Evaluating Photo Aesthetics Using MobileNetV3 and Feature Engineering

Frank Zhao
Stanford University
frankz24@stanford.edu

Scott Xu
Stanford University
scottxu@stanford.edu

Abstract

Automatic image aesthetics assessment has been an important area of research and has a wide variety of applications, such as image thumb-nailing and automatic correction of photographic images. Recently, the advances in deep CNN networks makes it promising to produce aesthetic rating of the images in an end-to-end scheme. In this project, we designed a new model architecture and feature engineering method to predict aesthetic rating over the AVA dataset by an aggregation of MobileNet [5] networks. Different from previous work, our method attempted to apply the saliency map of a pretrained model, image histogram and HOG (histogram of oriented gradients) as features to the network attached to a two-layer fully connected structure. Our model achieves an accuracy of 77.2% *on the AVA dataset and demonstrates correct behaviour in a range of applications such as automatic image enhancement and automatic cropping.*

1. Introduction

For a professional photographer, sorting through and ranking thousands of images after one day of shooting can be a time-consuming and inefficient task. Therefore, automated quality assessment of photo aesthetics can be a game changer for photo editors. It is also useful in "a wide variety of applications such as evaluating image capture pipelines, storage techniques", and album thumbnail composition. [1] We plan to use image prepossessing techniques as an feature engineering stages to extract features like image histogram, saliency maps, and histogram of oriented gradients to use for training deep neural networks. Specifically, we use these augmented features along with the image as our input and use an aggregated MobileNet architecture to predict the distribution of human ratings for a specific image in the AVA dataset [10]. We also plan to qualitatively evaluate what the model regards as good photography - how will the exposure, saturation, or composition change the overall rating.

2. Problem Statement

We define our problem as a regression problem, which takes in the image and its auxiliary features to predict a discrete probability distribution over the ratings $\hat{p}_i, 1 \leq i \leq 10$ such that $\sum_{i=1}^{n} \hat{p}_i = 1$. Some work prior to NIMA [1] also framed the same problem as a binary classification problem because of the lack of human-annotated aesthetic rating, so we will also apply the predicted probability distribution to produce a similar binary classification result (rating score bewteen 1-5 or between 6-10).

3. Related Work

Recent advances in convolutional neural networks architecture have been instrumental for assessing images' aesthetic values better. In 2016, Shu Kong et al. trained "a Siamese network that takes a pair of images as input and directly predicts relative ranking of their aesthetics in addition to their overall aesthetic scores." [6] They used a "Aesthetics with Attributes Database" that contains eleven attributes closely related to traditional photography rules on top of raw pixels. Their best CNN-based architecture achieved an accuracy of 73.33% on the AVA dataset, which achieves the state-of-the-art at that time without semantic labels of the photos. However, explicit attribute labels may not always be available "due to the high cost of manual annotation and the expert knowledge required for feature design". [11] Many of the future work tends to design more robust architectures that use less domain knowledge to predict aethestics score for the images.

In 2017, Hossein Talebi et al. used VGG16 [13] and Inception v2 [14], and MobileNet [5] to predict the distribution of human opinion scores on the AVA dataset using a novel squared EMD (earth mover's distance) loss function. They also used this quality assessment approach to "effectively tune parameters of image denoising and tone enhancement operators to produce perceptually superior results." [1]

In 2018, Kekai Sheng et al. improved the architecture by training an 18-Layer ResNet [2] on the raw AVA dataset to predict a binary class of whether an image is aestheti-

cally positive or negative. More so, they used "an attention-based objective to enhance training signals by assigning relatively larger weights to misclassified image patches." [11] They achieved the current state-of-art binary classification accuracy of 83%, and the highest linear correlation coefficient between predicted and ground-truth rating means of 0.636. Yet, predicting binary classes is still not reflective of real-life human ratings compared to distribution prediction in [1].

Beyond these major work, other teams have tried different model architectures and methods to extract key image features or components. For example, Vlad Hosu et al. improved upon the naive fully connected layer in previous architecture and tried different combinations fully connected structures at the classifier stage [3]. The best performing architecture is Pool-3FC, which uses a inception-type module with 3 stacked fully connected layers, batchnorm, and dropout.

Some other work focuses on incorporating local and global features into training. Shuang Ma et al. proposed an adaptive layout-aware architecture, which feed "the most discriminative and informataive patches" and a layer-aware attribute graph into separate CNNs and aggregate their representations [9]. The RGNet model proposed by Dong Liu et al. also applied a similar idea by extracting key components from the image and locally compute aesthetics-preserving features in each components and apply graph convolution over the composition graph formed by these features [8], achieving performance on par with SOTA.

It's notable that image aesthetics can also be framed as a reinforcement learning problem that gets wide range of applications such as automatic image adjusting in the fields of photography and multi-media content creation. For example, Debang Li et al. proposed the A2-RL model to find the best cropping strategy so that the cropped image produce the highest aesthetics rating [7].

4. Dataset

Our experiment uses the AVA (Aesthetics Visual Analysis) dataset [10], which contains approximately 255,000 images rated manually based on their aesthetic qualities by an average of 210 amateur photographers. The images are collected from www.dpchallenge.com, a digital photography contest. The dataset contains image ratings of the photos, semantic labels for over 60 categories, and labels related to photographic style. The image rating ranges from 1 to 10 with 10 being the highest and the mean rating is around 5.5. Our experiment only make uses of the images themselves and the rating labels, considering the fact that professional comments and style labels of the images are rarely accessible in real applications of the image aesthetic model.

In training time and testing time, we apply the same augmentation techniques following [1]. First, all images are rescaled so that the shortest dimension is 256 pixels, and we randomly selected crop it into a 224×224 image snapshot. Then, we apply random horizontal flip to the image and normalize the RGB channels of the image into the same scale. We believe that both the random cropping and random horizontal flip helps reduce overfitting on the image, and would still preserve aesthetic feature of the image (compared to other augmentation like vertical flips or color jittering, which we choose not to apply in this experiment).

As the AVA dataset covers a wide range of images from photographic challenges, the distribution of rating among different challenges and different types of photo may vary. In general, photos with higher mean rating are considered as having higher aesthetic quality, but it's noted that some unconventional techniques results in high variance of a photo's rating, creating extra difficulties for predicting the full rating distribution.

5. Baseline Method

We mainly use the method NIMA [1] as a baseline, which tried both a VGG-16 and Inception-v2 network pretrained on ImageNet. The network has its last layer removed and is attached with a linear and softmax layer to perform the regression task in this problem.

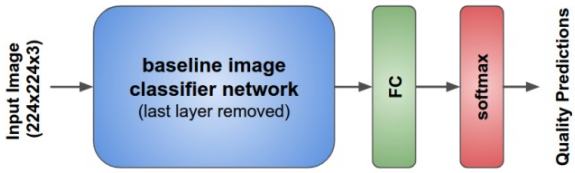


Figure 1. Baseline Model

The baseline method doesn't include auxiliary features of the model other than the original image pixels. The baseline methods minimizes the EMD (Earth Mover's Distance) loss to compare ordered classes that has more accurate representation than the cross entropy loss on unordered classes. Specifically, the loss is defined as the minimum cost to move the probability mass of one distribution to the other. For two distribution $p = (p_1, \cdots, p_N), \hat{p} = (\hat{p}_1, \cdots, \hat{p}_N)$

$$EMD(p, \hat{p}) = \left(\frac{1}{N} \sum_{k=1}^{N} |CDF_p(k) - CDF_{\hat{p}}(k)|^r \right)^{\frac{1}{r}}$$

where $CDF_p(k)$ is defined as the cumulative distribution $\sum_{i=1}^{k} p_i$.

To obtain this baseline model, we use a pretrained VGG16 network on the AVA dataset from an open-source PyTorch implementation on https://github.com/

yunxiaoshi/Neural-IMage-Assessment. The model follows from the design in [1] and is trained further until 85 epochs near convergence. We will refer to this model as the *baseline model* for the rest of the paper.

6. Proposed Methods

Our method aims to build on top of the baseline method by applying more feature engineering techniques as auxiliary input to our model and experimenting alternative model architectures on the inputs.

6.1. Feature engineering

Inspired by previous work that extract local and global features of the images, our project mainly wish to experiment whether auxiliary features about the images could help the model learn better to predict its aesthetic rating. Therefore, we apply three feature engineering techniques: saliency map, image color histogram, and HOG (Histogram of Oriented Gradients).

6.1.1 Saliency Map

First proposed in [12], the saliency map of the model is often considered as a visualization approach after the training phase to measure which parts of the visual information the model pays most attention to. However, our team hypothesizes that such information is also valuable as an auxiliary feature for the training stage by allowing the model to distinguish important and unimportant components of the images or learn the hierarchical structures between different scenes in the image. For example, the following is the saliency map of a very aesthetic image:

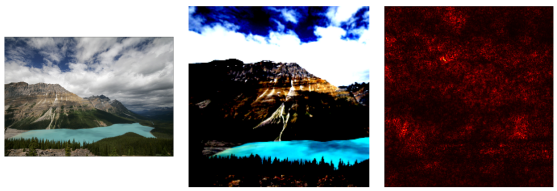


Figure 2. Example saliency map

The brighter spots indicate more important regions whereas the darker spots indicate less important regions. As we can see, the most important regions in the image are the mountain peaks and the lake. The saliency map contains information about the focus of the photo and may therefore contain some information about the composition of the photo - both of which are important for aesthetic predictions. This intuition comes from the common practice of featuring the main object in the center or in the rule-of-thirds spots in photography, as photos with less focuses may perform worse to engage its audience.

To extract the saliency map of the images, we use the fixed baseline model. We compute a backward pass to the baseline model and extract gradient of the unnormalized scores corresponding to the correct class with respect to each pixels of the image. We take the absolute value of the gradient and compute the maximum across the channel to produce our saliency map for each image, denoted as $S \in \mathbb{R}^{1\times 224\times 224}$.

6.1.2 Color Histogram

The color histogram of an image is a representation of the distribution of colors in an image. Specifically, the color histogram for each channel (R, G, B) records the number of pixels that have intensity in the range of integer values from 0 to 255. In this project, we created four color histogram from three channels (R, G, B) and also produce a grayscale histogram based on these RGB values, where the grayscale pixel strength is calculated as

$$\text{Gray} = 0.299 \times \text{Red} + 0.587 \times \text{Green} + 0.144 \times \text{Blue}$$

These features are concatenated to produce $H \in \mathbb{R}^{4\times 256}$ before feeding in the fully connected module. As an example, the following is the colour histogram for the image in Figure 2.

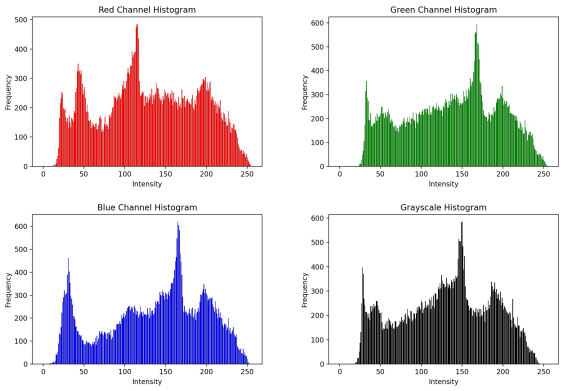


Figure 3. Example colour histogram

As an auxiliary feature to predict image aesthetics, we believe that the color histogram produces insights about exposure and colour balance on the photograph. For example, the colour histograms in Figure 3 show that the image has well spread-out pixel intensities for all the channels. This usually suggests that the image is well-exposed and has a good balance of colours. In contrast, if we see large peaks toward the left, the image may be underexposed and if we see large peaks toward the right, the image may be overexposed. These features are usually used by photographers in the editing process.

6.1.3 Histogram of Oriented Gradients

To account for the texture information of the images, we also extract the HOG (histogram of oriented gradients) feature for each image. The features ignores the color information of the image and only include local object appearances and shape within the images, which is described by the edge directions and intensity gradients within subdivided cells that span the images. Our implementation builds on top of the assignment code in https://cs231n.github.io/assignments2022/assignment1/#q5-higher-level-representations-image-features and we choose the cell size to be 16×16 to get less fine-grained information. For each image, the HOG feature extractor produces a HOG feature vector $O \in \mathbb{R}^{1764}$. Along with the color histogram described above, we hope that these two features can pre-compute many of the structural information from the image for the model to better identify key traits of the image.

6.2. Model Architecture

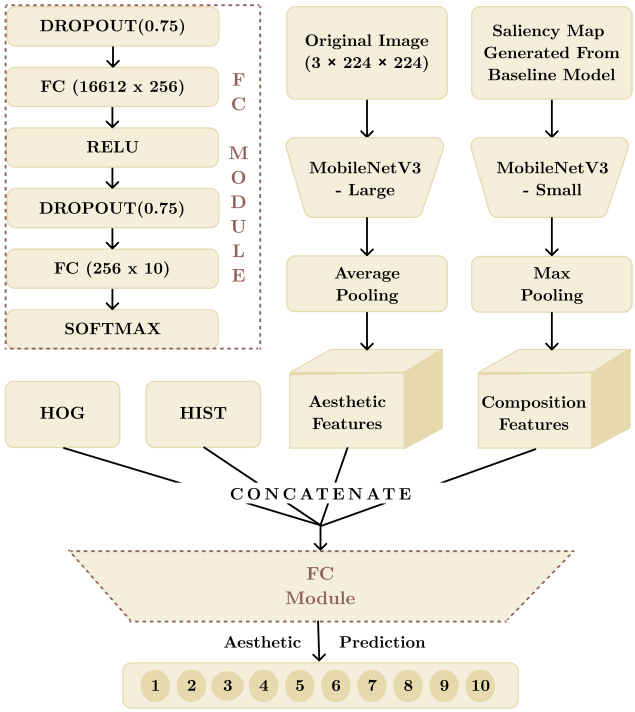


Figure 4. Proposed model architecture

To select our backbone CNN model for the AVA dataset, we choose to use MobileNetV3 [4] since it has fewer parameters and requires shorter training time considering the constraints of our training resources. The first stage of our model is to get the hidden representation of the image itself and its saliency map obtained from the pretrained VGG model in the baseline. For each image input $I_0 \in \mathbb{R}^{3\times224\times224}$, we pass in the feature extraction layers of a large MobileNetV3 model to get its hidden representation $I_h \in \mathbb{R}^{960\times7\times7}$. For the saliency map, we pass it into a small version of MobileNetV3 model and use its feature extraction layers to get a hidden representation $S_h \in \mathbb{R}^{576\times7\times7}$.

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Figure 5. Architecture of MobilenetV3 (large) [4]. The smaller architecture can be found in the same original source, and our experiment cut off the network before the pooling layer.

To reduce the parameter of the model and extract more generalizable information, we apply two pooling layers respectively on I_h and S_h. For image feature I_h, we choose the 2D average pooling layer (3×3 kernel, stride 2) to best preserve the neighboring information on image representation. For saliency feature S_h, we choose the 2D max pooling layer (3×3 kernel, stride 2) since the extreme values on the saliency map show more informative insights on where we should pay attention to the image. This produces $I'_h \in \mathbb{R}^{960\times3\times3}$ and $S'_h \in \mathbb{R}^{576\times3\times3}$ and we flatten these representation into $I_f \in \mathbb{R}^{8640}$ and $S_f \in \mathbb{R}^{5184}$ to be concatenated with our auxiliary features.

Finally, we concatenated I_f, S_f, the color histogram feature H, and the HOG feature O into $[I_f, S_f, H, O]$ as the input to our fully connected module. Different from the single layer in the baseline model, we designed two fully conneted layers with ReLU non-linear activation and layer-wise dropout probability 0.75. The input dimension of the first layer is $8640 + 5184 + 1024 + 1764 = 16612$ and the intermediate dimension is chose to be 256. The resulting model has roughly 8M parameters and each epoch takes 80 minutes to train. Notably, this has significantly less parameters compared to our baseline model (50M), and the training time is slightly reduced for each epoch.

7. Results and Evaluation

7.1. Results of Baseline Model and Proposed Model

To train our proposed method, we use a batch size of 128, and we use the SGD optimizer with 0.9 momentum, $5e^{-4}$ learning rate for the convolutional layers, and $5e^{-3}$ learning rate for the fully connected layers. These hyperparameters are proven to be suitable for the baseline model and also in our tests to overfit the proposed model on small subsets of the data. To prevent overfitting, we also introduce an exponential learning rate decay of 0.95 for every 10 epoches. And after every epoch, we evaluate the model on the validation set and we only store the epoch with the highest validation performance. We were able to train the model to 45 epoches where we noticed that the model has started converging.

We then compute some important evaluation metrics to compare the baseline model with our proposed model on the test set.

We can use a mean rating of 5 to binary-classify all the images in the testing set, i.e. if an image has an average rating of 5 or more, it is classified as a "good" image and vice versa. Then some common evaluation metrics are calculated for the baseline model below. Specifically, denote the number of true positives as TP, the number of false negatives as FN, the number of true negatives as TN, and the number of false negatives as FN. Then, accuracy= $\frac{TP+TN}{TP+FP+TN+FN}$, precision= $\frac{TP}{TP+FP}$, recall= $\frac{TP}{TP+FN}$, and F1 score= $\frac{2}{\frac{1}{\text{Precision}}+\frac{1}{\text{Recall}}}$.

	Baseline	Our model
Accuracy	76.5%	**77.2%**
Precision	78.4%	**79.1%**
Recall	**92.0%**	91.9%
F1 Score	84.6%	**85.0%**

Further, we can calculate the linear correlation coefficient (LCC= $\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$) and the Spearman's rank correlation coefficient (SRCC= $\frac{\text{cov}(R(X),R(Y))}{\sigma_{R(X)}\sigma_{R(Y)}}$, where $R(X), R(Y)$ are the rank variables.) of the predicted and ground-truth rating distribution mean and standard deviation. We can also calculate the Earth Mover's Distance (EMD) loss between the predicted distribution and the ground-truth distribution.

	Baseline	Our model
LCC of mean	0.595	0.608
SRCC of mean	0.577	0.598
LCC of std.dev	0.206	0.099
SRCC of std.dev	0.203	0.078
EMD Loss	0.0712	0.0718

We may also use the mean (rounded to the nearest integer), median or the mode to predict the final rating of an image. Then, we can regard the problem as a 10-class classification problem, where we are trying to predict a single rating for an image. We then calculate the accuracy below.

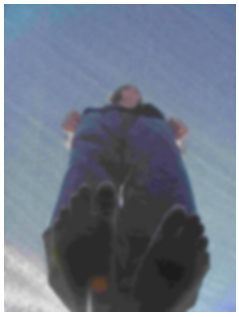
	Baseline	Our model
Mean Accuracy	0.568	**0.575**
Median Accuracy	0.593	**0.596**
Mode Accuracy	0.625	**0.628**

In terms of binary classification accuracy, our model is able to perform better than the baseline model. And note that we only had to train our model for 45 epochs whereas the baseline model had to be trained for 85 epochs. The dataset is slightly unbalanced as it contains more "good" images but the model still performs quite well in terms of precision, recall, and F1 score. Our model also has a higher LCC and SRCC of mean values, which shows that it can predict the mean aesthetic score quite well. However, our model performs worse in terms of prediction's standard deviations, which is the main shortcoming of our proposed method. This could be due to the fact that the AVA dataset itself is unbalanced. There are more mediocre images and so our prediction tends to stay in that narrow range of $4-7$ but fails to produce very low or very high scores. The other metrics are also comparable or slightly better than the baseline model, which again shows the effectiveness of our model design.

Lastly, to further demonstrate what the dataset consists and what the model does, we show the two images with the highest predicted mean rating and the lowest predicted mean rating here.

6.563 ($\pm$1.376)

Figure 3. Image with highest predicted rating



4.190 ($\pm$1.465)

Figure 4. Image with lowest predicted rating

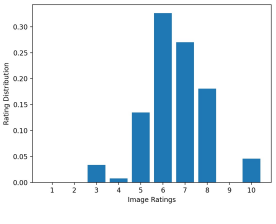


Figure 5. Rating distribution for Figure 3

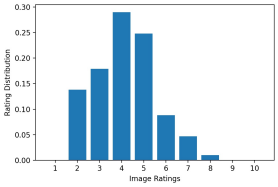


Figure 6. Rating distribution for Figure 4

7.2. Success Cases and Failure Cases

We present some success cases and failure cases in this section.

Prediction:	4.968 (±1.420)	4.986 (±1.404)	5.781 (±1.485)
Ground Truth:	4.968 (±1.453)	4.986 (±1.568)	5.781 (±1.285)

Figure 6. Top 3 success cases.

The success cases are mostly images with a simple main focus, somewhat mediocre content, and conventional styles. It is not surprising that the AVA dataset would contain many similar images in this category for the model to learn them very well.

Prediction:	4.917 (±1.437)	4.950 (±1.436)	4.779 (±1.441)
Ground Truth:	1.809 (±1.488)	8.000 (±1.670)	2.083 (±1.826)

Figure 7. Top 3 failure cases.

The failure cases contain images that are mostly monotone and almost entirely white. It is common for bright images to be rated higher because they are usually more visually attractive. So, perhaps the model is fooled by the exposure information and fails to recognise the lack of content information in the first and third images. In contrast, for the second image, perhaps the model is not able to recognize the unconventional composition choices and the small main subject of the image. Humans may be able to recognize the serenity and gradation in the image, which results in the disparate performance.

7.3. Feature Importance

Also, we evaluate whether the augmented features like saliency map, color histogram and HOG substantially affect the output of our model. To evaluate their contribution, we use the Integrated Gradients method in the pytorch captum module in https://captum.ai/api/integrated_gradients.html. This method approximates the integral of gradients of the model's output (of a given output class) with respect to the inputs, and a higher value of attribution score in absolute value means a higher influence of the input to the model's prediction. We randomly sample 1000 images from the test set, calculate the maximum attribution score in absolute value from each type of features (image, saliency, color histogram, and HOG) and average these maximum score over all the 10 classes and over all test samples. The results are shown in the table below:

Feature Type	Average attribution
image features	0.01465
saliency map	0.00136
color histogram	1.22×10^{-6}
HOG	2.22×10^{-6}

Table 1. Average maximum attribution score of training features

The result indicates that the color histogram and HOG plays a relatively minor role in affecting the output of the model. A possible explanation is that the deep CNN structure is already able to capture the color and edge distribution of the original image and developer deeper features from the image features than it would from the histogram and HOG features. But these auxiliary features may still be useful to provide a cold start for the model to learn approximate features and speed up its convergence. We could see that the saliency map outperformed the other two auxiliary features to affect the model's prediction, but not as greatly as the image features themselves.

7.4. t-SNE graph

Further, to confirm the ability of the model to separate the two different classes of images, we plot the t-SNE graph of the hidden representations over 50 random positive samples and 50 random negative samples. The representations are retrieved before the last fully connected layer and have dimension 256. We found that there is a large separability between the two classes which ensures the expressiveness of the model, despite a few outliers. The result also hints us that using two fully connected layers may not be necessary as the first fully connected layer already possess such separating ability.

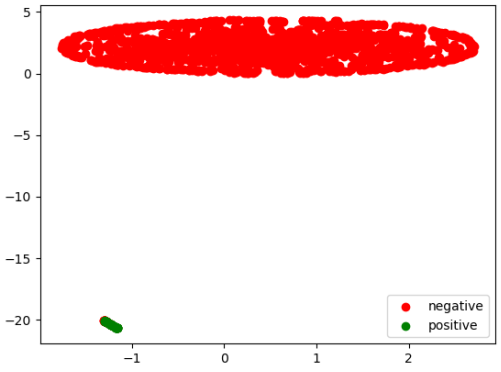


Figure 8. t-SNE graph of hidden representation in $\mathbb{R}^{256}$ before last FC layer

7.5. Effect of Image Adjustments on Aesthetic Scoring

After analysing our model, we want to test it in two applications. In this subsection, we change an image's exposure, contrast, and sharpness to see how these adjustments influence the model's aesthetic scoring. Based on these results, we can also provide auto-enhancing suggestions to edit images.

We use Pillow's ImageEnhance module [15] to change the original image's brightness/contrast/sharpness. An adjustment factor of 1 leaves the image untouched; an adjustment factor of less than 1 decreases the brightness/contrast/sharpness and an adjustment factor of more than 1 increases the brightness/contrast/sharpness. Some sample images after adjustment are shown in Figure 14, Figure 15, and Figure 16 in the Appendix. Correspondingly, we also show the graph of the model's aesthetic prediction in response to these adjustments in Figure 9, Figure 10, and Figure 11.

7.5.1 Changing Brightness

This graph shows how the aesthetic prediction score changes in response to brightness adjustments.

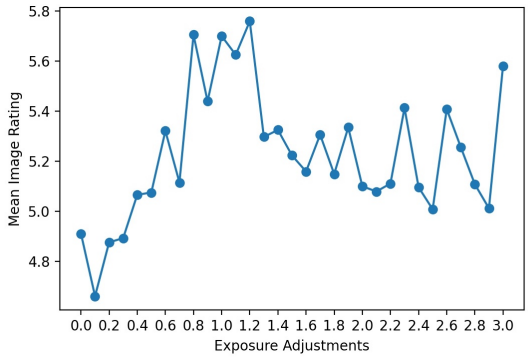


Figure 9. Changes of Aesthetic scores in response to changes in brightness adjustments.

From Figure 9, we can see that the response roughly follows an inverted "U" shape. When we decrease the brightness, the predicted aesthetics score decreases and when we increase the brightness too much, the predicted aesthetics also decrease. There is a sweet spot (1.2) in the middle where the image is most aesthetic. Note that this aligns with our human intuition very well as images that are too dark or too bright are not visually pleasing.

7.5.2 Changing Contrast

This graph shows how the aesthetic prediction score changes in response to contrast adjustments.

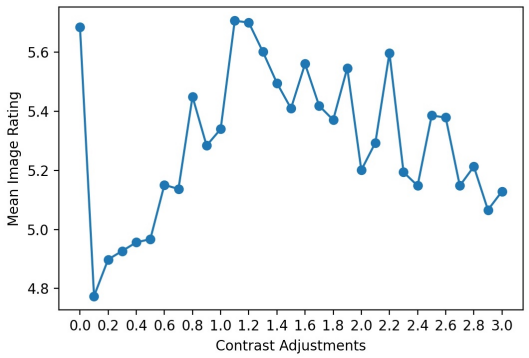


Figure 10. Changes of Aesthetic scores in response to changes in contrast adjustments.

From Figure 10, we can see that the responses to contrast adjustments roughly follow an inverted "U" shape as well. When we decrease the contrast, the predicted aesthetics score decreases and when we increase the contrast too much, the predicted aesthetics also decrease. There is a sweet spot (1.1) in the middle where the image is most aesthetic. Note that this again aligns with our human intuition. These two examples demonstrate the power of our model.

7.5.3 Changing Sharpness

This graph shows how the aesthetic prediction score changes in response to sharpness adjustments.

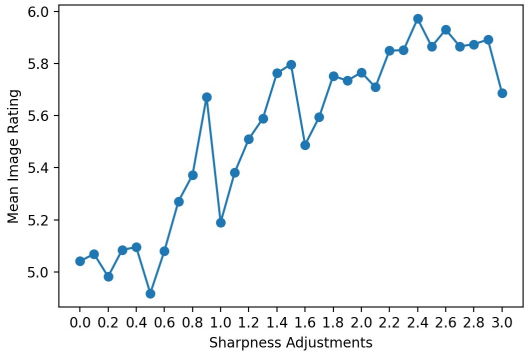


Figure 11. Changes of Aesthetic scores in response to changes in sharpness adjustments.

In contrast to brightness and contrast adjustments, the model seems to prefer more sharpness is a monotonic fashion. The higher the sharpness, the higher the model predicts its aesthetic score. This also tend to align with our perceptions. Further, perhaps more sharpness results in sharper edges, which allows the model to discern the main object and composition better.

7.5.4 Automatic Editing

Based on these results, we select the adjustment in each category that results in the best aesthetic rating (brightness = 1.2, contrast = 1.1, and sharpness = 2.4) to "automatically" enhance the original image. We can see that the enhanced image does look more visually pleasing and the predicted aesthetic score is higher.

Original Image
5.651 (±1.501)

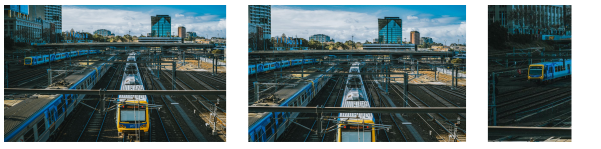
Auto-enhanced Image
5.735 (±1.501)

Figure 12. Comparing original and auto-enhanced images.

7.6. Random Cropping

Finally, in this subsection, we randomly crop different regions of the same image and feed them to our model to see which crop achieves the highest rating. This has the potential application to automatically crop a given photo. This can be useful for many applications such as "image cropping, image thumbnailing, view recommendation, and autonomous photo taking." [16]

Specifically, we used a sliding window approach to crop 3 different sizes of the original image in landscape orientation as well as in portrait orientation. This resulted in 93 cropped images overall. Out of all these crops, the best cropped image and worst cropped image are shown below



Original Image
5.708 (±1.494)

Best Cropped Image
5.757 (±1.484)

Worst Cropped Image
4.735 (±1.445)

Figure 13. Comparison between original image and best cropped image and worst cropped image.

We can see that the best cropped image does center the main subject a bit better and achieves a higher predicted aesthetic rating. The worst cropped image is a dark patch on the corner without a main focus, which is why it has a low predicted aesthetic rating. As further research, we can even train a auto-crop neural network whose objective is to get a crop that maximises our model's aesthetic rating.

8. Conclusion

In this paper, we propose a novel model design that uses two lightweight MobileNet architecture alongside some feature augmentations, including saliency maps, colour histograms, and histogram of oriented gradients. Notably, this novel design is able to achieve slightly higher performance than the baseline model in [1] with roughly half the training epochs. It is able to achieve an accuracy of 77.2% on the AVA dataset. Further, we analysed the usefulness of augmented features, concluding that saliency maps are the most useful whereas HOG and histograms are much less useful. We also found that the first fully connected layer already contains enough separating ability from "good" images to "bad" ones. Lastly, we tested our model on two important applications - automatic image enhancement and cropping. The model demonstrates excellent behaviour on these two tasks.

In our future work, we could update the model design based on our analysis. We could replace the MobileNetV3-Large model with a more powerful Inceptionv3 model and eliminate the HOG and hist features to see if it performs better. We could also look for ways to address imbalance in the dataset. Currently, there are too many mediocre images and the model tends to fail to predict very low or very high aesthetic scores. Perhaps by putting more weight on failure training cases or putting more loss on the two endpoints, we can make the model more robust. And lastly, we could further experiment with the model's application. We could build other neural networks for these applications whose aim is to maximise out model's predicted aesthetic scoring.

9. Contributions and Acknowledgements

We use the implementation of NIMA model in https://github.com/yunxiaoshi/Neural-IMage-Assessment as our baseline method. Our implementation of HOG feature extraction refernces the assignment code in https://cs231n.github.io/assignments2022/assignment1/#q5-higher-level-representations-image-features. And our implementation of saliency map extraction references the code in https://towardsdatascience.com/saliency-map-using-pytorch-68270fe45e80. Our team members work jointly on the project and make equal contribution with a rough division of the following tasks:

- Frank Zhao: Literature review, testing code, saliency map and color histogram extraction, training, evaluation (image adjustments, image cropping), model design, and report writing.

- Scott Xu: Literature review, model code, HOG feature extraction, training, evaluation (feature importance, t-SNE graph), model design, and report writing.

References

[1] Hossein Talebi Esfandarani and Peyman Milanfar. NIMA: neural image assessment. *CoRR*, abs/1709.05424, 2017. 1, 2, 3, 8

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 1

[3] Vlad Hosu, Bastian Goldlücke, and Dietmar Saupe. Effective aesthetics prediction with multi-level spatially pooled features. *CoRR*, abs/1904.01382, 2019. 2

[4] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. *CoRR*, abs/1905.02244, 2019. 4

[5] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. 1

[6] Shu Kong, Xiaohui Shen, Zhe Lin, Radomír Mech, and Charless C. Fowlkes. Photo aesthetics ranking network with attributes and content adaptation. *CoRR*, abs/1606.01621, 2016. 1

[7] Debang Li, Huikai Wu, Junge Zhang, and Kaiqi Huang. A2-RL: aesthetics aware reinforcement learning for automatic image cropping. *CoRR*, abs/1709.04595, 2017. 2

[8] Dong Liu, Rohit Puri, Nagendra Kamath, and Subhabrata Bhattacharya. Composition-aware image aesthetics assessment. *CoRR*, abs/1907.10801, 2019. 2

[9] Shuang Ma, Jing Liu, and Chang Wen Chen. A-lamp: Adaptive layout-aware multi-patch deep convolutional neural network for photo aesthetic assessment. *CoRR*, abs/1704.00248, 2017. 2

[10] Naila Murray, Luca Marchesotti, and Florent Perronnin. Ava: A large-scale database for aesthetic visual analysis. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2408–2415, 2012. 1, 2

[11] Kekai Sheng, Weiming Dong, Chongyang Ma, Xing Mei, Feiyue Huang, and Bao-Gang Hu. Attention-based multi-patch aggregation for image aesthetic assessment. pages 879–886, 10 2018. 1, 2

[12] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer, 2014. 3

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[14] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. 1

[15] P Umesh. Image processing in python. *CSI Communications*, 23, 2012. 7

[16] Zijun Wei, Jianming Zhang, Xiaohui Shen, Zhe Lin, Radomir Mech, Minh Hoai, and Dimitris Samaras. Good view hunting: Learning photo composition from dense view pairs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 8

10. Appendix

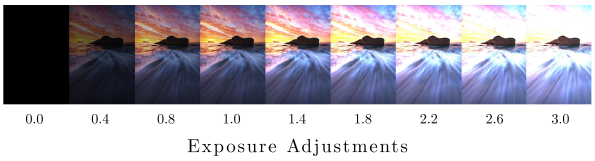


Figure 14. Different brightness adjustments on the image.

Figure 15. Different contrast adjustments on the image.

Figure 16. Different sharpness adjustments on the image.