Improving Your Instagrammability

Hyun Jii Cho, Sophie Guilleux, Mike Jiao

Lausanne, Switzerland

Abstract

Instagram has not only changed the way millennials capture and share photography, but it has redefined where people decide to spend their time, and how they interact with the world around them. Nowhere has the effect of Instagram been more strongly felt than at Museums, where instagrammable exhibits have driven record visitorship, and caused many, previously non-camera friendly, institutions to revise rules around permitting photography. For millenials, instagrammability is now an essential part of a successful museum experience.

This project will explore what it means to take an instagrammable photo in the context of Lausanne museums, and strive to enable museum curators and visitors alike to understand the instagrammability of their own museums and photos. An interface to input and score a photo for instagrammability will be provided. The interface will allow a user to take or upload a photo from their desktop or mobile web browser, and receive a numerical score from 0 to 100 predicting the instagrammable quality of their photo. Predictions will be made based on a predictive model generated from training a convolutional neural network with 7,704 Lausanne museum-related images, each of which are scored on the ratio of Instagram Likes over Followers.

Keywords: Museum, Instagram, Experience Design

1. Technical Documentation

1.1. Data Collection

In order to generate a predictive model for museum photo instagrammability through supervised learning, we required as prerequisite a set of images forming our training data. Therefore, we used web scraping to download all the Instagram images that are publically available online and tagged with the top 7 most popularly instagrammed museums in Lausanne:

- Le Musée Olympique
- Collection de l'Art Brut
- Musée de design et d'art appliqués contemporains (mudac)
- Musée de l'Elysée
- Espace Arlaud
- Fondation de l'Hermitage
- Musée Cantonal des Beaux-Arts

(Fig. 1). We used a Python application called Instagram Scraper [1] to facilitate the web scraping process. This application allowed for the downloading of only images with specific Instagram hashtags.

Web scraping enabled us to collect around ten thousand images of various sizes, along with metadata on the Likes and Followers associated with most of the images.

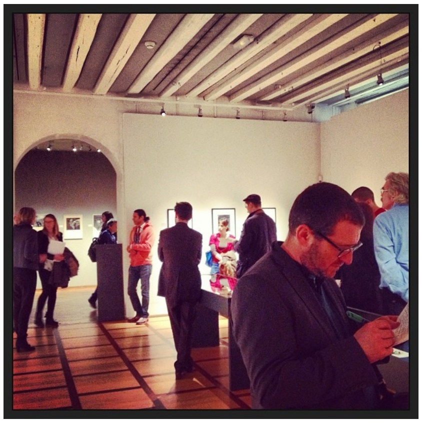


Figure 1: Example Instagram Training Set Image from Musée de l'Elysée

1.2. Labeling

Because we chose to use Convolutional Neural Networks, a supervised learning approach, to build our Instagrammability score system, we required our training data to be labeled. Using the "Likes" and "Followers" counts associated with each image in our dataset, we labeled each image with a instagrammability score, which is a ratio of Likes over Followers. We chose to use such a ratio to adjust for the skewed nature of Instagram accounts, with more followers receiving more exposure, and thus their images receiving more Likes all else the same. The labels were extracted in a similar way as the image datasets, via web scraping.

1.3. Image Processing

Images collected from Instagram were cropped and resized to a standard 224 x 224 pixels before being used as training data in our deep learning approach. This was done via Python scripts using libraries such as PIL (Python Imaging Library) and the Python Operating System module.

1.4. Exploratory Data Analysis

Before even to think about designing a tool able to classify Instagram photos, we first tried to find out more about the photos posted. The main question that guided us at this stage was probably to better understand what makes a photo more Instagramable? We therefore conducted an analysis based on a 800 photos database from the seven major museums of Lausanne. We discovered and listed the main features that improve the Instagramability of those panel photos. By looking at the top 25 liked photos for example, we discovered that images with strong elements of perspective or non-flat images have more success. Neutral photos in term of easy-recognisable subjects or not too personalized are also more often liked. Not least, bright or harmonious colours and elements that enhance the beauty of the region of Lausanne have a positive impact. For you to know, the results of this analysis are also visible on our new website (https://guilleuxsophie5.wixsite.com/monsite). This is also a unique opportunity to look at different results of the analysis such as, the top 25 liked Instagram photos, the top Instagram museums, the top hash tags, etc.

1.5. Machine Learning

Once a labeled Instagram training dataset was collected, we needed to select an appropriate method or machine learning technique for predicting image instagrammability. Although several options were explored, we ultimately proceeded with using a convolutional neural network technique due to limitations on time and feasibility. The section below details the machine learning approaches considered, and why they were or were not ultimately adopted.

1.5.1. Automated Feature Extraction

An obvious initial approach was to use existing image classification libraries to extract features from images, and predict an images instagrammability based on the existence of certain dominant features such as image colour, contrast, and composition. For example, if as a rule instagrammable images contained more highly saturated colours, more predictive instagrammability points would be awarded by our predictive algorithm to images with greater amounts of colour saturation (as determined by existing image classification libraries). However, we realized that it would be difficult to prove the validity and comprehensiveness of our hypothesized rules around instagrammability, especially in the context of Lausanne museums. For this reason, this method was not adopted.

1.5.2. Crowdsourced Feature Annotation

Another approach we considered was to crowdsource signals for image instagrammability, and use that data as the basis for an algorithm predicting instagramability. We tested building an interface for users to annotate images based on the image features they noticed, and their perception of an images instagrammability (Fig. 2). However, we realized that consistently labeling our training data in this fashion would be infeasible with the time we had. This approach would require an enormous amount of public participation to generate the necessary volume of crowdsourced instagrammability signals. In addition, there were several concerns around understanding the various biases introduced to the dataset by a crowdsourcing approach, and what that might mean for the predictive instagrammability signal we were hoping to generate.

1.5.3. Convolutional Neural Network

We ultimately decided on using deep learning, in particular Convolutional Neural Networks, to learn the relationship between images and an instagrammability score.

Convolutional Neural Networks are state-of-the-art deep learning algorithms that are powerful in analyzing images. Neural Networks are made up of layers of neurons with learnable weights and biases. In the simplest case, a CNN model is a list of layers that transforms the original pixel values of an image to an output score. The three main types of layers are Convolutional Layers, Pooling Layers, and Fully-Connected Layers.

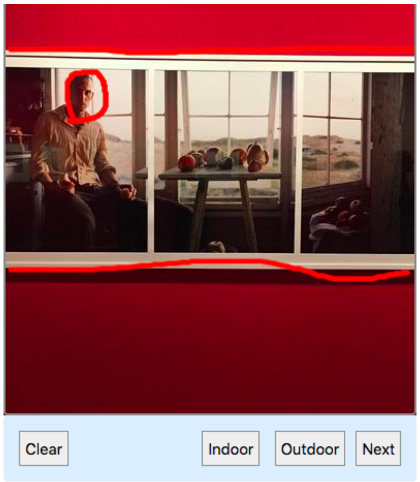


Figure 2: Example of Image Being Annotated in Crowdsource Annotation Tool

In our project, we would like to map images to instagrammability scores; thus, this can be viewed as a regression problem because our goal is to predict a value from a continuous set. For the purposes of the project and the time-constraint, we decided to use transfer learning to fine-tune a pre-trained model.

Transfer learning is a popular method that allows us to efficiently build accurate models. Instead of starting a learning process from scratch, we use a pre-trained model to leverage patterns that were learned when solving a different problem.

A pre-trained model is a model that was trained on a large benchmark dataset, such as the ImageNet images in our case. Due to the computational cost of training, it is common practice to import popular CNN architectures from published literature. Specifically, for our instagrammability scorer, we use the ResNet50 architecture. ResNet50 is a model consisting of 50 layers and it differs from traditional sequential models because it incorporates heavy batch normalization and ”identity shortcut connections” in which one or more layers are skipped.

In our model, we use the keras ResNet50 model pretrained on 14 million ImageNet images. We freeze all the layers except the last layer and train the last layer and predict with a sigmoid function for our image regression problem. Despite the computational power and time required for training, even a pretrained model, we trained the model for 50 epochs with a batch size of 32. In the end, the validation mean squared loss was 0.2094.

1.6. Post Processing

1.7. Interface

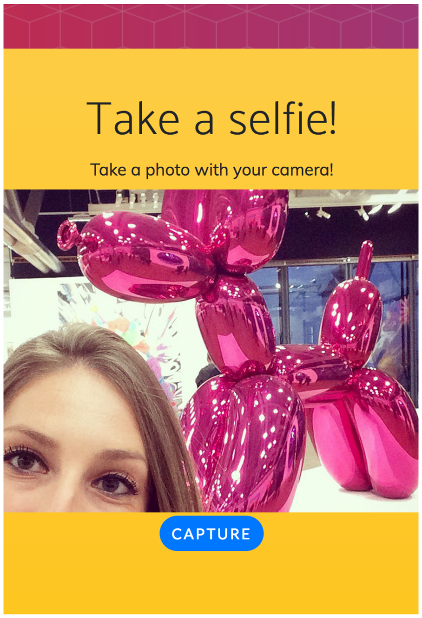
The interface for instagrammablity predictions can be accessed at https://trailerspoiler.com or by running

```
python3 instability\_init.py
```

in the terminal. The interface is built with the Flask backend and Bootstrap frontend frameworks.

1.7.1. Image Selection

To score an images instagrammability, users are given the option to upload an image, or take a picture from the web application. After selecting an input image (through the users device camera or file directory), the selected image will be displayed for the user to preview (Fig. 4). Users can change the selected input image as many times as they desire.



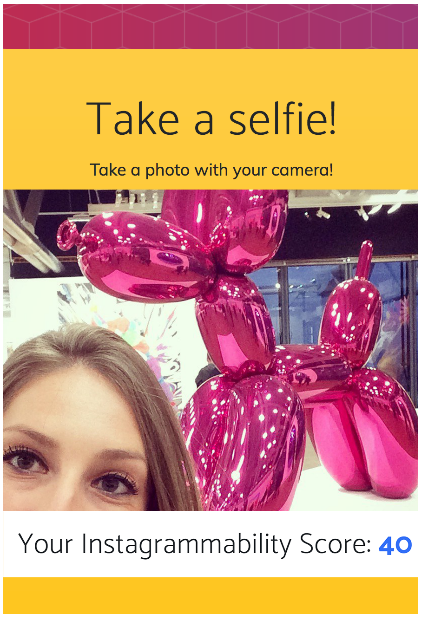
Take a selfie!

Take a photo with your camera!

Figure 3: Example of Image Being Selected for Instagrammability Prediction

1.7.2. Image Scoring

After an input image is selected, users can scroll down to the next section of the web application and press the Predict button to receive a score predicting the images instagrammability (Fig. 5). The score will range from 0 to 100, from best to worst possible prediction score.



Take a selfie!

Take a photo with your camera!

Your Instagrammability Score: 40

Figure 4: Example of Image Instagrammability Prediction Score Result

1.8. Possible Improvements

Due to the time constraints of this course, this project presents an initial investigation into understanding museum instagrammability. Many possible areas of improvement and extension exist for the project. These areas include improving the size of the training dataset used for the machine learning, improving the quality of the supervisory signal used to classify instagrammability (perhaps using an images comments, hashtags, and other factors to create a more comprehensive signal), and extending the web interface used to display instagrammability prediction scores to provide features such as the ability to select and score multiple images at once.

This project can also be extended beyond the scope of museum experiences to include zoos, parks, popular tourist attractions, and many more areas. In fact, it is arguable that there exist many more practical, and general areas to study instagrammability outside of the museum space.

2. Code Documentation

The code for this project exists in a set of Jupyter Notebooks, python scripts and web application files. You can find the GitHub repository containing these files at https://github.com/justcho5/instagrammability, in addition to the submitted zipped files. The instagrammability web application is hosted on https://trailerspoiler.com and the image analysis website is at https://guilleuxsophie5.wixsite.com/monsite

2.1. Notebooks

Jupyter Notebooks were used to collect, label, process, and train our Instagram image dataset for machine learning. Notebooks were also used for data exploration purposes, investigating the nature of the dataset we worked with and identifying insights from Lausanne museum images.

2.2. Flask and Bootstrap

Flask is used to run the backend of the web application showcasing the image instagrammability predictions. Bootstrap is used to build the frontend of the web application in a well-designed, minimalistic, mobile friendly fashion. Dependencies required to run the Flask and Bootstrap web application are listed in the requirements.txt file, and should be installed before running the project. After cloning the project repository and installing required dependencies, running the web application locally can be done by running ./instability_init.py within the flask-instability directory.

3. References

[1] Python Instagram Web Scraper Application, https://github.com/rarcega/instagram-scraper 2018

[2] Python Imaging Library, https://github.com/python-pillow/Pillow 2018

[3] Flask Web Backend Framework, https://github.com/mitsuhiko/flask 2018

[4] Bootstrap Web Frontend Framework, https://github.com/BlackrockDigital/startbootstrap-new-age 2018