the Scenes category ($r = 0.57$) and the second-highest correlations for the Objects and Interior Design categories ($r = 0.29$ and $r = 0.69$, respectively). As shown in Table 1, when we recomputed number of regions using the mean-shift segmentation algorithm implementation provided by Jean (2020), we obtained slightly higher correlations of $r = 0.63, 0.36$, and 0.71. (See Comaniciu and Meer (2002) for an introduction to the mean-shift segmentation algorithm.) Therefore,

Table 1: Performance of visual complexity metrics tested by Saraee et al. (2020) versus our number of distinct regions metric on SAVOIAS Scenes, Objects, and Interior Design images, as well as for all 3 categories combined. Performance is measured by the Pearson correlation coefficient between automated and groundtruth (i.e., human-generated) complexity scores. $p < 0.001$ for all self-computed metrics (i.e., those not from Saraee et al.) on all categories. Number of regions, as computed by Saraee et al. (2020), is the best low-level complexity metric on the Scenes category and the second-best low-level metric on the Objects and Interior Design categories, behind feature congestion and compression ratio, respectively.

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**

we focused on refining the number of regions metric to use as our groundtruth complexity measure on MS COCO.

In order to qualitatively evaluate the mean-shift-segmented number of regions metric, we ranked SAVOIAS Scenes, Objects, and Interior Design images by number of regions, followed by manual inspection of the resulting rankings. We found that using the raw number of regions per image as a measure of complexity overestimated the complexity of images that were highly textured or contained repeating patterns and shapes. Figure 2 shows two SAVOIAS images which have similar numbers of regions in their mean-shift segmentations, but do not intuitively share the same level of complexity. The image on the right in Figure 2 contains many repeated instances of red and green balls, whereas the image on the left contains greater diversity of colors, shapes, and object types, suggesting that it should be considered more complex.

Our number of distinct regions metric improves upon the number of regions metric by counting only those regions which have unique size and color. As shown in the bottom row of Figure 2, our distinct regions algorithm successfully excludes regions of similar size and color (such as many of the green and red balls in the image on the right) from being counted toward an image's complexity.

In §4.1, we give an overview of our distinct regions algorithm. In §§4.2 and 4.3, we discuss how similarity between region color and size is determined.

4.1 Distinct regions algorithm

In Algorithm 1, we introduce our procedure for counting distinct regions from the set of regions S in the mean-shift segmentation of an image. We begin by initializing a set D of distinct regions with a single arbitrarily chosen region from S. Next, we iterate through the regions $s \in S$. For each region s, we compare its size to all regions $d \in D$. Any region s with sufficiently different size from all $d \in D$ is added to D. (Note that region size acts as a more readily quantifiable proxy for region shape.) Otherwise, if s is similarly sized to any $d \in D$, we compare the color of s to the colors of all $d \in D$. If s has distinct color from all $d \in D$, even though it may be similarly sized to some $d \in D$, s is added to D. Finally, if s fails both the size and color similarity tests, s is not added to D. The algorithm terminates after returning D. The complexity of the image can be computed as $|D|$, i.e., number of distinct regions in the image.

We considered alternative approaches to measuring distinctness, such as computing a hash code for each image region, then making all possible region comparisons (which would require n^2 comparisons for n regions in all cases). Ultimately, we chose the two-stage region comparison approach presented here for its relative efficiency.

Note that the number of distinct regions in an image is necessarily dependent upon the size and color similarity thresholds, δ_s and δ_c, set by the user to define size and color distinct-

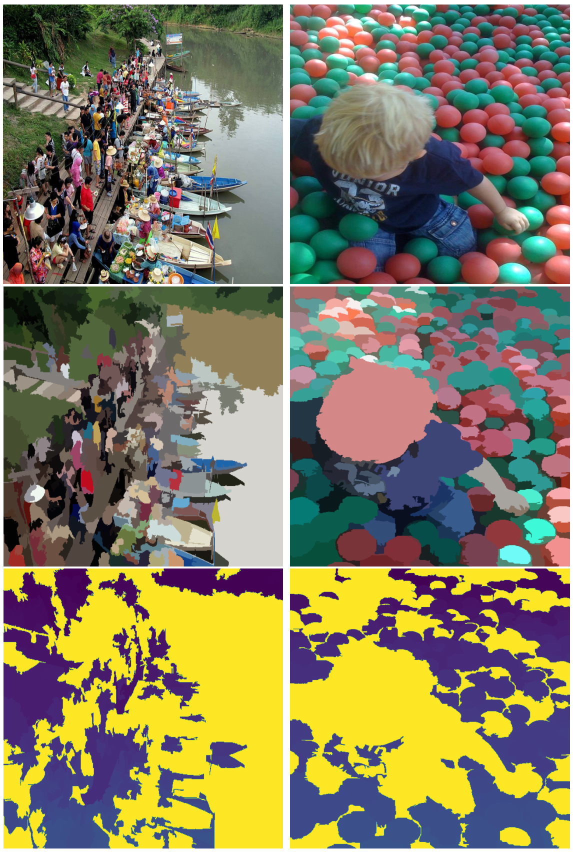


Figure 2: Two SAVOIAS images, one from the Objects category (left) and one from the Scenes category (right). The top row shows the original images, the middle row shows their mean-shift segmentations, and the bottom row shows distinct regions from the segmentations in yellow. Both images have similar complexities as measured by raw number of regions (261 regions for the image on the left and 263 regions for the image on the right). While both images contain many objects, the image on the left is more colorful and contains a greater variety of objects, suggesting that it should be considered more complex than the image on the right. Our number of distinct regions metric is derived from this observation, reducing the contribution of repeated instances of the same object (e.g., the many red and green balls in the picture on the right) to an image's complexity score.

ness. Larger values of δ_s and δ_c translate to stricter definitions of size and color distinctness, as the tests $difference(size(s), size(d)) > \delta_s$ and $difference(colors(s), color(d) > \delta_c$ become more difficult to pass.

In §§ 4.2 and 4.3, we discuss how size and color difference, $difference(size(s), size(d)) > \delta_s$ and $difference(colors(s), color(d) > \delta_c$, are computed.

Algorithm 1 Filtering for Distinct Regions from Mean-Shift Segmentation

```
Require: $S$: set of image regions, $\delta_c$: color similarity threshold, $\delta_s$: size similarity threshold
$D := \{s_{start}\}$ where $s_{start}$ some $s \in S$
$S := S - \{s_{start}\}$
for $s \in S$ do
    if $\forall d \in D \; difference(size(s), size(d)) > \delta_s$ then
        $D := D \bigcup \{s\}$
    else if $\forall d \in D \; difference(color(s), color(d)) > \delta_c$ then
        $D := D \bigcup \{s\}$
    else
        continue
    end if
end for
return $D$
end
```

4.2 Size difference

The difference D in sizes of two regions s_1 and s_2 in image I is computed as

$$difference(size(s_1), size(s_2)) = \left| \frac{size(s_1)}{size(I)} - \frac{size(s_2)}{size(I)} \right| \quad (1)$$

where $size(s_1), size(s_2)$ and $size(I)$ are computed as the number of pixels contained in the region or image, e.g.,

$$size(I) = width(I) * height(I) \quad (2)$$

Equation 1 is partially inspired by the region size similarity measure introduced in Uijlings et al. (2013).

4.3 Color difference

Region color is output by Jean's PyMeanShift segmentation algorithm in RGB colorspace, as the mean value of the pixels in the region. To compute color difference between two given regions, we convert both regions' colors to CIELAB color space via the Python colormath module (Taylor, 2021). We then compute the CIEDE2000 color difference between the two region colors (Luo et al., 2001), also using colormath.

5 Text-Based Prediction of Visual Complexity

In order to train text-based models to predict visual complexity, we need an image dataset with both (1) high-quality text descriptions of images and (2) groundtruth visual complexity scores for the images. Due to the lack of existing datasets meeting both criteria, we develop our own automated visual complexity metric (§4), use it to score the visual complexity of images in the Microsoft COCO dataset (§5.1), then train regression and classification models to predict image complexity from the image captions provided by COCO.

5.1 Data preprocessing

Because caption and instance annotations are not publicly available for MS COCO test set images, we resplit the MS COCO 2017 training and validation sets to produce our own training, validation, and test sets for training our classification and regression models. The MS COCO 2017 dataset split, including annotations, is available for download from Lin et al. (2021). Here, we report only performance on our validation set. We plan to report test set statistics in future work.

First, we segment the MS COCO training and validation set images using PyMeanShift, with a spatial radius of 3, range radius of 10, and minimum density of 300 (Jean, 2020; Comaniciu and Meer, 2002). Next, we count distinct regions per image using Algorithm 1, with $\delta_s = 0.05$ and $\delta_c = 4$. Correlations between SAVOIAS groundtruth complexity score, number of regions, and distinct number of regions for these choices of parameters can be found in Table 1.

Table 2 summarizes our resplit datasets. Each image has a single complexity label but multiple captions. Consequently, captions corresponding to the same image are assigned the same label but are treated as independent samples during training and testing.

For the classification model, we take the top and bottom 10% of MS COCO training images with the most/fewest distinct regions, labeling these as complex and noncomplex, respectively. We randomly sample 500 complex and 500 noncomplex images from the labelled MS COCO 2017 training set to use as our validation set. The remaining MS COCO training images not used in our validation set are used as our training data. Finally, we use the MS COCO 2017 validation set as our test set. We take the top and bottom 10% of validation images with the most/fewest distinct regions, and label these as complex and noncomplex, respectively.

For the regression model, we normalize the number of distinct regions r per image to yield a complexity score c in the range $(0,1)$, according to the following formula:

$$c = tanh(\frac{r}{80}) \qquad (3)$$

As shown in Figure 3, complexity scores for both the training and validation sets are approximately normally distributed.

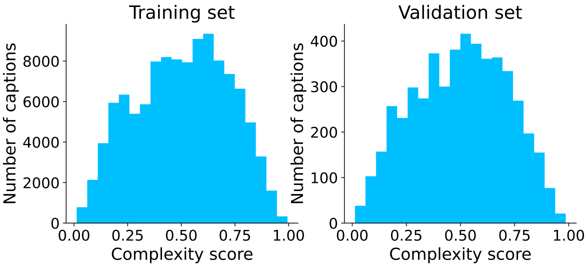


Figure 3: Distribution of complexity scores in regression training and validation sets. Complexity scores are computed as $c = tanh(\frac{r}{80})$ where r is the number of distinct regions per image.

Next, we randomly sample 5,000 images from the MS COCO 2017 training set to use as our validation set. The remaining MS COCO 2017 training set images not included in our regression validation set are used for training. We use the full MS COCO 2017 validation set as our test set.

For both the classification and regression models, COCO captions are tokenized using a pretrained BERT$_{\text{BASE}}$ (uncased) tokenizer, available from Hugging Face (Tokenizers; Kenton et al., 2019). The inputs to both classification and regression models are tokenized captions of size 128.

5.2 Finetuning the classification model

We fine-tune a pretrained BERT$_{\text{BASE}}$ (uncased) model to predict image complexity from image captions via mini-batch stochastic gradient descent, with a batch size of 10. We use the BERT$_{\text{BASE}}$ implementation available from the Hugging Face transformers library, which includes a sequence classification/regression head (Hugging Face Transformers; Kenton et al., 2019). The model inputs are tokenized COCO captions of size 128, each carrying a binary label of "complex" (1) or "noncomplex" (0) as described in §5.1. The model output is a scalar x which is normalized to yield the probability p that the input caption describes a complex image:

$$P(complex) = p = \sigma(x) = \frac{1}{1+e^{-x}} \qquad (4)$$

Our loss function is binary cross-entropy loss, where the loss l given probability p and binary

Table 2: Regression and classification dataset statistics. Because the MS COCO test set annotations are not publicly available, we create our own test set from the MS COCO 2017 validation set, and split the MS COCO 2017 training set into new training and validation sets.

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

label y is

$$l = y * log(p) + (1-y) * log(1-p) \tag{5}$$

We use the PyTorch AdamW optimizer with learning rate $\lambda = 2 * 10^{-5}$ and $\varepsilon = 1 * 10^{-8}$ (Loshchilov and Hutter, 2019, 2017). We decrease λ according to a linear schedule without warmup steps.

After fine-tuning for 4 epochs, we choose the model with the highest accuracy on the validation set.

5.3 Fine-tuning the regression model

The procedures for fine-tuning the regression model are identical to those for the classification model, except for the model inputs and loss function.

For regression, inputs are tokenized COCO captions of size 128, each labeled with a complexity score in the interval $(0,1)$, as described in § 5.1. As with the classifier, the regression model output is normalized to yield a scalar in the interval $(0,1)$, which is the predicted complexity score for the input caption rather than the probability that the image is complex.

The loss function for regression is mean squared error, computed as

$$l = (x-y)^2 \tag{6}$$

for true complexity score y and predicted complexity score x.

After fine-tuning for 4 epochs, we choose the model with the lowest loss on the regression validation set. For the purpose of computing average precision (as described in §6), we also evaluate the fine-tuned regression model on the classification validation set.

6 Results

6.1 Quantitative analysis of results

The classification model achieves an accuracy of 83.9% and an average binary cross entropy loss of 0.411 on the validation set. The ability of the text-based BERT$_{\text{BASE}}$ model to achieve such high accuracy on the visual complexity prediction task suggests substantial biases in how people describe complex and noncomplex images in COCO captions.

The regression model achieves a mean squared error loss of 0.030 and a Pearson's correlation of $r = 0.659$ $(p < 0.001)$ between predicted and true complexity scores on the validation set. When predicted complexity scores are averaged across all captions for each image, the correlation increases to $r = 0.716$ $(p < 0.001)$, demonstrating that access to multiple text inputs results in more precise predictions. However, even in the more realistic real-world scenario of access to only a single caption, the regression model seems to be extracting substantial information about the visual complexity of COCO images from their captions.

We also compute average precision for both classification and regression models with scikit-learn (Pedregosa et al., 2011). The average precision computation for the classifier is straightforward, requiring the model's probability estimates of the positive class ("complex") and labels on the classification validation set. For the regression model, we treat its predicted complexity scores as the output of a decision function for classifying images as complex or noncomplex. We compute average precision for the regression model only on the classification validation set, for which we have binary complexity labels.

Average precision for the classification model evaluated on the classification validation set is 0.913. Average precision for the regression model evaluated on the classification validation set is 0.951. The regression model's better performance can be attributed to its much larger training set, 566,747 captions compared to 113,342 captions for the classification training set (see Table 2).

6.2 Qualitative analysis of results

Table 3 shows images predicted by the fine-tuned BERT classifier to be complex (top row) and noncomplex (bottom row), based on their captions. (The results for the regression model are similar.) While the model's predictions are mostly correct and correspond to our intuitive sense of visual complexity, some biases are evident.

For example, BERT tends to predict that food images will be complex, as with the second and fourth images in Table 3. We hypothesize that this is due to two main factors. First, our number of distinct regions metric tends to count many regions for images of food, since they are typically colorful and contain a large number of objects, e.g., piles of fruit at an outdoor market or toppings on a pizza. Second, even discounting any bias in our groundtruth complexity algorithm, a substantial majority of images containing food in the COCO dataset are complex, as shown in Figure 4. Thus, the model learns to associate references to food in image captions with higher visual complexity scores.

In contrast, images containing objects of other COCO categories, e.g., the "airplane" category, have the opposite imbalance—i.e., more noncomplex than complex captions in the training set. The noncomplex airplane image in Table 3 is representative of most airplane images in the COCO dataset: the airplane is centered against the plain background of the sky. Thus, image captions containing the word "airplane" are much more likely to result in a noncomplex prediction. Image captions describing beach scenes, surfers, and skiiers also tend to be predicted noncomplex, for the same reason: images of these scenarios are frequently noncomplex in the COCO dataset, as shown by the example images in Table 3 and the dataset statistics in Figure 4.

The model's learned, MS COCO–specific biases may hurt its ability to generalize to other data. After all, not all real-world images of food are complex, and not all real-world images of airplanes are noncomplex. Indeed, the second noncomplex prediction in Table 3 illustrates model error presumably due to overreliance on captioning content for making its predictions. While most MS COCO images of skiiers are noncomplex, with the majority of the image taken up by uniformly colored snow or sky (which are segmented into very few regions when computing the complexity score), the image in Table 3 is clearly complex, featuring a large, colorful crowd. We would expect the model to pick up the words "huge audience" in the caption as a cue of high visual complexity, but instead the model likely takes the word "skiier" as a cue of low complexity.

We would like to determine if it is possible for the model to learn to predict visual complexity from syntax instead of vocabulary. In §7, we discuss our approach to mitigating the effects of complexity biases in the MS COCO dataset.

7 Mitigating Complexity Biases in the MS COCO Dataset

The classification and regression models trained on our full datasets learn content-related biases, e.g., that images of food in the MS COCO dataset tend to complex and that images of airplanes tend to be noncomplex. These biases harm model accuracy and generalizability.

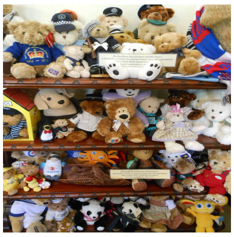
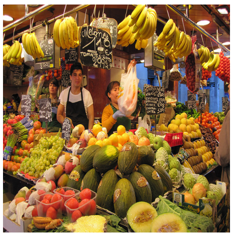
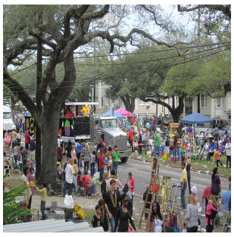
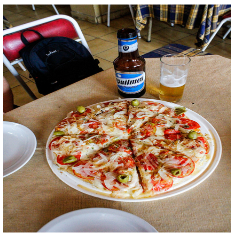
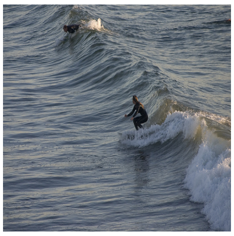
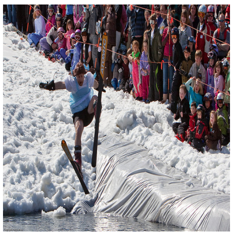
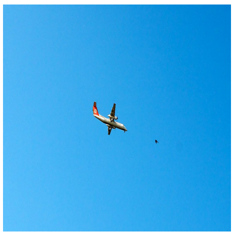
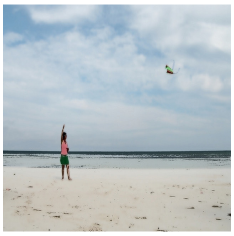
In this section, we describe two approaches to mitigating the complexity biases in the COCO dataset while fine-tuning our models to predict visual complexity from text. In Section 7.1, we describe the setup for our cross-domain evaluation experiments, and in Section 7.2, we describe our experiments fine-tuning with hypernymized captions. In Section 7.3, we discuss the results.

7.1 Cross-domain evaluation

We hypothesize that fine-tuning BERT on an object category–specific subset of the COCO captioning dataset, rather than the full set, should decrease the model's ability to leverage content-related complexity cues, forcing it instead to rely on sentence structure. For example, if a model is trained only on images containing food, it would be forced to focus on complexity cues other than the references to food in the captions in order to make its predictions. Furthermore, the model would be less prone to develop biases with respect to typically noncomplex COCO dataset object categories. For example, if the model has never seen captions for images of airplanes during training, it should not have a bias toward predicting that airplane images are noncomplex (despite the fact that this is true in the COCO dataset).

We carry out our cross-domain evaluation as follows. First, we filter our classification and regression datasets (see §5) using the publicly available COCO instance annotations, to create 12 new

Table 3: Examples of images predicted by BERT to be complex (top row) and noncomplex (bottom row) based on their captions.

Complex	 "several different types of stuffed animals arranged on shelves." $p(complex) = 0.994$, label = 1	 "a colorful farmers market has vegetables and fruit on display." $p(complex) = 0.995$, label = 1	 "a crowd gathered for a small-town parade looks on as the next float comes down the street." $p(complex) = 0.994$, label = 1	 "a plate with sliced pizza and a bottle of beer." $p(complex) = 0.991$, label = 1
noncomplex	 "a couple of surfers in wetsuits catching a gentle wave" $p(complex) = 0.002$, label=0	 "a skiier jumps into the air in front of a huge audience ." $p(complex) = 0.004$, label=1	 "the airplane is flying in the clear blue sky." $p(complex) = 0.002$, label=0	 "a woman on a sandy beach flying a kite." $p(complex) = 0.001$, label=0

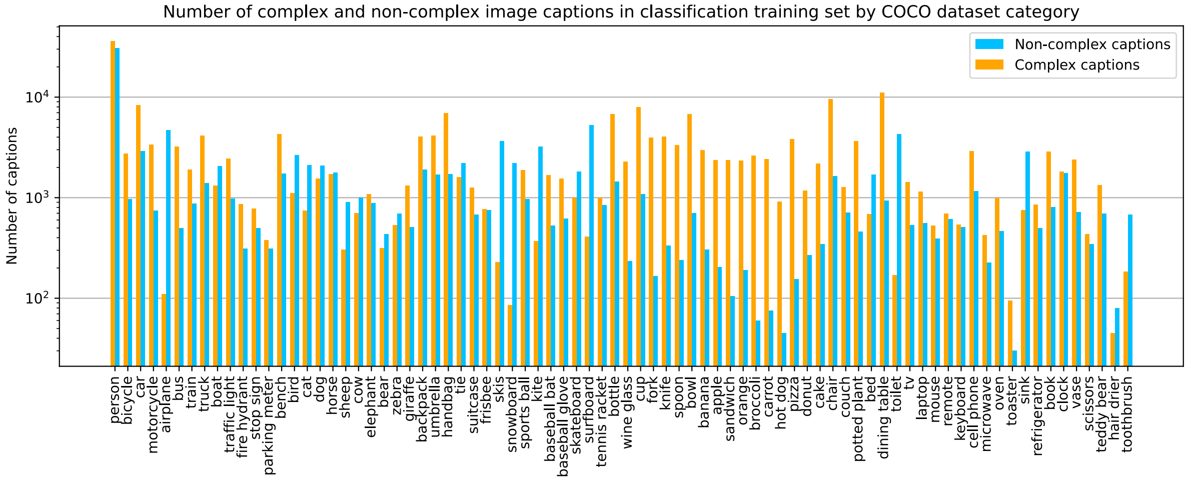


Figure 4: Complex and noncomplex captions in the classification training set by COCO dataset object category. Each pair of bars represents the number of image captions in the training set for images that include at least one instance of the corresponding category. Note that images containing food- and kitchen-related objects tend to be more complex, while images containing airplanes and certain sports- or bathroom-related objects tend to be less complex.

datasets. Eleven datasets contain captions for images that include objects belonging to each of the eleven COCO supercategories. The twelfth dataset contains captions for images that include instances of the COCO *person* category. Training set statistics are summarized in Table 4. Note that these category-specific datasets are not perfectly disjoint, as many COCO images contain multiple object categories.

We fine-tune $\text{BERT}_{\text{BASE}}$ classification and regression models on each of the 12 supercategory datasets, using mostly the same setup as described in §§5.2 and 5.3. To mitigate for the class imbalance in the classification datasets, we use the PyTorch Weighted Random Sampler to ensure balanced mini-batches during training. Given a list of 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

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

We still use mini-batch stochastic gradient descent, taking batches of size 10. Note that we sample with replacement, which means that a given caption may appear multiple times in the same batch.

We evaluate each of our fine-tuned models on the person dataset, all 11 supercategory datasets, and the full dataset. See Section 7.3 for results.

7.2 Training with hypernymized captions

Nouns, verbs, adjectives, and adverbs carry most of the object content cues in image captions. We hypothesize that fine-tuning BERT on captions where these parts of speech are hypernymized will result in a more generalizable model, with less complexity bias learned from the COCO dataset.

For the hypernymization, we first use NLTK's off-the-shelf part-of-speech (POS) tagger to label the parts of speech in our COCO captioning data with the Penn Treebank tagset (Loper and Bird, 2002). We then make the substitutions shown in Table 5, substituting generic placeholders for all nouns, verbs, adjectives, and adverbs.

To investigate how much the models rely on caption content, we create four versions of each dataset described in Section 7.1, with:

1. nouns hypernymized
2. nouns and verbs hypernymized
3. nouns, verbs, and adjectives hypernymized
4. nouns, verbs, adjectives, and adverbs hypernymized

Thus the caption

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

becomes

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

Limitations of our hypernymization approach include error in the POS tagging. We find that the default off-the-shelf NLTK POS tagger achieves 89.3% accuracy on the Penn Treebank Corpus and 61.9% accuracy on the Brown Corpus (Marcinkiewicz, 1994; Francis and Kucera, 1979). However, from qualitative observation, most sentences are still comprehensible after undergoing hypernymization.

Another limitation of our approach is the coarseness of our placeholder words. Substituting all nouns with the word "object" or "objects" obscures the distinction between uncountable nouns (e.g., "sky") and countable nouns (e.g., "objects"). Similarly, substituting all verbs with forms of the verb "act" obscures the distinction between transitive verbs (e.g., "throw [something]") and intransitive verbs (e.g., "walk"). And it is somewhat difficult to find appropriately generic placeholders for adjectives and adverbs, which are by definition descriptive.

As a possible future approach to the problem of hypernymizing COCO captions, we propose using either WordNet synsets or Glove word embeddings to select the appropriate substitution for each part of speech (Fellbaum, 2010; Pennington et al., 2014).

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

Table 4: Number of complex and noncomplex image captions per COCO (super)category for classification and regression datasets. Note that the majority of the classification datasets suffer from severe class imbalance between complex and noncomplex training examples. We mitigate class imbalance using the PyTorch Weighted Random Sampler during training.

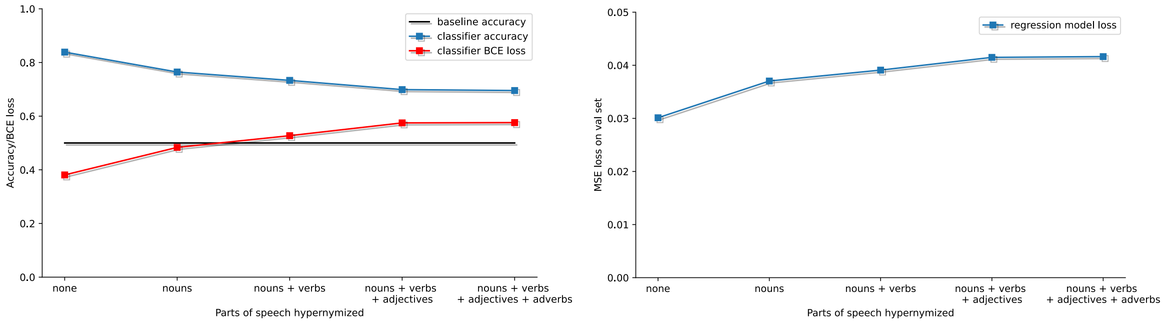


Figure 5: Distribution of complexity scores in regression training and validation sets. Complexity scores are computed as $c = tanh(\frac{r}{80})$ where r is the number of distinct regions per image.

7.3 Results

We report the results of our cross-domain evaluation experiments in Table 6. For each object category–specific dataset, as well as for the full captioning dataset, we report the training set of the best model evaluated on that dataset, baseline accuracy, validation set accuracy/loss, average precision, and (for regression models) the Pearson's correlation between predicted and true complexity scores.

As our baseline accuracy, we take the accuracy that would result from always guessing the majority class, i.e., the proportion of training captions which are labeled with the majority class (complex or noncomplex) for that category. For all 13 datasets in Table 6, the best classification model has higher accuracy on the validation set than baseline. On 4 out of 12 category-specific classification datasets and on the full dataset, the best classifier is the model trained on the full captioning data. For the regression datasets, the model trained on the full data is the best performing model on 8 out of 12 categories and the full dataset. The regression and classification models trained on the full captioning datasets do not suffer as substantial a loss in performance as expected when evaluated on image captions for only a single COCO category, contrary to our hypothesis.

Cases where models trained on a category-specific dataset outperform the models trained on the full dataset most likely result from the similarity of the training and validation data. For example, the best classifier on the "furniture" and "appliance" categories is trained on "indoor" image captions. In all cases where the best regression model on a category is not the model trained on the full set, it is instead trained on the same category.

In Figure 5 we plot accuracy and loss for the classification and regression models as we hypernymize increasing parts of speech in the COCO captioning data. As expected, accuracy decreases and loss increases for both the classification and regression models as more parts of speech are hypernymized. Note that hypernymizing adverbs when nouns, verbs, and adjectives are already hypernymized does not noticeably impact model

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

Table 5: Placeholder words for nouns, verbs, adjectives, and adverbs. By fine-tuning BERT classification and regression models on captioning data transformed with the above substitutions, we expect to produce a more generalizable, less biased model.

performance. This suggests that the majority of complexity-related vocabulary is already covered by nouns, verbs, and adjectives.

Additionally, even when all 4 parts of speech are hypernymized, the classification model still performs substantially better than baseline (random guessing), at 69.5% accuracy on the validation set. We therefore conclude that in addition to vocabulary, BERT must rely at least partially on sentence structure to make its complexity predictions.

8 Conclusion and future work

In this work, we introduced the task of text-based visual complexity prediction, hypothesizing that image descriptions typically contain enough cues of image complexity for a model to predict visual complexity solely from the descriptions. We defined a new visual complexity metric, number of distinct regions, and used it to groundtruth the complexity of images from the Microsoft COCO dataset. We found that BERT$_{\text{BASE}}$ models fine-tuned on COCO captioning data to classify images as complex/noncomplex or predict a continuous complexity score succeed on the task with a high degree of accuracy. BERT appears to primarily rely on vocabulary and, to some extent, sentence structure in its predictions. Future models of visual complexity may thus benefit from integrating language and vision inputs.

We also found biases in the COCO Dataset with respect to how different object categories are typically represented, and which combinations of objects tend to correspond to complex images. This has implications for other models trained on COCO, which may pick up the dataset's biases.

In future work, we plan to release test set performance statistics for our models, perform further analysis of COCO captions' sentence structure, and visualize BERT's attention to different parts of the captions. Possible alternative approaches to identifying complex images include using multimodal models or datasets other than COCO, thus expanding the study of visual and linguistic complexity to more domains. Future applications of complexity modeling could include identification of problematic images for object segmentation or detection models and evaluating aesthetic appeal and ease of use for artwork and web design (Saraee et al., 2018, 2020).

9 Code

Code for this project is available at `https://github.com/emlinking/visual-complexity`.

10 Acknowledgements

The author would like to express her gratitude to Vicente Ordonez and Ziyan Yang for their mentorship throughout this project and the writing of this report. 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 material is based upon work 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.

References

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Dorin Comaniciu and Peter Meer. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE*

Table 6: Results of cross-domain evaluation for classification (top) and regression (bottom) models trained on images annotated with 11 COCO supercategories, the person category, and the full captioning data. "Baseline accuracy" refers to the proportion of the majority class (complex or noncomplex) in the training data, i.e., the accuracy we would expect from guessing the majority class 100% of the time. Average precision for the regression models is computed by treating the model predictions on the classification validation set as decision function outputs. "Best classifier" refers to the model with the highest validation set accuracy. "Best regression model" refers to the model with the lowest average mean squared error on the validation set.

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

Transactions on pattern analysis and machine intelligence, 24(5):603–619.

Christiane Fellbaum. 2010. WordNet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.

W Nelson Francis and Henry Kucera. 1979. Brown corpus manual. *Letters to the Editor*, 5(2):7.

Hugging Face Transformers. 2022. Hugging Face Transformers: State-of-the-art Machine Learning for Pytorch, TensorFlow, and JAX.

IKEA. 2022. IKEA.

Mainak Jas and Devi Parikh. 2015. Image specificity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2727–2736.

Frédéric Jean. 2020. Python Module for Mean Shift Image Segmentation. Original-date: 2015-08-14T21:35:39Z.

Lee Kenton, Jacob Devlin, Ming-Wei Chang, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Tsung-Yi Lin, Genevieve Patterson, Matteo R. Ronchi, Yin Cui, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, Larry Zitnick, and Piotr Dollár. 2021. COCO - Common Objects in Context.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations 2019*, page 18, New Orleans. OpenReview.net.

M. R. Luo, G. Cui, and B. Rigg. 2001. The development of the CIE 2000 colour-difference formula: CIEDE2000. *Color Research & Application*, 26(5):340–350. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/col.1049.

Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of english: The Penn Treebank. *Using Large Corpora*, 273.

Aude Oliva, Michael L Mack, Mochan Shrestha, and Angela Peeper. 2004. Identifying the Perceptual Dimensions of Visual Complexity of Scenes. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26, pages 1041–6.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Ruth Rosenholtz, Yuanzhen Li, and Lisa Nakano. 2007. Measuring visual clutter. *Journal of vision*, 7(2):17–17.

Elham Saraee, Mona Jalal, and Margrit Betke. 2018. SAVOIAS: A diverse, multi-category visual complexity dataset. *arXiv preprint arXiv:1810.01771*.

Elham Saraee, Mona Jalal, and Margrit Betke. 2020. Visual complexity analysis using deep intermediate-layer features. *Computer Vision and Image Understanding*, page 102949.

Gregory Taylor. 2021. Python Color Math Module (colormath).

Tokenizers. 2022. Tokenizers. Original-date: 2019-11-01T17:52:20Z.

Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. 2013. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464.