Image Analytics
A Consolidation of Visual Feature Extraction Methods

Liu, Xiaohui; Liu, Fei; Li, Yijing; Shen, Huizhang; Lim, Eric T.K.; Tan, Chee-Wee

Document Version
Accepted author manuscript

Published in:
Journal of Management Analytics

DOI:
10.1080/23270012.2021.1998801

Publication date:
2021

Image Analytics: A Consolidation of Visual Feature Extraction Methods

Xiaohui Liu[a], Fei Liu[b], Yijing Li[b], Huizhang Shen[a], Eric T.K. Lim[c], and Chee-Wee Tan[d]*

[a]Antai College of Economics & Management, Shanghai Jiao Tong University, Shanghai, People's Republic of China; [b]Department of Management and Marketing, Hong Kong Polytechnic University, Hongkong, People's Republic of China; [c]School of Information Systems and Technology Management, University of New South Wales, Sydney, Australia; [d]Department of Digitization, Copenhagen Business School, Copenhagen, Denmark

*Corresponding author. Email: ct.digi@cbs.dk

Image Analytics: A Consolidation of Visual Feature Extraction Methods

Abstract

Revolutionary advances in machine and deep learning techniques within the field of computer field have dramatically expanded our opportunities to decipher the merits of digital imagery in the business world. Although extant literature on computer vision has yielded a myriad of approaches for extracting core attributes from images, the esotericism of the advocated techniques hinders scholars from delving into the role of visual rhetoric in driving business performance. Consequently, this tutorial aims to consolidate resources for extracting visual features via conventional machine and/or deep learning techniques. We describe resources and techniques based on three visual feature extraction methods, namely calculation-, recognition-, and simulation-based. Additionally, we offer practical examples to illustrate how image features can be accessed via open-sourced python packages such as OpenCV and TensorFlow.

Keywords: Image Analytics; Attribute Extraction; Computer Vision; Deep Learning; Python

1. Introduction

With the explosive growth of multimedia data in the digital space, visual cues play an increasingly prominent role in consumers' online decision-making process. Visual rhetoric, expressed through visual features of digital imagery, has been widely acknowledged as an indispensable element governing consumer behavior (Saraee et al., 2020; Nixon and Aguado, 2012; Solomon and Breckon, 2011). Indeed, the aesthetic appeal of visual presentation of products and/or services has been documented to be instrumental in boosting the click-through (Cheng et al., 2012) and conversion (Li et al., 2014; Zhang et al., 2016) rates of online advertisements. Likewise, Li et al. (2018) discovered that facial and textual features embedded in the listing image of service offerings can spur consumer purchases by inducing service tangibility, whereas the work of So and Oh (2018) attested to the rhetorical power of object richness in driving online sales of products and/or services.

Although the importance of visual rhetoric has been touted by academics and practitioners alike (Borges et al., 2021; Li et al., 2014; Li et al., 2018), we have a somewhat limited understanding of the business value of visual rhetoric. A major challenge in disentangling the business value of visual rhetoric stems from difficulties in discerning visual features embedded in digital images and quantifying their impact (Luo et al., 2013). In this regard, the maturity of computer vision coupled with recent breakthroughs in deep learning techniques have opened up unprecedented opportunities for enabling automated extraction of visual features (Garain et al., 2021; Guo et al., 2021; Jang & Lee, 2021; Lomov et al., 2021; Alguliyev et al., 2019; Khosla et al., 2015; Kong et al., 2016) and gauging their influence on human viewers (Santra et al., 2022; Tan et al., 2021; Druzhkov and Kustikova, 2016). An accessible, systematic consolidation of the abundant resources and techniques for extracting visual features is hence critical, especially for academics and practitioners in business disciplines, to overcome hurdles in ascertaining the impact of visual rhetoric. To this end, this study consolidates cutting-edge resources and techniques in the field of computer vision to serve as a tutorial for academics and practitioners who are keen to harness visual rhetoric in their work.

This study begins by reviewing previous empirical work on visual rhetoric in order to pinpoint opportunities for applying visual feature extraction methods in management research. We then give an overview of contemporary visual features together with the framework designed to extract these features from images. Particularly, we delineate visual features into three types (sensory, supraliminal, and subliminal feature) and classify the extraction methods of visual features into three categories, namely *calculation*-, *recognition*-, and *simulation*-based, to consolidate essential resources and techniques accompanying each method. Walkthroughs are then offered to illustrate key procedural steps in the implementation of each method to extract visual features via open-source, python frameworks.

2. Visual Rhetoric In Business Management Research

Digital images and videos, due to their 'attention-grabbing' nature, possess greater persuasive power than their textual counterpart (Nixon and Aguado, 2012; Solomon and Breckon, 2011). This in turn has given rise to the prevalence of visual rhetoric in business contexts nowadays, prompting management scholars to investigate the effects of visual features in digital imagery across multiple contexts. Indeed, past studies on visual stimuli have alluded to the pivotal role of visual rhetoric in dictating consumers' attitudes and behaviors (see Table 1). Whereas Jiang et al. (2016) have demonstrated, via survey questionnaire, that users' perception of the beauty of a website significantly affects their attitude toward the site, Ettis (2017) have conducted experiments to testify to the role of online atmospheric color (i.e., blue versus yellow) in fostering flow experience and its subsequent impact in inducing consumers' approach behavior (e.g., visit duration and purchase intention) during online shopping. Likewise, Kim (2019) have blended experimentation and surveys to illuminate how product depiction and textual description in digital images influence consumers' intention to remain with a given vendor. Even though the studies mentioned above have yielded invaluable insights on the effects of visual rhetoric, the majority of them tend to be grounded in experimentation and surveys, both of which are constrained in their scalability and hamper efforts to discern the impact of visual features in digital imagery on a broader scale. Advances in machine and deep learning techniques within the field of computer vision thus offer a window of opportunity for tackling the preceding challenge (Borges et al., 2021; Lu, 2019; Tan et al., 2021; Zhang & Lu, 2021). In light of the opportunities for management scholars to employ advanced computer vision techniques in business management research, this study delivers a detailed description of the extraction technique(s) associated with each type of visual features together with a step-by-step tutorial on how to apply each technique.

---Insert Table 1 here---

3. Visual Features: An Overview

We delineate visual features within extant literature into three categories according to the way images are processed by humans: *sensory*, *supraliminal*, and *subliminal* features. ***Sensory features*** denote elementary patterns of digital imagery that can be directly identified by human optical systems without cognitive interpretation (Pylyshyn, 1999). These elementary patterns range from *photographical properties of images* in the likes of colors, brightness, and texture, to *embedded componential entities* such as human faces (Garain et al., 2021; Nguyen et al., 2016), ordinary objects (Sunaina et al., 2021; Herranz et al., 2016), and text (Borges et al., 2021; Zhou et al., 2017). Both photographical and componential features of images have been documented to exert direct effects on viewers' decisions such as their inclination to click on visual-based online advertisement (Cheng et al., 2012; Luo et al., 2013) and participation toward crowdsourced services (Hou, 2019). In computer vision, the photographical properties of images are captured by global (e.g., overall brightness of images) and local (e.g., brightness of an object in the image) distribution of achromatic or chromatic pixels (Rasheed et al., 2015). Extracting these photographical properties of images hence relies on ***calculation-based methods*** that synthesize the statistical metrics of pixels (Cheng et al., 2012).

On the flip side, recognizing componential entities in an image demands intelligent processing (e.g., machine learning algorithms) that can learn and predict patterns of various objects based on massive labelled training image datasets (Druzhkov and Kustikova, 2016). We term such methods ***recognition-based.*** Unlike objective patterns of images, viewers' subjective perceptions of images, such as perceptions of image aesthetics (Jang & Lee, 2021; Sumathy et al., 2018; Khosla et al., 2015; Kong et al., 2016) and interestingness (Grabner et al., 2013b), tend to be subjected to viewers' cognitive assessment and interpretation of visual input (Pylyshyn, 1999). In line with Pylyshyn's (1999) work, we define such subjective properties of images as ***supraliminal features*** given the involvement of viewers' supraliminal cognition. Identical to the recognition of componential entities in images, extracting supraliminal visual features also relies on recognition-based methods to comprehend viewers' evaluation criteria when forming their subjective perceptions (Khosla et al., 2015).

Despite conscious interpretation of supraliminal visual features, viewers' processing of an image can manifest in a subtle manner without their conscious awareness (Pylyshyn, 1999). For example, memorability has been epitomized as a typical subliminal property of images that signifies viewers' unconscious meta-memory on the ease of visual encoding (Morris & Wickham, 2001). Such subliminal feelings of memorability, while cannot be explicitly reported by viewers (Hu & Borji, 2018), have been found to stimulate the latter's likes and preference (Schwarz, 2004). Due to viewers' inability to recognize the ***subliminal features*** of images, extracting such features usually demands ***simulation-based methods*** that equip deep learning algorithms with the capability to simulate attention-allocation and visual encoding processes inherently within the human brain (Jia et al., 2021; Gao et al., 2018). In this sense, we arrive at three categories of visual features (see Table 2 below) and elaborate on how core visual features can be extracted via *calculation-*, *recognition-*, and *simulation-based* methods in the next section.

---Insert Table 2 here---

4. Visual Feature Extraction Methods and Resources

4.1. Calculation-based extraction methods

Photographical properties of digital imagery can be directly calculated based on pixel matrixes encoded in diverse color spaces (Rasheed et al., 2015). Accordingly, we first expound two basic concepts in computer vision: conventional color spaces and histogram of images. Next, we epitomize the calculation methods of 24 most frequently used photographical features in relation to 11 core sensory features. Table 3 gives a brief breakdown of photographical features and their calculation methods accordingly.

---Insert Table 3 here---

A digital image is a multidimensional matrix of pixel values. We denote an image as I such that $|I|$ represents the size of the focal image (i.e., the number of pixels). Depending on its resolution, an image I is encoded as a $X * Y$ matrix in computer graphics where X and Y indicate the number of rows and columns of the focal image's pixel grid respectively. The intensity of each pixel at location (x, y) can be represented in distinct color spaces including but not limited to RGB (Red, Green, Blue), Grayscale, YUV, HSV (Hue, Saturation, Value/Brightness), and HSL (Hue, Saturation, Luminance or lightness) (Reinhard et al., 2001). RGB is a fundamental color space in which the value of each pixel encodes red (R), green (G) and blue (B) components of the color of the pixel (Reinhard et al., 2001). The RGB values of pixels can be directly converted into grayscale by employing colorimetry and photometry techniques. The transformed monochrome image possesses the same luminance as the original color image. YUV is a variation of RGB color space that compresses the R, G, B components of pixels in a more human-friendly manner (Ibraheem et al., 2012; Sural et al., 2002). Y measures the brightness of the color (i.e., the luminance) and U/V portrays the color itself (i.e., the chrominance). HSV and HSL, alternative representations of RGB, are cylindrical color spaces that take human vision into account in color encoding (Ibraheem et al., 2012; Sural et al., 2002) (see Figure 1).

---Insert Figure 1 here---

Besides conventional color spaces, the histogram of an image is another basic concept when extracting sensory features. The histogram of an image is indicative of the amount of pixels sharing the same color across diverse color spaces, and is usually shown as a graph or a plot representing the distribution of the pixel intensities of the focal image (Jeong, 2001). Given a select color space, a histogram plot divides the color space into a certain number of small intervals (i.e., ‘bins’) and portrays the number of pixels located in each bin accordingly. The histogram of an image acts as the basis for calculating various sensory features of the image in the likes of contrast and segment statistics (Ibraheem et al., 2012; Sural et al., 2002).

4.2. Recognition-based extraction methods and resources

Generally, componential and supraliminal visual features are extracted via recognition-based methods wherein machine learning serves as the predominant technique to recognize the patterns of images sharing similar properties (Druzhkov and Kustikova, 2016). Depending on the features of interest, recognition-based methods transform raw pixels of images into a targeted form of representation and leverages labelled dataset to train a machine learning model. The trained model is then deployed to label the visual features of target images (Druzhkov and Kustikova, 2016). We illustrate the basic procedures of r*ecognition-based extraction methods* in Figure 2, in which the *dataset*, *image representation*, and *model* constitute the key components of a pattern recognition system. Training datasets are employed to learn the parameters of machine learning models, whereas test datasets are employed to validate the trained model. An image representation encodes an image with either expert-engineered feature vector or raw pixel values according to machine learning models (e.g., conventional machine learning models or deep learning models). In this section, we elucidate four componential visual features (i.e., texts, human, ordinary object, and scene) and three supraliminal visual features (i.e., image aesthetics, interestingness, and image quality) to exemplify the resources for detecting/recognizing these features with machine/deep learning techniques from datasets, image representation, and models in Table 4.

---Insert Table 4 and Figure 2 here---

4.3. Simulation-based extraction methods and resources

While *recognition-based methods* are dependent on conventional machine learning or deep learning models to learn the patterns of componential and supraliminal visual features with abundant labelled data, s*imulation-based methods* train deep learning models to extract subliminal visual features by emulating human's visual processing. Particularly, two advanced techniques, namely attention mechanisms and adversarial training, have been devised to empower Deep Neural Networks (DNNs) with such simulating capability (Jia et al., 2021). Inspired by the human visual system whereby viewers' attention tends to be unconsciously attracted by more salient visual regions, attention modules have been implemented in Convolutional Neural Networks (CNNs) to identify attention-grabbing visual stimulus (i.e., visual saliency) and simulate how viewers memorize displayed images (i.e., image memorability) (Chen and Zhao, 2018). Generative Adversarial Network (GAN) is another technique that aids us in comprehending how deep learning algorithms emulate human thinking. This machine learning technique consists of two models: a generator and a discriminator. The former's objective is to generate data similar to the training data while the latter aims to identify if the data is real.

Subliminal visual features, such as image memorability and visual saliency, can be extracted by applying the above two techniques in deep neural networks. Following the feature extraction process depicted in Figure 2, Table 5 illuminates how the two subliminal visual features can be distilled by utilizing appropriate training datasets, image representation, and deep learning models.

---Insert Table 5 here---

5. Illustrative Examples

In this section, we present illustrative examples of how visual features can be extracted with python in accordance with *calculation*-, *recognition*-, and *simulation*-based methods mentioned above. Python is one of the most well-recognized programming languages for image processing and pattern recognition. *OpenCV*[1], an open-sourced computer vision library, can be unimpededly interfaced with python by utilizing *OpenCV-Python (henceforth cv2)* package. Besides, Python also offers manifold image analyses libraries such as *Python Image Library (PIL/PILLOW)* and *Matplotlib* (a plotting library). Table 6 enumerates common Python libraries for image analysis. The key Python packages employed in this tutorial consist of *OpenCV*, *NumPy*, and *Matplotlib*.

---Insert Table 6 here---

5.1. Calculation-based extraction methods

Importing Image: Importing an image in python is straightforward. Both *cv2* and *Matplotlib* provide a function named "*imread*" that reads images as matrices of pixels in RGB color space. Though both packages store RGB images as *NumPy* arrays, *cv2* returns 3-dimentional arrays in BGR order whereas *matplotlib* portrays pixels in original RGB system. *cv2* offers a function named *cvtColour*to convert imported images to RGB color space. We then implement a function named "*load_show_img*" that loads an image file into python with *cv2.imread* and displays the focal image utilizing *Matplotlib*. Taking a swimming pool picture as example, we illustrate the function mentioned above (see Appendixes Exemplary Code 1).

Understanding image properties: *OpenCV* by default reads an image as a three-dimensional matrix in which each number in the matrix represents the intensity of blue, green, and red colour of the corresponding pixel. The number varying from 0 to 255 indicates the shade of the colour where 0 denotes black and 255 implies white. Once the image is loaded, we can decipher the dimensions of the image data by getting its shape using "*img.shape*", including height, width, the number of channels (i.e., color spaces) and the pixel matrix. Additionally, we can examine the image size by calling *size* function (see Appendixes

[1] https://docs.opencv.org/master/d9/df8/tutorial_root.html

Exemplary Code 1). *OpenCV* has convenient functions for computing the statistics of images. Particularly, *cv2.mean* and *cv2.meanStdDev* can be employed to retrieve the mean and standard deviation of each channel's color intensity, which return a tuple of the statistics in RGB channels respectively (see Exemplary Code 2 in Appendices). *cv2.cvtColor(img, flag)* converts image data across different colour spaces, where *flag* indicates the type of conversion. For example, an RGB image can be converted into HSV space by setting *flag* as *cv2.COLOR_RGB2HSV*, whereas *cv2.COLOR_RGB2GRAY* reduces images into 2-dimentional grayscale format. (see Exemplary Code 2 in Appendices).

Extracting Image Histograms: Color-level, Gray-level, and Hue-level features are calculated based on histograms. Taking an grayscale image as example, we illustrate how histograms can be generated by the function named *cv2.calcHist*. *cv2.calcHist* reads parameters of image, channels, mask (optional, can be set as *None*), histSize (dimensions of bins) and ranges for each channel (typically 0-255). Computed histograms are typically normalized before proceeding to further calculation. *Matplotlib* can then be employed to visualize the extracted histogram. (see Exemplary Code 3 in Appendices)

Computing sensory features: We take Lightness/Brightness, Saturation, and Contrast as examples to illustrate how photographical properties can be computed from Python. They are derived from the statistics of values of all pixels in a single channel of the HSL/YUV space. After loading images with *cv2.imread*, the pixel matrix would be then transformed into YUV space for computing the mean and standard deviation of the "Y" values. We outline the steps and codes extracting the average and contrast of lightness (see Exemplary Code 4 in Appendices).

5.2. Recognition-based extraction methods

Recognition-based extraction methods are employed whenever conventional machine learning and deep learning methods are applied on labelled data to recognize or classify visual features. Alternatively, APIs, such as *Google Cloud Vision*[2], *Microsoft Face*[3], *IBM Watson Visual Recognition*[4], and *Amazon Recognition*[5] can be directly queried to perform image recognition task. In this section, we elucidate the procedure of recognition-based extraction method employing pre-trained deep learning models by taking extraction of facial emotion and image aesthetics as examples.

Facial emotion Recognition: In general, supervised machine learning methods are deployed in recognizing facial emotion in images. For supervised machine learning models, the labelled dataset is employed to fit the model in the object recognition task. In this sub-section, we rely on *Face Emotion Recognition (FER) database* (Zeiler and Fergus, 2014) and implement a pre-trained 6-layer CNN, *Mini Xceptionbuilt* (Arriaga et al., 2017), to predict facial emotions. By leveraging pre-trained ConvNet on ImageNet, the recognition procedure is introduced step-by-step and corresponding codes are showed in Exemplary Code 5 in the Appendices.

Aesthetics: Kong et al. (2016)'s regression network for rating image aesthetic is finetuned from Alex-Net with a large-scale image aesthetic dataset (i.e., AADB dataset which crowdsources human rating scores according to the overall aesthetical attributes of images[6]). Employing Kong et al. (2016)'s model which requires *Caffe* for implementation, scholars can compute a score predicting the aesthetics of interested images with subtle inferences on their color harmony, color vividness, brightness, repetition, balancing, object emphasis, and conformity to the rule of third. We take Kong et al. (2016)'s model as an example to predict image aesthetics (see Exemplary Code 6 in Appendixes).

---

[2] https://cloud.google.com/vision
[3] https://azure.microsoft.com/en-au/try/cognitive-services/?api=face-api
[4] https://www.ibm.com/watson/services/visual-recognition
[5] https://aws.amazon.com/cn/rekognition
[6] https://github.com/aimerykong/deepImageAestheticsAnalysis

5.3. Simulation-based extraction methods

Departing from recognition-based extraction methods which learn from labelled training datasets, simulation-based extraction methods predict visual features by simulating human thinking process and its training datasets do not have to be annotated manually. In this subsection, we take image memorability as example to exemplify how simulation-based extraction can be accomplished through python. MemNet is a CNN that learns to assess the memorability of images from Large-scale Memorability Dataset (LaMem). Khosla et al. (2015) have released an API and a pre-trained MemNet. We exemplify the procedure for predicting image memorability by reproducing their procedure (see Exemplary Code 7 in Appendixes). Khosla et al. (2015) have also written a demo which is accessible from the LaMem demo site[7]. The memorability score of an image can be established by uploading the image to the demo.

6. Conclusion

The importance of visual rhetoric cannot be understated in today's digital economy due to the indispensable role it plays in communicating the merits of products and/or services to online consumers. Extracting visual features from digital imageries can be regarded as the first step toward deciphering and optimizing the business value of visual rhetoric. Nevertheless, implementing appropriate methods of extracting core visual features from images can be challenging. In this tutorial, we attempt to proffer a comprehensive recipe for extracting three types of visual features (i.e., sensory, supraliminal, and subliminal) from images by consolidating relevant resources and techniques. This in turn paves the way for harnessing visual rhetoric in business value creation and management research.

To achieve the preceding objective, we synthesize extant literature to classify visual feature extraction methods into three categories, namely *calculation*-, *recognition*-, and

[7] http://memorability.csail.mit.edu/demo.html

simulation-based. We then prescribe actionable guidelines in conjunction with illustrative examples to illuminate how each method can be realized through python and deep learning libraries. For each feature extraction task, recommended datasets, ways of representing images, and learning models are showcased. Specifically, we stress that:

First, calculation-based methods can be deployed to extract photographical sensory features by performing calculations on pixel matrix of images encoded in diverse color spaces. Image processing libraries, such as *OpenCV*, supply built-in functions that can directly extract photographical properties of images.

Second, recognition-based methods can be deployed to recognize componential sensory (e.g., human, text, and scene) and supraliminal (e.g., aesthetics, interestingness, and quality) visual features of images. Conventionally, machine learning algorithms are widely applied models that learn the patterns of these visual features from sophisticated image representations designed by human experts. Nevertheless, with advances in deep learning and the availability of large-scale annotated image datasets (e.g., *ImageNet* and *Open Image*), DNNs (e.g., *AlexNet* and *InceptionV3*) have dominated conventional machine learning algorithms to extract componential and supraliminal visual features of images directly from their raw pixel values.

Third, simulation-based methods can be deployed to distil subliminal visual features (e.g., image memorability) that stimulate responses beneath viewers' conscious awareness. In comparison to recognition-based counterparts, simulation-based methods employ deep learning algorithms to emulate human's visual processing with advanced techniques like attention mechanism and adversarial training. Once the focal images are pre-processed into required formats, performing predictions on their subliminal visual features with pre-trained deep neural networks are relatively straightforward.

Disclosure statement

No potential conflict of interest was reported by the author(s)

Endnotes of Table 4 and Table 5

1. https://github.com/andreasveit/coco-text
2. http://www.iapr-tc11.org/dataset/MSRA-TD500/MSRA-TD500.zip
3. https://github.com/dengdan/ICDAR_2015_data_visualization
4. http://u-pat.org/ICDAR2017/index.php
5. https://github.com/eragonruan/text-detection-ctpn
6. https://github.com/whai362/PSENet
7. https://www.pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector
8. https://cloud.google.com/vision/docs/ocr
9. http://shuoyang1213.me/WIDERFACE
10. http://vis-www.cs.umass.edu/fddb
11. https://www.ics.uci.edu/~xzhu/face
12. http://www.cbsr.ia.ac.cn/faceevaluation
13. https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html
14. http://www.consortium.ri.cmu.edu/ckagree
15. http://www.dataonthemind.org/node/1617
16. https://ai.google/tools/datasets/google-facial-expression
17. https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data
18. https://paperswithcode.com/dataset/hico
19. https://research.google.com/ava/
20. https://storage.googleapis.com/openimages/web/factsfigures.html
21. http://vision.stanford.edu/Datasets/40actions.html
22. http://moments.csail.mit.edu/
23. http://cocodataset.org
24. http://host.robots.ox.ac.uk/pascal/VOC
25. https://arxiv.org/abs/1605.06409
26. https://github.com/matterport/Mask_RCNN
27. https://github.com/rbgirshick/fast-rcnn
28. http://datahacker.rs/od1-yolo-object-detection
29. http://places.csail.mit.edu/
30. https://vision.princeton.edu/projects/2010/SUN/
31. http://web.mit.edu/torralba/www/indoor.html
32. http://cvcl.mit.edu/Papers/IJCV01-Oliva-Torralba.pdf
33. http://scenenn.net/
34. https://github.com/aimerykong/deepImageAestheticsAnalysis
35. https://www.kaggle.com/hsankesara/flickr-image-dataset
36. http://live.ece.utexas.edu/research/ChallengeDB/index.html
37. http://vision.eng.shizuoka.ac.jp/mod/page/view.php?id=23
38. http://www.comlab.uniroma3.it/BattistiPapers/ACIVS2013_Battisti.pdf
39. http://database.mmsp-kn.de/koniq-10k-database.html
40. https://live.ece.utexas.edu/research/ChallengeDB/index.html
41. https://groups.csail.mit.edu/vision/SUN/hierarchy.html
42. http://memorability.csail.mit.edu/explore.html
43. http://figrim.mit.edu
44. http://cocosci.princeton.edu/jpeterson/objmem
45. https://github.com/tyshiwo/MemNet
46. http://saliency.mit.edu/results_cat2000.html
47. http://saliency.mit.edu/results_mit300.html
48. https://github.com/marcellacornia/sam
49. https://github.com/imatge-upc/salgan
50. https://github.com/mpatacchiola/deepgaze

ORCID

Xiaohui Liu https://orcid.org/0000-0003-4988-7469

Fei Liu https://orcid.org/0000-0001-7141-5105

Eric T.K. Lim https://orcid.org/0000-0003-3317-8406

Chee-Wee Tan https://orcid.org/0000-0003-4245-541X

References

Abousaleh, F. S., Cheng, W.-H., Yu, N.-H., & Tsao, Y. (2021). Multimodal deep learning framework for image popularity prediction on social media. *IEEE Transactions on Cognitive and Developmental Systems*, *13*(3), 679-692.

Alguliyev, R. M., Aliguliyev, R. M., & Abdullayeva, F. J. (2019). Privacy-preserving deep learning algorithm for big personal data analysis. *Journal of Industrial Information Integration*, *15*, 1-14.

Aloufi, S., Zhu, S., & Saddik, A. E. (2017). On the prediction of Flickr image popularity by analyzing heterogeneous social sensory data. *Sensors*, *17*(3), 631.

Anh, D. N. (2020). Interestingness improvement of face images by learning visual saliency. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *24*(5), 630-637.

Apostolidis, K., & Mezaris, V. (2019). Image aesthetics assessment using fully convolutional neural networks. *International Conference on Multimedia Modeling*, 11295(1), 361-373.

Arriaga, O., Valdenegrotoro, M., & Ploger, P. G. (2017). Real-time convolutional neural networks for emotion and gender classification. *arXiv: Computer Vision and Pattern Recognition*.

Azimi, J., Zhang, R., Zhou, Y., Navalpakkam, V., Mao, J., & Fern, X. (2012). Visual appearance of display ads and its effect on click through rate. *Proceedings of the 21st ACM international conference on Information and knowledge management*, 12(1), 495-504.

Bente, G., Baptist, O., & Leuschner, H. (2012). To buy or not to buy: Influence of seller photos and reputation on buyer trust and purchase behavior. *International Journal of Human-Computer Studies*, *70*(1), 1-13.

Berson, E., Duong, Q. N., & Demarty, C.-H. (2019). Collecting, analyzing and predicting socially-driven image interestingness. *European Signal Processing Conference (EUSIPCO)*, 1-5.

Bianco, S., Celona, L., Napoletano, P., & Schettini, R. (2016). Predicting image aesthetics with deep learning. *Advanced Concepts for Intelligent Vision Systems, Acivs 2016*, *10016*, 117-125.

Borges, A. F. S., Laurindo, F. J. B., Spínola, M. M., Gonçalves, R. F., & Mattos, C. A. (2021). The strategic use of artificial intelligence in the digital era: Systematic literature review and future research directions. *International Journal of Information Management*, *57*, 102225.

Bylinskii, Z., Isola, P., Bainbridge, C., Torralba, A., & Oliva, A. (2015). Intrinsic and extrinsic effects on image memorability. *Vision Research*, *116(Pt B)*, 165-178.

Chao, Y., Wang, Z., He, Y., Wang, J., & Deng, J. (2015). HICO: A benchmark for recognizing human-object interactions in images. *International Conference on Computer Vision*, 1017-1025.

Chen, C.-Y., & Cheng, K.-Y. (2005). A sharpness dependent filter for mesh smoothing. *Computer Aided Geometric Design*, *22*(5), 376-391.

Chen, S., & Zhao, Q. (2018). Boosted attention: Leveraging human attention for image captioning. *Proceedings of the European Conference on Computer Vision (ECCV),* 72-78.

Cheng, H., Zwol, R. v., Azimi, J., Manavoglu, E., Zhang, R., Zhou, Y., & Navalpakkam, V. (2012). Multimedia features for click prediction of new ads in display advertising. *Proceedings of ACM SIGKDD international conference on Knowledge discovery and data mining, 18(1), 777-785*.

Colliander, J., & Marder, B. (2018). 'Snap happy' brands: Increasing publicity effectiveness through a snapshot aesthetic when marketing a brand on Instagram. *Computers in Human Behavior*, *78*, 34-43.

Cyr, D., Head, M., & Ivanov, A. (2006). Design aesthetics leading to m-loyalty in mobile commerce. *Information & management*, *43*(8), 950-963.

Cyr, D., Head, M., & Larios, H. (2010). Colour appeal in website design within and across cultures: A multi-method evaluation. *International Journal of Human-Computer Studies*, *68*(1-2), 1-21.

Druzhkov, P., & Kustikova, V. (2016). A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognition and Image Analysis*, *26*(1), 9-15.

Deng, L., & Poole, M. S. (2010). Affect in web interfaces: A study of the impacts of web page visual complexity and order. *MIS Quarterly*, *34(4),* 711-730.

Ettis, S. A. (2017). Examining the relationships between online store atmospheric color, flow experience and consumer behavior. *Journal of Retailing and Consumer Services*, *37(1)*, 43-55.

Fan, S., Shen, Z., Jiang, M., Koenig, B. L., Xu, J., Kankanhalli, M. S., & Zhao, Q. (2018). Emotional attention: A study of image sentiment and visual attention. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7521-7531.

Fujiyoshi, H. (2007). Gradient-based feature extraction: Sift and hog. *Computer*, *107*(207), 211-224.

Gao, R., Xiong, B., & Grauman, K. (2018). Im2flow: Motion hallucination from static images for action recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 1-11.

Garain, A., Ray, B., Singh, P. K., Ahmadian, A., Senu, N., & Sarkar, R. (2021). GRA_Net: A deep learning model for classification of age and gender from facial images. *IEEE Access*, 9, 85672-85689.

Gelli, F., Uricchio, T., Bertini, M., Bimbo, A. D., & Chang, S. F. (2015). Image popularity prediction in social media using sentiment and context features. *ACM International Conference on Multimedia,* 1-4.

Guo, Z., Yang, G., Chen, J., & Sun, X. (2021). Fake face detection via adaptive manipulation traces extraction network. *Computer Vision and Image Understanding*, *204*, 103170.

Grabner, H., Nater, F., Druey, M., & Van Gool, L. (2013). Visual interestingness in image sequences. *ACM international conference on Multimedia*, 1017-1026.

Ha, Y., & Im, H. (2012). Role of web site design quality in satisfaction and word of mouth generation. *Journal of Service Management, 23(1),* 79-96.

Hao, W., & Zhang, Z. (2019). Spatiotemporal distilled dense-Connectivity network for video action recognition. *Pattern Recognition*, *92*, 13-24.

Hasler, D., & Susstrunk, S. (2003). Measuring colorfulness in natural images. *Human Vision and Electronic Imaging Conference*, *5007*, 87-95.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR) , 770-779*.

Herranz, L., Jiang, S., & Li, X. (2016). Scene recognition with CNNs: Objects, scales and dataset bias. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)* 571-579.

Hou, J.-R., Zhang, J., Zhang, K. . (2019). Can title images predict the emotions and the performance of crowdfunding projects? *Proceedings of Hawaii International Conference on System Sciences*, *52(1)*, 4439-4449.

Houtsma, A. J. (1997). Pitch and timbre: Definition, meaning and use. *Journal of New Music Research*, *26*(2), 104-115.

Hu, S., & Borji, A. (2018). Understanding perceptual and conceptual fluency at a large scale. *Proceedings of the European Conference on Computer Vision (ECCV),* 674-689.

Ibraheem, N. A., Hasan, M. M., Khan, R. Z., & Mishra, P. K. (2012). Understanding color models: a review. *ARPN Journal of science and technology*, *2*(3), 265-275.

Jang, H., & Lee, J.-S. (2021). Analysis of deep features for image aesthetic assessment. *IEEE Access*, *9*, 29850-29861.

Jeong, S. (2001). Histogram-based color image retrieval. *Infrared and Laser Engineering, 34,* 84-87.

Jia, X., Du, S., Guo, Y., Huang, Y., & Zhao, B. (2021). Multi-Attention ghost residual fusion network for image classification. *IEEE Access*, *9*, 81421-81431.

Jiang, M., Huang, S., Duan, J., & Zhao, Q. (2015). Salicon: Saliency in context. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 1072-1080.

Jiang, Z., Wang, W., Tan, B. C., & Yu, J. (2016). The determinants and impacts of aesthetics in users' first interaction with websites. *Journal of Management Information Systems*, *33*(1), 229-259.

Jin, X., Chi, J., Peng, S., Tian, Y., Ye, C., & Li, X. (2016). Deep image aesthetics classification using inception modules and fine-tuning connected layer. *International Conference on Wireless Communications & Signal Processing (WCSP)*, 1-6.

Jin, X., Wu, L., Zhao, G., Zhou, X., Zhang, X., & Li, X. (2018). IDEA: A new dataset for image aesthetic scoring. *Multimedia Tools and Applications*, *79(3)*, 1-15.

Lomov, I., Lyubimov, M., Makarov, I., & Zhukov, L. E. (2021). Fault detection in Tennessee Eastman process with temporal deep learning models. *Journal of Industrial Information Integration*, *23*, 100216.

Lu, Y. (2019). Artificial intelligence: A survey on evolution, models, applications and future trends. *Journal of Management Analytics*, *6*(1), 1-29.

Khosla, A., Raju, A. S., Torralba, A., & Oliva, A. (2015). Understanding and predicting image memorability at a large scale. *Proceedings of the IEEE International Conference on Computer Vision*, 2390-2398.

Kim, M. (2019). Digital product presentation, information processing, need for cognition and behavioral intent in digital commerce. *Journal of Retailing and Consumer Services*, *50(SEP.)*, 362-370.

Kong, S., Shen, X., Lin, Z., Mech, R., & Fowlkes, C. (2016). Photo aesthetics ranking network with attributes and content adaptation. *European Conference on Computer Vision*, 662-679.

Koo, D. M., & Ju, S. H. (2010). The interactional effects of atmospherics and perceptual curiosity on emotions and online shopping intention. *Computers in Human Behavior*, *26*(3), 377-388.

Lee, W., & Benbasat, I. (2004). Designing an electronic commerce interface: Attention and product memory as elicited by web design. *Electronic Commerce Research and Applications*, *2(3),* 240-253.

Leonardi, M., Celona, L., Napoletano, P., Bianco, S., Schettini, R., Manessi, F., & Rozza, A. (2019). Image memorability using diverse visual features and soft attention. *International Conference on Image Analysis and Processing*, 171-180.

Li, X., Wang, M., & Chen, Y. (2014). The impact of product photo on online consumer purchase intention: an image-processing enabled empirical study. *Proceedings of Pacific Asia Conference on Information Systems*, *325*.

Li, Y., Lim, T. K. E., Liu, Y., & Tan, C.-W. (2018). Tangiblizing your service: The role of visual cues in service e-tailing. *Proceedings of International Conference on Information Systems*, 39(8), 1-17.

Luo, W., Wang, X., & Tang, X. (2013). Content-based photo quality assessment. *IEEE Transactions on Multimedia*, *15*(8), 1930-1943.

Morris, P. E., & Wickham, L. H. (2001). Typicality and face recognition: A critical re-evaluation of the two factor theory. *The Quarterly Journal of Experimental Psychology Section A*, *54*(3), 863-877.

Nagpal, R., Nagpal, P., & Kaur, S. (2010). Hybrid technique for human face emotion detection. *International Journal of Advanced Computer Science and Applications*, *1*(6), 91-101.

Nguyen, D. T., Li, W., & Ogunbona, P. O. (2016). Human detection from images and videos: A survey. *Pattern Recognition*, *51*, 148-175.

Nixon, M., & Aguado, A. S. (2012). *Feature extraction and image processing for computer vision*. Academic Press.

Pylyshyn, Z. (1999). Is vision continuous with cognition?: The case for cognitive impenetrability of visual perception. *Behavioral and brain sciences*, *22*(3), 341-365.

Qu, Z., & Chen, Z. (2021). An intelligent vehicle image segmentation and quality assessment model. *Future Generation Computer Systems*, *117*, 426-432.

Rasheed, M. B., Javaid, N., Alghamdi, T. A., Mukhtar, S., Qasim, U., Khan, Z. A., & Raja, M. H. B. (2015). Evaluation of human activity recognition and fall detection using android phone. *IEEE 29th International Conference on Advanced Information Networking and Applications,* 163-170.

Reinhard, E., Adhikhmin, M., Gooch, B., & Shirley, P. (2001). Color transfer between images. *IEEE Computer Graphics and Applications*, *21*(5), 34-41.

Ren, J., Shen, X., Lin, Z., Mech, R., & Foran, D. J. (2017). Personalized image aesthetics. *Proceedings of the IEEE International Conference on Computer Vision*, 638-647.

Sasibhooshan, R., Kumaraswamy, S., & Sasidharan, S. (2021). WavNet - Visual saliency detection using Discrete Wavelet Convolutional Neural Network. *Journal of Visual Communication and Image Representation*, *79*, 103236.

Santra, B., Shaw, A., & Mukherjee, D. P. (2022). Part-based annotation-free fine-grained classification of images of retail products. *Pattern Recognition*, *121*, 108257.

Saraee, E., Jalal, M., & Betke, M. (2020). Visual complexity analysis using deep intermediate-layer features. *Computer Vision and Image Understanding*, *195*, 102949.

Schwarz, N. (2004). Metacognitive experiences in consumer judgment and decision making. *Journal of Consumer Psychology*, *14*(4), 332-348.

So, H., & Oh, W. (2018). Picture perfect: An image mining of advertising content and its effects on social targeting. *Proceedings of the 39th International Conference on Information System*.

Sohn, S. (2017). Consumer processing of mobile online stores: Sources and effects of processing fluency. *Journal of Retailing and Consumer Services*, *36*, 137-147.

Solomon, C., & Breckon, T. (2011). *Fundamentals of digital image processing: A practical approach with examples in Matlab*. John Wiley and Sons.

Sultani, W., & Shah, M. (2021). Human action recognition in drone videos using a few aerial training examples. *Computer Vision and Image Understanding*, *206*, 103186.

Sumathy, P., Shanmugavadivu, P., & Vadivel, A. (2018). *Image retrieval and analysis using text and fuzzy shape features: Emerging research and opportunities*. IGI Gobal.

Sunaina, S., Kaur, R., & Sharma, D. V. (2021). A review of vision-based techniques applied to detecting human-object interactions in still Images. *Journal of Computing Science and Engineering*, *15*(1), 18-33.

Sural, S., Qian, G., & Pramanik, S. (2002). Segmentation and histogram generation using the HSV color space for image retrieval. *Proceedings. International Conference on Image Processing,* 2.

Takimoto, H., Omori, F., & Kanagawa, A. (2021). Image aesthetics assessment based on multi-stream CNN architecture and saliency features. *Applied Artificial Intelligence*, *35*(1), 25-40.

Tan, W., Huang, L., Kataev, M. Y., Sun, Y., Zhao, L., Zhu, H., Guo, K., & Xie, N. (2021). Method towards reconstructing collaborative business processes with cloud services using evolutionary deep Q-learning. *Journal of Industrial Information Integration*, *21*, 100189.

Talebi, H., & Milanfar, P. (2018). Nima: Neural image assessment. *IEEE transactions on image processing*, *27*(8), 3998-4011.

Wang, L., Guo, S., Huang, W., Xiong, Y., & Qiao, Y. (2017). Knowledge guided disambiguation for large-scale scene classification with multi-resolution CNNs. *IEEE transactions on image processing*, *26*(4), 2055-2068.

Wang, L., & Mueller, K. (2008). Harmonic colormaps for volume visualization. *Proceedings of the Eurographics / IEEE VGTC Workshop on Volume Graphics*, 33-39.

Wang, M., & Li, X. (2017). Effects of the aesthetic design of icons on app downloads: evidence from an android market. *Electronic Commerce Research*, *17*(1), 83-102.

Wang, Z., Chang, S., Dolcos, F., Beck, D., Liu, D., & Huang, T. S. (2016). Brain-inspired deep networks for image aesthetics assessment. *Michigan Law Review, 52(1),*123-128.

Wu, S. Z., Kan, M. N., He, Z. L., Shan, S. G., & Chen, X. L. (2017). Funnel-structured cascade for multi-view face detection with alignment awareness. *Neurocomputing*, *221*, 138-145.

Yang, S., Luo, P., Loy, C. C., & Tang, X. (2017). Faceness-net: Face detection through deep facial part responses. *IEEE transactions on pattern analysis and machine intelligence*, *40*(8), 1845-1859.

Ye, Q., & Doermann, D. (2014). Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence*, *37*(7), 1480-1500.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *European Conference on Computer Vision*, 8689.

Zeng, H., Zhang, L., & Bovik, A. C. (2017). A probabilistic quality representation approach to deep blind image quality prediction. *Computer Vision and Pattern Recognition*.

Zhan, S., Tao, Q., & Li, X. (2016). Face detection using representation learning. *Neurocomputing*, *187*(187), 19-26.

Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. *Journal of Industrial Information Integration*, *23*, 100224.

Zhang, S., Lee, D., Singh, P. V., & Srinivasan, K. (2016). How much is an image worth? An empirical analysis of property's image aesthetic quality on demand at Airbnb, *Proceedings of the 37th International Conference on Information System*.

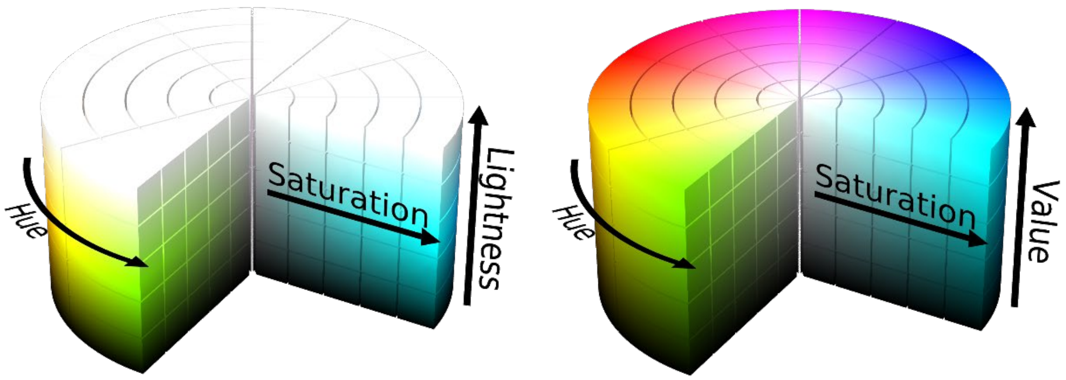
Zhang, S., Lee, D., Singh, P. V., & Srinivasan, K. (2017). How much is an image worth? Airbnb property demand estimation leveraging large scale image analytics. *Social Science Electronic Publishing,* 1-75.

Zhou, H., Yuan, Y., & Shi, C. (2009). Object tracking using SIFT features and mean shift. *Computer Vision and Image Understanding*, *113*(3), 345-352.

Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). EAST: An efficient and accurate scene text detector. *Computer Vision and Pattern Recognition*, 2642-2651.

Figures

Figure 1. HSL and HSV Cylinder



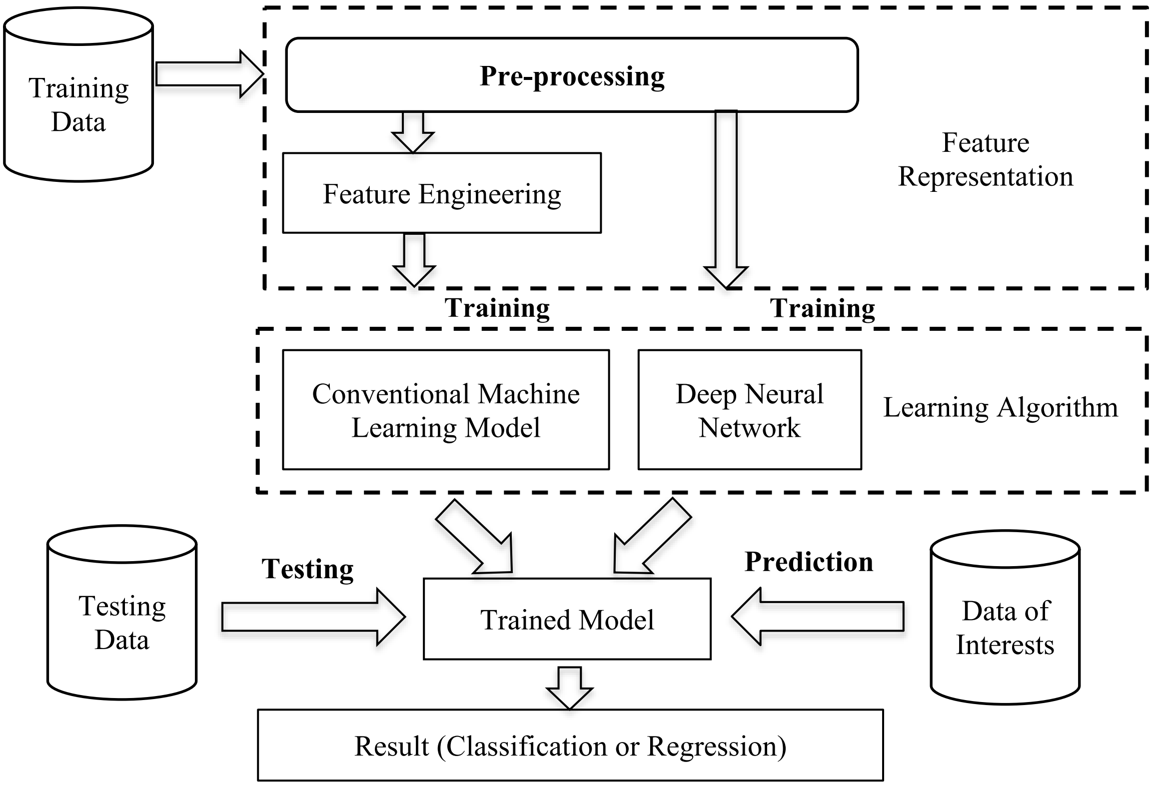
HSL Cylinder	HSV Cylinder

Notes:

- Colors are distributed in an angular dimension that starts from red primary (at 0°) and then rotate to green primary (at 120°), blue primary (at 240°), before wrapping back to red at 360°
- Hue of an image is measured by the rotation of a color point around the center
- Saturation resembles the purity of colors (i.e., the shades of hue) that can be captured by the proximity of a color point to the center
- Vertical axis of the cylinder encompasses the value/brightness of the achromatic colors; the distinction between HSV and HSL is that the lightness of a pure color (L) captures the lightness of medium gray in HSL color space (Ibraheem et al., 2012)

Source: Kon Karampelas. (2020, November 5). Lifting the Purple Haze: Color Theory Part I - Color Spaces and Perception. https://rgutzen.github.io/2020-11-05-color_spaces

Figure 2. Flowchart of Conventional Machine Learning and Deep Learning Algorithms



Tables

Table 1. Summary of Empirical Business Management Research Associated with Visual Rhetoric

Reference	Research Objective	Visual Stimuli	Method	Opportunities
Cyr et al. (2010)	▪ Explore how color scheme in website design influence users' color appeal, and further investigate its impact on users' e-loyalty.	▪ Color scheme	▪ Experiment ▪ Survey ▪ Interview	Adopt machine learning algorithms to extract such sensory feature.
Ettis (2017)	▪ Explore the role of online atmospheric color on consumers' approach behavior in the likes of purchase intention and revisit intention in online store settings.	▪ Online Atmospheric color	▪ Online Experiment	
Ha and Im (2012)	▪ Explore how background color, fonts and icon influence users' likelihood to recommend the focal website.	▪ Color of Website's background	▪ Survey	
Kim (2019)	▪ Explore how the visual and verbal stimuli of products affect customers' behavioral intention in e-commerce settings.	▪ Product picture size	▪ Online Experiment ▪ Survey	
Koo and Ju (2010)	▪ Explore how atmospheric cues embedded in the online store affect consumers' continuous intention to use.	▪ Graphics, colors, links and menu of the online store	▪ Survey	
Lee and Benbasat (2004)	▪ Explore the effect of product photo characteristics on consumers' information-searching behavior in e-commerce settings.	▪ Characteristics of product photo including image size, image clarity and image motion	▪ Experiment	
Zhang et al. (2017)	▪ Explore the economic impact of visual cues embedded in hotel picture on the demand for Airbnb property.	▪ Composition; Color; Figure-grounded relationship	▪ Secondary Data Analyses	
Li et al. (2018)	▪ Explore the effect human facial cues and verbal anchoring embedded in portal images in increasing sales.	▪ Human facial cue and textual cue	▪ Secondary Data Analyses	
Bente et al. (2012)	▪ Explore how seller reputation may interact with seller photo trustworthiness rating to influence consumers' purchase decision.	▪ Seller's photo trustworthiness	▪ Experiment ▪ Survey	Adopt deep learning algorithms to extract such subliminal features.
Colliander and Marder (2018)	▪ Examine the effect of snapshot aesthetics/traditional studio aesthetics of products photo on consumers evaluation of a brand and word of mouth.	▪ Photo with snapshot aesthetics	▪ Experiment ▪ Survey	
Dianne Cyr et al. (2006)	▪ Explore the impact of visual design aesthetics on perceived usefulness, ease of use and enjoyment, and further investigate its impact on users' e-loyalty.	▪ Visual design aesthetics	▪ Survey	
Jiang et al. (2016)	▪ Explore factors affecting website aesthetics and their impact on consumers' impression on the platform as well as cooperate image.	▪ Perceived beauty of website	▪ Survey	
Sohn (2017)	▪ Explore the role of processing fluency perceptions (i.e., perceived visual complexity and perceived visual congruence) in consumer experiences in mobile application market.	▪ Perceived visual complexity ▪ Perceived visual congruence	▪ Survey	Adopt deep learning algorithms to extract such supraliminal feature.
Deng and Poole (2010)	▪ Explore the role of visual complexity and order as central factors in the design of website, which further induces users' positive reactions.	▪ Visual complexity	▪ Experiment	

Table 2. Summary of Visual Features and the Corresponding Extraction Methods

Visual Feature	Definition	Exemplary Features	Extraction Method(s)
Sensory Feature	Photographical and componential properties of images that can be directly identified by human optical systems without the intertwinement of cognitive interpretation, namely photographical and componential features	▪ Color ▪ Brightness ▪ Objects	Calculation- and recognition-based methods
Supraliminal Feature	Subjective properties of images that are recognized by human's conscious assessment and interpretation	▪ Aesthetics ▪ Quality ▪ Interestingness	Recognition-based method
Subliminal Feature	Subliminal properties of images that cannot be recognized by humans' conscious awareness	▪ Memorability	Simulation-based method

Table 3. Summary of Photographical Features and the Corresponding Calculation Methods

Photographical Feature	Description	Calculation
Gray-Level Feature	▪ The intensity properties of an image.	
▪ Grayscale Contrast (f_1)	▪ Discrepancies in the intensity (e.g., amount of light) of an image	▪ Width of the middle 95% of the histogram.
▪ Grayscale Simplicity (f_2)	▪ Spread of luminance	▪ Number of dominant bins of the greyscale histogram of an image. Conventionally, a monochrome image can be compressed as a grayscale histogram that comprises 256 bins: $h_1, h_2, \cdots, h_i, \cdots, h_{255}$, where h_i denotes the number of pixels in ith bin. Accordingly, $f_2 = \sum_{j=0}^{255} I(h_j > \theta \max(h_i))$, where $I(h_j > \theta max_i(h_i))$ equal to 1 if $h_j > \theta max_i(h_i)$ and θ is a threshold parameter.
▪ Grayscale Dispersion (f_3)	▪ Level of variation of the amount of light in an image	▪ Standard deviation of the grayscale values of all the pixels in the image.
Color Simplicity and Dominancy	▪ Viewers' most intuitive feelings of the color distribution of an image that can be extracted from color histogram.	▪ In line with Azimi et al. (2012), we quantify each RGB channel into 8 values such that a histogram of 512 bins (8*8*8) can be represented as: $h_1, h_2, \cdots, h_i, \cdots, h_{512}$.
▪ Color Simplicity (RGB) (f_4)	▪ Spread of the composition of colors	▪ $f_4 = \sum_{j=1}^{512} I(h_j > \alpha \max_i(h_i))$, where h_j is the number of pixels in ith bin, $I(h_j > \alpha max_i(h_i))$ equal to 1 if $h_j > max_i(h_i)$ and α is a threshold value
▪ Color Dominancy (RGB) (f_5)	▪ Superior color of an image in RGB color space	▪ $f_5 = max_i(h_i)/\lvert I \rvert$, where $\lvert I \rvert$ represent the size of the image
Hue-Level Feature	▪ The degree to which a stimulus can be recognized as similar to elementary color stimuli such as red, green, blue and yellow (Cheng, et al., 2012)	▪ Hue-level features are estimated from the hue histogram of images portrayed in HSL or HSV color space. ▪ Quantify hues by eliminating the pixels of saturation and value that are less than 0.2 (Li and Chen, 2009).
▪ Hue Simplicity (f_6)	▪ Spread of hues of an image in HSV or HSL color spaces	▪ $f_6 = \sum_{j=1}^{20} I(h_j > \beta \max_i(h_i))$ indicates the number of dominate hues in an image, where $I(h_j > \beta max_i(h_i))$ equal to 1 if $h_j > max_i(h_i)$ and β is a threshold value
▪ Hue Contrast (f_7)	▪ Disparities in the dominant hue values of an image	▪ $f_7 = max_{i,j} \lVert \lvert h_i \rvert - \lvert h_j \rvert \rVert$, where $\lVert \cdot \rVert$ represents the length of the vector
▪ Hue Dispersion (f_8)	▪ Level of variation of hues of an image	▪ f_8 is measured by the standard deviation of the hue values of all the pixels in the image

Table 3. Continued

Photographical Feature	Description	Calculation
Color Harmony	▪ Coordination of color combination that generates a sense of aesthetics.	▪ Color harmony can be gauged by eight color harmonic distribution templets (Wang and Mueller, 2008). These are generated from the hue value of HSV system, which is defined as $D = \{d^1, d^2, \cdots, d^8\}$.
▪ First Order Color Harmony (f_9)	▪ Average deviation of the hue values of the focal image from the best fitted color harmonic model	▪ $\phi(d_\alpha^i, x)$ represents the hue value of the closest point in ith distribution to x (represent any arbitrary pixel in the image) after α degree rotation. We define $f_9 = \gamma(I, d^*) = \text{argmin}\frac{1}{\|I\|}\sum_{x\epsilon I}\lVert hue(x) - \phi(d^*, x)\rVert \cdot sat(x)$, which means the least distance between the hue distribution of image I and the best fitting model $d^* \in D$, where $d^* = argmin_{d^i}\left(argmin_\alpha \frac{1}{\|I\|}\sum_{x\epsilon I}\lVert hue(x) - \phi(d_\alpha^i, x)\rVert \cdot sat(x)\right)$, $hue(x)$ and $sat(x)$ indicate the hue and saturation at pixel x, and $\lVert\cdot\rVert$ denotes the arc-length distance.
▪ Second Order Color Harmony (f_{10})	▪ Average deviations of the hue values of the focal image from the best two fitted models (the best and second best fitted color harmonic models)	▪ $f_{10} = \gamma(I, d^*) = \text{argmin}\frac{1}{\|I\|}\sum_{x\in I}\lVert hue(x) - \phi(d^*, x)\rVert \cdot sat(x)$, which means the least distance between the hue distribution of image I and the best two fitting model $d^* \in D$
Lightness/Brightness	▪ Luminance or brightness of the color distribution of an image (Cheng et al., 2012)	▪ Derived from "L" in HSL or "Y" in YUV color space.
▪ Average Lightness (f_{11})	▪ Average of the tonal luminance and lightness value of an image	▪ f_{11} measured by the mean of the "L" value of all pixels of an image in HSL color space or the average of the "Y" values of all the pixels in YUV color space
▪ Dispersion of Lightness (f_{12})	▪ Variation of the amount of luminance and lightness value of an image	▪ f_{12} can be computed as the standard deviation of the "L" value of all pixels in HSL color space or the "Y" values of all the pixels in YUV system
Saturation	▪ Vividness of an image (Cheng et al., 2012).	▪ Derived from the "S" (i.e., saturation) dimension of the image pixels in HSV or HSL color spaces
▪ Average Saturation (f_{13})	▪ Average of the saturation value of an image	▪ f_{13} measured by the mean of the "S" value of the image pixels in HSV or HSL color spaces
▪ Dispersion of Saturation (f_{14})	▪ Variation of the amount of saturation value of an image	▪ f_{14} measured by the standard deviation of the "S" value of the image pixels in HSV or HSL color spaces
Contrast (f_{15})	▪ Visual distinguishableness of the elements in an image	▪ $f_{15} = std(\sum L(x, y))$, where the luminance $L(x, y)$ denotes the "L" value at x rows and y columns at the grid of image pixels

Table 3. Continued

Photographical Feature	Description	Calculation
▪ ***Sharpness*** (f_{16})	▪ Level of clarity of an image	▪ Laplacian function has been widely employed to calculate image sharpness based on the luminance of all pixels in an image (C.-Y. Chen & Cheng, 2005). Specifically, $f_{16} = \sum_{x,y} \frac{LP(x,y)}{u_{xy}}$, where $LP(x,y) = \frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}$ is a function of its Laplacian and u_{xy} denotes the average luminance around pixel (x,y).
▪ ***Colorfulness*** (f_{17})	▪ Visual chromatism of the color of the image	▪ According to Hasler and Susstrunk (2003), the colorfulness of images in RGB color space can be computed by a linear combination of $\{\sigma_{R-G}, \sigma_{\frac{R+G}{2}}, \mu_{R-G}, \mu_{\frac{R+G}{2}}\}$, where $R-G$ and $\frac{R+G}{2} - B$ are two newly built simple opponent colour space, μ and σ mean the average and standard deviation value of the pixel along the new axis (that is, $R-G$ or $\frac{R+G}{2} - B$), respectively. In this regard, $f_{17} = \sqrt{\sigma^2_{(R-G)} + \sigma^2_{(\frac{R+G}{2}-B)}} + 0.3 * \sqrt{\mu^2_{(R-G)} + \mu^2_{(\frac{R+G}{2}-B)}}$.
Texture	▪ Information about the spatial arrangement of color or their intensities in an image (Mohanaiah et al., 2013).	▪ Gray-Level Co-occurrence Matrix (GLCM) captures the main characteristics of image texture (Mohanaiah et al., 2013). ▪ A M × M sized GLCM stores histograms of co-occurring grayscale values at a given offset over an image, where M denotes the number of dimensions of the grayscale of the image.
▪ Texture Entropy (f_{18})	▪ Complexity of an image texture	▪ f_{18} measures the information entropy conveyed in the GLCM matrix of an image, which can be calculated by the equation $f_{18} = -\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} P[i,j] log_2 P[i,j]$, where $P[i,j]$ is the normalized GLCM matrix and denotes the joint probability distribution of the grayscale values of two displacement pixels i and j
▪ Texture Contrast (f_{19})	▪ Sharpness and the depth of texture grooves of images	▪ $f_{19} = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1}(i-j)^2 P[i,j]$
▪ Texture Homogeneity (f_{20})	▪ Level of intra-regional changes in image texture	▪ $f_{20} = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} P[i,j]/(1+\|i-j\|)$
▪ Texture Energy (f_{21})	▪ Crudeness of image textures	▪ $f_{21} = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} P[i,j]^2$
Segmentation Statistics	▪ Image segmentation algorithms partition a digital image into groups of pixels based on predetermined criteria that are enacted to distinguish primary components of images from each other (Ronneberger et al., 2015).	
▪ Segment Number (f_{22})	▪ Degree of segmentation in the image	▪ Assume m salient segments (a segment is dropped if it is smaller than 5% of the image), $S = \{S_1, \cdots, S_i, \cdots, S_m\}$, are identified, where S_i represents the set of pixels in segment i
▪ Segment Contrast (f_{23})	▪ Difference between the size of the largest and smallest segments	▪ $f_{23} = \max_i(\|S_i\|) - \min_i(\|S_i\|)$
▪ Segment Dominance (f_{24})	▪ Super segment in the image	▪ $f_{24} = \frac{max_i(\|S_i\|)}{\|I\|}$, where $\|I\|$ denotes the size of the image

Table 4. Summary of Exemplary Resources for Visual Features' Recognition-Based Extraction Methods

Visual Features	Task	Resources of Recognition-Based Extraction Methods		
		Dataset for Training Models	Image Representation as Input to Learning Algorithm	Models for Recognition Task
Text	▪ Optical character Recognition task. ▪ To locate and recognize digital or handwritten texts embedded in images.	▪ COCO-Text[1] ▪ MSRA-TD500[2] ▪ ICDAR2015[3] ▪ ICDAR2017[4]	▪ Scale Invariant Feature Transform (SIFT) (Zhou et al., 2009) ▪ Histograms of Oriented Gradients (HOG) (Fujiyoshi, 2007) ▪ Color feature, contour of a specific pattern, and edge feature(Ye and Doermann, 2014)	▪ Connectionist Text Proposal Network (CTPN)[5] ▪ Progressive Scale Expansion Network (PSE-Net)[6] ▪ API: EAST[7].; OCR API in Google's Could Vision[8]
Human ▪ **Presence of Human Face**	▪ Human detection task ▪ Identify human via facial cues embedded in images.	▪ WIDER FACE [9] ▪ FDDB[10] ▪ AFW dataset[11] ▪ MALF dataset[12]	▪ Conventional machine learning classifier: HOG and SIFT (Fujiyoshi, 2007) ▪ Convolutional Neural Network (CNN) feature extractor (Zhan et al., 2016)	▪ GRA_Net (Garain et al., 2021) ▪ Conventional machine leaning method: SeetaFace Detection (Wu et al., 2017); Haar Classifier[13]; ▪ Deep learning method: Faceness-Net (Yang et al. (2017)
▪ **Facial Emotion**	▪ Emotion recognition task ▪ Take bounded face as input and predicts probabilities of emotion categories	▪ Cohn-Kanade (CK) dataset[14] ▪ Extended Cohn-Kanade (CK+) database[15] ▪ Google facial expression comparison dataset[16] ▪ Facial Emotion Recognition (FER) dataset[17]	▪ Raw pixel values	▪ Deep convolutional neural network (DCNN) advanced by Barsoum et.al (2016) ▪ Hybrid neural networks proposed by Jain et.al (2018)
▪ **Human Action**	▪ Comprehension of what the person is doing in the image (Sultani & Shah, 2021)	▪ Humans Interacting with Common Object (HICO)[18] dataset ▪ AVA-Kinetics Localized Human Actions Video Dataset[19] ▪ Open Images V5[20] ▪ Stanford 40 Action[21] ▪ Moments in Time dataset[22]	▪ Raw pixel values	▪ Human-Object CNN (HOCNN) (Chao et al., 2015) ▪ Spatiotemporal Distilled Dense-Connectivity Network (SDDN) (Hao and Zhang, 2019)
Ordinary Object	▪ Object recognition task ▪ most common entities (e.g., animals, food and vehicles) embedded in images.	▪ ImageNet ▪ MS COCO dataset[23] ▪ ASCAL VOC dataset[24]	▪ Raw pixel values and Image representations produced by DNNs pre-trained on ImageNet	▪ R-CNN[25]and Mask R-CNN[26] ▪ Fast-CNN[27] ▪ Faster-CNN[27] ▪ YOLO[28]

Table 4. Continued

Visual Features	Task	Resources of Recognition-Based Extraction Methods		
		Dataset for Training Models	Visual Features	Task
Scene Attributes	▪ Definition of a context for object recognition	▪ Places[29] dataset ▪ SUN database[30] ▪ MIT Indoor67 database[31], Urban and Natural Scene Categories[32] and Scene-NN[33]	▪ Raw pixel values and Image representations produced by DNNs pre-trained on ImageNet	▪ Place-CNN (Herranz et al. (2016) ▪ MRCNN (Limin Wang et al., 2017)
Image Aesthetics	▪ Image aesthetics assessment task ▪ To predict the aesthetic scores of images with a high consistency to human's general sense of beauty. (Takimoto et al., 2021; Apostolidis & Mezaris, 2019)	▪ Aesthetics with Attributes Database (AADB)[34] ▪ AVA (Murray et al., 2012)	▪ Raw image pixels and Image representations produced by DNNs pre-trained on ImageNet	▪ Multi-stream CNN (Takimoto et al., 2021) ▪ DeepIA (Bianco et al. (2016) ▪ Brain-Inspired Deep Networks (BDN) (Z. Wang et al., 2016) ▪ ANE (Semantic-Aware Hybrid Network) ▪ ILGNet (Jin et al., 2018) ▪ PAM (Ren et al., 2017) ▪ NIMA (Talebi and Milanfar, 2018)
Image Interestingness	▪ Image interestingness assessment task ▪ To assess the extent to which a given image can be perceived as interesting by viewers. ▪ Viewer's perceived interestingness of a given image has been conceived as the primary driver of their attention (Amengual et al., 2015)	▪ Flickr Dataset (FD)[35] ▪ LaFin dataset (Berson et al., 2019) ▪ Webcam Dataset(WD) (Grabner et al., 2013a)	▪ Expert engineered features in relation to the componential and contextual factors: e.g., brightness and saturation, HSV values, RGB values, SIFT histograms, colorfulness, contrast and edge distributions ▪ Raw image pixels and Image representations produced by DNNs pre-trained on ImageNet	▪ CNNs via a combination of pre-trained image features and captioning-based features. (Berson et al., 2019) ▪ CNNs using visual saliency of face images as learning features (Anh, 2020)
Image Popularity	▪ Interestingness concept in social media context. ▪ Different image views and downloads in various social networking and photo sharing platform have promoted researchers study popularity attributes of the image	▪ Flickr Dataset (FD)	▪ Features includes both social cues, as well as image-specific cues. ○ Feature representation related image content; ○ User feature and context feature representation for popularity prediction.	▪ Support Vector Regression (SVR) (Gelli et al., 2015) ▪ Off-the-shelf Support Vector Machine (SVM) method provided by Aloufi et al. (2017) ▪ Visual-social CNN (VSCNN) (Abousaleh et al., 2021)
Image Quality	▪ Image Quality Assessment algorithm (IQA) (or blind image quality)	▪ LIVE IQA database[36], ▪ CSIQ database[37], ▪ TID2013[38]	▪ Image-quality assessment always relay on the structure and content of the images for personal evaluation	▪ Combination of Alex-Net and Res-Net pre-trained on the ImageNet (Kim et al. 2017)

	▪ make quality predictions that are in agreement with subjective opinions of human observers (Zeng et al., 2017)	▪ KonIq-10k[39] ▪ Live Challenge[40].	▪ Structure features: raw pixel data is usually transformed to global or local features of the image (e.g., HOG, SIFT or color feature) ▪ Content features: CNN feature extractors, which are employed to process raw images into CNN-adapted representations	▪ FR-IQA CNN model based on feature similarity index (Qu & Chen, 2021)

Table 5. Summary of Exemplary Resources for Visual Features' Simulation-Based Extraction Method

Visual Features	Description	Resources of Simulation-Based Extraction Methods		
		Dataset	Image Representation	Model
Memorability	▪ Image memorability task. ▪ Image memorability is an intrinsic property of images that determines the extent to which a depicted image can be remembered by viewer. ▪ To mimic human's attention allocation process by which visual input is encoded and rememorized.	▪ SUN[41] ▪ LaMem (short for Large-scale Memorability Dataset)[42] ▪ FIGIM Dataset[43] (Bylinskii et al., 2015) ▪ PASCAL-S[44]	▪ Raw image pixels and image representations produced by DNNs pre-trained on ImageNet	▪ MemNet[45] ▪ DNNs using diverse visual features and soft attention (Leonardi et al., 2019)
Visual Saliency	▪ Visual saliency prediction task. ▪ To predict where human gazes will be attracted when viewing a given image. ▪ Attention mechanism is important. ▪ Attention map serves as the output. (a scalar matrix that scores the number of fixations viewers are likely to allocate in a given image region)	▪ SALICON(M. Jiang et al.,2015) ▪ CAT2000[46] ▪ EMOd (Fan et al., 2018) ▪ MIT300[47]	▪ Raw image pixels and image representations produced by DNNs pre-trained on ImageNet	▪ Saliency Attentive Model (SAM)[48] ▪ SalGAN[49] ▪ DeepGaz[50] ▪ WavNet (Sasibhooshan et al., 2021)

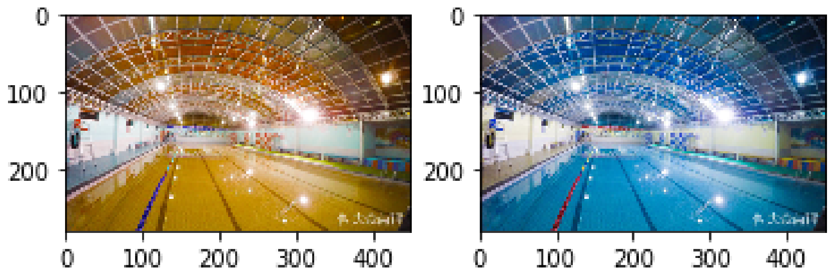
Table 6. Common Image Processing Libraries in Python

Name	Description	Source
Scikit-image	Fairly simple and straightforward open-source Python package that works with NumPy arrays.	https://scikit-image.org/docs/stable/user_guide.html
Numpy	Core library in Python programming and provide support for array. Image is essentially a standard NumPy array containing pixels of data points. Hence, you can modify the pixel values of an image using NumPy.	http://www.numpy.org
SciPy	Another Python's core scientific modules (like NumPy) and can be used for basic image manipulation and processing tasks.	https://docs.scipy.org/doc/scipy/reference/tutorial/ndimage.html#correlation-and-convolution
PIL/Pillow	Python Image Library, a free library for the python programming language but its development has stagnated. Pillow is an actively developed fork of PIL that is easier to install and support Python 3. This library contains basic image processing functionality, including point operations, filtering with a set of built-in convolution kernels, and color-space conversions.	https://pillow.readthedocs.io/en/3.1.x/index.html
OpenCV-Python	Open-source Computer Vision Library (OpenCV-Python), is the Python API for OpenCV and one of the most widely used library for computer vision application.	https://github.com/abidrahmank/OpenCV2-Python-Tutorials
SimpleCV	One open-source framework for building computer vision application; its learning curve is smaller than OpenCV's because of no needing to know about bit depths and color spaces.	http://examples.simplecv.org/en/latest
Mahotas	Contains traditional image processing function as well as more modern computer vision function for feature computation	https://mahotas.readthedocs.io/en/latest/install.html

Appendix: Codes of Illustrative Examples

Exemplified Code 1 and its output (Importing and Displaying an Image)

```
import cv2
from matplotlib import pyplot as plt
def load_show_img(img_file):
    img = cv2.imread(img_file)
    img_RGB = cv2.cvtColour(img, cv2.COLOUR_BGR2RGB)
    plt.subplot(1,2,1)
    plt.imshow(img)
    plt.subplot(1,2,2)
    plt.imshow(img_RGB)
    plt.show()
img_file = '/Users/Lillian/Desktop/img.png'
load_show_img(img_file)
print (img.shape, img.size)
(296,474,3), 420912  # Output
```

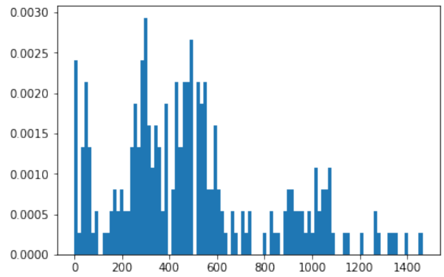


Exemplified Code 2 (Understanding image properties)

```
def statistics_img(img_file):
    img = cv2.imread(img_file)
    means = cv2.mean(img)
    (means, stds) = cv2.meanStdDev(img)
    return print("Means",means), print("Std",stds)
statistics_img(img_file)
Means: [[155.22125397], [124.0479127], [78.11589683]]
Std: [[50.87520102], [52.61902816], [72.16649679]]
# How to convert an RGB Represented Image into HSV/Graylevel Space
def cvt_RGB2HSV(img_file):
    img = cv2.imread(img_file)
    img_RGB = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    img_hsv = cv2.cvtColor(img_RGB, cv2.COLOR_RGB2HSV)
def cvt_RGB2GRAY(img_file):
    img = cv2.imread(img_file)
    img_RGB = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    img_gray = cv2.cvtColor(img_RGB,cv2.COLOR_RGB2GRAY
img_hsv = cv_RGB2HSV(img_file)
img_gray = cvt_RGB2Gray(img_file)
print("img_gray_shape:", img_gray.shape)
img_gray_shape: (280,450) # Output
```

Exemplified Code 3 and its output (Image Histogram Generation)

```
def histogram(img_file, mask=None):
    img = cv2.imread(img_file)
    img_gray = cv2.cvtColour(img, cv2.COLOUR_BGR2GRAY)
    hist = cv2.calcHist([img_gray], [0], None, [256], [0,255])
    return hist.ravel()
h = histogram (img_file, mask = None)
plt.hist(h, normed = True, bins = 100)
plt.show()
```



30.

Exemplified Code 4 and its output (Calculation of Lightness Features)

```
import numpy as np
import cv2
def brightness_yuv(img_file):
# Step1:Load image with cv2
    img = cv2.imread(img_file)
# Step2: Convert image from BRG to YUV colour space
    yuv = cv2.cvtColour(img,cv2.COLOUR_BGR2YUV)
#Step3: Retrieve Y values of all pixels
    img_Y = yuv[:,:,0]
    img_Y = np.array(img_Y)
#Step4: Compute the mean and standard deviation of all Y values
    img_Y_mean = img_Y.mean() #f_11
    img_Y_std = img_Y.std() # f_12
    return img_Y_mean,img_Y_std

def satuation_hsv(img_file):
# Step1:Load image with cv2
    img = cv2.imread(img_file)
# Step2: Convert image from BRG to HSV colour space
    hsv = cv2.cvtColour(img, cv2.COLOUR_BGR2HSV)
#Step3: Retrieve S values of all pixels
    img_S = hsv[:,0,:]
    img_S = np.array(img_S)
 # Step4: Compute the mean and standard deviation of all S values
    img_S_mean = img_S.mean() #f_11
    img_S_std = img_S.std() #f_12
    return img_S_mean,img_S_std
```

Exemplified Code 5 (Facial Emotion Recognition with Pre-trained model)

```
# Step 1: Download the pre-trained Mini Xceptionalhuil Model from
https://github.com/abhijeet3922/FaceEmotion_ID/tree/master/models to your working directory
# Step 2: Load the model "_mini_XCEPTION.106-0.65.hdf5".
import numpy as np
from keras.models import load_model
import cv2
model = load_model("./emotion_detector_models/_mini_XCEPTION.106-0.65.hdf5")
# Step 3: Prepare the input image and resize it into 48*48 pixels.
dim = (48,48)
img = cv2.imread(img_file)
resized_face = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
# Step 4: Pass the input image into the loaded model and predict probabilities of the inclusion of the
seven emotions (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral) in the input
image.
pred =  model.predict(resized_face)
# Step 5: Output the predicted facial emotion by taking the arguments of the maxima (i.e., argmax)
from the probability list.
predicted_class = np.argmax(pred)
```

Exemplified Code 6 (Image Aesthetics Assessment)

```python
# Step 1: Download and load the trained model architecture and its weights from https://github.com/aimerykong/deepImageAestheticsAnalysis
# import libraries or modules for such task
import caffe  # If you get "No module named _caffe", either you have not built pycaffe or you have the wrong path.
import os
import glob
import cv2
import caffe
import numpy as np
from caffe.proto import caffe_pb2
# Step 2: Download the AVA dataset and a simplified version of AADB with resized images from https://www.ics.uci.edu/~skong2/aesthetics.html. Then load the mean image matrix and define image transformers for feeding into networks.
AVA_ROOT = '/Datasets/AVA/'
# Step 2.1. Reading caffe model and its weights
#caffe.set_mode_gpu()
caffe.set_mode_cpu()
Deploy = AVA_ROOT + 'initModel.prototxt'
Model_File = AVA_ROOT + 'initModel.caffemodel'
net = caffe.Net(Deploy, Model_File, caffe.TEST)

# Step 2.2. Data preparation
IMAGE_MEAN= AVA_ROOT + 'mean_AADB_regression_warp256.binaryproto'
IMAGE_FILE = AVA_ROOT + "*jpg"
#  Reading mean image
mean_blob = caffe_pb2.BlobProto()
with open(IMAGE_MEAN) as f:
    mean_blob.ParseFromString(f.read())
    mean_array = np.asarray(mean_blob.data, dtype=np.float32).reshape((mean_blob.channels, mean_blob.height, mean_blob.width))
# Define image transformers
def transform_img(img, img_width=IMAGE_WIDTH, img_height=IMAGE_HEIGHT):
    img = cv2.resize(img, (img_width, img_height), interpolation = cv2.INTER_CUBIC)
    return img
#Size of images
IMAGE_WIDTH = 227
IMAGE_HEIGHT = 227

input_layer = 'imgLow'
# Image processing helper function
def transform_img(img, img_width=IMAGE_WIDTH, img_height=IMAGE_HEIGHT):
     img = cv2.resize(img, (img_width, img_height), interpolation = cv2.INTER_CUBIC)
     return img
# Print("Shape mean_array : ", mean_array.shape)
# Print("Shape net : ", net.blobs[input_layer].data.shape)

net.blobs[input_layer].reshape(1,   # batch size
                                    3,   # channel
                                  IMAGE_WIDTH, IMAGE_HEIGHT)# image size
transformer = caffe.io.Transformer((Ettis))
transformer.set_mean(input_layer, mean_array)
transformer.set_transpose(input_layer, (2,0,1))

# Step 3. Make prediction of aesthetic score in terms of the input image.
test_img_paths = [img_path for img_path in glob.glob(Image_File)]
test_ids = []

preds = []
best_image = ''
best_score = 0.0

for img_path in test_img_paths:
    img = cv2.imread(img_path, cv2.IMREAD_COLOUR)
    img = transform_img(img, img_width=IMAGE_WIDTH, img_height=IMAGE_HEIGHT)
    net.blobs[input_layer].data[...] = transformer.preprocess(input_layer, img)
```

```
    out = net.forward()
    print(Houtsma)
    pred_score = out['fc11_score'][0][0]
    print (img_path, '\t', pred_score )
    if pred_score > best_score:
        best_score = pred_score  #print "Better score !"
        best_image = img_path
print ("Best image, based only on fc11_score = ", best_image )
```

Exemplified Code 7 (Image Memorability Simulation)

```
import numpy as np
import matplotlib.pyplot as plt
import sys
import caffe
# Step 1. Pre-trained model preparation. Download the pre-trained CNN named MemNet using Caffe deep learning toolbox (http://memorability.csail.mit.edu/download.html) , and load model into memory.
caffe.set_mode_cpu()
caffe_root = '/home/caffe/'
model_def = caffe_root + 'memory/model/memnet/deploy.prototxt'
model_weights = caffe_root + 'memory/model/memnet/memnet.caffemodel'
net = caffe.Net(model_def,      # defines the structure of the model
                model_weights,  # contains the trained weights
                caffe.TEST)     # use test mode (e.g., don't perform dropout)
from skimage import io;
io.use_plugin('matplotlib')
# Step 2. Data preparation
# load the mean ImageNet image for subtraction
mu = np.load(caffe_root + 'python/caffe/imagenet/ilsvrc_2012_mean.npy')
mu = mu.mean(1).mean(1) # average over pixels to obtain the mean (BGR) pixel value
print("mean_subtracted value:", zip('BGR',mu))
# define image transformation called 'input_layer'
transformer = caffe.io.Transformer({'input_layer': net.blobs['input_layer'].data.shape})
transformer.set_transpose('input_layer',(2,0,1)) # change image channels
transformer.set_mean('input_layer',mu) # subtract dataset-mean value in each channel
transformer.set_raw_scale('input_layer',255) # rescale to [0,255]
transformer.set_channel_swap('input_layer',(2,1,0)) #swap channels from RGB to BGR
net.blobs['input_layer'].reshape(50,3,277,277) # represent 50 batches, 3-channel and 277*277 image size separately
# Step 3. Prediction for memorability scores, HR (hit rate) is used as the memorability score
# load file list and loop
test_img_paths = [img_path for img_path in glob.glob(model_weights)]
test_ids = []
preds = []

for img_path in test_img_paths:
    img = cv2.imread(img_path, cv2.IMREAD_COLOUR)
    img = transform_img(img, img_width=277, img_height=277)
    net.blobs[input_layer].data[...] = transformer.preprocess(input_layer, img)
    out = net.forward()
    print(Houtsma)
score1 = out.values()[0][0]
socre2 = net.blobs['fc8-euclidean'].data[0][0]
print(img, score1,score2)
```