Estimating Intended Color from its Name

Lyndon White, Roberto Togneri, Wei Liu,
Mohammed Bennamoun
lyndon.white@research.uwa.edu.au, roberto.togneri@uwa.edu.au,
wei.liu@uwa.edu.au,
mohammed.bennamoun@uwa.edu.au
The University of Western Australia. 35 Stirling Highway, Crawley, Western Australia

2018-06-11

Abstract

When a speaker says the name of a color, the color that they picture is not necessarily the same as the listener imagines. Color is a grounded semantic task, but that grounding is not a mapping of a single word or phrase to a single point in color-space. Proper understanding of color language requires the capacity to map a sequence of words to a probability distribution in color-space. A distribution is required as there is no clear agreement between people as to what a particular color describes – different people have a different idea of what it means to be "very dark orange".

Learning how each word in a color name contributes to the color described, allows for knowledge sharing between uses of the words in different color names. This knowledge sharing significantly improves predicative capacity for color names with sparse training data. The extreme case of this challenge in data sparsity is for color names without any direct training data. Our model is able to predict reasonable distributions for these cases, as evaluated on a held-out dataset consisting only of such terms.

1 Introduction

Color understanding is an important subtask in natural language understanding. It is a challenging domain, due to ambiguity, multiple roles taken by the same words, the many modifiers, and shades of meaning. Due to its difficulty, texts containing color descriptions such as `the flower has petals that are bright pinkish purple with white stigma` are used as demonstrations for state of the art image generation systems (**??**). The core focus of the work we present here is addressing these linguistic phenomena around the descriptions of the color, in a single patch, as represented

in a color-space such as HSV (**?**). Issues of illumination and perceived color based on visual context are considered out of the scope.

2 Related Work

The understanding of color names has long been a concern of psycholinguistics and anthropology (**????**). It is thus no surprise that there should be a corresponds field of research in natural language processing.

The earliest works revolve around explicit color dictionaries. This includes the ISCC-NBS color system (**?**) of 26 words, including modifiers, that are composed according to a context free grammar such that phrases are mapped to single points in the color-space; and the simpler, non-compositional, 11 basic colors of **?**. Works including **??????** which propose methods for the automatic mapping of colors to and from these small manually defined sets of colors. We note that **??** both propose systems that discretize the color-space, though to a much courser level than we consider in this work.

More recent works, including the work presented here, function with much larger number of colors, larger vocabularies, and larger pools of respondents. In particular making uses of the large Munroe dataset **?**, as we do here. This allows a data driven approach towards the modelling.

? and **?** present color naming methods, mapping from colors to to their names, the reverse of our task. These works are based on defining fuzzy rectangular distributions in the color-space to cover the distribution estimated from the data, which are used in a Bayesian system to non-compositionally determine the color name. **?** maps a point in the color-space, to a sequence of probability estimates over color terms. They extends beyond, all prior color naming systems to produce a compositional color namer based on the Munroe dataset. Their method uses a recurrent neural network (RNN), which takes as input a color-space point, and the previous output word, and gives a probability of the next word to be output – this is a conditional language model. In this work we tackle the inverse problem to the creation of a conditional language model. Our distribution estimation models map from a sequence of terms, to distribution in color space. Similarly, our point estimation models map from sequence of terms to single point in color-space.

? propose another compositional color naming model. They use a per-character RNN and a variational autoencoder approach. It is in principle very similar to **?**, but functioning on a character, rather than a word level. The work by Kawakami et al. also includes a method for generating colors. However they only consider the generation of point estimated, rather than distributions. The primary focus of our work is on generating distributions. The datasets used by Kawakami et al. contain only very small numbers of observations for each color name (often just one). These datasetsare thus not suitable for modelling the distribution in color space as interpreted by

the population. Further, given the very small number of examples they are not well suited for use with word-based modelling: the character based modelling employed by Kawakami et al. is much more suitable. As such we do not attempt comparison to their work.

? presents a neural network solution to a communication game, where a speaker is presented with three colors and asked to describe one of them, and the listener is to work out which is being described. Speaker and listener models are trained, using LSTM-based decoders and encoders respectively. The final time-step of their model produces a 100 dimensional representation of the description provided. From this, a Gaussian distributed score function is calculated, over a high dimensional color-space from **?**, which is then used to score each of the three options. While this method does work with a probability distribution, as a step in its goal, this distribution is always both symmetric and unimodal – albeit in a high-dimensional color-space.

The generation of color from text has not received a signficant ammount of attention in prior work. In particular the generation of probability distributions in color space, to our knowledge has not been considered at all. Conversely, there has been several works on the reverse problem: the generation of a textual name for a color from color space point. The work presented here closed that gap.

Consider that the word `tan` may mean one of many colors for different people in different circumstances: ranging from the bronze of a tanned sunbather, to the brown of tanned leather; `green` may mean anything from `aquamarine` to `forest green`; and even `forest green` may mean the rich shades of a rain-forest, or the near grey of the Australian bush. Thus the color intended cannot be uniquely inferred from the color name. Without further context, it does nevertheless remain possible to estimate likelihoods of which colors are be intended based on the population's use of the words. The primary aim of this work is to map a sequence of color description words to a probability distribution over a color-space. This is required for a proper understanding of color language. We also consider the more basic point estimation of colors, though it's value is questionable.

Proper understanding requires considering *the color intended* as a random variable. In other words, a color name should map to a distribution, not just a single point or region. For a given color name, any number of points in the color-space could be intended, with some being more or less likely than others. Or equivalently, up to interpretation, it may intend a region but the likelihood of what points are covered is variable and uncertain. This distribution is often multimodal and has high and asymmetrical variance, which further renders regression to a single point unsuitable. We do produce results point estimate results for interest in **??**, however for any form of precise work the use of such systems is limited A single point estimate, does not capture the nature of the problem adequately. The mean of a multimodal distribution (one with two peaks) will lie in the valley between

the – a less likely color. Similarly it will be off to the side of the mode, in an asymmetrical distribution Thus whi

We estimate a probability distribution over the color-space. To qualify our estimate of the distribution we discretize the space into a large number of patches, and produce an output much like a histogram. This allows us to take advantage of the well-known softmax based methods for estimating a probability mass distribution using a neural network.

Estimating color probabilities has a clear use as a subsystem in many systems. For example, in a human-interfacing system, when asked to select the `dark bluish green` object, each object can be ranked based on how likely its color is according to the distribution. This way if extra information eliminates the most-likely object, the second most likely object can immediately be determined. Further, if the probability of the color of the object being described by the user input is known, a threshold can be set to report that no object is found, or to ask for additional information. More generally, the distribution based on the color name alone can be used as a prior probability and combined with additional context information to yield better predictions.

3 Method

3.1 Tokenization

For all the term based methods, we perform tokenization. Tokenization

3.1.1 HSV color-space

We use the HSV color-space (**?**). through-out this work. We use the format which the data is originally provided in. In this format hue, saturation and value all range between zero and one. Unlike many other colors spaces (CIELab, Luv etc.) the gamut is square. This regular space means that errors on all channels can be considered equally, and unlike other color spaces. The scaling of hue in particular to be between zero and one (like the other channels) makes the modelling task more consistent as were the hue to range between 0 and 2π it would be over-weighted compared to the other channels.

When working with the hue, all measures need to take into account the wrap-around effect. When ever we refer to mean squared error, mean or mode on the HSV space in this paper, we are referring to the angularly corrected forms given in Section 3.2.

Unlike the RGB color space, the HSV channels do correspond to how humans perceive colors. However, it is not designed to be a perceptionally uniform color space across hue (unlike CIELab), which does suggest using mean squared error for the point estimates is not optimal. However, given

the other issues outlined above with point estimates we do not judge this a major concern. There are no such issues for our distribution estimations.

One of it's important advantages over other color spaces is that it best meets the assumption that for a given color name each channel is statistically independent.

3.2 Angularly Correct Calculations on HSV

When performing calculations with the HSV color space it is important to take into account that hue is an angle. As we are working with the color space regularized to range between zero and one for all channels, this means a hue of one and a hue of zero are equivalent (in radian's this would be 0 and 2π).

The square error of two hue values is thus calculated as

$$SE(h_1, h_2) = \min\left((h_1 - h_2)^2, (h_1 - h_2 - 1)^2 \right) \tag{1}$$

(using radians the -1 would be -2π) to take into account the error could be calculated clockwise or counter clockwise.

The mean of a series of N hue values is calculated as

$$\bar{h} = \text{atan2}\left(\frac{1}{N} \sum_{i=1}^{i=N} \sin(h_i), \frac{1}{N} \sum_{i=1}^{i=N} \cos(h_i) \right) \tag{2}$$

As we are using the regularized angle, we use the regularized versions of the trigonometric functions (though this shift is no different from degrees to radians).

4 Distribution Estimation

4.1 Conditional Independence Assumption

We make the assumption that given the name of the color, the distribution of the H, S and V channels are independent. That is to say, it is assumed if the color name is known, then knowing the value of one channel would not provide any additional information as to the value of the other two channels. The same assumption is made, though not remarked upon, in **?** and **?**. This assumption of conditional independence allows considerable saving in computational resources. Approximating the 3D joint distribution as the product of three 1D distributions decreases the space complexity from $O(n^3)$ to $O(n)$ in the discretized step that follows.

Superficial checks were carried out on the accuracy of this assumption. Spearman's correlation on the training data suggests that for over three quarters of all color names, there is only weak correlation between the channels (Q3 = 0.187). However, this measure underestimates correlation for

values that have circular relative value, such as hue. HSV had the lowest correlation by a large margin of the 16 color-spaces evaluated. Full details, including the table of correlations, are available in supplementary materials. These results are suggestive, rather than solidly indicative, on the degree of correctness of the conditional independence assumption. We consider the assumption sufficient for this investigation.

4.2 Discretization

For distribution estimation, our models are trained to output histograms. By making use of the conditional independence assumption Section 4.1, we output one histogram per channel.

4.3 Kernel-Density Based Smoothing

We make use of the **?**

5 Experimental Setup

5.1 Implementation

The implementation of the CDEST and baseline models was in the Julia programming language (**?**). The full implementation is included in the supplementary materials. can be downloaded from the GitHub repository.[1] It makes heavy use of the MLDataUtils.jl[2] and TensorFlow.jl,[3] packages. the latter of which we enhanced significantly to allow for this work to be carried out.

5.2 Common Network Features

Dropout(**?**) is used on all layers, other than the embedding layer, with threshold of 0.5 during training. The network is optimized using Adam **?**, using a learning rate of 0.001. Early stopping is checked every 10 epochs using the development dataset. Distribution estimation methods are trained using full batch (where each observation is a distribution) for every epoch. Point Estimation trains using randomized mini-batches of size 2^16 observations (which are each color-space triples). All hidden-layers, except as otherwise precluded (in side the convolution, and in the penultimate layer of the point estimation networks) have the same width 300, as does the the embedding layer.

[1] Implementation source is at `https://github.com/oxinabox/ColoringNames.jl`
[2] MLDataUtils.jl is available from `https://github.com/JuliaML/MLDataUtils.jl`
[3] TensorFlow.jl is available from `https://github.com/malmaud/TensorFlow.jl`

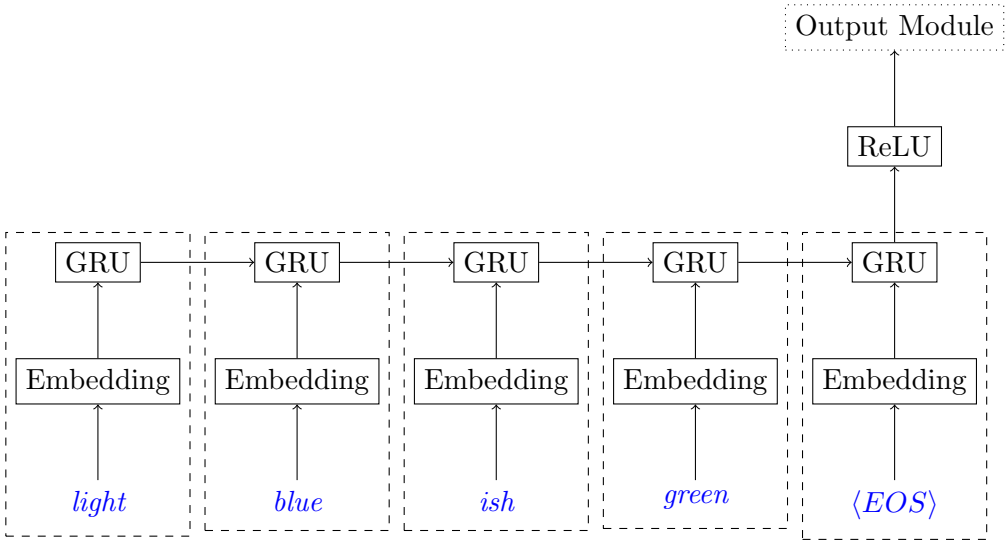

Figure 1: The RNN Input module for the example input `light greenish blue`. Each dashed box represents 1 time-step.

5.2.1 Embeddings

All our neural network based solutions incorporate an embedding layer. This embedding layer maps from tokenized words to vectors. We make use of 300d pretrained FastText embeddings **?**[4].

The embeddings are not trained during the task, but are kept fixed. As per the universal approximation theorem (**??**) the layers above allow for arbitrary non-linear continuous transformation.

5.3 Input Modules

5.3.1 Recurrent Neural Network(RNN)

A Recurrent Neural Network is a common choice for this kind of task, due to the variable length of the input. The general structure of this network, shown in **??** is similar to **?**, or indeed to most other word sequence learning models. Each word is first transformed to an embedding representation. This representation is trained with the rest of the network allowing per word information to be efficiently learned. The embedding is used as the input for a Gated Recurrent Unit (GRU) (**?**). The output of the last time-step is fed to a Rectified Linear Unit (ReLU) (**?**).

[4]Available from `https://fasttext.cc/docs/en/english-vectors.html`

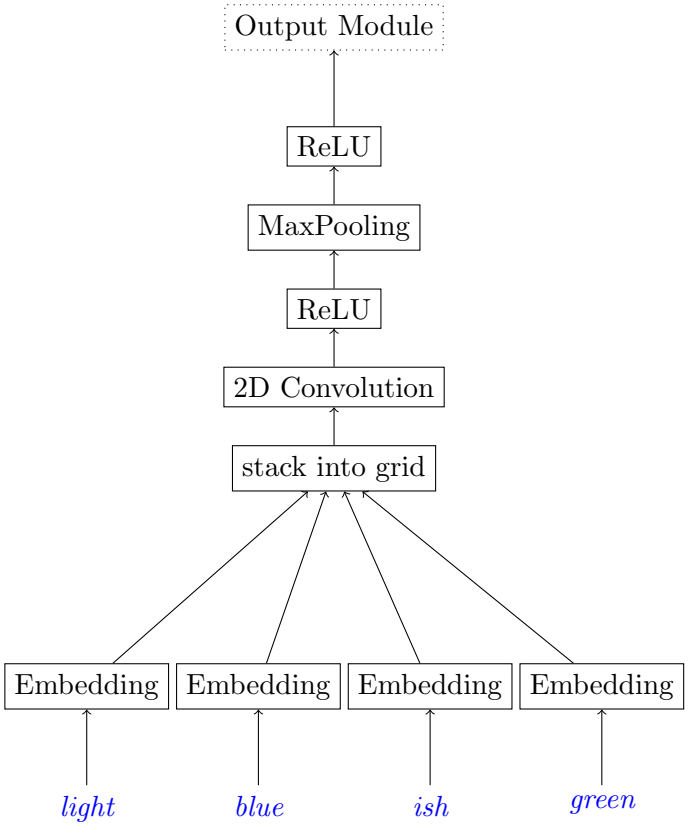

Figure 2: The SOWE input module for the example input `light greenish blue`

5.3.2 Sum of Word Embeddings (SOWE)

Using a simple sum of word embeddings as a layer in a neural network is less typical than an RNN structure. Though it is well established as a useful representation, and has been used an input to other classifiers such as support vector machines. Any number of word embeddings can be added to the sum. However, it has no representation of the order. The structure we used is shown in Figure 3.

5.3.3 Convolutional Neural Network(CNN)

We apply a convulutional neural network to the task by applying 2D convolution over the stacked word embeddings. **??** We use 64 filters of size between 1 and the length of the longest padded embedding (5).

5.3.4 Non-term based Baseline

To baseline the performance of our models we propose

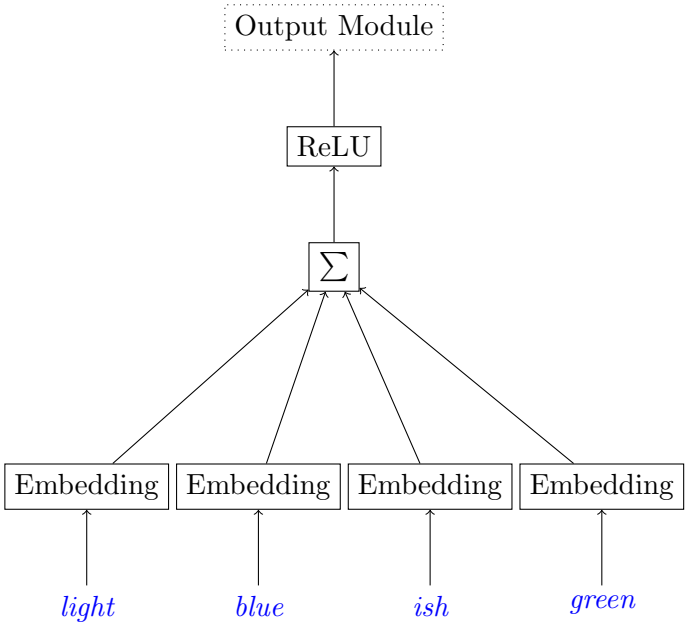

Figure 3: The SOWE input module for the example input `light greenish blue`

Histogram Baseline (Distribution Estimation, only)

Mean Squared Error Baseline (Point Estimation, only)

5.4 Datasets

5.4.1 Full Training and Testing set

We make use of the Munroe dataset as prepared by **?** from the results of the XKCD color survey. The XKCD color survey (**?**), collected over 3.4 million observations from over 222,500 respondents. McMahan and Stone take a subset from Munroe's full survey, by restricting it to the responses from native English speakers, and removing very rare color names with less than 100 uses. This gives a total of 2,176,417 observations and 829 color names. They also define a standard test, development and train split.

5.4.2 Extrapolation Training and Testing Set

The primary goal in constructing using the term based models is to be able to make predictions for never before seen descriptions of colors. For example, based on the learned understanding of `salmon` and of `bright`, from examples like `bright green` and `bright red`, we wish for the systems to make predictions about `bright salmon`, even though that description

never occurs in the training data. To evaluate this generalisation capacity, we define an extrapolation sub-dataset for both testing and training. This is defined by selecting the rarest 100 color descriptions from the full dataset, with the restriction that every token in a selected description must still have at least 8 uses in other descriptions in the training set. The selected examples include multi-token descriptions such as: `bright yellow green` and also single tokens that occur more commonly as modifiers than as stand-alone descriptions such as `pale`.

The extrapolation training set is made up of the data from the full training set, excluding those corresponding the the rare descriptions. Similar is done for the development set, so as no direct knowledge of the combined terms can leak during early-stopping. Conversely, the extrapolation test set is made up of only the observations from the full test set that do use those rare descriptions.

By training on the extrapolation training set and testing on the extrapolation test set, we can assess the capacity of the models to make predictions for color descriptions not seen in training. A similar approach was used in `?`. We contrast this to the same models when trained on the full training set, but tested on the extrapolation test set, to see how much accuracy was lost.

5.5 Order Testing set

It is known that the order of words in a color description to some extent matters. `greenish brown` and `brownish green` are distinct, if similar, colors. To assess the models on there ability to make predictions when order matters we construct the order testset. This is a subset of the full test set containing only descriptions with terms that occur in multiple different orders. There are 76 such descriptions in the full dataset. Each of which has exactly one alternate ordering. This is unsurprising as while color descriptions may have more than 2 terms, normally one of the terms is a joining token such as `ish` or `-`.

5.6 Output Modules

5.6.1 Distribution Estimation

For all the distribution estimation systems we investigate here, we consider training both on the binned-data, and on the smoothed data (as described in Section 4.3). Making use of the conditional independence assumption (see Section 4.1), we output the three discretized distributions. This is done using 3 softmax output layers.

The output module for distribution estimation

Contrasting to estimating continuous conditional distributions, estimating a discrete conditional distributions is a significantly more studied appli-

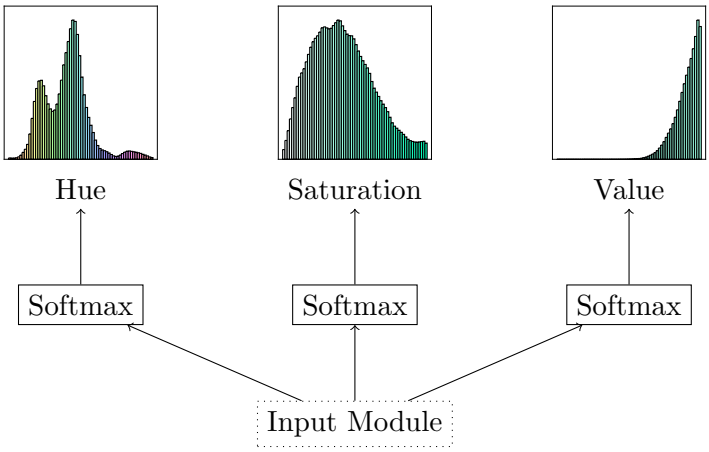

Figure 4: The Distribution Output Module

cation of neural networks – this is the basic function of any softmax classifier. To simplify the problem, we therefore transform it to be a discrete distribution estimation task, by discretizing the color-space. Discretization to a resolution of 64 and 256 bins per channel is considered.

For the case of the machine learning models, the output is produced using softmax layers.

5.6.2 Point Estimation

Our point estimation output model for the neural network is shown in Figure 5. The hidden-layer from the top of the input module is feed to an 4 single output neurons.[5] Two of these are used the sigmoid activation function (range 0:1) to produce the outputs for the saturation and value channels. The other two use the tanh activation function (range -1:1), they produce the intermediate output that we call y_{shue} and y_{chue} for the sine and cosine of the hue channel respectively. We use these intermediate values when calculated this loss function as it results in a loss function that is continuous and correctly handles the wrap-around nature of the hue channel.

During training we use the following loss function for each observation

[5]Equivalently these 4 single neurons can be expressed as a layer with 4 outputs and 2 different activation functions.

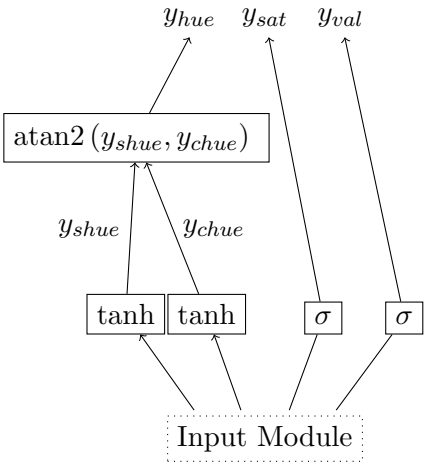

Figure 5: The Point Estimate Output Module. Here atan2 is the quadrant preserving arctangent, outputting as a regularized angle (as per in all evaluations).

$y^\star$, and each corresponding prediction y.

$$loss = \frac{1}{2}\left(\sin(y^\star_{hue}) - y_{shue}\right)^2 \tag{3}$$

$$+ \frac{1}{2}\left(\cos(y^\star_{hue}) - y_{chue}\right)^2 \tag{4}$$

$$+ \left(y^\star_{sat} - y_{sat}\right)^2 \tag{5}$$

$$+ \left(y^\star_{val} - y_{val}\right)^2 \tag{6}$$

A On the Conditional Independence of Color Channels given a Color Name

As discussed in the main text, we conducted a superficial investigation into the truth of our assumption that given a color name, the distributions of the hue, value and saturation are statistically independent.

We note that this investigation is, by no means, conclusive though it is suggestive. The investigation focusses around the use of Spearman's rank correlation. This correlation measures the monotonicity of the relationship between the random variables. A key limitation is that the relationship may exist but be non-monotonic. This is almost certainly true for any relationship involving channels, such as hue, which wrap around. In the case of such relationships Spearman's correlation will underestimate the true strength of the relationship. Thus, this test is of limited use in proving the conditional independence. However, it is a quick test to perform and does suggest that the conditional independence assumption may not be so incorrect as one might assume.

For the Monroe Color Dataset training data given by $V \subset \mathbb{R}^3 \times T$, where $\mathbb{R}^3$ is the value in the color-space under consideration, and T is the natural language space. The subset of the training data for the description $t \in T$ is given by $V_{|t} = \{(\tilde{v}_i, t_i) \in V \mid t_i = t\}$. Further let $T_V = \{t_i \mid (\tilde{v}, t_i) \in V$ be the set of color names used in the training set. Let $V_{\alpha|t}$ be the α channel component of $V_{|t}$, i.e. $V_{\alpha|t} = \left\{v_\alpha \mid ((v_1, v_2, v_3), t) \in V_{|t}\right\}$.

The set of absolute Spearman's rank correlations between channels a and b for each color name is given by $S_{ab} = \left\{\left|\rho(V_{a|t}, V_{b|t})\right| \; t \in T_V\right\}$.

Color-Space	$Q3(S_{12})$	$Q3(S_{13})$	$Q3(S_{23})$	max
HSV	0.1861	0.1867	0.1628	0.1867
HSL	0.1655	0.2147	0.3113	0.3113
YCbCr	0.4005	0.4393	0.3377	0.4393
YIQ	0.4088	0.4975	0.4064	0.4975
LCHab	0.5258	0.411	0.3688	0.5258
DIN99d	0.5442	0.4426	0.4803	0.5442
DIN99	0.5449	0.4931	0.5235	0.5449
DIN99o	0.5608	0.4082	0.5211	0.5608
RGB	0.603	0.4472	0.5656	0.603
Luv	0.5598	0.6112	0.4379	0.6112
LCHuv	0.6124	0.4072	0.3416	0.6124
HSI	0.2446	0.2391	0.6302	0.6302
CIELab	0.573	0.4597	0.639	0.639
xyY	0.723	0.5024	0.4165	0.723
LMS	0.968	0.7458	0.779	0.968
XYZ	0.9726	0.8167	0.7844	0.9726

Table 1: The third quartile for the pairwise Spearman's correlation of the color channels given the color name.

We consider the third quartile of that correlation as the indicative statistic in Table 1. That is to say for 75% of all color names, for the given color-space, the correlation is less than this value.

Of the 16 color-spaces considered, it can be seen that the HSV exhibits the strongest signs of conditional independence – under this (mildly flawed) metric. More properly put, it exhibits the weakest signs of non-independence. This includes being significantly less correlated than other spaces featuring circular channels such as HSL and HSI.

Our overall work makes the conditional independence assumption, much like n-gram language models making Markov assumption. The success of the main work indicates that the assumption does not cause substantial issues.