Processing Images *in scale* with Go

Jyotiska NK, DataWeave
@jyotiska_nk

Image Processing

Any form of signal processing for which the input is an image, the output of image processing may be either an image or a set of characteristics or parameters related to the image. (Wikipedia)

Compression

Segmentation

Histogram

Scaling

Stitching

Enhancement

Extraction

Smoothing

Noise Reduction

Interpolation

Feature Detection

Face Detection

Edge Detection

Background / Foreground Detection

Go Libraries

- image (github.com/golang/go/src/image)
- imagick (github.com/gographics/imagick)
- go-opencv
  - github.com/hybridgroup/gobot
  - github.com/lazywei/go-opencv
- imaging (github.com/disintegration/imaging)
- resize (github.com/nfnt/resize)
- go-colorful (github.com/lucasb-eyer/go-colorful)

image

- Part of standard libraries.
- Implements a standard 2D image library.
- Supports multiple formats - JPEG, GIF, PNG.
- image/color implements the basic color library along with color palettes.
- image/color/palette provides Plan9 and WebSafe color palettes.

imagick

- Go bindings for ImageMagick C API.
- Provides various ImageMagick methods -
  - Resize
  - Grayscale
  - Tiling
  - Rotation
  - Text Effects

go-opencv

- OpenCV bindings from C APIs.
- Basic OpenCV methods -
    - Hooking up a webcam or a camera
    - Accessing frames from camera
    - Face detection
- lazywei/go-opencv provides more methods -
    - Canny Edge Detection
    - Cropping
    - Resizing

imaging

- Basic image manipulation library.
- Depends on the standard “image” library.
- Provides following functions -
  - Image encoding, decoding
  - Cropping and overlaying
  - Image flipping, rotating, transforming
  - Image blurring, sharpening
  - Image resizing

resize

- Image resizing library written in pure Go.
- Provides method to create thumbnails preserving the aspect ratio.
- Offers common interpolation methods -
    - NearestNeighbor
    - Bilinear
    - Bicubic
    - MitchellNetravali
    - Lanczos2 / Lanczos3

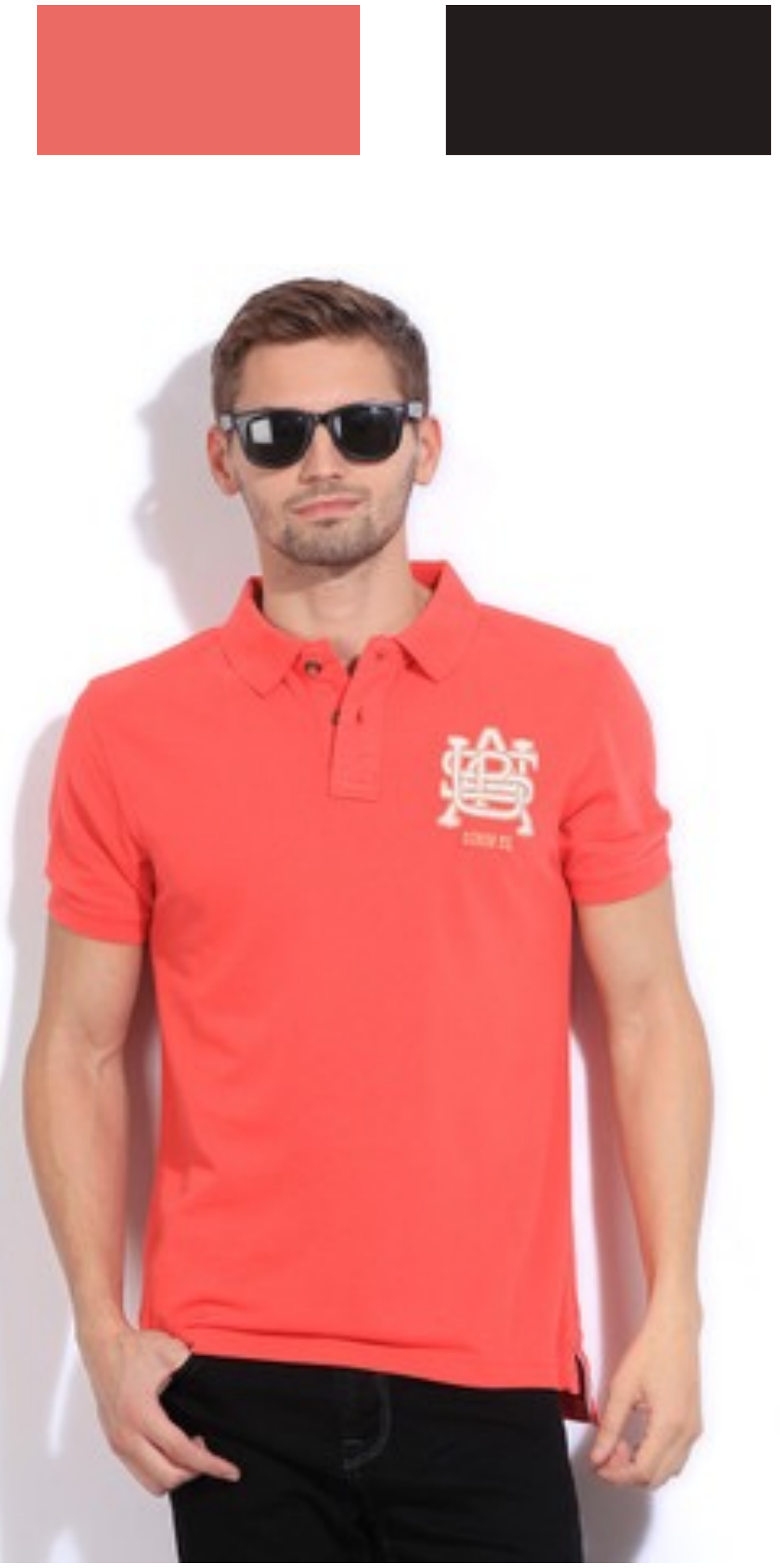
go-colorful

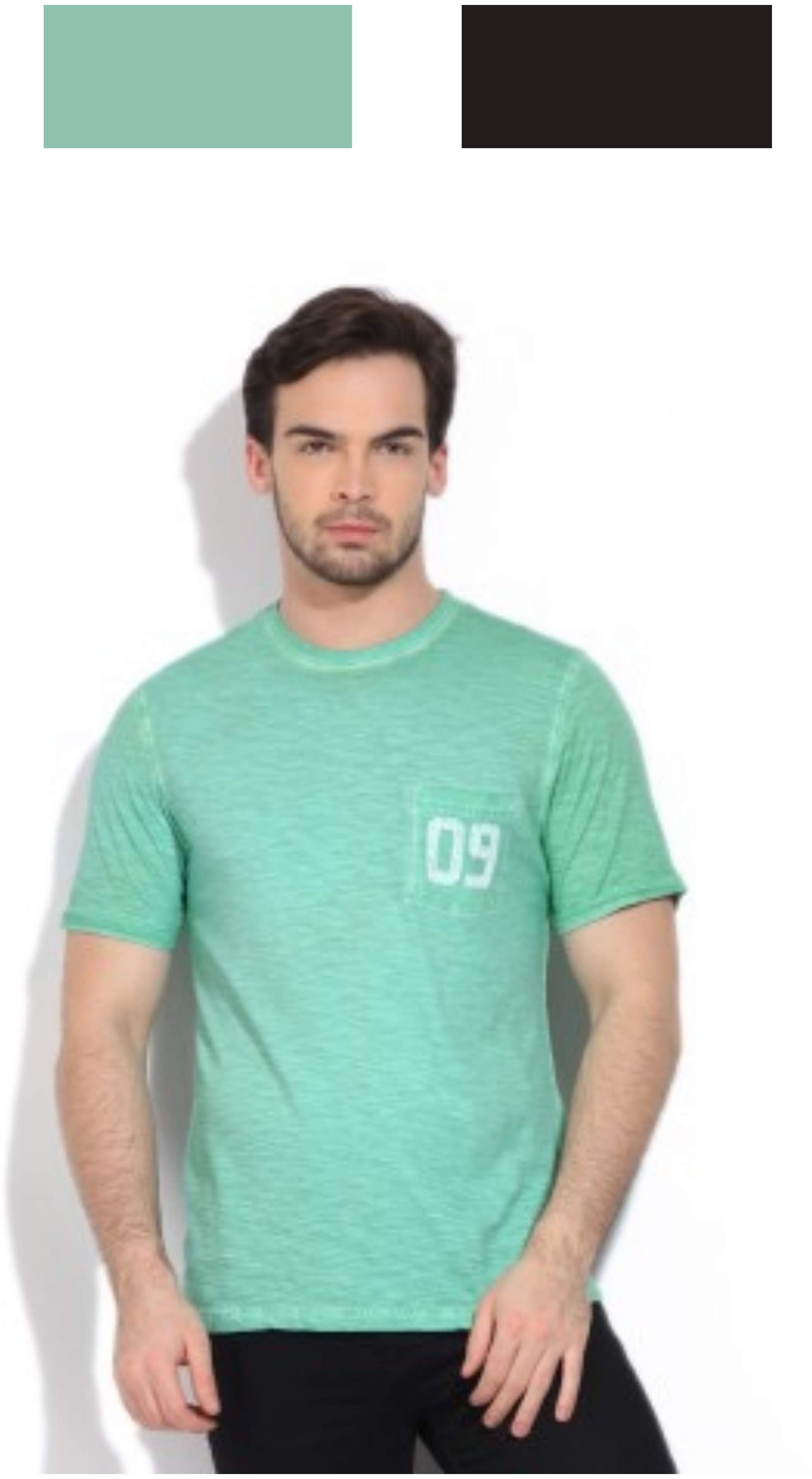
- Library for working with colors written in Go.
- Stores colors in RGB and provides methods to convert colors in different color spaces - Hex RGB, HSV, Linear RGB etc.
- Can be used to convert between color spaces, generate random colors or create color palettes.

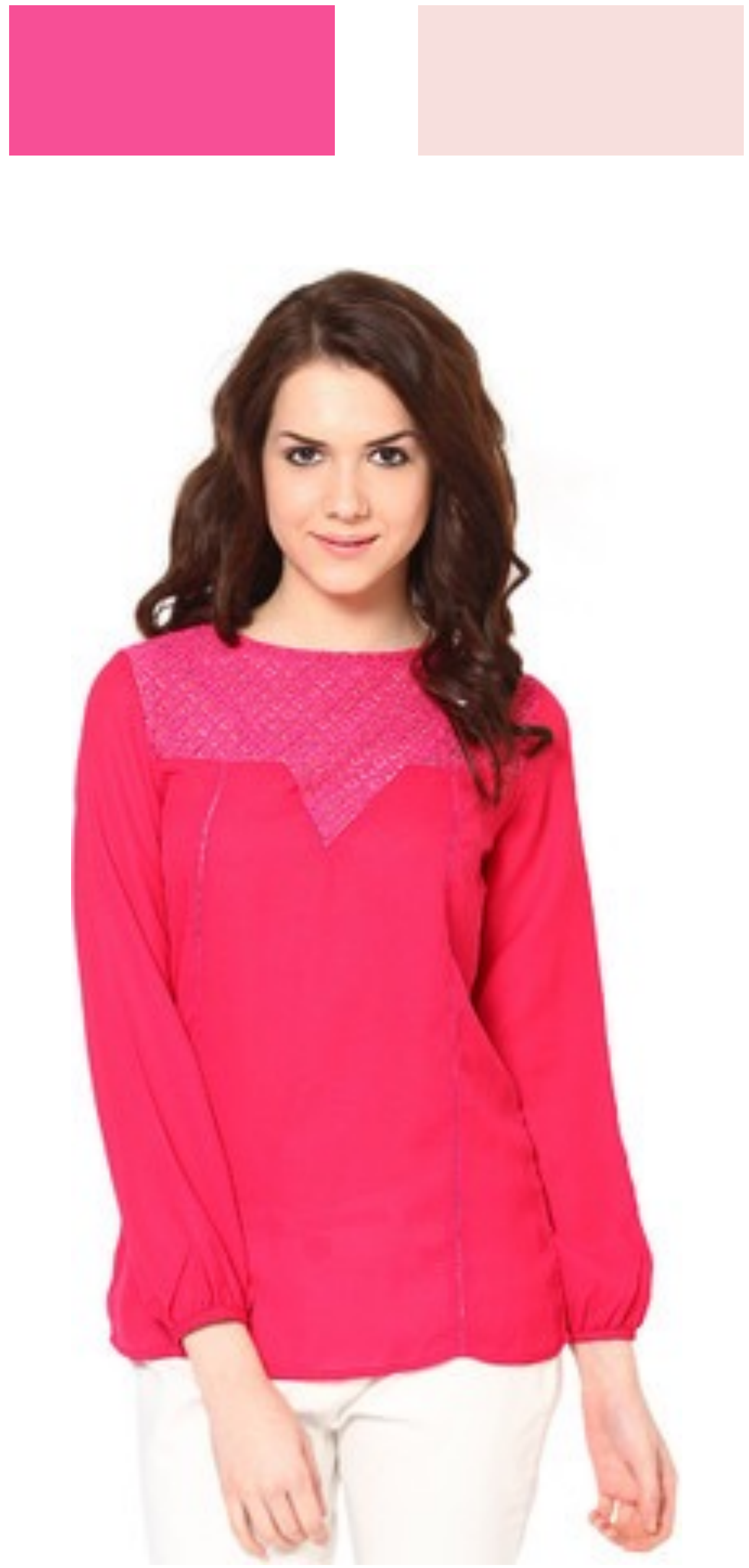
Use case at DataWeave

Extract Dominant Colors from Images

Extract dominant colors from images

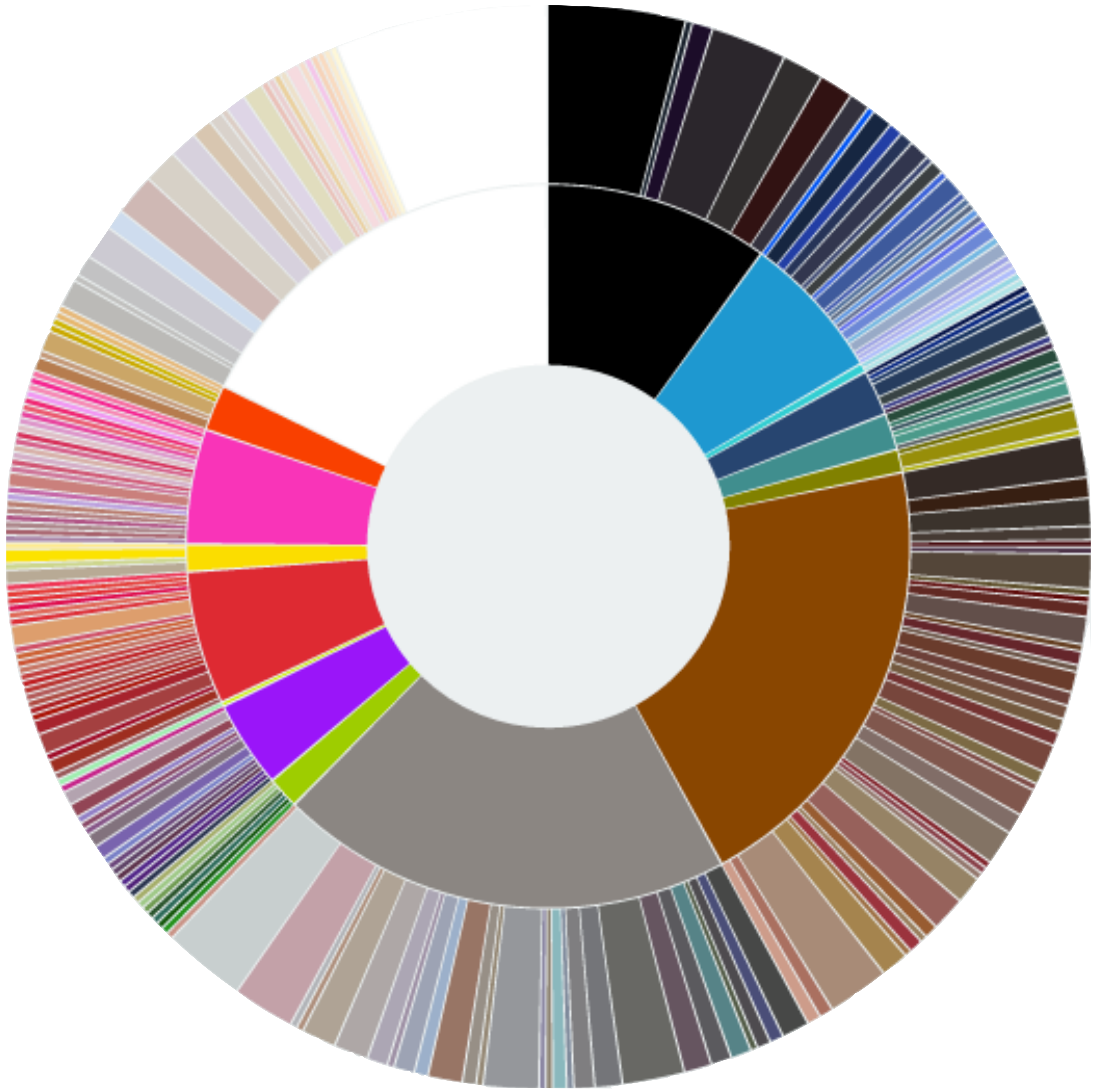




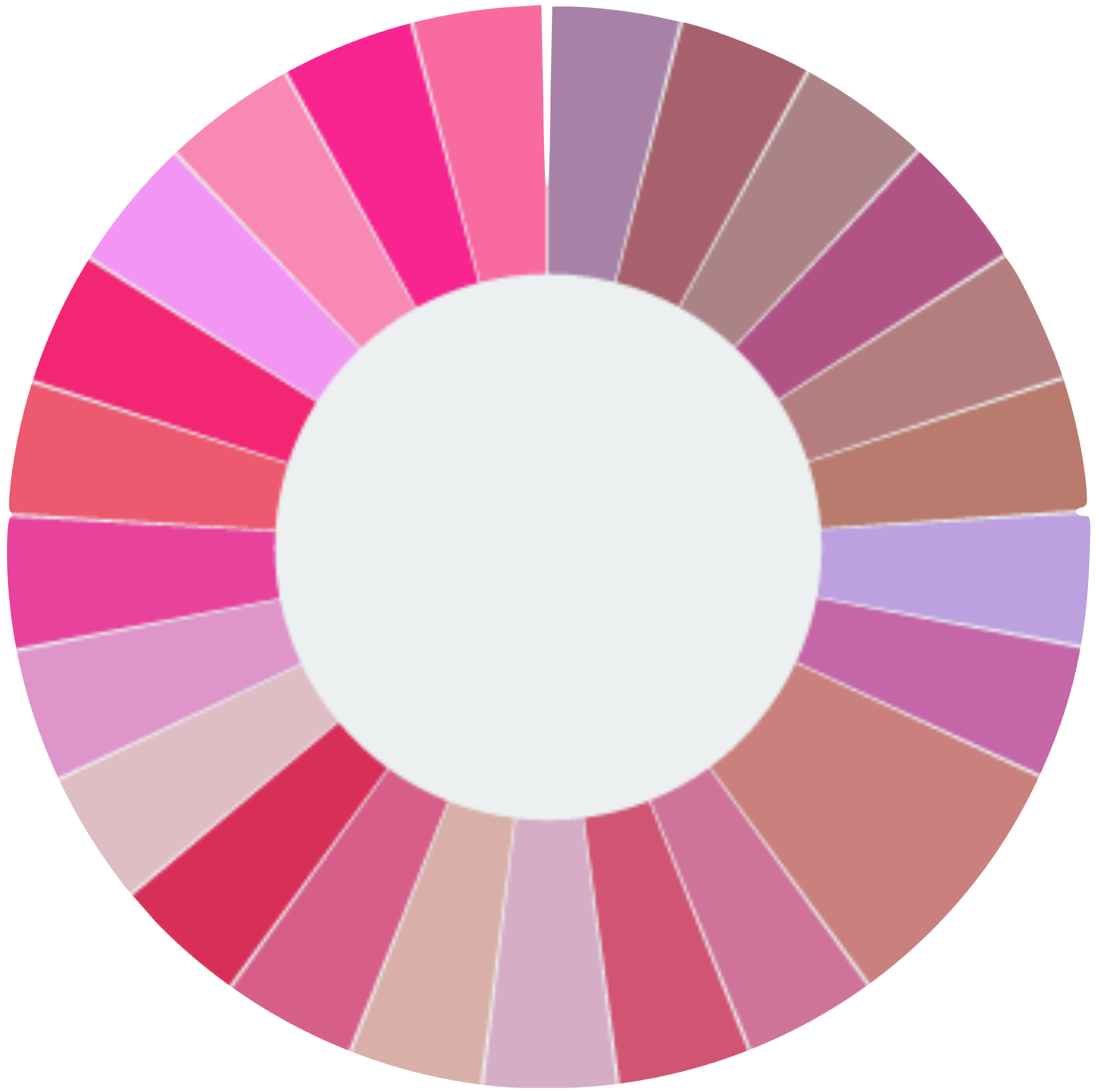


Generate Insights based on Product Colors

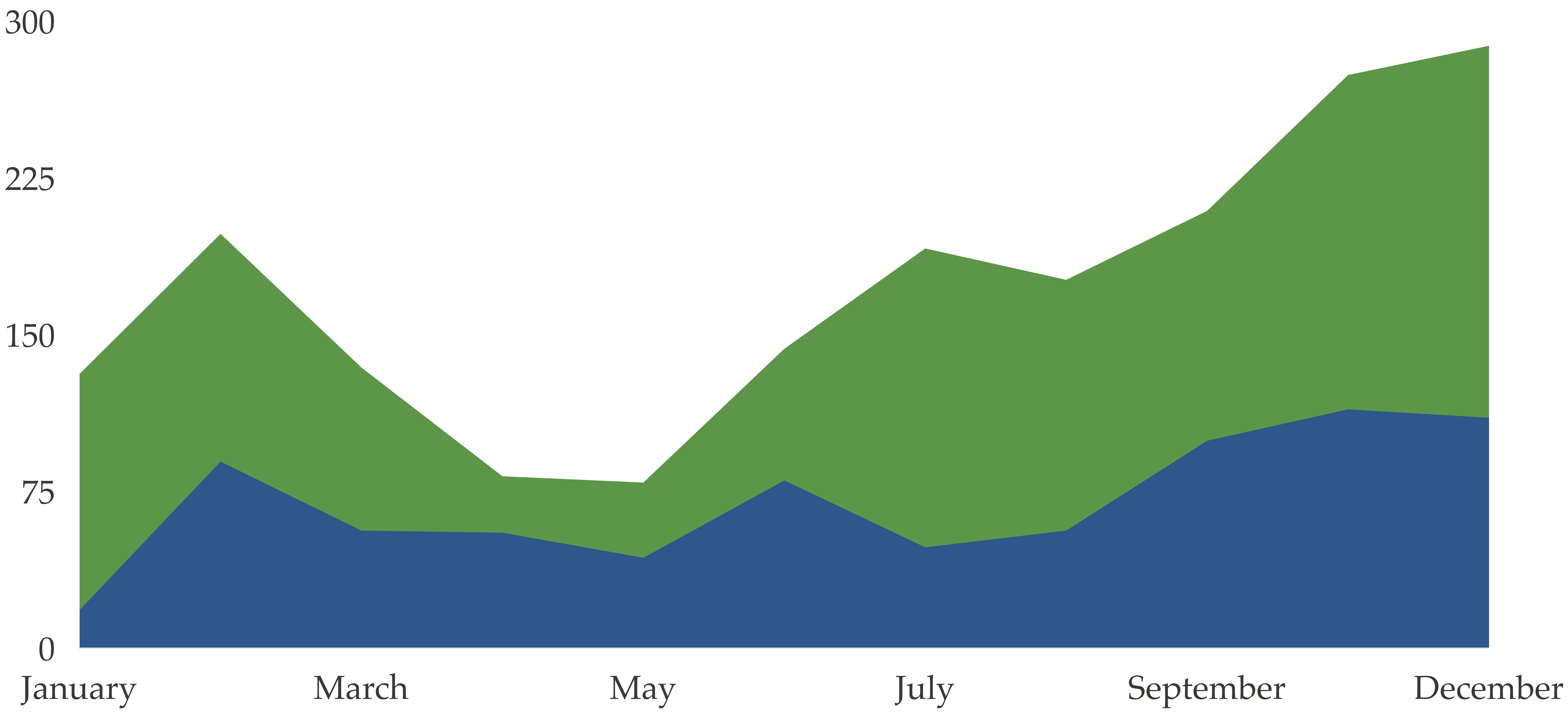
Color Distribution for *Apparels* on Store #1



Color Distribution for *Pink Tops* on Store #1



Time Series data over 12 months for Store #1



Cluster Similar Products based on Color Histogram

Clustering Similar Products across Stores

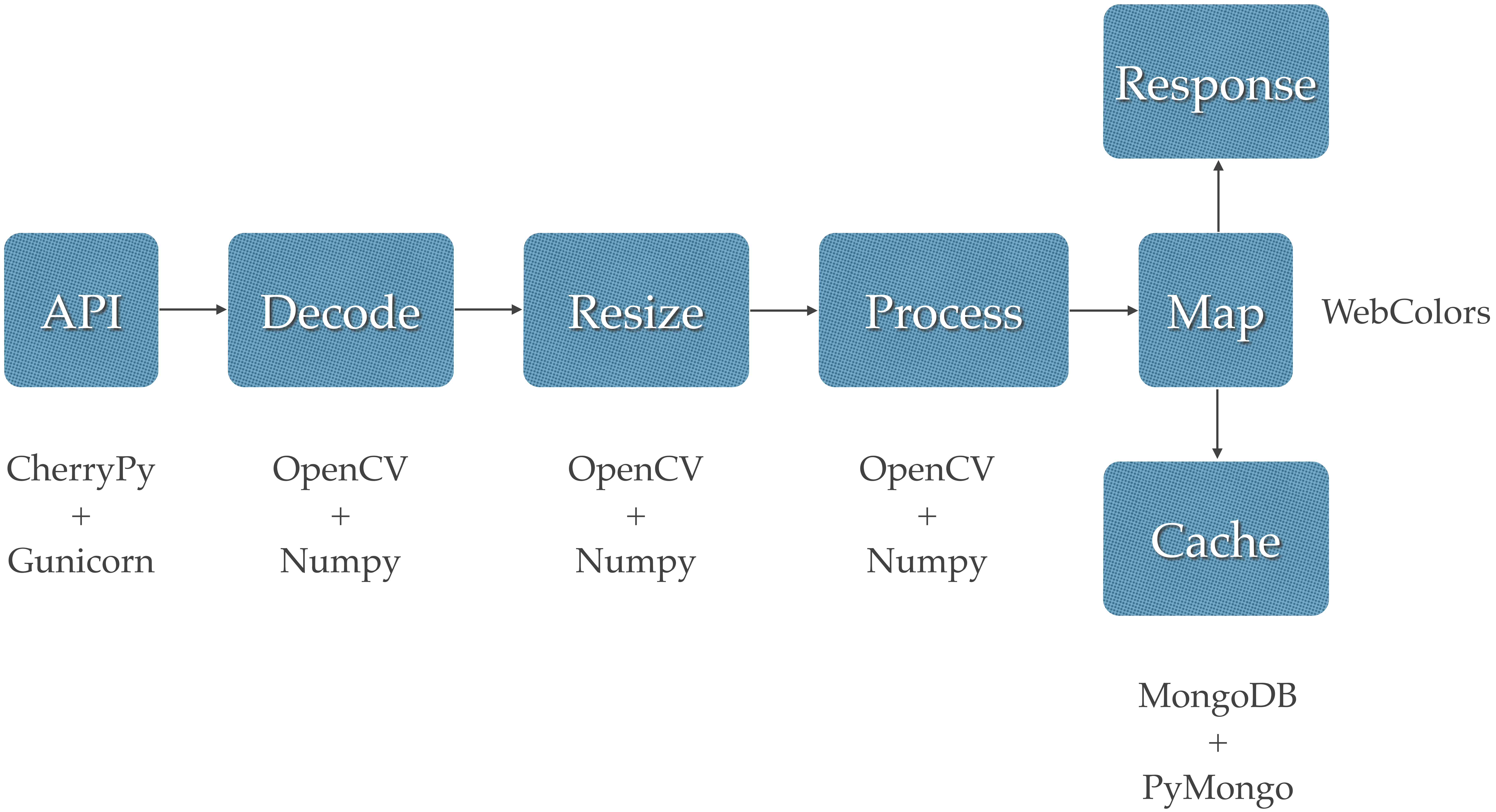
Product A in Store #1

Product A in Store #2

Doing things at scale

- 15 million webpages crawled and refreshed globally everyday.
- 40% of crawls are Apparels and Lifestyle products.
- 30% of daily crawls are new introduced products.
- 2 servers shared with bunch of other services.

Existing Architecture



API

CherryPy
+
Gunicorn

Decode

OpenCV
+
Numpy

Resize

OpenCV
+
Numpy

Process

OpenCV
+
Numpy

Map

WebColors

Response

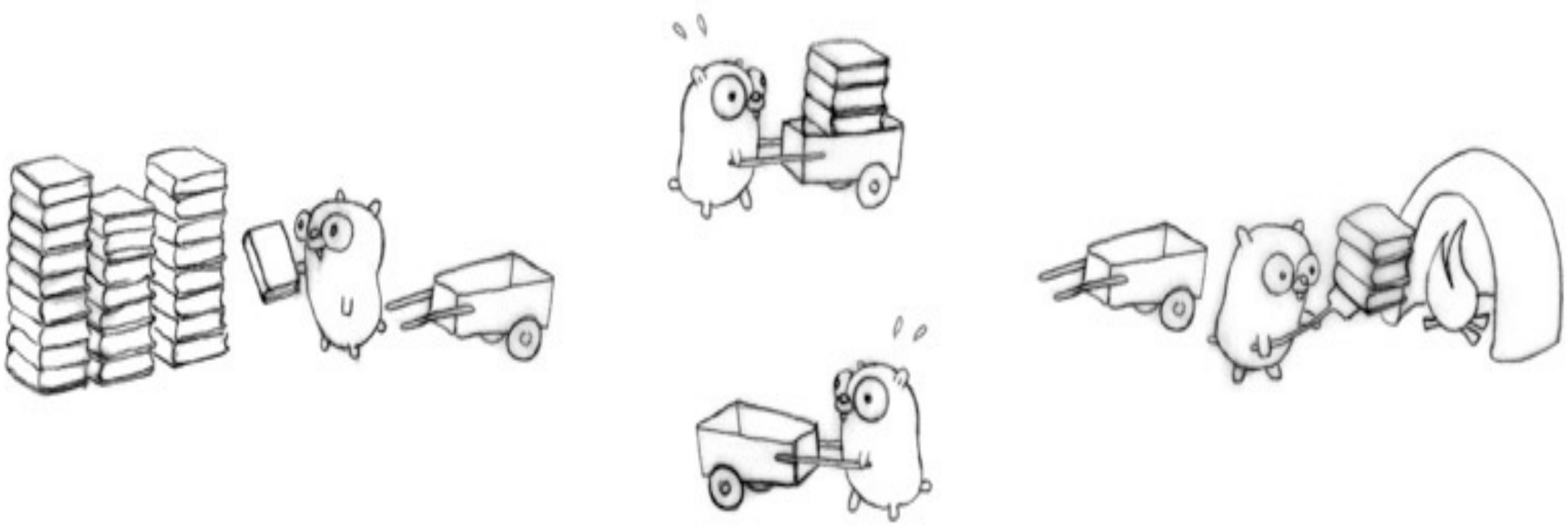
Cache

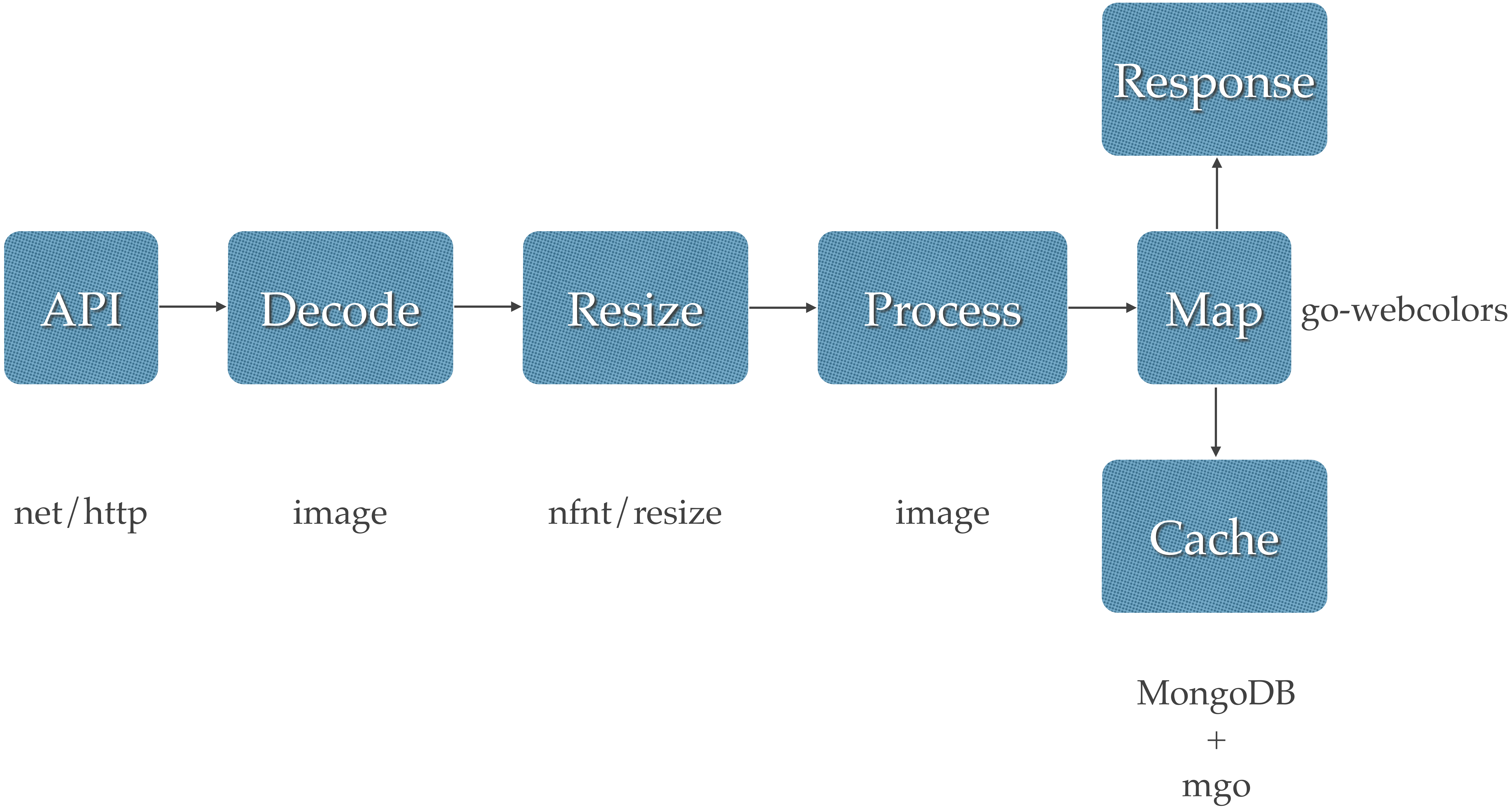
MongoDB
+
PyMongo

Pain Points

- High system usage
  - Servers are shared with other services (Celery, RabbitMQ etc.)
  - Running on 4 separate processes maxes out the CPU usage
  - Average memory usage takes over 70% of entire memory available

Bringing Gophers to the action…





API

net/http

Decode

image

Resize

nfnt/resize

Process

image

Map

go-webcolors

Response

Cache

MongoDB
+
mgo

Advantages

- Cheap concurrency - API is now able to serve more requests.
- Cooler servers - System usage never exceeds 50-60% with GOMAXPROCS set to 4.
- Easier deployment to multiple servers (thanks to binaries)
- Awesome standard libraries - bringing down the external package dependencies.

Still a long way to go…

- Numerical computing packages.
- Not many pure Go libraries for image processing or computer vision related works.
- Major OpenCV bindings are lacking, using SWIG to hook up C++ methods is pain!
- Need more adopters, more people playing and experimenting :)