Office of Graduate Studies

Dissertation / Thesis Approval Form

This form is for use by all doctoral and master's students with a dissertation/thesis requirement. Please print clearly as the library will bind a copy of this form with each copy of the dissertation/thesis. All doctoral dissertations must conform to university format requirements, which is the responsibility of the student and supervising professor. Students should obtain a copy of the Thesis Manual located on the library website.

Dissertation/Thesis Title: Predicting E-commerce Item Popularity Using Image Quality Features

Author: Stephen Zakrewsky

This dissertation/thesis is hereby accepted and approved.

Signatures:

Examining Committee

Chair ______ Ali Shokoufandeh

Members ______ Dario Salvucci

______ Kamelia Aryafar

______

______

______

Academic Advisor ______

Department Head ______

Predicting E-commerce Item Popularity Using Image Quality Features

A Thesis

Submitted to the Faculty

of

Drexel University

by

Stephen Zakrewsky

in partial fulfillment of the

requirements for the degree

of

Master of Computer Science

June 2016

Dedications

This thesis is dedicated to my children -

May you cherish the pursuit of education.

Acknowledgments

I would like to thank all of the people whose help and inspiration made this work possible. I would like to thank my family for providing love and support . My loving parents Donna and Victor were an inspiration and instilled the values of education and hard work. They taught me to create goals and work as hard as I can to accomplish them. I would like to thank my wife Megan, who supported our family through out this process and put up with the countless hours spent on school work.

I greatly appreciate the time with my advisors Dr. Shokoufandeh and Dr. Aryafar without whose advise this would have never have happened. In fact, a long time ago, Dr. Shokoufandeh actually put the thought of graduate school in my mind. Dr. Aryafar was a big help in bringing this idea to fruition and proving guidance along the way.

Table of Contents

List of Tables

List of Figures

Abstract

Predicting E-commerce Item Popularity Using Image Quality Features

Stephen Zakrewsky
Advisor: Ali Shokoufandeh, Ph.D.

In order to traverse the plethora of items for sale online, searching, ranking, and recommendation systems must be built, and the quality of these systems can make the difference between boom or bust. In all of these methods, being able to distinguish between popular and non-popular items is very important. Traditionally, these systems have only utilized textual metadata, however, images represent first order information to the shopper, and are composed of a variety of signals that shoppers respond to. In this thesis we look at the problem of predicting item popularity on a popular e-commerce site using image quality features, and show that these features provide complementary information to the textual features in making this prediction.

Chapter 1: Introduction

In order to traverse the plethora of items for sale online, searching, ranking, and recommendation systems must be built, and the quality of these systems can make the difference between boom or bust. In all of these methods, being able to distinguish between popular and non-popular items is very important. Traditionally, these systems have only utilized textual metadata, however, images represent first order information to the shopper, and are composed of a variety of signals that shoppers respond to. In this thesis we look at the problem of predicting item popularity on a popular e-commerce site using image quality features, and show that these features provide complementary information to the textual features in making this prediction.

Online e-commerce has a market size valued at over a trillion dollars worldwide. It is projected that this will continue to grow at a fast pace, from 7.3% of the total retail market to 12.4% in four years [1]. This growth can be attributed to increasing mobile connectivity in rural areas, a growing Asian-Pacific market, and increased competition by the largest online retailers. However, according to the popular e-commerce blog site LemonStand[2], over 94% of the 12 million e-commerce sites do less than $1000 per year in sales. In addition to price competition, these small stores need to ensure that they are providing the same rich user experience as the larger online retailers or risk losing out on this growing market. Accurate online search, ranking, and recommendation systems are critical to quickly connecting users to the items they want to buy in order to capture the purchase.

Distinguishing between popular and non-popular items is important to these tasks but is hard to define because it is subjective depending on the context. For example, from the perspective of multimedia, what makes an artistic image popular such as photographic quality and color, while important to a journalistic photo is not as important as the journalistic subject matter. Additionally, these qualities seem far from what would make a viral videos popular, such as cats or babies doing funny things. In the context of e-commerce, popularity is defined by what we want, or simply what

[1] https://www.internetretailer.com/2015/07/29/global-e-commerce-set-grow-25-2015

[2] http://blog.lemonstand.com/just-how-big-is-the-ecommerce-market-youll-never-guess/

people buy or are likely to buy. Luckily, purchases are tracked by any typical e-commerce site. Other common popularity metrics used in e-commerce are number of times items have been favorited, or viewed.

Traditionally, systems to search, rank, and recommend have been built utilizing only textual metadata. For example, a typical e-commerce search engine will build an inverse index mapping query terms to items for sale. The results are then sorted using a statistic that is designed to put items that are more relevant to the query first. A common way to do that is by scaling the counts of terms in each item description by the counts in all of the descriptions. This way items are associated with the terms that are most frequent but are also not common among all of the items. On some e-commerce sites, all of the textual information is provided by sellers, so the opportunity to optimize the search process is limited. However, there is a large amount of additional untapped information in the images that the sellers upload.

Images represent first order information to the shopper, and are composed of a variety of signals that shoppers respond to such as simple signals such as color, abstract signals like mood, and even specific objects. Extracting these features are an important task for any system looking to utilize the image information. Various global image features such as color histograms, easily segmentable regions and popular localized features such as SIFT, SURF and HOG have been proposed for image content retrieval and search. These features are also popular for general computer vision tasks because they operate as a low level, such as identifying edges, corners, and other local image areas with a lot of signal change. The concepts that trained photographers use to evaluate the photographic quality of images such as light, color, rule of thirds, texture, smoothness, blurriness, depth of field, and scene composition, also influence the way we experience an image and are the features we examine in this thesis.

In order to study our hypothesis on real data we partner with the e-commerce site Etsy. Etsy[3] is a marketplace where people around the world connect, both online and offline, to make, sell and buy unique goods. With over 35 million items for sale from 1.6 million sellers, and with 24 million active

[3] https://www.etsy.com/about/

buyers, effectively delivering the information that connects both parties to each other is paramount. This task is made additionally challenging. The items for sale on Etsy are handmade, vintage, items and crafts that can't be found anywhere else. The nuances that describe what makes each item unique must be provided by the sellers because they are marketing their own goods and providing their own item descriptions. Therefore, it is important for Etsy to provide the tools that accurately connection buyers to the items they want to buy. For example, understanding user preferences and which listings should appear higher in search results are areas Etsy are continuously working to improve.

In this thesis we look at the problem of predicting item popularity on the e-commerce site Etsy. In mathematical terms, this is a binary classification problem, where given an item we want to know whether it is popular or not popular. We define popularity using metrics collected by Etsy, and build a classifier to predict popularity using image quality features. We show that classification using image quality features are not only statistically significant but that these features provide complementary information to the textual features when making this prediction.

The structure of this thesis is as follows. Chapter 2 provides background on the related concepts needed to understand this thesis. Some of these have been mentioned briefly already, such as the classification problem. In Chapter 3 we detail the image quality features that were used and show examples. Then the experiment setup, how we collected data, the classifier, and the results are presented in Chapter 4. Lastly we summarize this work, and discuss further research in this area.

Chapter 2: Background

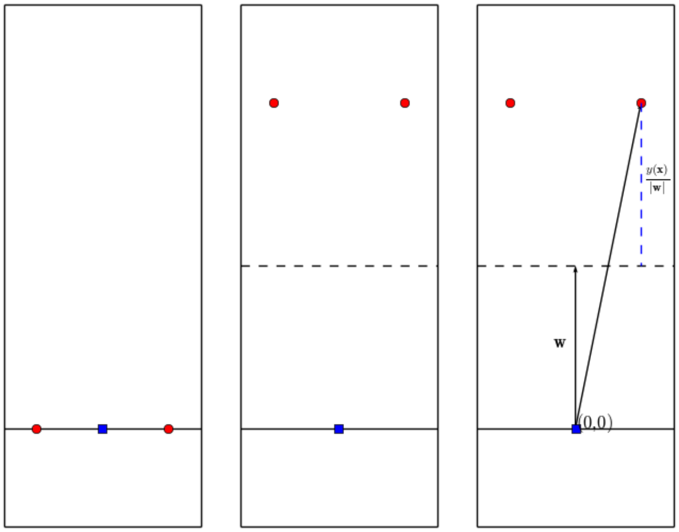
In this chapter we present the background material needed to understand our work. First and foremost, we are building a classifier, therefore, we start with some background on machine learning and common classification tools. Then we talk about what it is that we are classifying, namely the concept of popularity, and we look at examples of previous work. Finally, we discuss the state of the art in multimodal systems similar to the one we are describing in this thesis.

2.1 Classification

The classification problem that we focus on is a subproblem in the field of machine learning. Machine learning is a field where the parameters of a data model are learned from a collection of existing data. Let's look at the very common Gaussian model, $\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. If we have some data and know that the Gaussian is a good representative model, then all we need to do is find the mean μ and standard deviation σ. Finding the sample mean and standard deviation from a set of discrete observations is pretty straight forward, but more advanced models require more advanced techniques.

The Gaussian model shown in the example above is a statistical model; it models the probability density of the random variable. However, machine learning models don't need to be based on statistics at all. Discriminant functions directly map the input to the output, and are often linear functions with respect to the model parameters. For example, let $\mathbf{x} \in \mathbb{R}^m$ be the input and $\mathbf{a} \in \mathbb{R}^n$ be the model parameters, then a discriminant function might be $y = \sum_{i=1}^{n} a_i \phi_i(\mathbf{x})$. The literature often uses a nonlinear feature function ϕ to allow linear models in feature space to be nonlinear in input space. Figure 2.1 shows a simple example where two classes of data in a one dimensional space are not linearly separable, but by transforming the input into two dimensional space using a quadratic function, it becomes linearly separable.

Data is a critical component of machine learning and is used to learn the optimal model pa-



(a)

Figure 2.1: On the left are two classes of nonlinearly separable data in one dimensional space, and in the center is the same data that has been transformed into two dimensional space using a quadratic function ϕ. This data is now linearly separable. On the right is a visualization of the geometry of the linear discriminant function; $\mathbf{w}$ is perpendicular to the separating plane, and the distance from the point to the separating plane is $\frac{y(\mathbf{x})}{\|\mathbf{x}\|}$.

rameters. This is also called training the model, or fitting the model. When the training data is labeled, that is to say, the training step takes n samples $X = \langle x_1, \ldots, x_n \rangle$ and their corresponding n labels $L = \langle l_1, \ldots, l_n \rangle$, the process is known as supervised learning. Examples of supervised learning include the discrete classification problem, where labels are known as classes, and the regression problem. In both of these we build a model to predict the label from new unlabeled data. When the training data is unlabeled then the process is referred to as unsupervised learning. Examples of unsupervised learning include clustering, density estimation, and data visualization.

Classification is a supervised learning model used to predict class labels of new unlabeled data. A two class linear Support Vector Machine (SVM) [1;2] is a very popular type of classifier that supports very high dimensional feature spaces given by the discriminant function $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$. It is a type of maximum margin classifier. That is we would like to define the class boundary such that the distance to the closest point in each class is maximized. This will provide the best generalization of the model for new unseen data. Solving for $\mathbf{w}$ is a quadratic programming problem. In the traditional sense, it is usually first converted to its dual form using Lagrange multipliers. The parameter b can be determined by $\mathbf{w}$ so it is dropped from the optimization problem. Figure 2.1 shows the relationship between $\mathbf{w}$, the decision boundary, and the distance to the the closest data points.

The two class SVM only outputs binary class labels, however, sometimes it is necessary to have the likelihood estimate assigned with each class $\{C_1, C_2\}$. Logistic regression[3] is a discriminative model for classification that returns the probability of the class label, $p(C_1|\phi) = \sigma(\mathbf{w}^T\phi)$, where $\sigma(\cdot)$ is the logistic sigmoid function. It is popular because it is a simple model that works under the assumption that the classes have a shared covariance matrix and it has a linear number of model parameters compared to a similar generative model which has a quadratic number of parameters with respect the to feature space dimensionality.

This is a maximum likelihood problem which can be solved by minimizing an error function. For n samples $X = \langle x_1, \ldots, x_n \rangle$ and their corresponding n labels $L = \langle l_1, \ldots, l_n \rangle$, where $l_i \in \{0, 1\}$, the likelihood function is $p(\mathbf{L}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{l_n}\{1-y_n\}^{1-l_n}$ and the error function is the negative logarithm $-\log p(\mathbf{l}|\mathbf{w})$. The parameters of this model can be solved using stochastic gradient decent, iterative reweighted least squares, or other technique.

The material presented above is generally well known and many implementations are available for public use. We will not go into more detail here, but curious readers are urged to take a look at the references, or a machine learning text book such as Pattern Recognition and Machine Learning[4]. Now that we understand that the goal of classification is to predict some label, that we need labeled sample data to train our model, and we know a few basic techniques, lets look at the feature space we will be dealing with.

2.2 Popularity

We are building a classifier to predict the popularity of items for sale on the popular e-commerce site Etsy. In particular, we want to extract features from the item images that are highly correlated with this goal. This section outlines previous work in extracting quality or popularity features from images, and defines popularity how we are using it.

Early work defined popularity as quality[5] or aesthetics[6] and use data from photography rating websites where users who have interest in photography upload their photos and rate others. Popularity has also been defined as memorability[7], and interestingness[8;9]. More recent work has directly tackled popularity. Khosla et al.[10] defines popularity as the number of views on Flickr, and Aryafar et al.[11] uses favorited listings on Etsy.

Popularity tends to be predicted using SVM classification or regression[6 10 12 13]. Datta et al.[6] uses a two class SVM classifier with a forward selection algorithm to find good feature sets. By using elastic net to rank feature relevance to aesthetics, and a best first algorithm to find feature sets that minimize the root mean squared error cross validation error, Wang et al.[13] are able to achieve a 30.1% improvement compared to Chen et al.[12]. A few have explored other machine learning techniques. Ke et al.[5] uses a naive Bayes classifier, not SVM. Aryafar et. al[11] studied the significance of color in favorited listings on Etsy using logistic regression, perceptron, passive aggressive and margin infused relaxed algorithms.

The features used in popularity prediction model the same qualities professional photographers use such as light, color, rule of thirds, texture, smoothness, blurriness, depth of field, scene composition[5 6 12 13]. Most of these features are unsupervised, but some such as the spacial edge distribution and color distribution features of Ke et al.[5] require all of the labeled training data. Some recent work has looked at semantic object features. Khosla et al.[10] used the popular Convolution Neural Net ImageNet to detect the presence of 1000 difference object categories in the image. The presence/absence of these categories is used as the feature.

In our work, we define popularity as listings that have been favorited, clicked on, or purchased, and we show that unsupervised image popularity features are statistically significant when combined with traditional text metadata features in predicting popularity. This combination of text and image features presents the classification problem as multimodal.

2.3 Multimodal

It is a natural human process to combine multiple modes of input into perceiving everyday tasks. Consider the five senses of sight, touch, taste, smell, and hearing. Has a smell ever triggered a memory? Can you learn simply by seeing something, does it help for someone to explain it too, or do you prefer to try it yourself? These are just a few examples of the multiple inputs we interact with every day. Multimodal machine learning is a new and growing field based on this idea that humans learn from multiple sensory inputs. The alternate input streams provide complementary information and compensate for noisy or low quality signals in another. In the rest of this section we discuss other work regarding multimodal machine learning.

Multimodal music and text has been studied for cross-modality information retrieval, and classification[14],[15],[16]. Cross-modality retrieval retrieves data in one modality from queries in another, for example a multimodal lyrics and audio machine can retrieve lyrics from audio queries. The result of using Canonical Correlation Analysis (CCA), an unsupervised technique to define a linear transformation that maximizes the correlation between the two multimodal spaces, improved the mean average precision by 10 percentage points over a baseline non-CCA cross-modality retrieval system. The second example is of a multimodal classifier to predict the artist from an audio query, but it is trained with multimodal lyrics and audio features. This multimodal fusion classifier improved the average classification accuracy rate by 5.76 percentage points over a similar audio features only classifier. The third example of genre classification predicts the genre given audio using a variant of SVM. An ℓ_1 sparsity-eager SVM is based on the concept of replacing the optimization objective of the classic SVM of maximizing the margin, with one that make the result as sparse as possible. In the research the average classification accuracy rate improved by 2.1 percentage points over the single modality classifier.

The examples so far have been multimodal with respect to audio and text. For our work we build a model from both image and textual features. Lynch et al.[17] apply multimodal image and text features to the task of ranking. In their work, the image features are deep neural net features taken from the last fully connected layer in the VGG-19 network pre-trained on ImageNet. The multimodal feature vector is simply a concatenation of the text and image features. We use the same concatenation approach in our work. Other state of the art work such as that done by Vinyals et al.[18] generate natural sentences describing an image using image and text features as training data. The system Neural Image Caption (NIC) encodes an image using a Convolution Neural Net, and then processes it with a Recurrent Neural Net to generate sentences. Given an image I, the system is trained to maximize the probability $p(S|I)$ of the sequence of words $S = S_1, S_2, \cdots S_n$.

2.4 Conclusion

This chapter presented the background material needed to understand our work. We discussed the basics of machine learning, and described the supervised classifiers SVM, and Logistic Regression. Then we talked about what it is that we are classifying, namely the concept of popularity, and we looked at examples of previous work. Finally, we discussed the state of the art in multimodal systems similar to the one we are describing in this thesis. In the next chapter we will describe the feature space of our classifier, namely the image quality features.

Chapter 3: Features

Careful choice of features is an important step in machine learning. Large dimensionality of the input space makes solving model parameters directly a difficult task, therefore, a smaller set of discriminate features must be chosen. However, care must be taken because not everything will correlate well with the task. Let's consider a simple example of finding a known object in an image. We could try and scan the object through the image in all possible scales, rotations, lighting and occlusions; with all of the infinite possibilities, this doesn't sound very promising. Alternatively, we can try to identify important points and match the geometry of those points. The latter makes use of features and is the method used in some of the best known object matching algorithms.

In this thesis we focus on image popularity features. Early work defined popularity as quality[5], aesthetics[6], memorability[7], and interestingness[8;9]. More recent work has directly tackled popularity. Khosla et al.[10] defined popularity as the number of views on Flickr, and Aryafar et al.[11] uses favorited listings on Etsy. In our work, we define popularity as listings that have been favorited, clicked on, or purchased. The features used in popularity prediction model the same qualities professional photographers use such as light, color, rule of thirds, texture, smoothness, blurriness, depth of field, and scene composition[5;6;12;13]. These features are unsupervised, meaning that their values can be computed directly from the prescribed algorithm without first training it.

This chapter presents the image features used by our classifier to predict popularity. First it introduces basic image concepts such as in-memory image representation, color spaces, simple filters, line detection, scale space pyramids, and frequency analysis. Using these basic concepts, we then discuss the individual features such as spacial edge distribution, blur, rule of thirds, and texture used to predict popularity.

3.1 Image Basics

An image I is a multi-channel matrix of pixel intensity values. For example, a black and white image is one channel with values constrained such that $I_{ij} \in \{0, 1\}$. The most popular format, Red, Green, Blue (RGB), is an 8-bit 3 channel image with pixel values $\{I_{ij} \in \mathbb{R}^3 | 0 \leq \mathbb{R} \leq 255\}$. A gray scale image is similar, but with only one channel.

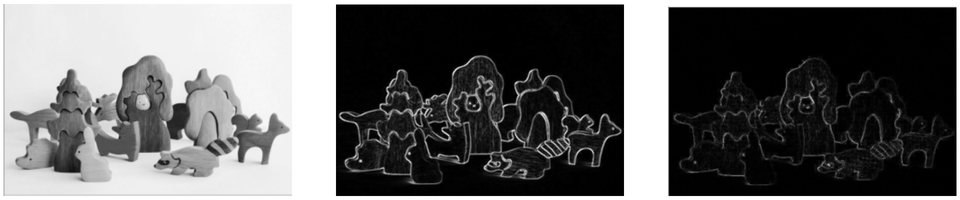
The RGB color space is popular because it has historical ties to computer monitors and sensory input devices, however, it also has some disadvantages. First, all the combinations of RGB can't produce all human distinguishable colors. Second, luminance is integrated into all three channels, therefore changing luminance requires balancing all three channels. The Lab color space was developed to be a perceptually uniform color space that overcomes the limitations of RGB. In this color space $0 \leq L \leq 100$ is a separate luminance channel that closely matches the way humans perceive changes in luminance. Other color spaces exist too. For example, Hue, Saturation, Value (HSV) is the common color wheel notation in which $0 \leq H < 360$ is the angle around the wheel, $0 \leq S \leq 1$ is a vector distance from the origin, and $0 \leq V \leq 1$ is luminance. In HSV, pure black is defined when $V = 0$, and pure white when $V = 1$ and $S = 0$.

Given an image representation, we would like to use it to do something useful. Common low level image operations include smoothing, scaling, edge detection, and pyramids. All of these utilize a technique called filtering $g(i, j) = \sum_{k,l} f(k, l)h(i - k, j - l)$ in which each output pixel value is a sum of a neighborhood of input pixel values weighted by a shifted filtering kernel h centered on the output pixel location. This operation is also known as the convolution operation. The choice of kernel, determines the type of operation. For example, a Gaussian kernel will smooth the image because it is a weighted average of neighboring pixels, where as specialty kernels such as the Sobel kernel shown in Figure 3.1 will approximate the gradient.

Gradients measure the rate of change. In an image, changes in pixel values usually mean different objects or parts, or more precisely, edges. Mathematically, the derivative is the tool to compute rate of change, and the second derivative can be used to find the extrema points. While the gradients usually do a good job enhancing edges, an alternative is by taking the second derivative. The

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Figure 3.1: Sobel kernel for approximating first derivative with respect to x



(a) **(b)** **(c)**

Figure 3.2: Figure b. shows the combined Sobel image gradient image, and figure c. shows the Laplacian edge detection image.

Laplacian is a common edge detection method defined as the sum of the second derivatives in the x and y directions $\Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$. Figure 3.2 shows a sample image with output from a Sobel gradient along side a Laplacian.

When working with images, scale is a factor. A popular technique for dealing with features at different scales is to build an image pyramid by repeatedly downsampling the image, shown in Figure 3.3. A useful side-effect of this image pyramid it that it can be used to build complementary pyramid of edge detections in scale space. A Laplacian pyramid is computed by subtracting the image at one level of the pyramid with the upsampled image from the next lower resolution level of the pyramid. This method is also known as a Difference of Gaussians (DoG); it is an approximation of the Laplacian mentioned above. A Laplacian pyramid is shown in Figure 3.3.

Wavelets are similar to DoG, and are also used in pyramids. Wavelets localize space and frequency and consist of a lowpass filter and a series of highpass filters that are computed from subtraction of lower levels in the pyramid. Typically, wavelets produce three images, HH, HL, and LH which isolate different high frequency areas of the image. Again, high frequencies indicate change and changes indicate edges. The defacto standard in frequency analysis is the Fourier Transform. The Fourier transform is a technique from signal processing where any signal can be decomposed into basic sinusoidal signals. The magnitude of the frequency space measures how much of that frequency

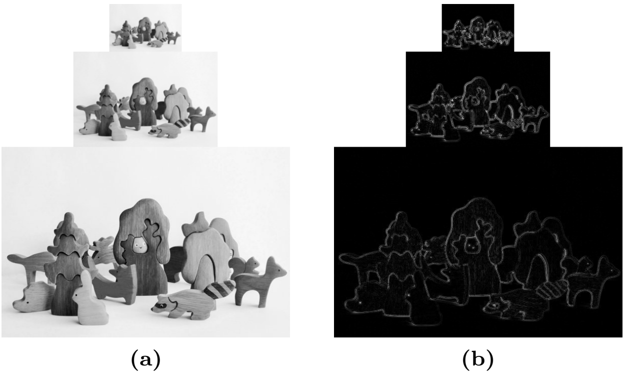


Figure 3.3: Image pyramids; a. is a regular image pyramid, and b. is its complementary Difference of Gaussians pyramid.

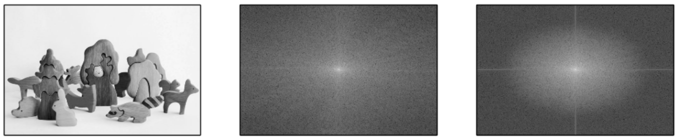


Figure 3.4: Fourier transform of a sample image. The center image is the frequency magnitude spectrum of the sample image. The image on the right is the frequency magnitude spectrum after applying a Gaussian blur. Notice that the higher frequencies further away from the center are dampened.

is in the original image. The figure 3.4 shows an example.

So far we understand how an image is represented as pixels, and different color formats that are supported. We learned about some basic image operations, and how we can detect edges and build scale invariant pyramids. In the rest of this chapter we use these tools to extract popularity image features.

3.2 Simplicity

High quality photos are typically simpler than others. They often have one subject placed deliberately in the frame. Sometimes the background is out of focus to emphasize the subject. Poor quality photographs tend to have cluttered backgrounds and it may be difficult to distinguish the subject of the scene. We used the four measures of simplicity from Ke et al.[5], spatial edge distribution, hue count, contrast and lightness, and blur.

3.2.1 Spatial Edge Distribution

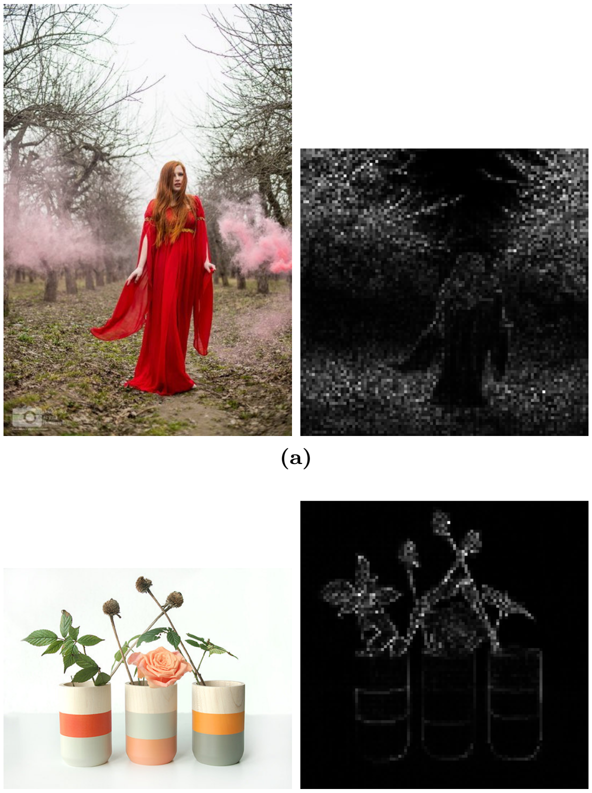
Spatial edge distribution measures how spread out sharp edges are in the image. A single subject is expected to have a small distribution while an image with a cluttered background would have a large distribution. An edge in computer vision is defined as high rate of change between neighboring pixels. Edges in this feature are detected by applying a 3x3 Laplacian filter and taking the absolute value. The Laplacian is defined as the sum of the second derivatives in the x and y directions $\Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$. The filter is applied to each RGB channel independently and the final image is computed as the mean across all three channels. The Laplacian image is resized to 100x100 and normalized to sum to 1. Then, the edges are projected onto the x and y axis independently. Let w_x, and w_y be the width of 98% of the projected edges centered on the center of mass of each projection respectively. The image quality feature $f = 1 - \frac{w_x w_y}{100}$ is the percent of area outside the majority of edges; low values of f represent a cluttered scene. Figure 3.5 shows the edges detected from two different images and their respective feature value.

3.2.2 Hue Count

Professional photographs look more colorful and vibrant, but actually tend to have less distinct hues because cluttered scenes contain many heterogeneous objects. We use a hue count feature by filtering an image in HSV color space. For this feature, only pixels with V in the range of [0.15, 0.95] and S greater than 0.2 are considered. A 20 bin histogram is computed on the remaining H values. Let m be the maximum value of the histogram and let $N = \{i | H(i) > \alpha m\}$, be the set of bins values greater than αm. The quality feature $f = 20 - |N|$ is 0 when there are a many different hues and larger as the number of distinct hues in the image goes down. We used the same $\alpha = 0.05$ as in the original work.

3.2.3 Contrast and Lightness

Brightness is a well known variable that professional photographers are trained to understand and adjust. We use an average brightness feature[5;12] computed from the L channel of the Lab color



(a)

(b)

Figure 3.5: The Laplacian image for computing spacial edge distribution for two images. The feature for figure a. is 0.013 and for b. is 0.30.

space. Contrast is similar, and is the ratio of maximum and minimum pixel intensities. We sum the RGB level histograms, and normalize it to sum to 1, and then take the width of the center 98% of the histogram[5]; larger values mean more contrast.

3.2.4 Blur

Blurry images are almost always considered to be of poor quality. We use two blur features. In Ke et al.[5] blur is modeled as $I_b = G_\sigma * I$ where I_b is the result of convolving a Gaussian filter with an image. The larger the σ the more high frequencies are removed from the image. Assuming the frequency distribution of all I is approximately the same, then the maximum frequency can be estimated from $C = \{(u, v) \mid \|FFT(I_b)\| > \Theta\}$ as $|C|$. The feature is $f = |C| \sim 1/\sigma$, after normalizing by the image size.

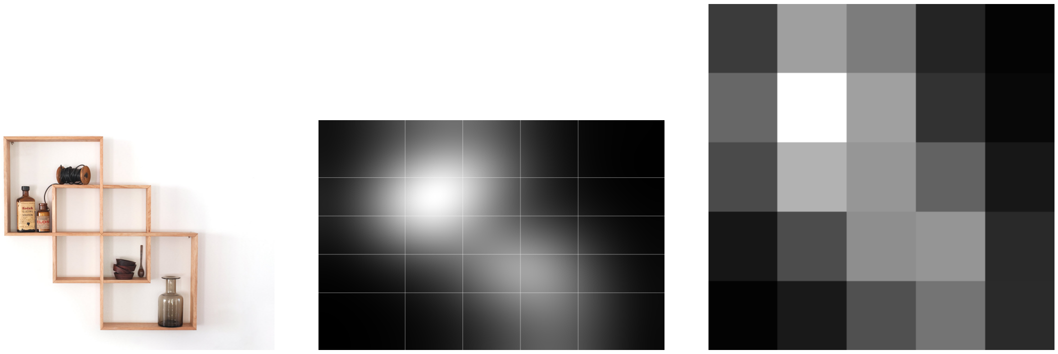
In Tong et al.[19], blur estimation is done based on changes in the edge structures. The blur operation will cause gradual edges to lose sharpness. Assuming that most images have gradual edges that are sharp enough, the blur is measured as the ratio of gradual edges that have lost their sharpness.

3.3 Rule of Thirds

The rule of thirds is an important composition technique. Thirds lines can be visualized as the horizontal and vertical lines that divide an image into a 3x3 grid of equal sized cells. The rule of thirds states that subjects placed along these lines are aesthetically more pleasing and more natural than subjects centered in the photograph. In order to segment the subject of the image from the background, we use the Spectral Residual saliency detection algorithm[20]. The feature is a 5x5 map where each cell is the average saliency value[21]. Let w_p be the saliency value of the pixel and $A(W_i)$ is the area of the cell, then the value of each cell is

$$w_i = \frac{1}{A(W_i)} \sum_{p \in W_i} w_p. \tag{3.1}$$

To compute the feature, the image is divided into a 5x5 grid with emphasis on the thirds lines; the horizontal and vertical regions centered on the thirds lines are 1/6 of the image size. Figure 3.6



(a) (b) (c)

Figure 3.6: Example of Rule of Thirds feature. Figure b. shows the SR saliency detection, and c. shows the thirds map feature.

shows the saliency detection with the 5x5 grid overlay, and the thirds map feature for an image.

3.4 Texture

A smooth image may indicate blur or out-of-focus, and the lack of which may indicate poor film, or too high an ISO setting. In contrast, texture in the scene is an important composition skill of a photographer. Smoothness may indicate the lack of texture. Texture and smoothness are some of the most statically correlated features with predicting image quality/popularity [10;13]. We use three smoothness/texture features from Wang et al. [13] and Khosla et al. [10].

A three level wavelet transform is applied to the L channel of the Lab color space. We only use the bottom level of the pyramid. The result is squared to indicate power. Let $b = \{HH, HL, LH\}$ be the bottom level of a wavelet transform, the feature is

$$f = \frac{1}{3MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{b} w^b(m, n) \tag{3.2}$$

where w is the square of the wavelet value. Because the Laplacian is often used as a pyramid of different scales, another feature

$$f = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} l(m, n) \tag{3.3}$$

is also used. This time l is the second level from the bottom of a Laplacian pyramid.

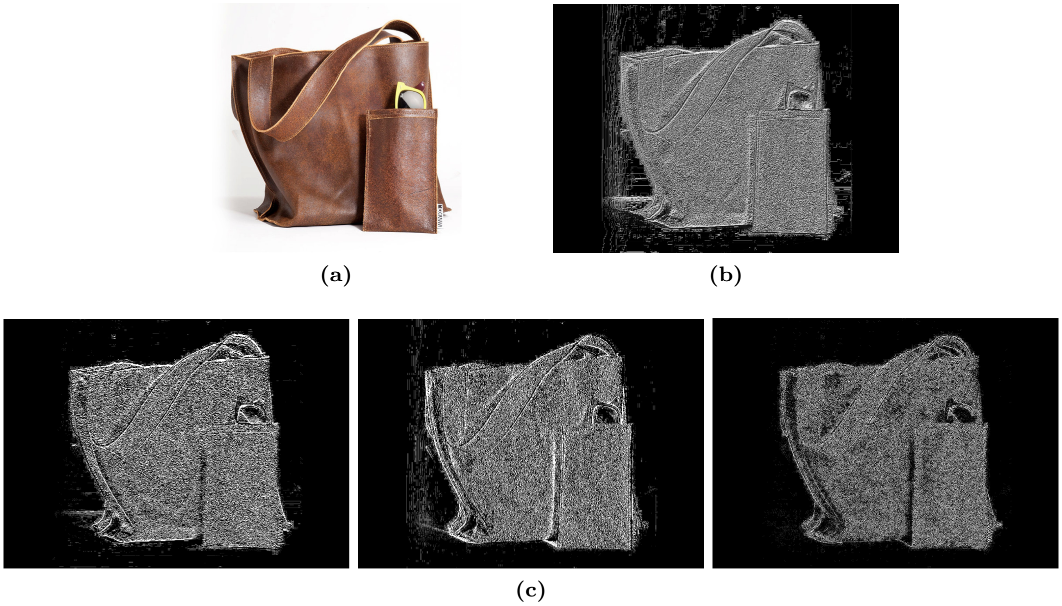


Figure 3.7: Smoothness and texture features. Figure b. shows Local Binary Pattern (LBP) feature image, and c. shows the 3 channels of the DB1 wavelet transform.

Another texture feature is computed using local binary pattern (LBP). Then a pyramid of histograms are computed as first described by Lazebnik et al.[22]. Figure 3.7 shows the similarities of LBP features and the three channels of Daubechies db1 wavelet.

3.5 Depth of Field

Depth of field is the distance between between the nearest and farthest objects that appear in sharp focus. A technique of professional photographers is to use low depth of field to focus on the photographic subject while blurring the background. We used the feature first described by Datta et al.[6] of the ratio of high frequency detail in center regions of the image compared to the entire image. Let w be the bottom level of a wavelet transform, the feature is

$$f = \frac{\sum_{(x,y)\in M_6 \cup M_7 \cup M_{10} \cup M_{11}} w(x,y)}{\sum_{i=1}^{16} \sum_{(x,y)\in M_i} w(x,y)}, \tag{3.4}$$

where $\{M_i | 1 \leq i \leq 16\}$ are the cells of a 4x4 grid. The same feature is also reapplied using the Laplacian pyramid l instead of the wavelet transform pyramid w[13]. These features only look at

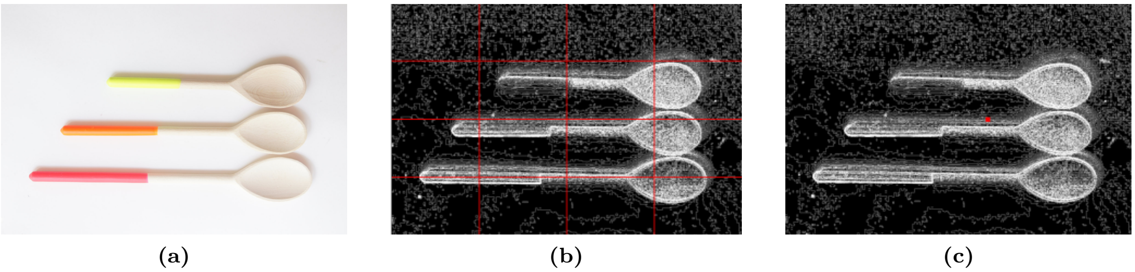


Figure 3.8: Figure b. shows the Low Depth of Field features in the center grid region for the Laplacian image. Figure c. shows the same image with its center of mass.

the center region of the image. A third feature looks at the spacial distribution of high frequency details[13]. Let l be the bottom layer of a Laplacian pyramid and c_{row}, c_{col} be the center of mass; the feature is defined as

$$f = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N} l(m,n)\sqrt{(m-c_{row})^2+(n-c_{col})^2}. \tag{3.5}$$

Figure 3.8 visualizes how these features are computed for an image.

3.6 Text

Maximally Stable Extremal Regions (MSER)[23] can be used to detect text because characters are typically single solid colors with sharp edges that standout from the background[24]. Additionally, texture patterns are also often detected by MSER, like bricks on a wall. We used the experimental feature of the count of the number of MSER regions. We would like to continue this experiment into other features based on text in images.

3.7 Chapter Summary

This chapter presented the image features used to predict popularity. First it introduced basic image concepts such as in-memory image representation, color spaces, simple filters, line detection, scale space pyramids, and frequency analysis. Using these basic concepts, we then discussed the individual features such as spacial edge distribution, blur, rule of thirds, and texture. In the next chapter we present the details of the classification algorithm using these features, and discuss the results.

Chapter 4: Experiment

So far, we have been discussing general concepts and ideas around building classifiers; machines that automatically qualify unseen data as belonging to some category. This chapter details our popularity classifier, the data, feature extraction pipeline, and the results of our experiment.

4.1 Data

Etsy has over 35 million items for sale, called listings. Each listing has one or more images for displaying different angles and details, and associated with each listing image is its display order. The first ordered image is taken to be the main image for the listing and the one that we use to extract image features. In addition, each image has a number of different sizes and resolutions available. We used the full size images which are guaranteed to be no wider than 1500px and variable height. The full size images provide the best detail for the feature extraction pipeline.

The images are an important component, but they are also basically useless as training data without the corresponding class labels. A common approach is to use Amazon Mechanical Turk[1] or other crowd sourcing platform where people are paid to label the data. This is not without challenges; whether an item strikes a chord with a user is subjective and the users that end up labeling the data may not be a good representation of the e-commerce site's users. Etsy has solved this problem by building an internal mechanism to infer class labels for listings based on historical user data. Let L be the listing with main image I. As an e-commerce site, items that have higher sales are more popular than those that don't. Each listing's number of purchases $P(L_I)$ is the first criteria in our label. Etsy has a favorites feature that allows users to favorite items that they like; to save for later or find others like it. The number of times each listing has been favorited $F(L_I)$ is the second criteria in our label. In additional to both of these, Etsy also tracks the number of clicks each listing has $C(L_I)$, this data is the third criteria in the label. The last criteria is the number of views $V(L_I)$.

[1] https://www.mturk.com/mturk/welcome

Using the labeled training data, we extracted the features and trained the classifier. The features we use are discussed in more detail in Chapter 3. Here we briefly outline the pipeline used to process the images. Etsy provides a developer API to access listings and images. We used the API to build a pipeline to download listing images and extracted the features shown in Chapter 3. We used over 50,000 listings for two experiments. The images were downloaded in sets of 10,000 images per set. Then each set was run through the feature extractor using 10-parallel subsets of 1,000 images. Once all the sets were extracted they were combined into the final extracted features. Overall, the feature extraction took less than 5 hrs on a 16 core machine.

4.2 Classifier

Given the extracted popularity features $\mathbf{x}$ for a listing, we would like to predict $y \in \{1, 0\}$ whether the listing is popular 1 or not 0. To do this we use logistic regression which as mentioned in Chapter 2 is a discriminative model for classification that returns the probability of the class label,

$$p(t = 1|\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}).$$

To use this, we first have to learn the model parameters $\mathbf{w}$ such that the probability of the classifier predicting the true labels is maximized. The likelihood of the true labels $\mathbf{L} = \langle l_1, \ldots, l_n \rangle$ being predicted given a choice of $\mathbf{w}$ is given by the equation,

$$p(\mathbf{L}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{l_n} \{1 - y_n\}^{1-l_n}.$$

This is equivalent to minimizing the negative log error function $-\log p(\mathbf{L}|\mathbf{w})$. The choice of a logarithmic error function provides a nice property in which the sigmoid function cancels out in the partial derivative with respect to $\mathbf{w}$,

$$\Delta E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\mathbf{x}_n.$$

The error function is convex, so the parameters of this model can be solved with the help of the gradient function using stochastic gradient decent, iterative reweighted least squares, or similar technique.

4.3 Evaluation

The results presented in the next section are evaluated using common analytic metrics. No metric is perfect, therefore these metrics have been chosen to provide a good range of information. Testing the model on the same data that is was trained on does little but show that it remembered. Therefore, the data is split into training and testing parts. However, it is still possible that the data is split in such a way that the results are no good. To solve this, we introduce the k-fold cross validation technique. K-fold cross validation is a technique to average the results of a number of runs over different training and testing sets. In k-fold cross validation, the data is split into k sets where each set takes a turn as the testing set on a model trained using the $k-1$ remaining sets. The reported result is the average over all k runs. All metrics reported below are computed using 10-fold cross validation in order to better generalize the results.

For classification, the basic metrics are precision and recall. The precision is the ratio of true positives to all predicted positives $\frac{\text{tp}}{\text{tp+fp}}$. For a binary classifier such as the one presented in this paper, it is a measure of how much to trust the predicted positive labels. On the other hand, recall is a measure of the completeness of the predicted positive labels. It is the ratio of true positives to all correct positives $\frac{\text{tp}}{\text{tp+fn}}$. The f-measure is a weighted average of the two, $\frac{2*\text{precision}*\text{recall}}{\text{precision+recall}}$.

Similar to precision and recall, are the true and false positive rates. The true positive rate (TPR) is the percent of positives that were predicted as positive, and the false positive rate (FPR) is the percent of negatives that were predicted as positive. These are particularly useful to plot against each other. Our logistic regression classifier returns the probability of the binary class label. In order to assign a class label, a decision threshold needs to be applied; for example a positive label is assigned when the probability is greater than 50%. The choice of this threshold may change the accuracy of the classifier. In order to visualize this relationship complementary Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC) metrics are used. The ROC is a visualization and the AUC is a numerical number that describes it. The ROC plots the relationship between the true and false positives rates with respect to the decision threshold. The area under the curve is a quantitative measure of this curve. This also visualized how separable the data is.

So far these metrics have been solely based on binary labels, however, our popularity score can take on a range of values between 0 and 1 that we also use as the probability of being popular. Since logistic regression returns the predicted probability we can directly compare its accuracy to these values. A common accuracy measurement is Mean Squared Error (MSE),

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

Because MSE doesn't weight all errors equally, in particular errors below 1 get smaller when squared, we also use the Mean Absolute Error (MAE),

$$\text{MAE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i|.$$

4.4 Results

We ran two experiments to evaluate the results of the classifier using the image quality features; one with image features alone, and the other in a multimodal setting using existing Etsy text features.

The first experiment utilized the public Etsy API to evaluate the image quality features to predict popularity. We collected 3305 listings from Etsy trying to keep an equal distribution of popular, non-popular, old and new listings. From the public API we created a popularity score $Popularity(L_I) = F(L_I) + V(L_I)$from the number of times the listing has been favorited and the number of views. This score is normalized to be between 0 and 1 and is a baseline popularity probability score. In order to assign the true binary labels we threshold the popularity score by the median which in the dataset was 0.00099. This created a set of 1670 positive labels.

Using this data we trained a logistic regression model and evaluated it using 10-fold cross validation. The results are shown in tables 4.1, 4.2. As shown in the ROC curve in Figure 4.1 the results show that image quality features are statistically significant in predicting popularity, and therefore

Table 4.1: Precision, recall, and F-measure for logistic regression classifier with image only features. Both 0.5 and median thresholding are shown.

	Precision	Recall	F1
0.5	0.518	0.516	0.517
median	0.519	0.654	0.578

Table 4.2: Mean squared error and mean absolute error for logistic regression classifier with image only features.

MSE	0.476
MAE	0.502

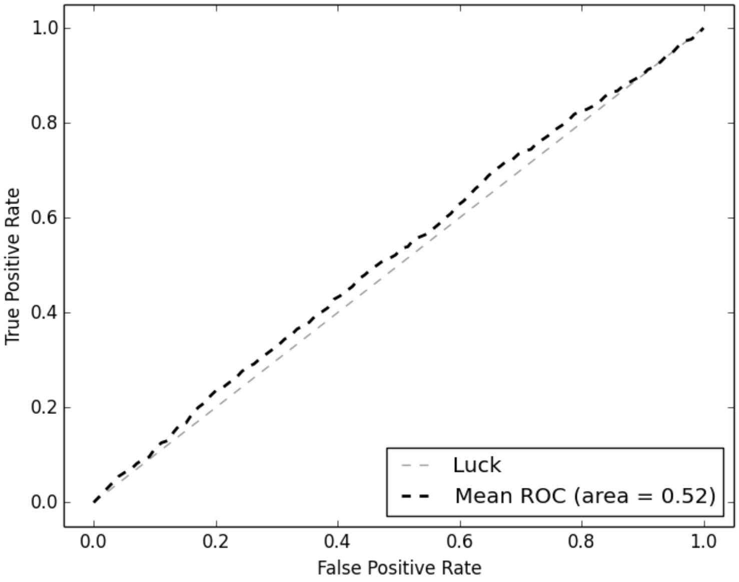
may have complementary information for multimodal classification.

The second experiment[25] tested whether in a multimodal setting a classifier using image quality features in addition to text features would perform better than text features alone. We used text features already employed by Etsy such as listing title and description unigrams and bigrams. This experiment also used logistic regression with 10-fold cross validation but the dataset contained 50,000 listing images. We were able to use additional information proprietary to Etsy such as purchase information, making our popularity score $Popularity(L_I) = F(L_I) + C(L_I) + P(L_I)$. We normalized it the same as in the first experiment. The results are shown in figure 4.3.

Table 4.3: Lift in accuracy rate using a logistic regression, relative to text-only baseline (%), on the sample dataset is shown in image-only and multimodal settings.

Modality	Image	Image+Text (MM)
Relative lift in AUC	+1.07%	+**3.45**%

This chapter presented the results of our experiments on predicting the popularity of e-commerce listings use a classifier built with image quality features. Using data from the e-commerce site Etsy, we created a true popularity score for each listing. Then we built a pipeline to extract the image quality features described in Chapter 3, and learned the parameters for a logistic regression model. We showed that the combination of logistic regression and image quality features has statistical significance in popularity prediction when used alone, and improves classification accuracy when used in a multimodal setting with existing listing text features.



(a)

Figure 4.1: ROC curve and AUC metric

Chapter 5: Conclusion

In this thesis, we have shown that the popularity of items for sale on e-commerce sites can be predicted using image quality features extacted from images of the items for sale. These features provide complementary information to traditional text only features used by sites' search, ranking, and recomendation systems. This is important because there is an overwhellmingly large number of items for sale online. Making sense of all of this data to connect users with what they want to buy is critical for an e-commerce site.

The image quality features we used capture concepts that professional photographers look for when evaluating the quality of a photograph, such as spacial edge distribution, light, color, rule of thirds, texture, smoothness, blurriness, depth of field, and scene composition. Using basic image concepts such as in-memory image representation, color spaces, simple filters, line detection, scale space pyramids, and frequency analysis, we extracted these features from images of items for sale.

The task of predicting popularity is a classification problem in the field of machine learning. Using the image quality features extracted from our Etsy dataset, we trained a Logistic Regression classifier to predict popularity. We defined the popularity score for a listing as the sum of the number of times it has been clicked, favorited, or purchased.

Using 10-fold cross validation, we showed that the classifier is statistically significant when predicting item popularity using image quality features. Additionally, when these features are added to a popularty classifier using features from text metadata only, the accuracy is improved.

We would like to extend this work in the future. One thought is to use state of the art CNN image features or high level semantic image features. There has been recent work on capturing image style. Predicting style might be a useful tool for online e-commerce as well, and would follow a similar experiment design.

Bibliography

[1] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[2] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[3] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 215–242, 1958.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

[5] Yan Ke, Xiaoou Tang, and Feng Jing. The design of high-level features for photo quality assessment. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 419–426. IEEE, 2006.

[6] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Studying aesthetics in photographic images using a computational approach. In *Computer Vision–ECCV 2006*, pages 288–301. Springer, 2006.

[7] Phillip Isola, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. What makes an image memorable? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 145–152. IEEE, 2011.

[8] Sagnik Dhar, Vicente Ordonez, and Tamara L Berg. High level describable attributes for predicting aesthetics and interestingness. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1657–1664. IEEE, 2011.

[9] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, Fabian Nater, and Luc Van Gool. The interestingness of images. 2013.

[10] Aditya Khosla, Atish Das Sarma, and Raffay Hamid. What makes an image popular? In *Proceedings of the 23rd international conference on World wide web*, pages 867–876. ACM, 2014.

[11] Kamelia Aryafar, Corey Lynch, and Josh Attenberg. Exploring user behaviour on etsy through dominant colors. In *2014 22nd International Conference on Pattern Recognition (ICPR)*, pages 1437–1442. IEEE, 2014.

[12] Ming Chen and Jan Allebach. Aesthetic quality inference for online fashion shopping. In *IS&T/SPIE Electronic Imaging*, pages 902703–902703. International Society for Optics and Photonics, 2014.

[13] Jianyu Wang and Jan Allebach. Automatic assessment of online fashion shopping photo aesthetic quality. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2915–2919. IEEE, 2015.

[14] Kamelia Aryafar and Ali Shokoufandeh. Fusion of text and audio semantic representations through cca. In *Multimodal Pattern Recognition of Social Signals in Human-Computer-Interaction*, pages 66–73. Springer, 2014.

[15] Kamelia Aryafar and Ali Shokoufandeh. Multimodal sparsity-eager support vector machines for music classification. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 405–408. IEEE, 2014.

[16] Kamelia Aryafar and Ali Shokoufandeh. Multimodal music and lyrics fusion classifier for artist identification. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 506–509. IEEE, 2014.

[17] Corey Lynch, Kamelia Aryafar, and Josh Attenberg. Images don't lie: Transferring deep visual semantic features to large-scale multimodal learning to rank. *arXiv preprint arXiv:1511.06746*, 2015.

[18] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

[19] Hanghang Tong, Mingjing Li, Hongjiang Zhang, and Changshui Zhang. Blur detection for digital images using wavelet transform. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 1, pages 17–20. IEEE, 2004.

[20] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[21] Long Mai, Hoang Le, Yuzhen Niu, and Feng Liu. Rule of thirds detection from photograph. In *Multimedia (ISM), 2011 IEEE International Symposium on*, pages 91–96. IEEE, 2011.

[22] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[23] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[24] Huizhong Chen, Sam S Tsai, Georg Schroth, David M Chen, Radek Grzeszczuk, and Bernd Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2609–2612. IEEE, 2011.

[25] Stephen Zakrewsky, Kamelia Aryafar, and Ali Shokoufandeh. Item popularity prediction in e-commerce using image quality feature vectors. *arXiv preprint arXiv:1605.03663*, 2016.