Material Normalization Using Computer Vision
The Capstone Experience

Team Herman Miller

Josh Bhattarai
Ritwik Biswas
Joseph Smith
Ted Stacy
David Xuan

Department of Computer Science and Engineering
Michigan State University
Fall 2018
Functional Specifications

• Herman Miller needs an efficient way to normalize their fabric dataset
• Currently, there is no process in place for sorting aesthetic/subjective categories
• When Herman Miller receives a custom order with a proposed fabric, a manual search is done to find a Herman Miller fabric similar to the customers proposed fabric
• Our system will create a predictive model to categorize newly acquired Herman Miller fabrics
• Our predictive model will also be leveraged for categorizing a non-Herman Miller fabric, and suggesting Herman Miller fabrics that are similar
Design Specifications

• Our material normalization system contains three primary components:
  ▪ A predictive model to perform fabric categorization
  ▪ A web component to serve our predictive model as an API
  ▪ A user interface with the ability to upload fabric images, return the images categorization tags, and if applicable, return similar Herman Miller owned images
Screen Mockup: Image Upload

Upload *images* for classification

Drag and drop or click here
to upload your image (max 2 MiB)

☐ check box for material recommendations

Submit
Screen Mockup: Classification

Material Classification Engine

Green, Yellow, & Beige
Striped Pattern

Color Distribution

- Green
- Yellow
- Beige

See Recommendations
Screen Mockup: Image Match
Screen Mockup: Recommendations

Material Classification Engine

Your submitted image

We didn’t find your material. Here are some alternatives.

- Material ID: 283992, 86% match
- Material ID: 439472, 65% match
- Material ID: 264284, 48% match
- Material ID: 103748, 32% match
Technical Specifications

• Color Classifier (79.0% accuracy)
  ▪ Color Adaboost Classifier (Decision tree base model)
    Trained it on 11,000 test images from Herman Miller
  ▪ We scan an input image by 3x3 pixel blocks and create a color distribution pattern

• Pattern Classifier (84.4% accuracy)
  ▪ Used transfer learning to retrain the bottleneck layer of ImageNet Inception v3 (State of the art CNN image classifier)

• Recommendation Engine
  ▪ Our engine will be able to compare user submitted images to fabrics in the database based on classification metadata
Color Classification
Color Classification

RGB values of 3x3 block

210 159 155 191 140 136 228 177 173
255 222 218 221 170 166 204 153 149
207 156 152 241 190 186 169 118 114
Color Classification

RGB values of 3x3 block

210 159 155  191 140 136  228 177 173
255 222 218  221 170 166  204 153 149
207 156 152  241 190 186  169 118 114

RGB mean

214 165 161
Color Classification

<table>
<thead>
<tr>
<th>RGB values of 3x3 block</th>
</tr>
</thead>
<tbody>
<tr>
<td>210 159 155 191 140 136 228 177 173</td>
</tr>
<tr>
<td>255 222 218 221 170 166 204 153 149</td>
</tr>
<tr>
<td>207 156 152 241 190 186 169 118 114</td>
</tr>
</tbody>
</table>

RGB mean

214 165 161

AdaBoost Classifier
Color Classification

RGB values of 3x3 block:

- 210 159 155
- 191 140 136
- 228 177 173
- 255 222 218
- 221 170 166
- 204 153 149
- 207 156 152
- 241 190 186
- 169 118 114

RGB mean:

- 214 165 161

AdaBoost Classifier:

Color Distribution:

PINK

0 25 50 75 100 125 150 175 200
Color Classification

RGB mean 228 86 76

AdaBoost Classifier

Color Distribution

- RED
- PINK
- ANGE
- TOWN
- BEIGE
- GRAY
- BLUE
-_MISC
- TOILET
Color Classification

RGB mean
134 158 168

AdaBoost Classifier

Color Distribution
Color Classification

RGB mean
225 54 63

AdaBoost Classifier

Color Distribution

PINK RED ANGE GRAY BLUE VIOLET TOWN GREEN MISC BEIGE
System Architecture

Flask Web Client

AWS API Gateway

AWS Lambda Functions

AWS SageMaker
  - Color Analysis Model Endpoint
  - Pattern Analysis Model Endpoint
  - Recommendation Engine

TensorFlow

GraphQL

Fabric Database

S3 Model Storage Bucket

S3 Data Storage Bucket
System Components

• Software Platforms / Technologies
  ▪ Amazon Web Services
    o S3 – Storage for training dataset
    o SageMaker – Makes use of Jupyter Notebooks instances running TensorFlow transfer learning network to train and deploy machine learning models
    o Lambda – Interfaces model endpoints
    o API Gateway – Endpoint for client requests
  ▪ Machine Learning
    o TensorFlow – Used for training neural network for pattern classification
    o Scikit-learn – Used for training RGB based color classification
  ▪ Flask
    o Client side framework used to interface with AWS
  ▪ GraphQL
    o A schema definition language used for querying their fabric database
Risks

• Inconsistent Tags
  ▪ The dataset that we are using to train our machine learning models have incorrect tags that will negatively affect models
  ▪ **Solution:** Create a script that will assist in manually retagging images in the dataset

• API Efficiency
  ▪ Our API and classification models take a significant amount of time and the process needs to scale for batch classification
  ▪ **Solution:** Utilize asynchronous calls so the calls execute faster

• Pattern Scale Feasibility (**Stretch Goal**)
  ▪ Herman Miller wants a categorization called pattern scale which is the size of the pattern on a fabric, which is near impossible to determine with the given constraints
  ▪ **Solution:** Look into EXIF or other image metadata that will determine scale or request scale in API, there are also other computer vision theories
Questions?