FaceLift: An explainable urban transformation pipeline

ABSTRACT

Interpretable and explainable insights from deep-learning models is still a working problem. This problem is also apparent in the cross-disciplinary work which uses deep-learning to quantify urban environments. In this paper we propose a Deep Learning driven pipeline which works on streetview images, called *FaceLift*. The pipeline allows us to learn, transform and explain, intangible concepts in urban settings like beauty, safety etc. For the sake of this paper we work with the dimension of beauty in urban scenes. We do so with the help of generative models, that generates approximate transformations of streetview images to maximize or minimize beauty. We further develop literature driven urban design metrics to explain the properties in a scene that makes it beautiful. We implement these metrics using computer vision driven algorithms. We validate our pipeline's transformational capabilities using crowd sourced experiments. We conclude by summarizing actual expert views about the use of such a pipeline and metrics, and discuss the broader implications.

ACM Reference format:

. 2017. FaceLift: An explainable urban transformation pipeline. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 9 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The development of new technologies behind deep neural networks has progressed at an amazing pace over the past decade [16], thus enabling the research community to achieve groundbreaking results across fields and in a wide variety of tasks including object detection [30], scene detection, reinforcement learning [17], and language modeling [3]. Where deep learning excels in performance – in many cases surpassing human ability – it often falls short in producing models that are interpretable. As a result, unlike in traditional machine learning, neural networks are often used as black-box functions whose decisions cannot be supplemented by any human-readable explanation about *why* those decisions were taken. This limitation is known as the *explainability gap* or the *black-box problem* of neural networks. The intrinsic complexity behind deep neural networks makes this problem hard to solve. However, as the adoption deep neural networks spreads across domains and as their use becomes more common to solve increasingly sophisticated duties with humans in the loop, it becomes crucial to annotate any decision with a human-readable explanation. The rapid growth in feasibility and potential of deeplearning models have also spurred several cross disciplinary application oriented research. We particular bring to notice the works like [15, 18, 19, 27, 28] which have used deeplearning in the areas of quantification of urban perception and urban spacial reasoning . These works draw inferences using deeplearning driven classifiers about several aspects of urban environment such as scenicness , deprivation or changes in terms of land use. However these also suffer from the problem of explaining why a particular inference is drawn. These insights are of significant importance as they can be fed back into the collection of intervention systems like urban planning departments, urban activists, municipal authorities etc. More so these explainable insights also need to be interpretable by the urban intervention agents. In this paper we propose a generalizable pipeline powered by deep learning that produces interpretable insights on what makes an urban environment arouse a particular perception. We do so by reasoning with the knowledge learnt by a generative deep learning model, trained to quantify a particular dimension of urban environment such as scenic-ness, beauty, safety etc. We reason using urban design metrics drawn from literature which are measured in actual streetview images using different computer vision techniques. For the sake of focus, in this paper we concentrate on the urban perception of beauty, but essentially this pipeline is generalizable to any aforementioned dimension, given a required dataset.

In the following sections, we will explore the related work in the fields of deep learning driven aesthetic computing and urban analysis. We then elaborate on the data, design and technical details of the *FaceLift* pipeline. We then describe the design and implementation of explainable urban design metrics. We then evaluate the behaviour of these metrics with respect to the data. Finally we discuss the expert views about such a tool and discuss the biases and implications.

2 RELATED WORK

We explore related work in the fields of computational aesthetics and in the area of data driven inferences in urban environments. Early work in the field of computational aesthetics done by Datta [5] looked at the beauty aspect of images using hand-crafted visual features and datasets collected from photo-contest websites. It showed that subjective properties like beauty can be estimated using computer vision techniques, provided we have good data. The introduction of deep-learning in this field boosted the activity. Post deep-learning works [13, 26, 32] explored the dimensions of beauty, aesthetics and their linkages to popularity and engagement over the web. Despite being very subjective dimensions, these works showed impressive performance in quantifying them.

Extending quantification of subjective information to the realm of maps was explored by works such as [1, 21–23]. These works took the subjective dimensions such as beauty, loudness, and smelly-ness and augment this information onto real world maps to present a new dimension in which one can explore their world. Works like [19, 25], collected and analyzed responses to images of urbanscapes across different subjective dimensions including safety, depression, beauty and built deep-learning models on their data. An extension

Conference'17, Washington, DC, USA
© 2017 ACM. 978-x-xxxx-xxxx-x/YY/MM...$15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

symbol	**stands for**
X	Georeferenced urban image dataset
I_i	Georeferenced image $\in X$
Y	Annotations classes for X (e.g. beautiful/ugly)
y_i	Class in Y (e.g. beautiful)
$\hat{I}_j$	Template image
I'	Target Image
C	Image Classifier
R	Images acquired by rotating street view camera
T	Images acquired by translating street view camera
ρ	Similarity bound below which smart augmentation chooses translated images

term	**stands for**
Template Image $\hat{I}_j$	A synthetic transformation of input image I towards the class y_j
Target Image I'	The natural image which is most visually similar to the template image
Data Augmentation	A process of data expansion which looks for images taken in the surroundings of the georeferenced images in X
Classifier	A deep-learning framework that is able to classify images into one of the classes in Y
Generator (GAN)	A deep-learning based image generator
$DGN - AM$	A framework that, given the GAN and the Classifier, transforms an input image into the template image.

Table 1: Notations and Terms.

of this work [8] used deep learning to train models capable to rank urban images according to these subjective dimensions. Other works used deep-learning [15, 18, 27, 28] to not just quantify urban environments, but to draw inferences and insights regarding outcome variables like poverty, mental health etc. The limitation of these works is that the models developed for urban perception are not interpretable, i.e., they provide predictions regarding subjective qualities of urban images, without explaining the reasoning behind the predicted score. This motivates our work as we aim at making these inferences interpretable to most agents who are learned in the science of urban design.

3 FACELIFT FRAMEWORK

We present here Facelift, an end-to-end framework for image beautification. The framework embeds a model trained on a set of urban images annotated with beauty scores. It takes as input a geolocated urban image and gives as output its transformed (beautified) version. Although we refer here to specific urban properties (i.e. beauty) and datasets, Facelift is generalizable to any labeled dataset of geolocated images.

For the sake of berevity, we summarise the notations used in Table 1 and the pipeline steps in Figure 1.

In general terms, the framework allows anyone with an arbitrary set of *geolocated* images $X = I_1, I_2....I_n$ annotated in classes $Y = y_1, y_2, ..., y_k$, to transform natural images between classes: the algorithm can transform an image I_i belonging to class $y_i \in Y$, to image I_j from class $y_j \in Y$. Both I_i and I_j are natural, non-synthetic images. Despite having another *meaning* (i.e. category), I_j maintains the structural characteristics of I_i (e.g. point of view, layout). This allows to visually reason about the discriminative properties between classes $y_i, y_j \in Y$, and visually understand the salient characteristics that drive a classifier to distinguish between classes y_i, y_j.

The transformation framework consists of three phases (see Fig. 1). In the first phase, we classify images from X into the corresponding categories Y with high accuracy , using a convolutional neural network C. In our case, y_i and y_j are the beautiful/ugly classes. In the second phase, we transform am image from class y_i to class y_j, using Generative Adversarial Networks[24]. The output of this phase is a synthetic image $\hat{I}_j$, which summarizes the basic traits of the destination class $y_j \in Y$. The last phase matches the synthetic image $\hat{I}_j$, with the closest natural image in X. Finally, to quantitatively reason about the beautification process, we perform aggregated analysis of the differences between original images and resulting target images.We do this by quantifying the presence of 5 urban design metrics in uglified and beautified images.

For the rest of this section, we would delve deeper into the specifics of the image beautification framework.

3.1 Phase 1: Classifying Beauty

We design here a classifier C able to correctly assess the beauty category y_i of an image in X using a deep learning network. To reliably train a convolutional neural netowrk we need first make sure we have enough reliable data to train the classifier **[REF]**. We do this by augmenting the available geolocated image data.

3.1.1 Dataset and Beauty Judgements. Our seed dataset comes from Place pulse, a research work on urban affective dimensions [8]. The dataset in total contains 100k images across 56 cities around the world from Google StreetView[1]. Images are annotated through pair-wise comparison for qualities such as beauty, depression, richness , safety etc. For the purpose of our work, we use the beauty judgements. To train our classifier C to detect beauty categories Y, we need to transform pairwise votes into absolute scores, then discretize absolute scores into a finite set of categories y_i. We transform the pairwise votes into ordinal scores using the TrueSkill [12] algorithm. To ensure reliability of absolute judgements, we filter out images with less than 3 votes. To discretize the resulting scores, we heuristically partition the data into two classes with maximum separation: beautiful and ugly. Figure 2 shows the distribution of Trueskill score estimates with the threshold scores at which we decide beauty or ugly class boundary.

3.1.2 Data Augmentation similarity bound. Despite over a hundred thousand images in the original data, only 20,000 has more than 3 judgements. This is non-ideal to train classifiers with substantial number of parameters such as convolutional neural network, since smaller data size implies that a machine learning model has a risk of over-fitting **[REF]**. We choose to augment the dataset by exploiting the geo-located nature of the image dataset. We also take advantage of the fact that urban places in close proximity look quite similar to each other [6] To develop a better way of augmenting images which can be transferred with scores of the original annotated ones, we make one heuristic assumption : "**[A1]***the composition of a StreetView image does not change considerably for small rotations of the camera angle*". This assumption was tried and tested over several samples both manually and using image similarity measures. An example of one such sample can be seen in Figure 3a. This assumption allows us to do a basic expansion of our dataset without adding a lot of noise. However we cannot to a similar assumption when it comes to translation of camera. All images in the PlacePulse dataset are taken with a default camera rotation which depends on

[1] https://maps.googleapis.com/maps/api/streetview

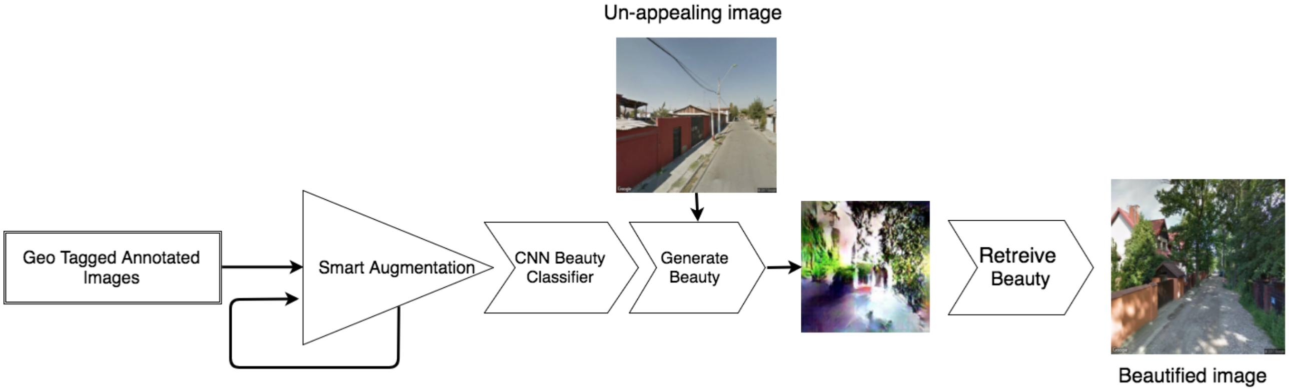


Figure 1: Architecture of the Beautification Pipeline

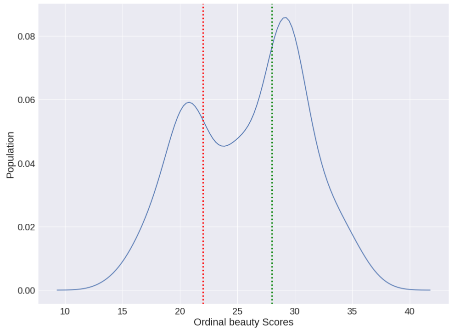


Figure 2: Distribution of ordinal scores for images with at-least 4 votes. The red and green line represent the threshold below and above which images are tagged Ugly or beautiful. Images in between are dropped for separability

the location. The Streetview API allows to specify the preferred camera rotation angle. This gives us the opportunity to take snapshots of the same location at different camera angles. We rotate the camera across different values $\theta \in -30°, -15°, 15°, 30°$ and derive a first set $R = \{R(I)_\theta \forall I \in X\}$, which consists of images acquired by rotating the camera angle for each annotated image in X. Following **A1**, For these images, the beauty score of each rotated image $R(I)_\theta$ can be safely transferred from I.

Next, for each image in X, we translate the location of the Streetview camera: we select points on the map at a distance of $d \in \{10, 20, 40, 60\}$ meters and acquire the resulting set of images $T = \{T(I)_d \forall I \in X\}$. Although possibly very similar, transferring the beauty score from I to each $T(I)_d$ might result in very noisy data. To understand the extent to which beauty scores can be transferred from images to their translated version, we use a smart augmentation technique.

In a nutshell, this technique computes the similarity between the translated and the original image, and transfers the beauty scores only if the similarity is acceptably high. To do so, we represent each images from both sets using visual features extracted from the FC7 layer of PlacesNet [33]. We then calculate the cosine similarities $S_t = \{s(I, T(I)_d) \forall I \in X\}$ between each original image I and all images in the augmented set $T(I)_d$. We also calculate another set of cosine similarities $S_r = \{s(I, R(I)_\theta) \forall I \in X\}$ between rotated and origianl images. We define a similarity bound as the median similarity betwen rotated and original images.

$$\rho = median(S_r) \text{ where } S_r = \{s(I, R(I)_\theta) \forall I \in X\} \qquad (1)$$

Following the assumption **[A1]** , we only transfer beauty scores to translated images who look as similar to the original as their rotated counterparts: $s(I, T(I)_d) < \rho$. We discard translated images not fullfilling this requirement and retain the resulting images in the smartly translated image set $\hat{T}$.

3.1.3 Semantics of Augmentable images.

Given that we now had a similarity bound to decide whether to augment or not a particular image, we wondered whether certain types of scenes are more prone to augmentation compared to others. So we partitioned our data in two sets

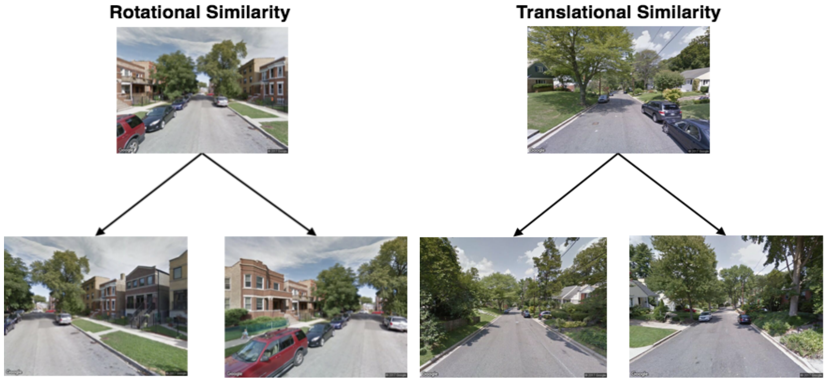
- $setA$: contains images where the median similarity between translated images and the original image $s(I, T(I)_d) < \rho$.
- $setB$: Images whose similarity with their translated set is farther apart i.e. $s(I, T(I)_d) > \rho$.

We describe each image in both sets according to the scene depicted, by collecting the PlacesNet [33] labels with the top5 confidence scores. We then aggregate such labels at a set level by computing a TF-IDF metric. The resulting set of {label,Count} pairs reflect essentially how common or uncommon is a particular scene label in $setA$ compared to $setB$

The resulting prevalences of scene types can be seen in Fig 4. The plot shows that scenes like highways, fields and bridges, typically more uniform and open, don't change despite translation. Other urban scenes like gardens, residential neighbourhoods , plazas and skyscrapers are more sensitive to change in perspective by translation.

3.1.4 The Beauty Classifier.

Once we have enough data, we train a deep convolutional network to classify images into the Y beauty classes. One may use several successful deep convolutional neural network architectures, which work for other use cases like AlexNet [14] , PlacesNet [33] or GoogLeNet [29]. For our paper we use CaffeNet which is a modified version of AlexNet. This trained classifier is a important component in the next phase, which is generation of images.



(a) (b)

Figure 3: Fig 3a shows an example set of images showing similarity of streetview scapes, when the camera is rotated by a small angle. Fig 3b shows the translational similarity where the angle is less than the established bound ρ

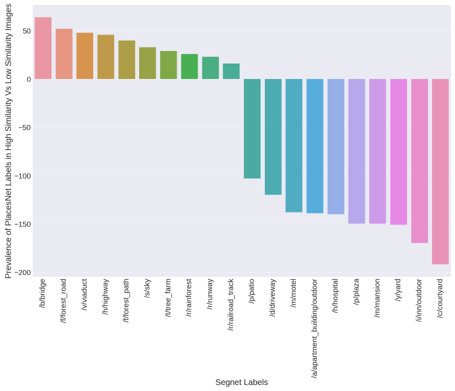


Figure 4: Prevalence plot of types of scenes prevalent in Similar images compared to dissimilar ones.

We employ the above observations to train the beauty classifier model. We start with the base dataset of 20k images (X), as described in Section 3.1.1. We then progressively augment the data: first with rotation across 5 angles (X,R), then rotation with uniform translation for all images (X,R,T), and then rotation and translation for only images which satisfy similarity bound as evaluated as shown in Equation 1 (X,R,$\hat{T}$). We then train a Convolutional neural net model based on AlexNet architecture [29] on each of these augmentated datasets. The training is done on 70% split of the data and tested on the 30%.

We see considerable improvement in classifier accuracy 2, with the best model performing at 73% accuracy for classifying images in two classes of Beauty and Ugly. This model represents the concept of beauty, learned from annotated and augmented streetview images.

Policy	Accuracy (Percentage)
No augmentation (X)	63
Rotation only	68
Rotation + translation	64
Rotation + Smart Translation	73.5

Table 2: Performance differences based on different augmentation policies

3.2 Phase 2: Generating Images

We now want to design a framework to transform any image I_i of class y_i into template image $\hat{I_j}$ (as shown in Figure 1), namely a synthetic version of the original image, with added features and motifs that maximize class y_j. To produce the template image $\hat{I_j}$, we need the following components in place:

- *Classifier*. We need a deep classifier C able to classify I into Y, i.e. between ugly and beautiful images.
- *Generator*. We train a generative adversarial network (GAN) which can generate an approximate natural looking image drawn from distribution of a particular class of images, similar to the one in [7].
- *Activation Maximization*. We plug in the GAN and the classifier network into an Activation Maximisation (AM) framework. Given these components, an input image I, and a target beauty class y_i, the AM transforms I in an ideal image $\hat{I_j}$ (that maximizes the activation for beauty class y_i).

We have described the design and performace of Classifier in Sec. 3.1.4. We will delve deeper into the other two below

3.2.1 Generator. Generative Adverserial networks are an extremly useful tool when it comes to generating samples from a learned distribution[24]. GANs consist of a pair of networks where the *generator* generates image samples similar to an input space using de-convolutional layers, and the *discriminator* learns to discriminate between natural images from the training set and synthetic images generated by the generator. Since GANs are known to be very tricky to train [11] we first try to use a pre-trained GANs on Imagenet from [20].However, because of the vast difference between images in Imagenet and Streetview images, our initial results were not very optimal. We therefore retrained the generator on the StreetScore dataset. This improved the visual quality of the generated images considerably.

3.2.2 Activation Maximization. We build on top of the Activation Maximization technique elaborated by Nguyen et. al [20] (DGN-AM). DGN-AM utilizes the property of locality of codes:Generator codes which are close to each other would create similar looking images. DGN-AM was initially built to visualise the concept learnt by CNNs, by finding the code which maximised the activation of a particular output class in the classifier network. The maximization is achieved by doing gradient descent on the input generator codes with respect to the classifier neuronal activation, keeping everything else locked. The result is a synthetic image that has a high activation for a pre-determined output neuron. We modify this method by starting the maximization method from a code K which corresponds to the a-priori input image, for example, an ugly urban image I_i. So for a given image I_i which belongs to class y_i (which could be the beauty neuron or the ugly neuron of the classifier C), the DGN-AM algorithm would perform Stochastic gradient descent on the generator codes of the a-priori image I_i so as to maximize the target neuron y_j (which could be beauty of ugly neuron of C resulting in a synthetic image $\hat{I_j}$ generated by the generator G from the code $\hat{K}$. The whole optimization can be expressed as Equation 2.

$$\hat{I_j} = G(\hat{K}) : \arg\max_{\hat{K}}(C_j(G(\hat{K})) - \lambda||\hat{K}||) \qquad (2)$$

Here C_j corresponds to the activation of the neuron j of the classifier C , and G is the generator network. λ is the L_2 regularization factor. The resulting output image $\hat{I_j}$ is a natural-like image, which maximizes the beauty neuron for our classifier. We hypothesize that because the process begins from an a-priori image, the resulting image is closest possible template to the ugly input image, but with the beauty neural activation maximized. Figure 5 shows the activation maximization output in the center.

3.3 Phase 3: Retrieving Images

In this final step we find a target image I' from the dataset that is closely aligned, in terms of some visual similarity metric $E(I_1, I_2)$, with the generated template image $\hat{I_j}$. The result of this exercise is to find the most similar looking image to an input image I that maximizes a particular annotation class y_j. The problem of finding images which are visually similar can be solved using image embeddings in a N dimensional space R^N We use a pre-trained deep network, which is trained to classify scene types to a very high accuracy [33] to extract the image embeddings. We extract a 4096 dimensional feature vector from the FC7 layer of the network to describe the the template image. We then extract feature vectors from the complete test dataset using the same process. We can now use the L_2 Norm to find pairwise distances in the R^{4096}. Formally with N test natural images and a template image $\hat{i}$ we extract $v_{\hat{i}} \in R^{4096}$ and and find pairwise distances $\{d_j \; \forall j \in N\}$ where $d_j = L_2(v_j, v_{\hat{i}})$ We then find the target image by finding the $min(\{d_j\})$. For the sake of redundancy, we find the top 5 such matches for every template $\hat{i}$ generated from every ugly image i. These target images are what we call the transformed images.

3.4 Pipeline Validation

Because beauty is a subjective notion, we need to understand if our framework actually correctly transforms images into more or less perceptually beautiful. For this, we run a user study to check how often humans agree with the machines inference. We designa a crowd-sourcing experiment to understand how much do real humans agree with the pipeline's transformations. We randomly select 200 images, 100 beautiful and 100 ugly as per their TrueSkill scores. To have a reliable seperation in terms of visual appeal, we only select ugly images with scores less than 15 and beautiful images with scores greater than 30. This means that we are selecting images from the bottom 10 and top 10 percentiles according to the Trueskill distribution as shown in figure 2. These images are then transformed to the opposite side of the spectrum of beauty using FaceLift. As a result, a beautiful image would be transformed into an ugly image and vice versa. Then we design an Amazon Mechanical Turk experiment, where we ask the turkers to choose the beautiful image between the original and the transformed images, without giving any hints of the transformation. We pay 0.1$ per human intensive task and we make sure that each Turker is a verified master, which assures that the Turker has a HIT approval rate above 90% for the past 30 days. We make sure that we have at least 3 votes on each image comparison, there by allowing us to choose majority voting.The results show that over all , the Turkers agree with the model **77.5%** of the times. Besides the overall agreement, the turkers agree **70%** of the time with the process of beautification and **85%** of the time with uglyfication. These results show that the facelift pipeline is learning the concept of beauty and then doing agreeable transformations on images.

4 VALIDATION METRICS

We found our pipeline to be an effective tool to beautify images of urban spaces. We now want to understand what the algorithm is looking at when transforming images. One way to do so would be to look at the template images and infer color and texture patterns. However, this approach is not scalable, as it would involve a substantial manual effort, and would be subject to personal interpretation.

One of the main contributions of this work is to develop metricsto explain what the network is learning as discriminatory features of beauty, in a fully automatic way. Table 3 shows a list of 5 evaluation metricstaken from urban design literature[2, 9]. These represent interpretable, measurable urban elements whose presence drives the aesthetic value of an urban environment. We design computer-based methods to map each of these theoretical metrics into a computationally measurable form. To measure the metrics, we select 500 ugly and 500 beautiful images from the test dataset based on their base TrueSkill scores. We then transform these images towards the opposite side of the beauty specturm using the FaceLift pipeline. We compare the values of these features between the two changed samples (beauty -> ugly , ugly->beauty), thus inferring statistics regarding which types of urban elements are added/removed by the beautification pipeline. In this Section, we present the computational methods used to map these 5 metrics, as well as the results that we get on the pre and post transform images.

4.1 Computational Methods

To validate the presence before/after transformation of the 5 urban elements in Table 3 we use a set computer-vision based tools as well as traditional data analysis techniques.

4.1.1 Measuring Walkability and presence of Landmarks. To quanitfy elements of Walkability, Greenery, and Landmarks, we proceed as follows. We use PlacesNet [33] to extract information regarding the scene type (e.g. beach, garden, etc.). PlaceNet's output is the SoftMax distribution over 205 scene labels. We retain for an image the top 5 labels with higher confidence scores. We classify

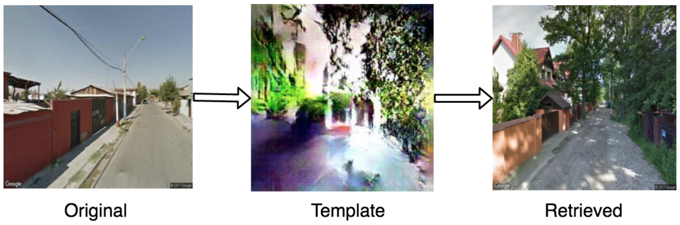


Figure 5: Example of Beautification Process

Metric	Description
Walkability	Walkable streets are rated high on an aesthetic scale [9]. Walkable streets increase the social capital of a place and appeal to the exploring nature of human psyche. This implies that the urban space needs to address the fundamental need of people to walk and explore. This also implies that a walkable street must also be perceived as safe.
Green Spaces	Presence of Greenery is always pleasing to the eye. The literature always links urban beauty to curated and well maintained green spaces, where social interactions can happen [2]. This 'social' aspect of greenery implies that dense forests or unkempt greens are not always related to the sense of beauty in urban scenes.
Landmarks	Loosing a bearing in the city is not a very pleasant experience. Hence presence of recognisable and guiding landmarks influences the perception of an urban space [9].
Privacy-Openness	A sense of privacy and a complimentary sense of openness are both influential in our perception of a place[9]. These values also tend to be related in an inverse 'U' fashion with beauty.
Visual Complexity	Visual complexity is a measure of how diverse a urban scene is. It manifests in terms of various design materials, textures and objects. Generally, visual complexity has an inverse 'U' relation with aesthetic values. The beauty and aesthetics of a place increases until it starts dropping because of 'too much' complexity[9].

Table 3: Urban Design Metrics

the 205 scene labels of PlacesNet into 4 categories, **L**andmarks , **A**rchitectural , **W**alkable , **N**atural. Each category is inspired from urban design literature [10]. Labels like *Abbey , Plaza , Courtyard, Garden, Picnic Area, Park , etc* fall into the category of *Walkable*, where as labels like *Mansion, Castle, Dam , Airport, etc* fall in the category of *Landmarks*. Labels like *Residential neighborhood, Motel, hotel, restaurant, etc* fall in the category of *Architectural* and labels like fields, pasture , forest, ocean, beach etc fall in the category *Natural*. For an image, we then measure its Wakability according to how many of the top-5 labels fall in category W. Similarly, we quantify presence of Greenery and Landmarks according to the frequency of N and L labels.

4.1.2 Measuring Openness and Green spaces. To measure Openness, we resort to Segnet [4], a semantic segmentation algorithm, which is trained on dashcam images from a real driving dataset , to detect 12 different elements in the image namely road, sky , trees , buildings , poles , signage , pedestrians, vehicles ,bicycles , pavement, fences and road markings. At the risk of over-simplifying, we can approximate that the openness of a street scene with the portion of sky visible in the scene, green cover by the portion of greenery detected in an image etc. We therefore quantify openness as the number of pixels labelled as 'sky' by segnet, and green cover as the portion of pixels labelled as greenery by segnet.

4.1.3 Measuring Visual Complexity. Visual complexity is a measure used in urban design measurement [9] to understand the diversity of a particular place. Ideally the complexity has a more granular nature, right from the texture of the roads and walls, to the groomed gardens or lack thereof. Again with a risk of oversimplication, but to approximate a computational metric, we define complexity as the amount of disorder in terms of distribution of urban elements in the scene. As described before, we use SegNet [4] , to extract a 12 dimensional stochastic vector consisting of the proportion of pixels belonging to each element for a given image. For a given image, we store these proportions into a stochastic vector $XH(X)$ on that vector:

$$H(X) = -\sum p(i)\log p(i) \tag{3}$$

The i in Eq 3 is the SegNet dimension for one of the 12 objects. The entropy value $H(X)$ would become a proxy for visual complexity of the urban scenes. What we want to understand is not the absolute nature, but the trend in the variation of this value across the beautification process.

4.2 Testing Hypothesis

We now try to reason about different hypothesis inspired from the literature [2, 9, 10] using the computational approaches mentioned above. The main outcome of this excercise is either to confirm or refute whether we can explain how our model understands beauty using popular urban design concepts. Without the loss of generalization, we want to probe whether the elements that the pipeline deems beautiful or ugly, are grounded in literature.

4.2.1 Walkability and Green spaces.

As mentioned in table 3 , Walk-ability of streets has high impact on the beauty and other aesthetic qualities of a place. We test this by quantifying the amount of greenery and walkable elements added by the beautification process. First, we transform 500 images from both sides of the beauty spectrum to either beautiful (if ugly) or ugly (if beautiful) urban scenes.

After detecting LAWN categories through PlacesNet labels, we compute, for each image, the difference between the category frequency before and after transformation (e.g. how many 'Walkable' labels are added after beautification?). We then plot the aggregated difference-distributions for beautified and uglified image sets in Fig 6.

[H1] Walkable streets is favoured in beautiful urban scenes

To test that *H1* is valid, we first plot a prevalence count of different categories of labels for Beautified and Uglified images. It can be seen from Fig 6, that walkable scene types are highly favoured in the beautification process. Ugly images are transformed into Walkable spaces almost twice as frequently in beautification compared uglification.

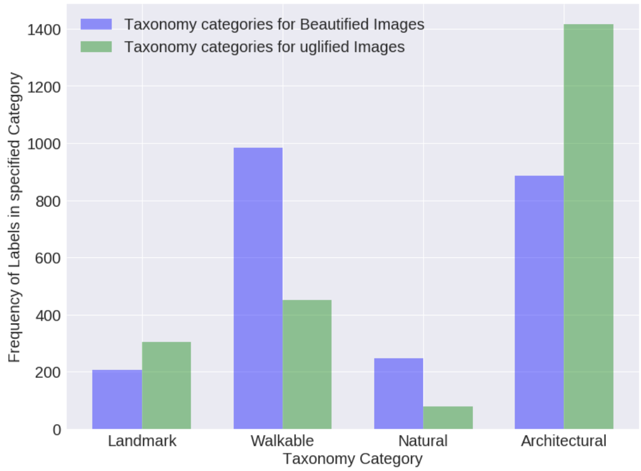


Figure 6: Prevalence plot of categories of scenes prevalent in Beautified against Ugly-fied images

[H2] Green spaces are favourable for beauty in urban scenes.

Figure 6 and 7 implies that natural scenes are twice as likely in beautified images than in uglified images. To test this hypothesis further, we further analyze the percentage of 'tree' pixels (according to SegNet) added by the beautification process: in average **[XXX% pixels]** of greenery are added after beautification.

4.2.2 Privacy and Openness .

From the literature, it is conjectured that privacy is great when one looks at personal spaces, but when it comes to public settings, there is an inverse 'U' relation with how private a place feels like. Too much privacy discourages the fundamental human urge to explore a mystery. Too much openness alerts the primal urge to feel safe.

[H3] Sense of Privacy has an inverse 'U' relationship with the sense of beauty

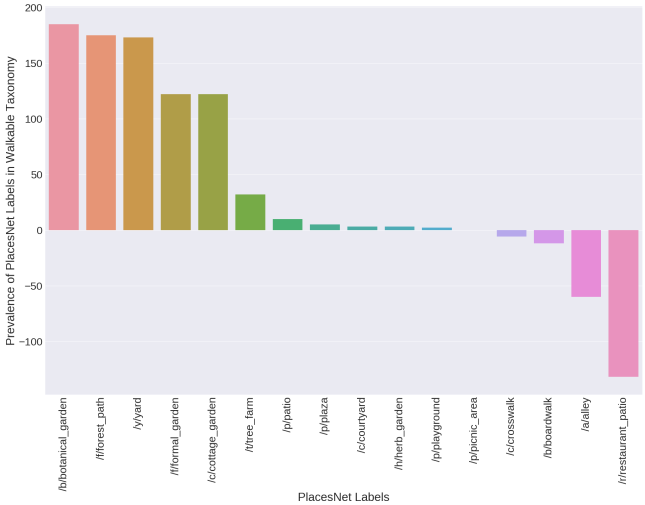


Figure 7: Prevalence of Walkable labels in Beautified images against Ugly

What *H3* suggests is that sense of privacy is not always associated with beauty. To understand the relationship between openness and beauty in urban scenes, we employ a technique called binned plots **CITE []**. We bin the range of sky pixels into **XXX Bins**Each image is then assigned to the bin corresponding to its proportion of sky pixels. Among the 1000 images (500 uglified, 500 beautified) in our data, we then repeatedly sample 100 images across bins, and count how many of the sampled images fall in either beautified or uglified transformed category. We plot the mean and standard deviation in a plot for these occurrence frequencies. The resulting plot gives a trend about how likely is the presence of pixels favoured or disfavoured by the beautification process.

It can be seen that our model prefers lack of openness in the beautification process. The inverse 'U' Relationship is completely absent and cozy urban places are actually favoured in the beautification process.

4.2.3 Visual Complexity .

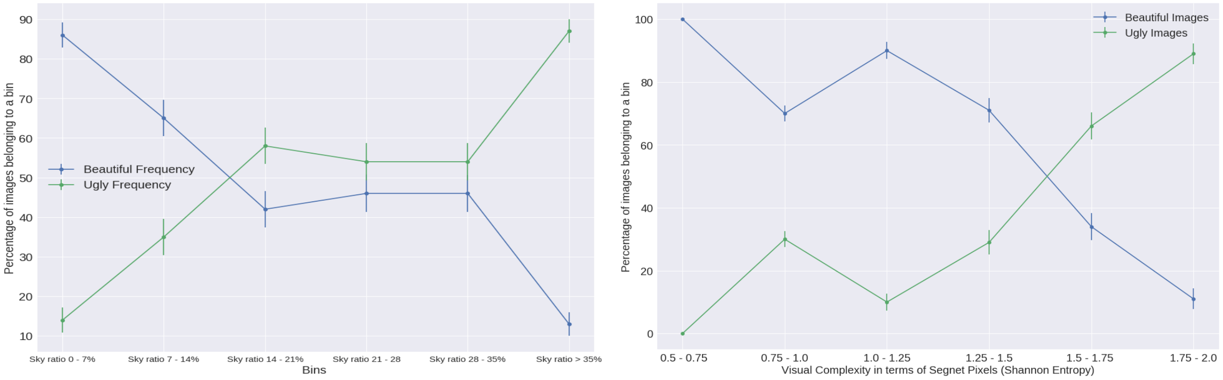
Visual complexity is a metric to measure the diversity of an urban scene. There is a trade off when it comes to balancing visual complexity with beauty. Too much diversity overwhelms the cognition and makes it hard to establish bearing.

[H4] Visual Complexity has a inverse 'U' relation with the sense of beauty

To test this hypothesis, we compute the binned plot (see Sec. 4.2.2) of our complexity metric. It can be seen from Fig 8b that visual complexity does peak in beautiful images but then deteriorates rapidly.

4.3 Interdependence of Urban elements

To understand the predictability and interdependence between the most influencing objects in an image and the probability of finding an image beautiful, we adapt the approach as described in [31], which proposes using logistic regression coefficients as a measure for upper bounds on influence of a variable. This method is also helpful



(a)

(b)

Figure 8: Figure 8a shows the Binned Plot for Sky pixels across transformed images and Figure 8b shows the Binned Plot for Visual Complexity across transformed images

in understanding the interdependence of variables for a particular outcome. Using the approach we perform a logistic regression over the two variables V_1 and V_2 which denote the ratio of a particular type of object pixels to the total area of image in pixels. So these ratios basically represent how much of the total image area is dominated by a particular object. The objects in this study are limited to the 12 urban object labels supported by SegNet [4]. Assuming dependence, we introduce a third term, which represents the factor that measures the dependence of V_1 and V_2 and is simply the product $V_1 * V_2$. The logistic regression would try to fit a line

$$L = invLogit(\alpha + \beta_1 * V_1 + \beta_2 * V_2 + \beta_3 * V_1.V_2) \qquad (4)$$

Here $invLogit$ is the inverse Logistic function. According to the rule of fourth described in [31] , the coefficients $\beta_1, \beta_2 and \beta_3$ represent some properties about the influence of V_1, V_2. An increase in 1% area of the amount of pixels belonging to object V_2 would correspond to a maximum increase/decrease of $\frac{\beta_2}{4}$ percent in the likely hood of the image being beautiful. Same rule applies for β_1. In case of β_3 the co-efficient corresponds to mutual dependence. An intuitive explanation is that if a 1% change in value of V_1 would add β_3 to the value of β_2. Hence β_3 links changes in variable V_1 to changes in influence of variable V_2 and vice versa. We perform the logistic regression on the 5 prime influences namely *Sky , Buildings , Road , Vehicles , Trees*. The results of pairwise regression along with the dependency term are summaries in the table 4

Object pair	β_1	β_2	β_3	Error Rate (Percentage)
Roads - Vehicles	-0.015	-0.05	0.023	40.6
Sky - Buildings	-0.08	-0.11	0.064	14.4
Sky - Trees	0.03	0.11	-0.012	12.8
Buildings - Trees	-0.032	0.084	0.005	12.7
Roads - Trees	0.04	0.10	-0.031	13.5
Roads - Buildings	-0.05	-0.097	0.04	20.2

Table 4: Regression coefficients

5 DISCUSSION

5.1 Explainability validation

As a part of a follow up project, we designed a web-app that explores the city of Boston and visualizes the urban beautification metrics in terms of changes that the pipeline suggested. As of now, this app is not available to a wider audience for the sake of preserving anonymity. However, the app was used to understand what experts think about the designed metrics and the insights. We sent this app to a set of 30 experts in the fields of architecture, urban planning and some from data visualization to use. Out of the addressed, 20 experts participated in the validation process and filled out a survey that quantified their understanding of the changes visualized in the beautification process. They also answered free from questions about the utility of such a pipeline for urban planners. Some comments like" *The maps reveal patterns that might not otherwise be apparent* " or " *The tool helps focusing on parameters to identify beauty in the city while exploring it.* " or " *The metrics are nice. It made me think more about beautiful places needing a combination of criteria, rather than a high score on one or two dimensions. It made me realize that these criteria are probably spatially correlated.* " pointed to the fact that such a tool does add value to the understanding of urban design from the aspect of a wider perception of beauty. The work on a larger analysis of the survey and the visualization techniques is under review

5.2 Limitations and biases

Like any supervised deep learning based framework, this work is only able to learn what is present in the data. Hence the method of acquiring annotations for urban images can introduce huge biases in the model. The current model is trained on images acquired from the study on streetscore [19]. However their annotation is open to general public and there is not way we can remove biases that come with culture and location, in a highly subjective effect like beauty. Moreover because the pair wise choice is simply done by clicking one of the two images, the data might have noise introduced by non-serious participants. Such biases are bound to be picked up by the deep learning model. One can argue that the preference of our model for greenery , is a form of bias in the data. Another Limitation of our work is in the metric formation. The computational metrics developed to capture the real urban design metrics are designed using heuristics. There needs to be more crowd and expert validation to establish the validity of their formulation.

5.3 Future work

The pipeline is generalizable for geotagged and annotated images. The aim of this paper is to propose a pipeline with uses state of art methods in generative models to understand perception of emotions in urban images and explain them. As an extension, understanding how intervention would look like against outcome variables such as depression, safety or mental well-being in general would be very valuable.

REFERENCES

[1] Luca Maria Aiello, Rossano Schifanella, Daniele Quercia, and Francesco Aletta. 2016. Chatty maps: constructing sound maps of urban areas from social media data. *Royal Society open science* 3, 3 (2016), 150690.

[2] Christopher Alexander, Sara Ishikawa, Murray Silverstein, Joaquim Romaguera i Ramió, Max Jacobson, and Ingrid Fiksdahl-King. 1977. *A pattern language*. Gustavo Gili.

[3] Ebru Arisoy, Tara N Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. 2012. Deep neural network language models. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*. Association for Computational Linguistics, 20–28.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2015. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561* (2015).

[5] Ritendra Datta and others. 2008. Algorithmic inferencing of aesthetics and emotion in natural images: An exposition. In *2008 15th IEEE ICIP*. IEEE, 105–108.

[6] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. 2012. What makes paris look like paris? *ACM Transactions on Graphics* 31, 4 (2012).

[7] Alexey Dosovitskiy and Thomas Brox. 2016. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4829–4837.

[8] Abhimanyu Dubey, Nikhil Naik, Devi Parikh, Ramesh Raskar, and César A Hidalgo. 2016. Deep learning the city: Quantifying urban perception at a global scale. *arXiv preprint arXiv:1608.01769* (2016).

[9] Reid Ewing and Otto Clemente. 2013. *Measuring urban design: Metrics for livable places*. Island Press.

[10] Clemente Otto Ewing Reid. Measuring Urban Design - Metrics for Livable Places. (????).

[11] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028* (2017).

[12] Ralf Herbrich, Tom Minka, and Thore Graepel. 2007. TrueSkillâĎć: a Bayesian skill rating system. In *Advances in neural information processing systems*. 569–576.

[13] Aditya Khosla and others. 2014. What makes an image popular?. In *Proceedings of the 23rd WWW*. International World Wide Web Conferences Steering Committee, 867–876.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[15] Stephen Law, Yao Shen, and Chanuki Seresinhe. 2017. An Application of Convolutional Neural Network in Street Image Classification: The Case Study of London. In *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery (GeoAI '17)*. ACM, New York, NY, USA, 5–9. DOI:http://dx.doi.org/10.1145/3149808.3149810

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, and others. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[18] Nikhil Naik, Scott Duke Kominers, Ramesh Raskar, Edward L Glaeser, and César A Hidalgo. 2017. Computer vision uncovers predictors of physical urban change. *Proceedings of the National Academy of Sciences* 114, 29 (2017), 7571–7576.

[19] Nikhil Naik, Jade Philipoom, Ramesh Raskar, and César Hidalgo. 2014. Streetscore-predicting the perceived safety of one million streetscapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 779–785.

[20] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. 2016. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*. 3387–3395.

[21] Daniele Quercia. 2015. Chatty, Happy, and Smelly Maps. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 741–741.

[22] Daniele Quercia, Rossano Schifanella, and Luca Maria Aiello. 2014. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *Proceedings of the 25th ACM conference on Hypertext and social media*. ACM, 116–125.

[23] Daniele Quercia, Rossano Schifanella, Luca Maria Aiello, and Kate McLean. 2015. Smelly maps: the digital life of urban smellscapes. *arXiv preprint arXiv:1505.06851* (2015).

[24] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[25] Philip Salesses, Katja Schechtner, and César A Hidalgo. 2013. The collaborative image of the city: mapping the inequality of urban perception. *PloS one* 8, 7 (2013), e68400.

[26] Rossano Schifanella and others. 2015. An Image is Worth More than a Thousand Favorites: Surfacing the Hidden Beauty of Flickr Pictures. In *Proceedings of THE 9TH ICWSM 2015*.

[27] Chanuki Illushka Seresinhe, Tobias Preis, and Helen Susannah Moat. 2015. Quantifying the impact of scenic environments on health. *Scientific reports* 5 (2015), 16899.

[28] Chanuki Illushka Seresinhe, Tobias Preis, and Helen Susannah Moat. 2017. Using deep learning to quantify the beauty of outdoor places. *Royal Society open science* 4, 7 (2017), 170170.

[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[30] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. 2013. Deep neural networks for object detection. In *Advances in neural information processing systems*. 2553–2561.

[31] Brandon K Vaughn. 2008. Data analysis using regression and multilevel/hierarchical models, by Gelman, A., & Hill, J. *Journal of Educational Measurement* 45, 1 (2008), 94–97.

[32] Yilin Wang and others. 2015. Unsupervised Sentiment Analysis for Social Media Images. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*. AAAI Press. http://dl.acm.org/citation.cfm?id=2832415.2832579

[33] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. 2014. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*. 487–495.