Seasonal Color Classification: Is ia a real thing? Can we train a model to do it from a picture?

Introduction

- Stylists sometimes use in their work the idea of classifying appeareances by color to find out which colors would suit the person the best.

- One of those classifications is the four seasons:
- Summer – muted and cool colors
- Winter – bright and cool colors
- Spring – bright and warm colors
- Autumn – muted and warm colors

Like this: typical looks of each season:

How It Is Done In Real Life

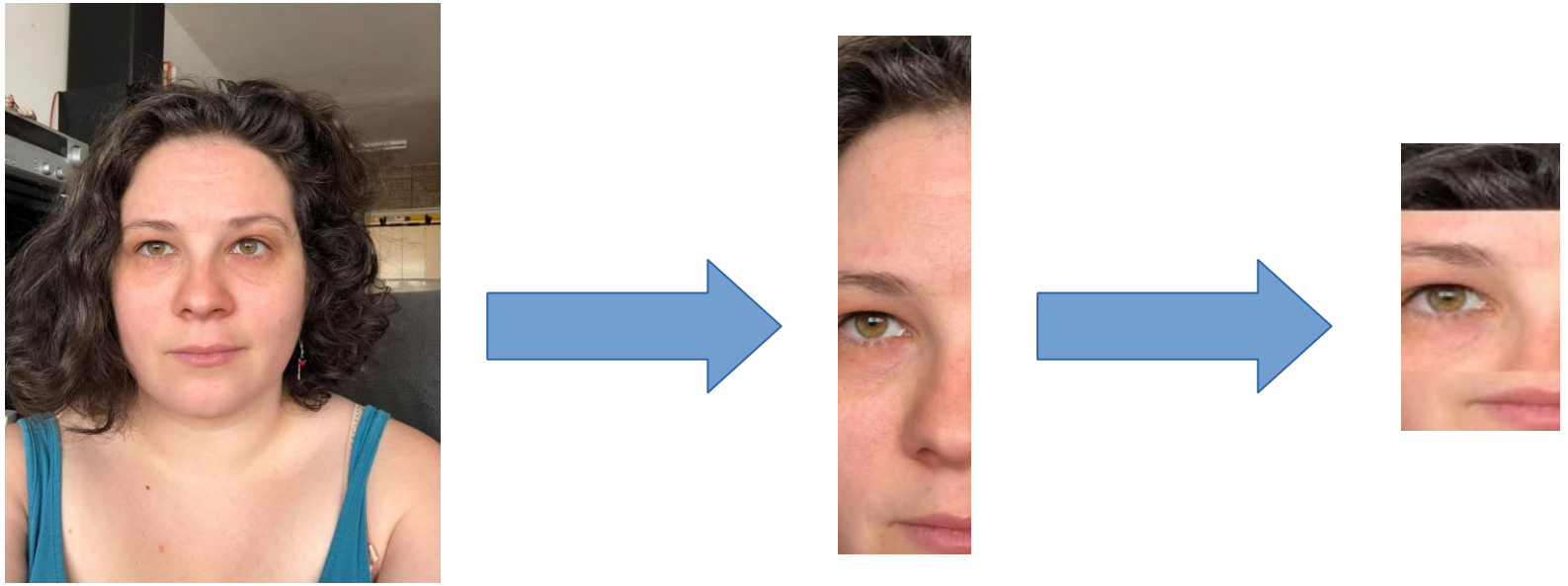
- Fabrics of different colors are places near one`s face and the peson prforming the classificatioin observes which families of colors are more flattering

Getting the data:

A friend of mine is a stylist that uses this color classificaion method.
So from her I could obtain a dataset of ~300 tagged images, which is a really small dataset for this, but I wanted to see what I can achive with this data.

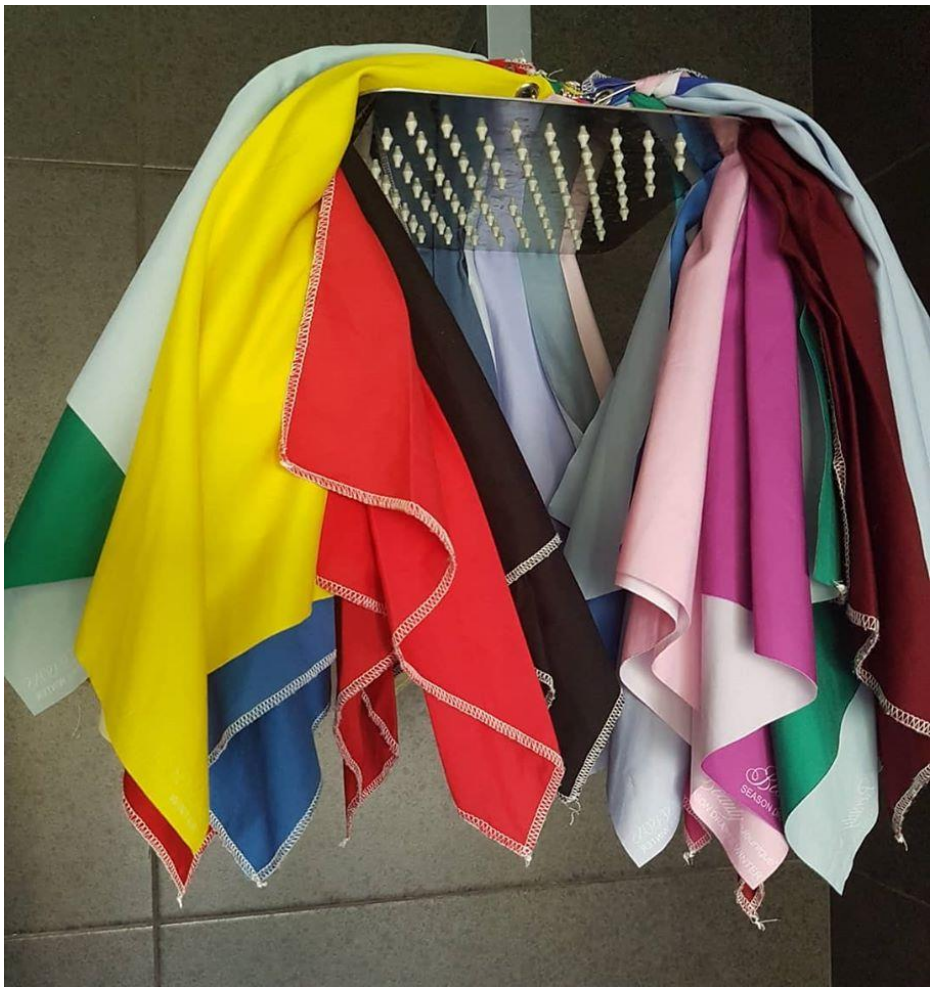
Removing irreletant parts of the image:

Initial image -> Cropped+Auto-Normalized -> Cropped again:

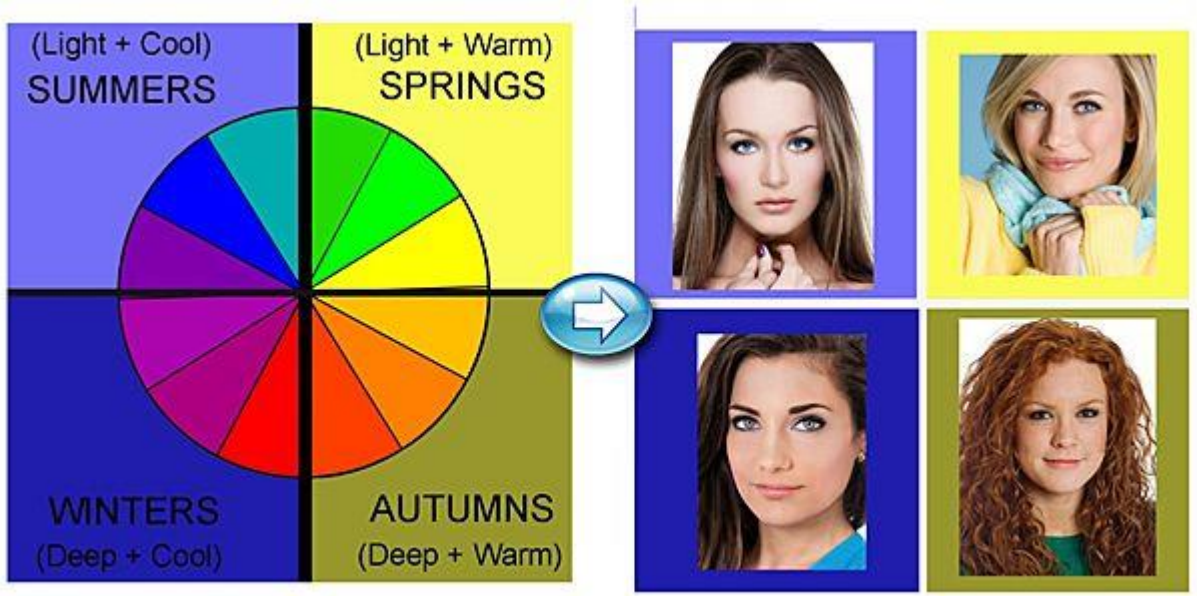


Sounds weird!

- Althougt flattering is a higly subjlective thing, I was curious to check if this classification can be done by a neural networkwhich will imply this classification criteria is real. A typical argument against this classification is that all faces have all kinds of colors in than, so this division is virtually impossible



How to classify?



2 possible approaches:

1) Two binary classifiers: one for cool vs warm, one for light vs deep (or bright vs muted)

2) One categorical classifier

First approach gives much more data in each category, but it is possible to miss color features this way.

Let`s try both.

Overcoming (somehow) small and not-so-balanced dataset:

1) Adding weights to classes – classes with smaller datasets get more weight

2) ImageDataGenerator – which allows to create more data from a given data set by modifing some of its properties, like scaling and rotating.

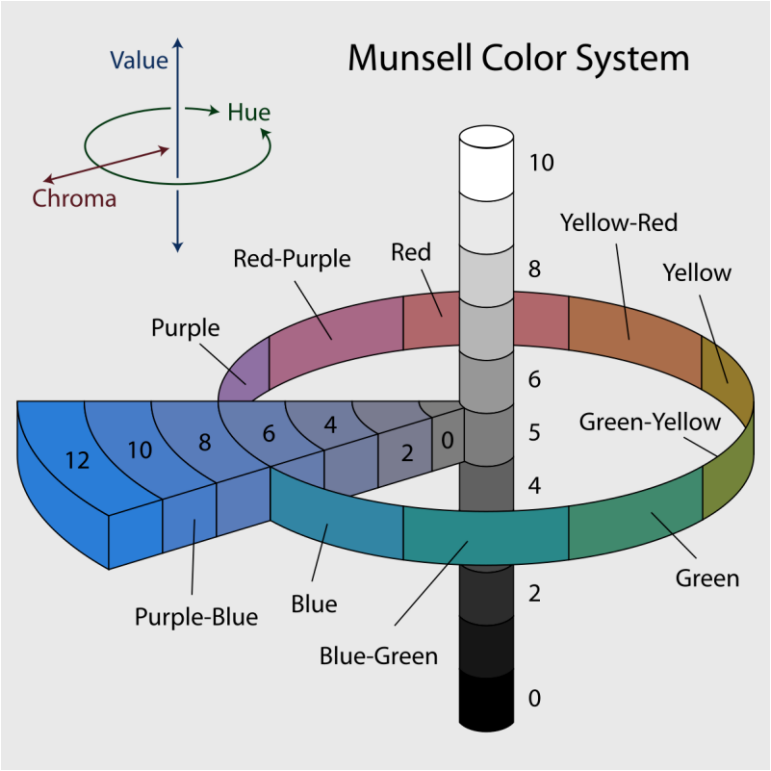
3) Transfer learning.
I found a model that was trained on a data that looked rather similar to me, but with a bigger dataset:

Rain, Sunsine, Sunrise, Cloudy:




A few words about color models:

The original theory is based on the following color model:



So it may be a good idea to use HSV (Hue, Saturation, Value) model, instead of RGB

Classification: First Approach: Two Binary classifiers:

First Attempt:

Model: CNN with three Convolution-ReLU-MaxPooling/AveragePooling block of layers.

MaxPooling layers were used for cool-warm classification, AveragePooling – for Bright-Muted

Results: Accurasy between 0.72 and 0.78

Interpretation:

Those results are better than guessing, which proves that the initial method is not based on pure guessing, but still not good enoght to be use for predictions.

Classification: First Approach: Two Binary classifiers:

Second Attempt:

FNN with two layers each consists of 1024 neurons, with histograms of the images in different color modes as inpu

Results: Accurasy between 0.6 and 0.7

Interpretation:

Those results are also better than guessing, but has no advantage oven the firts attempt

Classification: Second Approach: Categorical Classification

First Attempt:

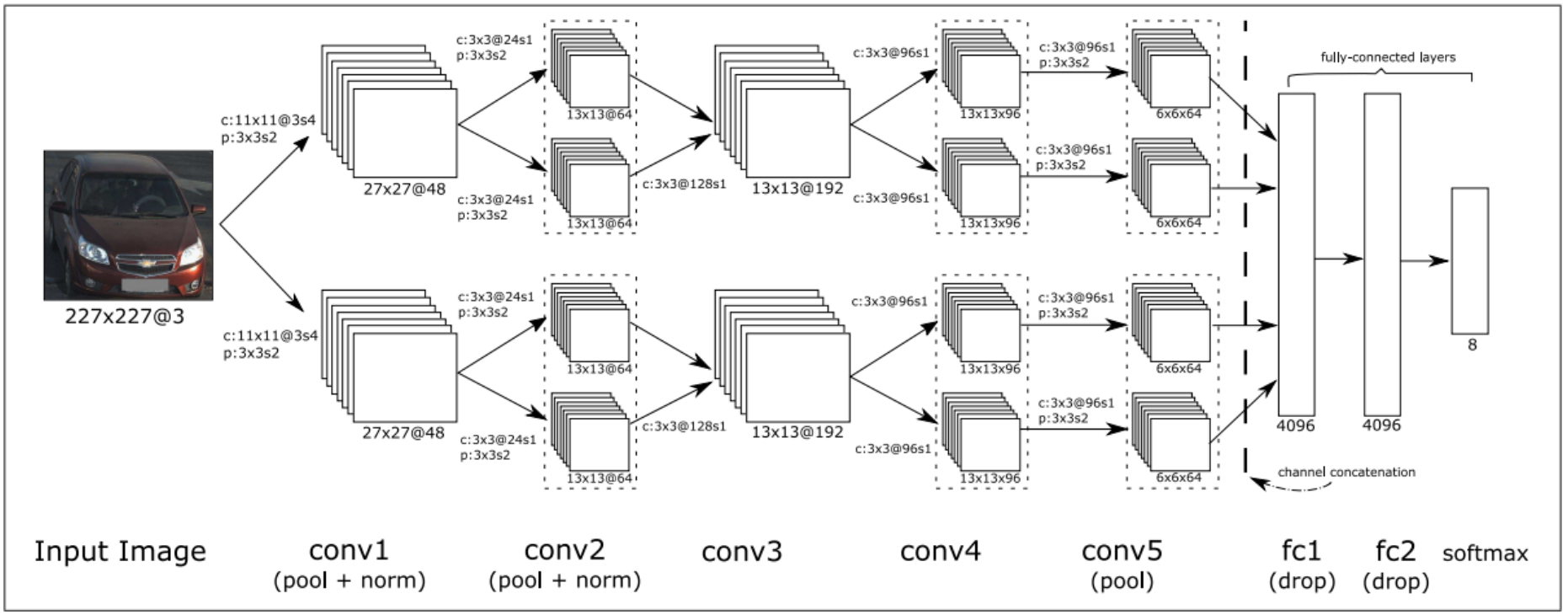
Simple CNN, resembling the ones I used for binary classificators.

Results: 0.5-0.6 accuracy. There are four categories, so this is much better than a guess, but still not good enough for making predictions

Classification: Second Approach: Categorical Classification

Second attempt:

Using a network from the paper ‘Vehicle Color Recognition Using Convolutional Neural Network’ by Reza Fuad Rachmadi and I Ketut Eddy Purnama:



Results: 0.5 accuracy.

Classification: Second Approach: Categorical Classification

Third attempt:

Using a network from the tutorial ‘Building Convolutional Neural Networks in Python using Keras’ Roman Paolucci:

```
model.add(Conv2D(32, kernel_size=3, activation='relu', input_shape=(IMAGE_HEIGHT, IMAGE_WIDTH, 3)))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(64, kernel_size=3, activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(128, kernel_size=3, activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(256, kernel_size=3, activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(4, activation='softmax'))
```

This model was created to categorize weather conditions into four categories,that resembled my data. First I tried to use the model as is.

Accuracy was 0.65.

Classification: Second Approach: Categorical Classification

Fourth Attempt:

Using a network from the tutorial ‘Building Convolutional Neural Networks in Python using Keras’ Roman Paolucci for transfer learning: first train it on its original data, and then use it for my classification data.

```
model.add(Conv2D(32, kernel_size=3, activation='relu', input_shape=(IMAGE_HEIGHT, IMAGE_WIDTH, 3)))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(64, kernel_size=3, activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(128, kernel_size=3, activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(256, kernel_size=3, activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(4, activation='softmax'))
```

I was sure this will improve the results a little, but instead, accuracy became 0.4.

Classification: Second Approach: Categorical Classification

Fifth Attempt:

FNN with two layers each consists of 1024 neurons, with histograms of the images in different color modes as input, as used for binary classification before, in different color modes.

Results:
0.82 accuracy for RGB and HSV modes, 0.74 for LAB mode.

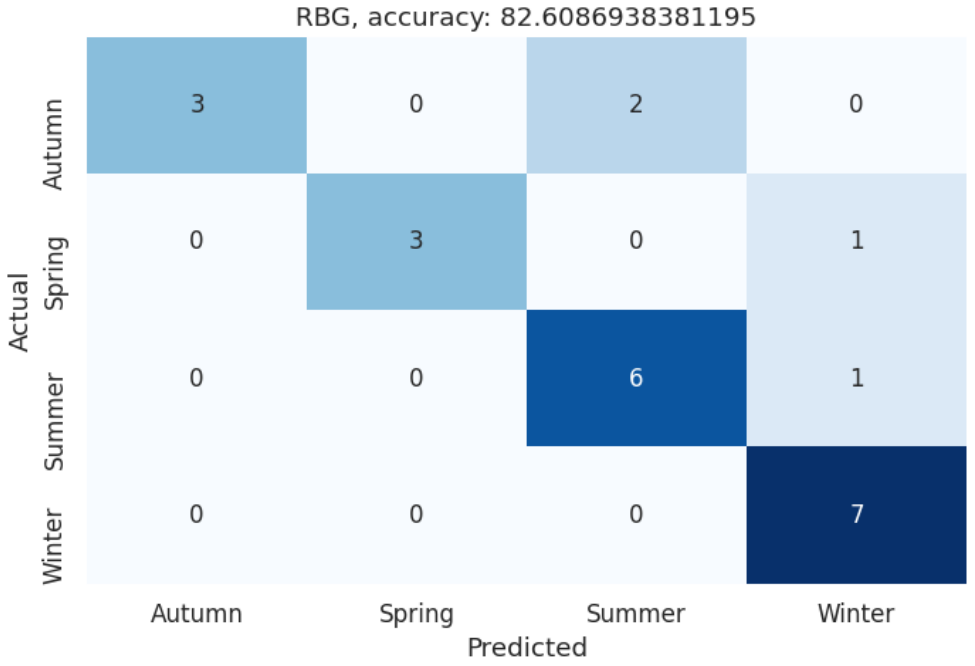
This already looks line something more practical to use. Yet all the color modes tend to misclassify Autumn as Summer

Example of the results:

/home/data_test/class_summer/summer_image_7.png => Summer (expected Summer)
/home/data_test/class_spring/spring_image_1.png => Winter (expected Spring)
/home/data_test/class_aut/aut_image_5.png => Summer (expected Autumn)
/home/data_test/class_aut/aut_image_4.png => Autumn (expected Autumn)
/home//data_test/class_winter/winter_image_1.png => Winter (expected Winter)
/home/data_test/class_winter/winter_image_3.png => Winter (expected Winter)

Classified correctly:

Cla nc :



Results and summary:

In every method I used, the classification rsults were significantly better then a simple guess, yet not good enough for actual predictions to be valuable, besided using the categorical classification with historgrams.

Although I tried many known ways to overcome the problem of having a small dataset.

Yet, getting an accuracy of 0.826 using a dataset of ~400 images convices me that if needed, given a bigger dataset (which is practically possible to obtain, a network can classify photos this was rather successfuly.

Another conclusion that dividing by brightness and temperarure produces (at least) no better results, so it is possible that when a human looks on such picture during testing, he doesn`t really takes into account each of this paramters independently.

Summary:

- This classification can be verified by Neural network.
- Prediction needs more work
- 2 binary classificators gives more or less same result as a categorical one when using CNN. FNN works much better with one categorical classifier and allows to achieve an accuracy of 0.826