IMAGION
HAROON CHOUDEERY, ROIANA REID, SUE YANG
Purpose

- To predict the success of Instagram images.
- Why Instagram? Very popular social media platform, used by individuals, small and large businesses, combines pictures and text, fast growing, global reach. High impact.
- Images scored on scale of 0-10. Scores are function of likes and performance of similar images in the past.
Target Demographic

- Advertisers
- Influencers
- Everyday users as well (everybody wants more Instagram likes 😄👍)

Common question: “Which of these two images will get more attention?”
Impact

- Quick, efficient and effective selection of quality pictures
- Gives users a better idea of characteristics and features that make image ‘better’
- Advertisers: Reduces advertise spend for companies
- Influencers: Increases popularity
Application Design

- Purpose boldly and clearly stated
- Attractive, intuitive, enjoyable experience for user
- Main feature are easy to find
- Quickly access results once inputs are uploaded
- Output displayed in simple and elegant manner
Upload Your Pictures (5 at most)

Analysis Result

4.8

Suggestion: This image is associated with an average score. Including a cat, fox, reflection, shoulder, shirt, jeans, leg, people.
## Google Vision API

### Top 5 labels for each image score

<table>
<thead>
<tr>
<th>Score</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>purple</td>
<td>product</td>
<td>art</td>
<td>text</td>
<td>red</td>
</tr>
<tr>
<td>1</td>
<td>room</td>
<td>plant</td>
<td>product</td>
<td>meal</td>
<td>dish</td>
</tr>
<tr>
<td>2</td>
<td>landmark</td>
<td>face</td>
<td>black</td>
<td>tree</td>
<td>meal</td>
</tr>
<tr>
<td>3</td>
<td>fashion access</td>
<td>purple</td>
<td>waterway</td>
<td>eyebrow</td>
<td>man</td>
</tr>
<tr>
<td>4</td>
<td>motor vehicle</td>
<td>room</td>
<td>yellow</td>
<td>blue</td>
<td>white</td>
</tr>
<tr>
<td>5</td>
<td>facial hair</td>
<td>mountain</td>
<td>face black &amp; white</td>
<td>plant</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>vacation</td>
<td>mountain</td>
<td>eyewear</td>
<td>swimwear</td>
<td>beauty</td>
</tr>
<tr>
<td>7</td>
<td>cat</td>
<td>fun</td>
<td>reflection</td>
<td>shoulder</td>
<td>sea</td>
</tr>
<tr>
<td>8</td>
<td>swimwear</td>
<td>room</td>
<td>jeans</td>
<td>leg</td>
<td>people</td>
</tr>
<tr>
<td>9</td>
<td>road</td>
<td>t shirt</td>
<td>girl</td>
<td>mammal</td>
<td>plant</td>
</tr>
<tr>
<td>10</td>
<td>dessert</td>
<td>rock</td>
<td>girl</td>
<td>mammal</td>
<td>photograph</td>
</tr>
</tbody>
</table>
Model

- **Model type:** Custom ConvNet with ~17mil parameters
- Designed for regression output (linear activation on final Dense layer)
- *See Appendix A for illustration of model*
- **Inputs:** Sample of ~12k images from Instagram
- **Labels:** Image score (more info on next slide)
- **Trained for 30 hours to ensure good fit using NVIDIA Tesla P100 GPU**
- **Loss:** ~0.02 MSE after about 120 epochs
Image Score

- Based on historical likes for user (over past x posts)
- Number of likes for each image is standardized for each user
- Standardized scores across users are normalized into [0,10] range to give final Image Score
- Excellent image = 10. Bad image = 0.
Evaluation

- Scoring accuracy function that ranks two randomly selected pictures from same user.
- Then accuracy calculated based on share of correct comparisons.
- Baseline is pro-Instagram user's ranking.

```
beverlyjoubert_10
beverlyjoubert_14
0
0
equal
a
False

1694 comparisons made
Accuracy score is: 0.27
```
Demo

www.ucbimagion.com
Results

- Model performance: 50.24% on our test data set (~1459 comparisons made)
- Expert performance: 67.5% on test data (40 comparisons)
- Average person performance: 47.5% on test data (40 comparisons)
Recap

- Successfully built out end-to-end pipeline for evaluating user-uploaded images
- Made several improvements to model but still not satisfactory enough
Challenges

- Model adjustment is time-consuming and somewhat mysterious (several iterations of model)
  - Parameters, pre-training Y/N, activation functions, optimizers, normalization, etc. needed to be tuned with every iteration

- Likes are not only a reflection of how good the image was (many other factors involved)

- Developing a quality score that reflects success of image for particular user, but that is consistent across users.
Future Work

- Gain access to a larger sample of images (perhaps IG API)
- Concatenate more relevant variables to image input
- Model image scores separately based on different use cases – advertising, models, sports, singers, actors
- Get tangible results to evaluate model
  - i.e. using two accounts – one with lower ranked images and another with higher ranked images (high-rank account should get more likes)
Thank You!

– The Imagion Team
Appendix A - Model Architecture

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Param #</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv2d_1(Conv2D)</td>
<td>(None, 100, 100, 32)</td>
<td>896</td>
</tr>
<tr>
<td>activation_1(Activation)</td>
<td>(None, 100, 100, 32)</td>
<td>0</td>
</tr>
<tr>
<td>conv2d_2(Conv2D)</td>
<td>(None, 98, 98, 32)</td>
<td>9248</td>
</tr>
<tr>
<td>activation_2(Activation)</td>
<td>(None, 98, 98, 32)</td>
<td>0</td>
</tr>
<tr>
<td>max_pooling2d_1(MaxPooling2)</td>
<td>(None, 49, 49, 32)</td>
<td>0</td>
</tr>
<tr>
<td>dropout_1(Dropout)</td>
<td>(None, 49, 49, 32)</td>
<td>0</td>
</tr>
<tr>
<td>conv2d_3(Conv2D)</td>
<td>(None, 49, 49, 64)</td>
<td>18496</td>
</tr>
<tr>
<td>activation_3(Activation)</td>
<td>(None, 49, 49, 64)</td>
<td>0</td>
</tr>
<tr>
<td>conv2d_4(Conv2D)</td>
<td>(None, 47, 47, 64)</td>
<td>36928</td>
</tr>
<tr>
<td>activation_4(Activation)</td>
<td>(None, 47, 47, 64)</td>
<td>0</td>
</tr>
<tr>
<td>max_pooling2d_2(MaxPooling2)</td>
<td>(None, 23, 23, 64)</td>
<td>0</td>
</tr>
<tr>
<td>dropout_2(Dropout)</td>
<td>(None, 23, 23, 64)</td>
<td>0</td>
</tr>
<tr>
<td>flatten_1(Flatten)</td>
<td>(None, 33856)</td>
<td>0</td>
</tr>
<tr>
<td>dense_1(Dense)</td>
<td>(None, 512)</td>
<td>17336784</td>
</tr>
<tr>
<td>activation_5(Activation)</td>
<td>(None, 512)</td>
<td>0</td>
</tr>
<tr>
<td>dropout_3(Dropout)</td>
<td>(None, 512)</td>
<td>0</td>
</tr>
<tr>
<td>dense_2(Dense)</td>
<td>(None, 1)</td>
<td>513</td>
</tr>
<tr>
<td>activation_6(Activation)</td>
<td>(None, 1)</td>
<td>0</td>
</tr>
</tbody>
</table>

Total params: 17,400,865
Trainable params: 17,400,865
Non-trainable params: 0