

Machine Learning 41204-85

Professor Mladen Kolar

Winter 2020

**ARTS - Aesthetic Restaurant Training System Proposal**

March 22, 2020

Arman Bhuiyan

Hikaru Sugimori

Spencer Reynolds

Shadmn Minir

**Executive Summary**

The present digital era has a strong appetite for the advancement of photography and image optimization through software, particularly in selecting the best photos. Of specific interest to this team was the use of this optimal photo selection feature in relation to improving a startup restaurant app. The team utilized a data scraper and a vectorization algorithm to create a digestible dataframe with a response variable created from picture ratings. Using this dataset, the team created a machine learning algorithm to predict the best pictures for restaurants, with a convolutional neural network model performing the best. The team recommends the use of the neural network model for the startup, with considerations to include comment utility to improve the capability of the ideal picture predictions.

**Introduction**

The playwright Henrik Ibsen once said, “A thousand words leave not the same deep impression as does a single deed.”1 While well meaning and morally righteous, this did not hold the staying power of its quickly misconstrued successor, the popular adage “a picture is worth a thousand words.” The latter statement has never been more evident that it has in today’s digital era, dictated by social relationships developed over Wi-Fi signals and advertisements dominating browser pages. In 2014 alone, over 657 billion photos were uploaded to the internet2, as people looked to offer the best depictions of their food, their weddings, and of course, their many selfies.

With the steady march towards digitization and the efforts of engineers and data scientists everywhere to improve efficiency in all aspects of life, photos have become a source of particular interest. From being able to identify objects in pictures to distinguish human faces from each other just a few years ago, AI and machine learning algorithms have looked to advance even further. The world is now at the point where algorithms are helping to computationally support the very camera taking the picture, with the intention to make any picture taken have the ideal balance of features, colors, and contrasts.

While taking consistently good photos is a nice advantage of living in 2020, there also exists a steady pressure for tools that work in sifting through a selection of photos, with the ultimate goal being to select the best photo. The key question becomes, can machine learning tools, using just the aesthetics in a photo, choose the most appealing photo for the targeted viewer? While people may disregard the value of being told optimal photo selection as the dream of a vain teenager on Instagram, there are in fact industries and careers that are directly influenced by the benefit of this kind of technology. Social media influencers looking for the ideal picture to represent their brand, a forsaken lover looking for the perfect shot to bounce back with on a dating app, and a writer looking for the right images to have their blog post go viral share the same interest in finding the photo that matches their goals.4 Of course, forward thinking corporations like Google have already seen these desires and have algorithms like their NIMA system hard at work suggesting the most emotionally and aesthetically appealing photos, with inputs from both statistical and subjective realms.5

**Objective**

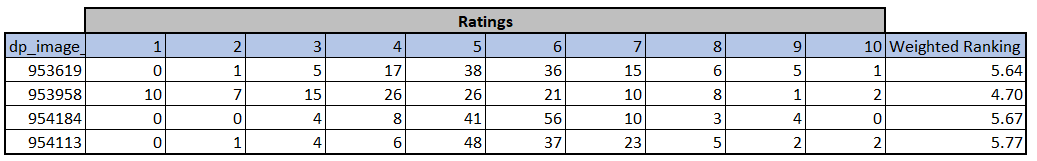
An optimal photo selection algorithm is of specific importance to the group due to the value it can provide for a startup headed by member Arman Bhuiyan. In his startup, a team works on the development of a restaurant rating application, similar to that of Yelp on initial observation. It differs in that it exists more to recommend restaurants based on the likes and interests of multiple users, looking to suggest a new brunch place or dinner spot that meets the palette interests of an entire friend group. A key factor in restaurant selection is the photos of the restaurant and the food it provides, with images provided by both the restaurant and reviewers being the primary selection perused by potential future customers. As a result, this group looks to understand the determining factors customers value when looking at pictures, particularly when it comes to restaurant selection.

To accomplish this goal, the team will acquire a dataset of rated photos that will help train machine learning algorithms that look for standout characteristics or common combinations of classes in popular images. After these tools are refined, they will help develop predictive models for the database of restaurant images of importance to the startup. Finally, recommendations for the best photos will be provided, as well as results that detail out errors within the final models.

**Data**

The data used to train the models was gathered from the DP Challenge website (dpchallenge.com). The DP challenge is an online website that developed concept challenges for photographers to aim to take pictures of.6 After entering their best pictures onto the site, registered site members would vote on the quality of the photo, between a ranking of 1 and 10. By taking in and vectorizing these images, the team was also able to have a response variable of picture rating for thousands of images, and understand not only which images were ranked the highest, but what traits within their borders caused them to stand out. The response rating is a weighted average of the individual rankings they received. The original data was listed as the number of users for each rating between 1 and 10. We then took those values to calculate a single rating value for each image.

Example calculation of weighted rankings:



To create a larger dataset to train the final model on for reduced bias, individual members of the team spent about a week downloading additional, unclassified images from the website to have a wide set of source data to model. This was done through the development of a web scraper that would crawl the DP Challenge website then locate and download the images. The length of time to acquire the data was due to trying to avoid being forced off the website server for overloading its ability to funnel through photos while downloading.

To vectorize the downloaded images, the team leveraged python’s ability to read images into a session. With that, all of the downloaded images were of different sizes. Additionally, due to some being in black-and-white while others were in full color, they had different dimensions of their pixels. As such, each image with first resized to be 300 X 300, and only images in color (which was the majority of the dataset) were kept. This would allow for more streamlined analysis and modeling later on. After this process, the images that were 300 X 300 X 3 became a single array of 270,000.

Once the images were re-sized and vectorized, the team utilized a smaller dataset on the DP Challenge website (~3000 images) that were identified by experts on the website as possessing anywhere from no significant aesthetic qualities to multiple. This dataset with the consistently sized vectors was used to build an aesthetic classification model after it was split into a training and validation set. Leveraging a neural network, the team was able to build out a model based on these 3000 images which maximized accuracy on the validation set. This model was then further applied to the large, unclassified dataset (~130,000 images) which did not have given classifications. Concerning the nature of how classifications were reported in the training datasets, images were biased either towards or against the respective aesthetic categories. The specific aesthetic characteristics following vectorization were:

Data Vectors

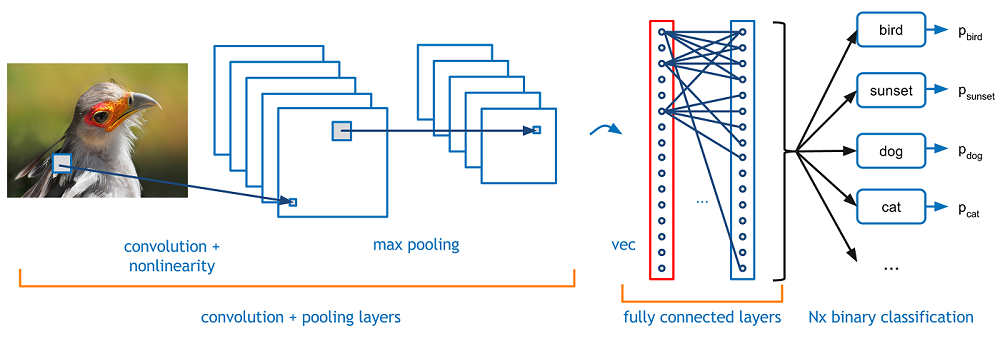
1. ID: identification number attached to picture name
2. Complementary Colors: Colors that strongly contrast and are “opposites”; if merged together, they result in a grayscale/black color7
3. Duotones: Used to bring middle tones and highlights of an image through superimposing contrasting color halftones over another color halftone8
4. HDR: High dynamic range; technique to improve luminosity over dynamic range9
5. Image Grain: random variation of brightness or color in images; electronic noise10
6. Light on White: a balance setting that neutralizes non-white colors such as yellow artifical light in order to create a neutral white background to make photos look natural11
7. Long Exposure: long duration shutter speed done to focus on stationary elements of an image and blur/smear out moving parts12
8. Macro: extreme close-up photography13
9. Motion Blur: streaking or moving objects in a frame due to rapid movement14
10. Negative Images: an image where lightest areas appear darkest and darkest areas appear lightest, reversed complementary colors15
11. Rule of Thirds: rule that viewers break down image into three parts vertically and horizontally(similar to a tic-tac-toe square), helps center a picture16
12. Shallow DOF: shallow depth of field, a small focus zone, background is often blurrier17
13. Silhouettes: an outline that appears dark against a light background18
14. Soft Focus: lens flaw where blurred images created by spherical aberration19
15. Vanishing Point: point in background where objects are so distant that the seem to disappear, or parallel lines converge20
16. Ratings: for training data, retrieved from average rating reported on the DPChallenge website; for test data, predicted as response variable. These rating values were the only ones not predicted on the larger, unclassified dataset, as they are not aesthetic qualities and were merely reported alongside the picture it was downloaded alongside.

Once the model was used to predict the above aesthetic qualities, the team developed an auto-encoding model to reduce the dimensionality of the images. The team was planning on building one final model for the predictions of ratings, and a dataset of 270,000+ independent variables per observation would be far too much for our computers. As such, an auto-encoder was built to reduce the size to only 500. The model read in images, tried various times to reduce the dimensions and then recreate the original images. The model with the closest resemblance to the initial image was chosen and used to auto-encode the entire dataset before the final modeling stages.

**Models**

Next, we created a “merged” csv dataframe that aggregated vectorized images, predicted classifications, and the actual weighted average score of these images. This dataframe was imported into python and we built a keras model with the vectorized images and classifications as the inputs and the weighted average score as the response.

Ultimately we concluded that a convolutional neural network (CNN) would be the most appropriate way to model this data as image processing is the classic, and arguably most popular, use case of these networks. Also, the Keras library in Python makes it pretty simple to build a CNN model.



CNNs are Deep Learning algorithms that take images as inputs and reduce the dimensionality of the image by passing images through a series of convolutional, nonlinear, pooling, and fully connected layers that scan the whole image and assign importance to various characteristics of the image. The end result is an output that best describes the image. This dimensionality reduction is what makes CNN especially adept at handling images since images have high dimensionality (computers consider each pixel to be a feature). In short, a CNN model allows a computer to “view” an image of a bird as a just bird instead of thousands of pixels. The architecture of CNNs is comparable to that of the connectivity pattern of neurons in the human brain. 21

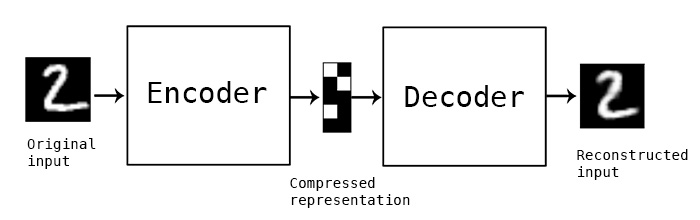
We built three models:

* An autoencoder to take an image and reduce it from 270,000 pixel representations to 500.
* A characteristic prediction model that was built off of the 3,000 images which had aesthetic qualities associated with them.
* A score prediction model that was built off of the entire image dataset of ~255,000.

For each of these, some of our key assumptions included the optimizer used, the batch size, and the number of epochs.

**Encoder Model**

Our autoencoder model was built using the Keras Python package. We decided to use rectified linear unit (ReLU) activation for this encoder, as our input variables were booleans, with either positive or zero values, and it is also one of the most common activation functions in CNNs. We did not have any negative values in our dataset which allowed us to leverage this mode. Additionally, since this was just for a part of our model and we were running this on a couple hundred thousand images, we appreciated that it was cheap to compute. Separately, we used the sigmoid activation function on the decoding part of the model. The input variables were all scaled to be between zero and one before being run through the auto-encoding model.



Once the encoder and decoder were set up, the vectorized images were scaled to between zero and one, and then they were split into a test and train dataset. These were then run with 5 epochs through the auto-encoder model, a batch size of 256, and shuffling enabled. With the finalized model, we applied the model to the full dataset for the further models.

**Characteristic Model**

Our characteristic model was a Sequential model, or a linear stack of layers. As we had resized all of our images, we were able to specify our input shape. The model required 3 inputs in order to be compiled - an optimizer, a loss function, and a list of metrics.

For our optimizer, we initially tried stochastic gradient descent (SGD), but our model was resulting in a number of empty results when we used it for predictions, even against the original dataset. We ultimately used Adaptive Moment Estimation, or Adam, to compute adaptive learning rates for each parameter. SGD was limited to using a single learning rate for all weight updates that did not update during training. Adam was more effective because of its use of two extensions of SGD in adaptive gradient algorithm, which maintains a per-parameter learning rate that reduces errors with sparse gradients, and root mean square propagation, which maintains a per-parameter learning rate that adapts to recent magnitudes of gradients for the weight. Adam used uncentered variance, utilizing the average of the second moments of gradients. It calculates the exponential moving average of the gradient and gradient squared, along with parameters that control the decay rates of the moving averages, before then correcting for an initial bias towards moment estimates of 0. When gradients become sparser, Adam often proves to be the most effective optimization tool for deep learning.22

**Score Prediction Model**

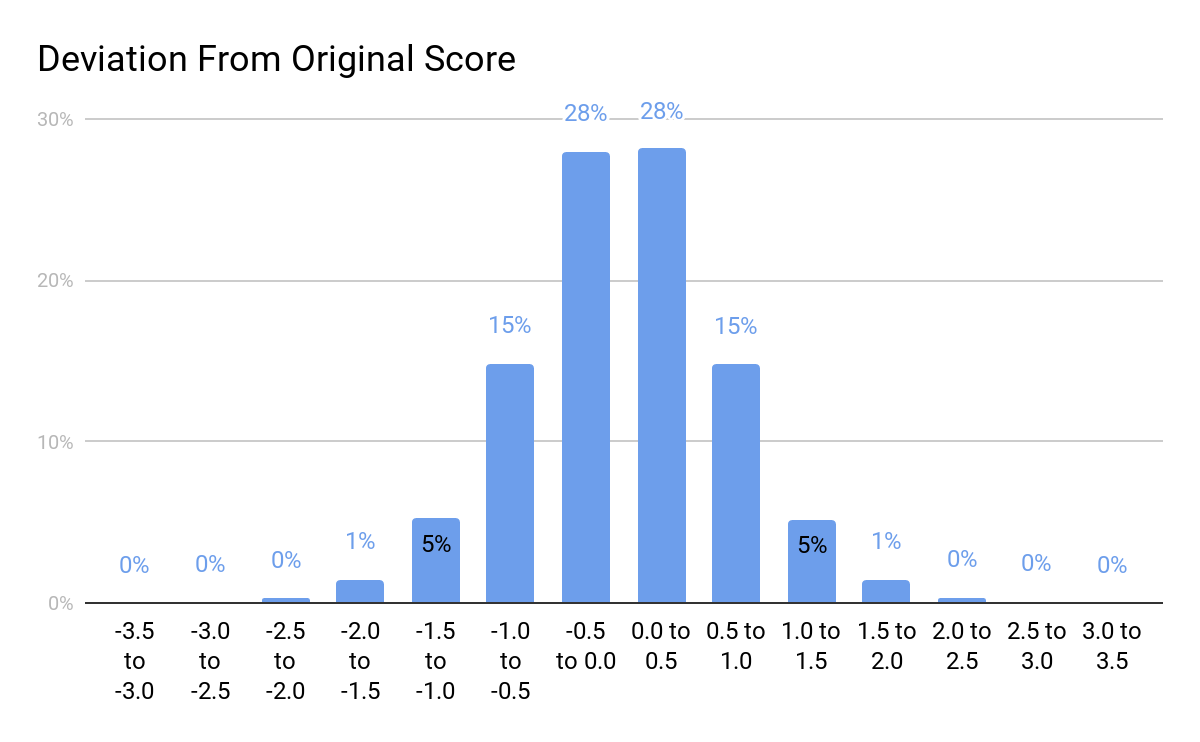
Finally, our score prediction model was, once again, a Sequential model, built very similarly to our characteristic model. However, as our dataset was much larger, we were willing to use a smaller split for the training set and include a validation set for the model. We set the training set to 60%, the validation set to 20% and the test set to 20%.

Some of the key variables for our model include a small batch size of 32, and 200 epochs. The batch size is the number of samples that the model works through before updating its internal parameters, and we targeted a low number. Several examples across the internet used batch sizes of 32, 64 and 256, and we chose the smallest of those for this particular problem. Our choice of epoch was driven by a target amount of time we desired for the model to be built. There are several pieces of literature that include a number of epochs in the thousands, but we did not have the processing power or time to run a model with that many iterations.

**Conclusions & Results**

Once the models were complete, we built a function to take any image, encode it, predict its characteristics, and predict a score for it out of 10.

The final score model was relatively close to the scores from the original dataset, with the same mean score (5.3765 out of 10) and a root mean squared error of 0.695, which is also the standard deviation, implying that 68% of scores were within 0.7 out of 10 out of the original score. In the below chart, a negative implies that the model had a lower score than the original, whereas a positive figure implies that the model scored the image above the original score. For example, if an image was originally scored at 6.42 and our model scored it at 5.63, then the deviation would be 0.79 and this particular image would fall into the range of “-1.0 to -0.5” in the below chart. Roughly 56% of the images fell within a score of 0.5 from the originals.

****

We ran several restaurant images, downloaded from various sources using Google Image Search, of varying subjective quality, through the model and predicted the below scores:

|  |  |
| --- | --- |
| **Image** | **Predicted Score** |
|  | 7.35 |
|  | 7.00 |
|  | 6.93 |
|  | 6.79 |
|  | 6.07 |
|  | 5.51 |
|  | 5.09 |

The scores that our external images received were within a very small range of about 4.5 to 7.0, with significant variation within the 5.0 to 6.0 range. There were a number of images that we would otherwise expect to be ranked very highly that were not, as well as a few that were not great images but were ranked average or above average. Please see the below examples of images that our model did not rank well, based on our subjective judgement:

|  |  |
| --- | --- |
| **Image** | **Predicted Score** |
|  | 5.53 |
|  | 5.29 |
|  | 4.04 |
|  | 3.75 |

**Recommendations**

Through this project we have developed an machine learning image classification model that can recognize, with a high degree of accuracy, the characteristics of an image as well as how aesthetically appealing an image is. In order to improve this model further, we believe that we can take the following steps to improve its performance with external images:

* Rather than reducing the autoencoder’s pixel representations from 270,000 to 500, we can perhaps look at using a larger number, like 2,000. This is a figure that we will likely want to experiment through multiple iterations.
* Our training/test/validation set for the final scoring model was split 60%/20%/20%, but we would like to build a new model that has a larger training set, such as an 80% training set and 20% test set.
* Our hypothesis was that including aesthetic classifications would be accretive to the model’s performance. However, we would like to run a version of the model without the aesthetic quality classifications to see how much of a difference it makes to include them.

To further the capability of the model, the modeling of the attractiveness of facial features (apply to aesthetics) could be implemented to further its robustness in weighing classifications for the optimal picture selection. 23 The model could also further be enhanced through the use of additional traits related to the image, such as most recent date of upload, comparison to pictures just like it within a small gallery, and the strength, or lack of, comments regarding a picture.

**References**

1. “A Picture Is Worth a Thousand Words.” *Wikipedia*, Wikimedia Foundation, 12 Feb. 2020, en.wikipedia.org/wiki/A\_picture\_is\_worth\_a\_thousand\_words.
2. Eveleth, Rose. “How Many Photographs of You Are Out There In the World?” *The Atlantic*, Atlantic Media Company, 3 Nov. 2015, [www.theatlantic.com/technology/archive/2015/11/how-many-photographs-of-you-are-out-there-in-the-world/413389/](http://www.theatlantic.com/technology/archive/2015/11/how-many-photographs-of-you-are-out-there-in-the-world/413389/).
3. Byford, Sam. “How AI Is Changing Photography.” *The Verge*, The Verge, 31 Jan. 2019, [www.theverge.com/2019/1/31/18203363/ai-artificial-intelligence-photography-google-photos-apple-huawei](http://www.theverge.com/2019/1/31/18203363/ai-artificial-intelligence-photography-google-photos-apple-huawei).
4. “Visual Aesthetics: Judging a Photo's Quality Using AI Techniques.” *By*, hackernoon.com/visual-aesthetics-judging-a-photos-quality-using-ai-techniques-6f2551cc9b0b.
5. Reporter, Dailymail.com. “Google's AI Software Is Learning What Makes a Good Photo.” *Daily Mail Online*, Associated Newspapers, 22 Dec. 2017, [www.dailymail.co.uk/sciencetech/article-5207101/Googles-AI-software-learning-makes-good-photo.html](http://www.dailymail.co.uk/sciencetech/article-5207101/Googles-AI-software-learning-makes-good-photo.html).
6. “About DPChallenge - A Digital Photography Contest.” *DPChallenge*, [www.dpchallenge.com/help\_about.php](http://www.dpchallenge.com/help_about.php).
7. “Complementary Colors.” *Wikipedia*, Wikimedia Foundation, 5 Mar. 2020, en.wikipedia.org/wiki/Complementary\_colors.
8. “Duotone.” *Wikipedia*, Wikimedia Foundation, 4 Oct. 2019, en.wikipedia.org/wiki/Duotone.
9. “High-Dynamic-Range Imaging.” *Wikipedia*, Wikimedia Foundation, 17 Feb. 2020, en.wikipedia.org/wiki/High-dynamic-range\_imaging.
10. “Image Noise.” *Wikipedia*, Wikimedia Foundation, 16 Feb. 2020, en.wikipedia.org/wiki/Image\_noise.
11. Lawton, Rod. “The A to Z of Photography: White Balance.” *TechRadar*, TechRadar, 27 July 2018, [www.techradar.com/news/the-a-to-z-of-photography-white-balance](http://www.techradar.com/news/the-a-to-z-of-photography-white-balance).
12. “Long-Exposure Photography.” *Wikipedia*, Wikimedia Foundation, 28 Jan. 2020, en.wikipedia.org/wiki/Long-exposure\_photography.
13. “Macro Photography.” *Wikipedia*, Wikimedia Foundation, 5 Mar. 2020, en.wikipedia.org/wiki/Macro\_photography.
14. “Motion Blur.” *Wikipedia*, Wikimedia Foundation, 6 Jan. 2020, en.wikipedia.org/wiki/Motion\_blur.
15. “Negative (Photography).” *Wikipedia*, Wikimedia Foundation, 12 Nov. 2019, en.wikipedia.org/wiki/Negative\_(photography).
16. “Rule of Thirds in Photography.” *Digital Photography School*, 25 May 2018, digital-photography-school.com/rule-of-thirds/.
17. “Understanding Depth of Field for Beginners.” *Digital Photography School*, 26 Oct. 2018, digital-photography-school.com/understanding-depth-field-beginners/.
18. “Silhouette Photography.” *PictureCorrect*, [www.picturecorrect.com/tips/silhouette-photography/](http://www.picturecorrect.com/tips/silhouette-photography/).
19. “Soft Focus.” *Wikipedia*, Wikimedia Foundation, 13 Dec. 2019, en.wikipedia.org/wiki/Soft\_focus.
20. “Understanding Vanishing Points - NYIP Photo Articles.” *New York Institute of Photography*, [www.nyip.edu/photo-articles/photography-tutorials/understanding-vanishing-points](http://www.nyip.edu/photo-articles/photography-tutorials/understanding-vanishing-points).
21. Deshpande, Adit. “A Beginner's Guide To Understanding Convolutional Neural Networks.” *A Beginner's Guide To Understanding Convolutional Neural Networks – Adit Deshpande – Engineering at Forward | UCLA CS '19*, <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>
22. Brownlee, Jason. “Gentle Introduction to the Adam Optimization Algorithm for Deep Learning.” *Machine Learning Mastery*, 13 Nov. 2019, machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.
23. Shulga, Dima. “How Attractive Are You in the Eyes of Deep Neural Network?” *Medium*, Towards Data Science, 22 Apr. 2018, <https://towardsdatascience.com/how-attractive-are-you-in-the-eyes-of-deep-neural-network-3d71c0755ccc>