Can Robots Love Kittens?
A Study in Cuteness and Computer Vision

I609 Final Project
Sam Goree

In the pursuit of artificial intelligence (AI), we hope that the artificial agents we create will exhibit robust, multifaceted intelligence, similar to humans. Despite tremendous recent progress in artificial intelligence research in areas like machine learning, our understanding of topics related to other facets of intelligent behavior, in particular, emotional intelligence is still limited.

It seems reasonable to believe that that will change though. Popular media imagines AI systems like virtual assistants or collaborative robots which can engage with humans in a robust, multifaceted manner. It imagines these systems to have the capacity for thoughts and feelings in a humanized way. Unfortunately, real progress in AI research is on finding better learning algorithms for classification or regression problems and does not directly invite humanization. While there is no doubt that better learning algorithms are highly valuable, if we as a field want to meet the popular conception of AI technology, we may have to increase the priority of research into more humanizing topics like artificial emotional intelligence.

In this paper, we make an argument for why research on detecting emotional stimuli would be an important contribution towards humanized AI systems, then pose the question: do current methods in computer vision have the capacity to effectively solve these detection problems? To address that question, we consider a particular problem, cuteness detection, and attempt to solve it using current methods in computer vision. We then critically evaluate whether those methods constitute an effective solution, and make suggestions for future computer vision research in this area.

1 Affective Computing and Emotional Stimuli

In this section, we will define affective computing and explore its relationship to AI. We argue that detecting and responding to emotional stimuli in an environment is crucial to an artificial agent's ability to afford humanization.

The term affective computing was coined by Rosalind Picard. She provides the definition, "computing that relates to, arises from, or influences emotions," [20]. Picard also provides a framework for thinking about different ways affect may present itself in computer systems:

I Computers which cannot perceive or express affect, and are "neither personal nor user-friendly" [20].

II Computers which express affect, through voices, faces or other interfaces.

III Computers which perceive the user's affective state and use it somehow.

Computer	Cannot express affect	Can express affect
Cannot perceive affect	I.	II.
Can perceive affect	III.	IV.

Table 1: Four categories of affective computing, focusing on expression and recognition.

Table 1: Picard's categories of affective computing.

IV Computers which provide personalized, user-friendly computing by perceiving and expressing emotions. Picard notes that these kinds of machines do not need to be emotive in a human sense.

While some topics in affective computing are fundamentally design problems (e.g. creating user-friendly voice interfaces) or engineering problems (e.g. detecting emotions from biometric sensors), there is a large intersection with AI research on topics like synthesizing affect in machines. Category II, in particular, holds several interesting AI problems related to creating artificial agents which respond to emotional stimuli in believable ways. A smiling face on a computer screen is not enough to make us think an artificial agent can express affect (Microsoft's agent Clippy, which used a smiling paper clip to try to be fun and helpful ended up being infamously annoying, see [21]), we will only ascribe emotions to agents that recognize emotional stimuli and respond in believable ways.

We discuss ascribing emotions to agents rather than agents that have emotions in some absolute sense for two reasons. First, determining whether something, even a human, has emotions at all is not a solved problem in philosophy, so even if we managed to build an agent with emotions, we would not be able to verify it. Second, when we are interested in emotions in artificial agents, for applications like virtual assistants or collaborative robots, we are really interested in the emotional or affective aspect of a social interaction between that agent and a human [7]. This approach turns a deep psychological problem into an easier algorithmic one: how do we create computer programs which afford humanization (i.e. encourage the user to ascribe emotions to them)?

Like other areas of AI, approaches to emotional intelligence will likely not attempt to explicitly model the brain, even if they try to replicate the brain's behavior. When trying to recreate human problem-solving abilities, for example, AI researchers have prioritized well-defined problems (like Chess, see [8]) with clear evaluation metrics and tried to come up with algorithms to solve them. Affective computing research is difficult to do within that framework because there is a lot of inherent subjectivity in both what the problems should be and how we should evaluate a computer's answers. While defining our way through that subjectivity may be difficult, finding a way to replace the existing framework of AI research seems more difficult, since it is unclear what the alternative would be. It seems reasonable

to believe that path of least resistance towards artificial emotional intelligence is through formalizations of subjective problems, like emotion recognition, rather than a paradigm shift in the way we solve those problems.

If our goal is to create algorithms which respond to stimuli and afford humanization, it seems natural to develop algorithms to recognize stimuli which should elicit specific emotional responses. If an artificial agent can recognize scenes which are happy, sad, scary or funny, it can react appropriately. We imagine this kind of technology to be helpful in two contexts. First, consider a companion robot for a child which has vision sensors and enough motor control over its facial expressions to smile, frown and look concerned or frightened. When a human user brings the robot into contact with a smiling human child, the robot should know to smile back. Alternatively, if the robot sees that child fall while playing, the robot should show concern and comfort the child (after getting help, of course). Second, consider a virtual assistant which has access to a user's photo collection. If a user asks the assistant for "a cute photo of my dog" the assistant system should be able to distinguish between cute photos and other photos of the user's dog.

The stakes surrounding appropriate reactions are high, though. If an agent reacts inappropriately (e.g. the companion robot sees the child fall and laughs), it may break the illusion that the agent is responding to emotions, or worse, really hurt the human involved. As a result, recognizing emotional content in scenes correctly, or recognizing when the emotional content is uncertain, is essential to creating realistic affective agents.

2 Cuteness Detection

In the following sections, we will explore a particular emotional stimulus, cuteness, as a computer vision problem. Before discussing exactly how we intend to frame and solve cuteness detection, we will discuss the existing literature on cuteness and cuteness detection in psychology, design and computer science.

The human visual system is highly related to several schemas which trigger emotional responses in humans, including an instinctive response to infantile features that Konrad Lorenz called *Kindchenschema* (literally "baby schema") [16]. Humans exhibit a positive affective response when they see cute things, including human children, animals and inanimate objects that display infantile features [10], and this response is specific enough in the brain to be detected using fMRI [11], but has remarkable effects on our behavior which can be exploited to achieve societal effects from cartoon marketing to animal conservation [9]. Cuteness provides an interesting case of a low-level instinctive response to stimuli that can be modeled without building a complex artificial sensory-motor system, but is central enough to our own emotions to be instantly recognizable.

Marcus et al. argue that cuteness is a much broader, socially constructed and culturally relative concept which is sometimes related to gender. They enumerate a taxonomy of cuteness, including 24 different reasons something can be cute spanning different kinds of exaggeration, cultural associations and gender-related concepts. They also claim that cuteness as we understand it today is a relatively recent phenomenon: the word "cute" itself 19th

century American slang's bastardization of the word acute, meaning sharp, and unlike other aesthetic concepts like beauty, cuteness does not have a deep philosophical tradition discussing it. Cuteness already has an influential role in design, though, as interfaces which are designed to be cute are used to calm and persuade us. Cute designs are particularly prevalent in China and Japan, where cuteness engineering has seen extensive corporate attention [17].

We found two examples of related work specifically on cuteness detection. Bao et al. use a support vector machine (SVM) method to predict whether pictures of dogs, cats or rabbits were cute or not [2]. They use a sophisticated feature extraction process, involving a Fisher vector representation of SIFT features, and treat cuteness detection as a classification problem. A SVM works well, classifying images with 84% accuracy. Notably, they do not describe how their dataset was labeled.

Wang et al. constructed a dataset of human faces with labeled cuteness scores, then extracted classical computer vision features (Gabor, LBP and HOG features) and also use a SVM method to predict those cuteness scores as a regression problem rather than a classification one [23]. They arrived at the ground truth scores by asking forty human participants to rank small subsets of the dataset, then infer cuteness scores on a 0-10 scale which preserve those ranks. This approach seems to work well, resulting in 1.27 mean absolute error on the ten point scale. They do not discuss the demographic makeup of their participants, nor do they investigate how cultural biases may have played into their scores.

In a related domain, You et al. use convolutional neural networks (CNNs) to classify images based on their associated emotion [24]. They gather a large dataset of art photographs and pay Amazon Mechanical Turk (AMT) human workers to label them with one of eight emotional categories: amusement, anger, awe, contentment, disgust, excitement, fear and sadness. They are able to solve the eight-class classification problem with reasonable success, achieving 58% accuracy using their AlexNet-style classifier.

Ahres and Volk complete a similar task using a more sophisticated active learning approach to train a CNN to classify 19 abstract or emotion-related concepts, including cuteness, on the NUS-WIDE image tagging dataset [1]. [1] Unlike the dataset used by You et al., this dataset allows a single image to have multiple concept labels, and this added complexity means that their approach is mostly unsuccessful, achieving an 18% F1 score. They report 37% recall and 5% precision on the "cute" tag specifically.

3 Methods

In order to approach cuteness detection within the framework of AI research, we will frame it as a binary classification problem, where our goal is to produce a model which classifies images into one of two categories, cute or not cute, and solve it using a supervised learning algorithm trained on a large image dataset, where each image is labeled cute or not cute.

[1] This paper is a student report from a graduate-level class, not a publication, and hasn't been peer reviewed. If I was going to publish this paper, I would have to get permission from the authors before citing their unpublished work.

Traditionally, researchers would evaluate the quality of a model by measuring its accuracy on an unseen test set and comparing its output with the labels.

The state-of-the-art models for image classification in computer vision are deep CNNs, which learn progressively higher-level representations of image data via the backpropagation algorithm [15]. While CNNs are not an accurate model of the human visual system, they are extremely effective for a variety of computer vision tasks, including image classification [14].

To mitigate the large data requirements for training deep neural networks, we use an existing model architecture, the 18 layer version of ResNet [13], pre-trained on the ImageNet dataset [6], a large-scale image classification challenge. Instead of using the existing last layer, which classifies images into one thousand categories, we replace it with a 2-class softmax activation output layer and fine-tune the network by training on a dataset of images labeled "cute" or "not cute" using a binary cross-entropy loss function and stochastic gradient descent. We implement our model in Python using the standard PyTorch implementation of ResNet [18].

Since none of the existing literature has approached cuteness detection posed in this particular way ([1] uses the same dataset, but is trying to solve a multi-class classification problem on more than just these tags, so it would be an unfair comparison), we use a K Nearest Neighbors (KNN) classifier on ResNet features as a baseline. KNN is a simple classifier which predicts the class of a new image by taking a majority vote of the k nearest training images. We calculate that "nearness" by measuring the euclidean distance between the ResNet features of those images. We show results for k values of 5, 10 and 20. Also for the sake of comparison, we use the Viola-Jones face detection algorithm pre-trained to detect full frontal faces, which is a reasonable if we assume that the cuteness of an image correlate with the presence of faces in that image [22]. For Viola-Jones, we use the OpenCV implementation [3].

We use the NUS-WIDE multi-class classification dataset, which contains over two hundred thousand images collected from Flickr, split into training and test sets [5]. The dataset uses a curated set of one thousand tags from Flickr, one of which is "cute" which we use as a ground-truth labeling. We sample 3600 images from the NUS-WIDE training set, half of which are labeled "cute" and half of which aren't, and a balanced test set of 2358 images from the NUS-WIDE test set, then further split the training images into training and validation sets with an 80-20 split.

Like any crowdsourced data labeling, the NUS-WIDE tags are applied to images in a subjective and inconsistent manner. As a result, a consistent classifier with an accuracy value of 100% is not always possible, since the subjective definition of cuteness that a model learns may still disagree with some Flickr users' definitions of cuteness. This limitation also means that a classifier like ours should not be taken as an "objective" or "crowdsourced" definition of cuteness, since such a thing does not exist.

While analyzing our results, we make use of another kind of supervised learning model, a decision tree classifier. Decision trees use a series of decision rules to arrive at classifications for input data, but are not used as image classifiers often because they tend to overfit on

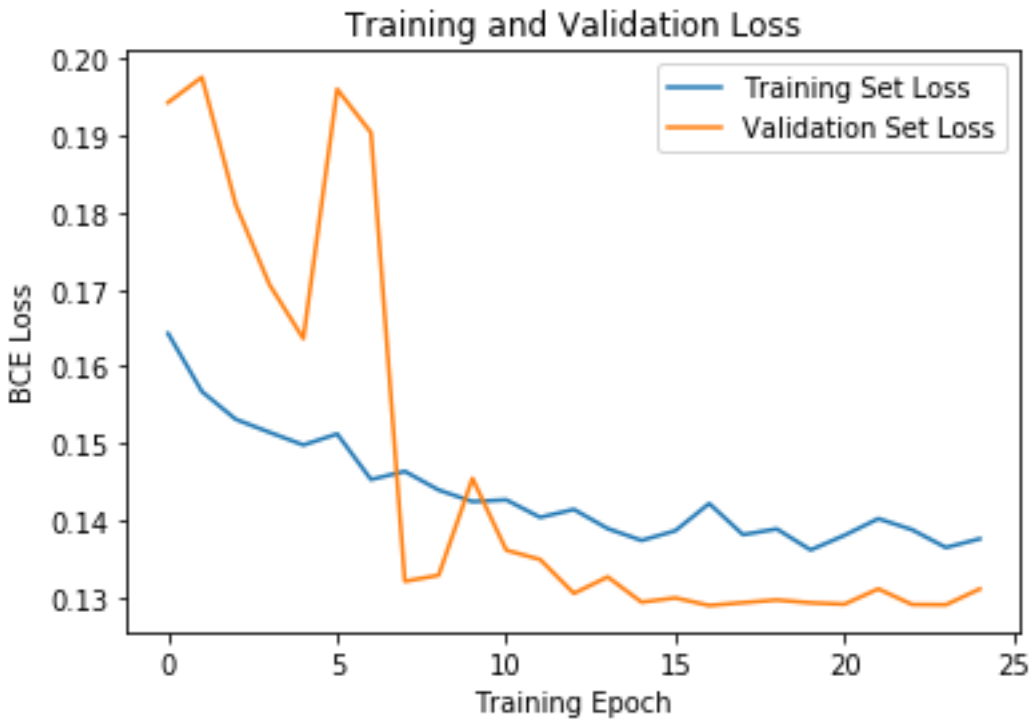

Figure 1: Training and validation loss over 25 epochs (passes through the training dataset).

complex data. These models are very useful, though, for producing "explanations" for the output of less interpretable models, which is how we use them here. While our CNN model is classifying cuteness from image pixels, we use decision trees to classify both whether an image is cute, and whether our CNN model will classify it correctly, with the other 999 NUS-WIDE tags as features. We build these trees using the Scikit Learn implementation of the CART algorithm [19] [4]. Note that in order to avoid training the decision trees on CNN training outputs, all decision tree training and test examples are taken from a split of the NUS-WIDE test set.

4 Results

Our model converges quickly and appears to generalize well, indicated by low validation loss (fig. 1). Accuracy on the held-out test set is 79.8%. The confusion matrix for the test set reveals that classification errors are evenly distributed between the classes (table 3). Our model significantly outperforms both KNN, which struggles to outperform a random classifier, and the Viola-Jones face detector.

A decision tree trained on the other tags has similar performance, around 79%, see table 4. Interestingly, the tree disagrees with our CNN's classifications (which remain at 79% on this smaller test set) about 30% of the time, usually because the CNN predicts positive while the

Model	Accuracy	Precision	Recall
Random	0.5	0.5	0.5
KNN ($k = 5$)	0.5	0.5	**1.0**
KNN ($k = 10$)	0.49	0.48	0.13
KNN ($k = 20$)	0.5	0.5	0
Viola-Jones	0.54	0.59	0.23
CNN	**0.79**	**0.79**	0.81

Table 2: Results on our NUS-WIDE test set (balanced with the 1179 positive examples and a consistent set of 1179 randomly selected negative examples). The KNN classifier with $k = 5$ achieves perfect recall because it outputted "cute" on every test example.

	Predicted Negative	Predicted Positive
Actual Negative	926 (TN)	253 (FP)
Actual Positive	224 (FN)	955 (TP)

Table 3: Confusion matrix for CNN classifier on NUS-WIDE test set.

tree predicts negative. Some example images are shown in fig. 2. We can examine the exact decision rules the tree uses, and find that the most important tags in order start with: pet, baby, cat, adorable, pretty, funny, animal, dog, young.

In order to investigate the performance of our classifier further, we gathered a handful of difficult examples, including one adversarial example constructed using the BFGS method described in [12]. Those results are shown in fig. 3 and fig. 4.

5 Discussion

At first glance, our classifier appears to perform well, achieving almost 80% accuracy on unseen examples. Since this classification problem is quite noisy and our labels are necessarily inconsistent, 80% accuracy is likely close to the ceiling for classifier performance based on only visual information. Our classifier significantly outperforms both KNN, which struggles to surpass a random baseline, and the Viola-Jones face detector. The poor performance of KNN likely is due to the curse of dimensionality: in high dimensional spaces, the amount of data necessary to fill that space increases exponentially with the number of dimensions. Since we compute distances over 512 dimensional ResNet feature space, 3600 training images are insufficient for a nearest neighbors algorithm. The poor performance of Viola-Jones

	Predicted Negative	Predicted Positive
Actual Negative	209 (TN)	20 (FP)
Actual Positive	79 (FN)	164 (TP)

Table 4: Confusion matrix for decision tree classifier evaluated on a subset of the NUS-WIDE test set.

(a) Ground truth: cute
CNN: cute
Tree: not cute

(b) Ground truth: cute
CNN: cute
Tree: not cute

(c) Ground truth: not cute
CNN: cute
Tree: not cute

(d) Ground truth: cute
CNN: not cute
Tree: cute

(d) Ground truth: cute
CNN: not cute
Tree: not cute

(d) Ground truth: not cute
CNN: cute
Tree: cute

Figure 2: Some examples of where our CNN, decision tree and ground truth labels disagreed.

(a) Our CNN classifier outputted "not cute" with 89% probability for an image which we consider cute.

(b) Our CNN classifier outputted "cute" with 94% probability for an image which we do not consider cute.

(c) Our CNN classifier outputted "cute" for both of these images with over 99.999% probability.

Figure 3: Images found on the Internet which we deemed difficult to classify. [25] [26] [27].

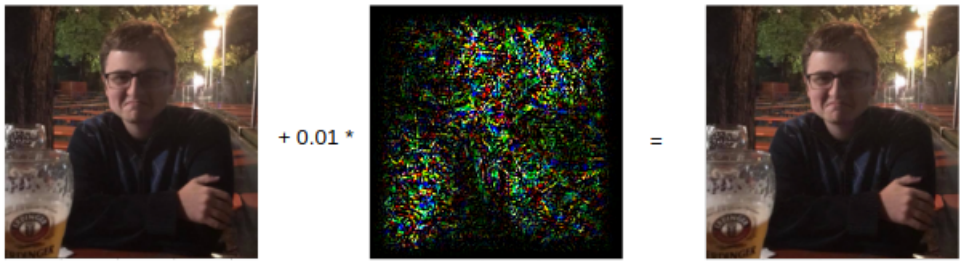

Figure 4: An image of the author, classified "not cute" with probability 0.983. When the distortion in the center image is added, it produces the right image, which is classified as "cute" with probability 0.975.

likely indicates that detecting faces is not an adequate shortcut, our classifier must be doing something more sophisticated.

The decision tree results show that our model is also not taking an object detection shortcut based on the presence of specific objects which tend to be cute, since a decision tree trained to predict cuteness from the other tags, which mostly indicate objects, often disagrees with our model. Decision tree results also show that there is no specific object or combination of objects which we are systematically misclassifying, as it is difficult to predict our models' performance from the other tags.

Despite our apparent success at classifying images correctly, we would argue that it is premature to consider cuteness classification a solved problem. We provide four counterexamples (fig. 3, fig. 4):

- Gollum from *Lord of the Rings* has infantile features and large eyes, but is not cute.
- A small tart which does not have clearly cute features, but is cute by virtue of its size.
- Photos of a muffin and a chihuahua which have very similar visual features, but should be classified differently, since a muffin isn't nearly as cute as a chihuahua.
- A photo of the author, which is not particularly cute, modified slightly on the pixel level until the network classifies it as cute.

Our classifier classifies Gollum, the muffin and the chihuahua as cute and the tart as not cute, all with high probability. True adversarial attacks, as was done with the picture of the author, These results might indicate that while our classifier has a good understanding of which visual patterns indicate cuteness, but their decision boundary may not be as complex or nuanced as that of a human. More research should be done to determine whether a difference of opinions about cuteness with respect to Gollum or a tart, which are within the realm of human subjectivity, would actually break the illusion that an artificial agent using this classifier has a human-level understanding of cuteness. If these sorts of discrepancies do have a significant difference, then we may have to rethink the evaluation metrics that we use.

An important limitation shown by these counterexamples is that the class probability that our model outputs is not analogous to an estimate of how cute the image is. All of the training data has been labeled "cute" or "not cute" in a binary sense, so there is no information comparing two cute examples to one another, or any information indicating that a non-cute example would be "arguably cute," so when the model outputs .99999 for the muffin, that only indicates that it is far from the model's learned decision boundary on the cute side, not an estimate of how cute that image is.

We would encourage future research in this area to focus on better modeling uncertainty, or predicting when humans might disagree about the cuteness of an image, rather than just predicting whether the majority would say cute or not. This problem is also a good candidate for model visualization research: if a generative model was trained to maximally excite individual neurons of a cuteness detector, we could learn more about what particular image features capture, and analyze the last layer's weights like coefficients of logistic regression.

6 Conclusion

In this paper, we argued that AI research must contend with affective computing problems like detecting emotional stimuli in order to create artificial agents which afford humanization. We then described an experiment to investigate whether current methods in computer vision have the capacity to perform cuteness detection, and investigated the learned model's solution to verify that it is learning to detect cute images, without taking any shortcuts. Our experiments show that our model does indeed appear to be learning to detect cuteness, but that doesn't necessarily mean that this problem is solved. Cuteness detection, and other emotional stimuli detection problems, may be useful test cases for developing new vision algorithms which are able to model their own uncertainty in order to avoid causing artificial agents to react inappropriately to their environments.

References

[1] Y. Ahres, N. Volk. "Abstract Concept and Emotion Detection in Tagged Images with CNNs." Unpublished Report, accessed from `http://cs231n.stanford.edu/reports/2016/pdfs/008_Report.pdf`.

[2] Y. Bao et al. "Cuteness Recognition and Localization in the Photos of Animals." ACM Multimedia. 2014.

[3] G. Bradski. "The OpenCV Library." *Dr. Dobb's Journal of Software Tools*. 2000.

[4] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and Regression Trees", Wadsworth, Belmont, CA, 1984.

[5] T. Chua et al. "NUS-WIDE: A Real-World Web Image Database from National University of Singapore." CIVR. 2009.

[6] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database." CVPR. 2009.

[7] P. Dumouchel and L. Damiano *Living with Robots*. Harvard University Press. 2017.

[8] N. Ensmenger "Is Chess the Drosophila of Artificial Intelligence? A Social History of an Algorithm" *Social Studies of Science* Vol 42, 1, pp. 5-30. 2011.

[9] G. Genosko. "Natures and Cultures of Cuteness." *InVisible Culture* 9. 2005.

[10] M. Glocker et al. "Baby Schema in Infant Faces Induces Cuteness Perception and Motivation for Caretaking in Adults." *Ethology* vol. 115, 3, pp. 257-63. March 2009.

[11] M. Glocker et al. "Baby Schema Modulates the Brain Reward System in Nulliparous Women." PNAS vol 106, 22, pp. 9115-9119. June 2009.

[12] I. Goodfellow, J. Shlens, C. Szegedy. "Explaining and Harnessing Adversarial Examples." ICLR. 2015.

[13] K. He, X. Zhang, S. Ren, J. Sun. "Deep Residual Learning for Image Recognition." CVPR. 2016.

[14] A. Krizhevskyd, I. Sutskever, G. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." NIPS. 2012.

[15] Y. LeCun, Y. Bengio, G. Hinton. "Deep Learning" *Nature* vol 521. May 2015.

[16] K. Lorenz. *Studies in Animal and Human Behavior*. Cambridge, MA: Harvard Univ Press. 1971.

[17] A. Marcus, M. Kurosu, X. Ma, A. Hashizume. *Cuteness Engineering: Designing Adorable Products and Services.* Springer. 2017.

[18] A. Paszke et al. "Automatic Differentiation in PyTorch." NIPS. 2017.

[19] Pedregosa et al. "Machine Learning in Python." JMLR 12, pp. 2825-2830, 2011.

[20] R. W. Picard "Affective Computing" MIT Media Laboratory Technical Report No. 321 November 1995.

[21] R. W. Picard "Affective Computing: Challenges" International Journal of Human-Computer Studies, Vol 59, 1-2, July 2003, pp. 55-64.

[22] P. Viola and M. Jones. "Rapid Object Detection using a Boosted Cascade of Simple Features." CVPR. 2001.

[23] K. Wang, T. Nguyen, J. Feng, J.Sepulveda. "Sense Beyond Expressions: Cuteness." ACM Multimedia. 2015.

[24] Q. You, J. Luo, H. Jin, J. Yang. "Building a Large Scale Dataset for Image Emotion Recognition: The Fine Print and the Benchmark." AAAI. 2016.

[25] Photo of Gollum from *The Lord of the Rings*. New Line Cinema. Retrieved from `https://www.vulture.com/2018/12/gollum-lord-of-the-rings-cgi-history.html`.

[26] Photo of tiny tarts. Instagram. Retrieved from `https://www.instyle.com/lifestyle/food-drink/tiny-foods-trend`

[27] Photos of Chihuahuas and Muffins. Instagram. Retrieved from https://blog.cloudsight.ai/chihuahua-or-muffin-1bdf02ec1680.