# 2 Methods

## Problem setting

We cast the current task of determining facial attractiveness as a *supervised learning problem*. Therefore, a successful supervised learning algorithm should be able to, given an input image of a face, provide a reasonable measure of attractiveness, as specified by the training data. Indeed, this is the approach taken by most previous research endeavors (Altwaijry & Belongie, 2013; Fan et al., 2012).

Since attractiveness, considered as the “ground truth” of the task, is not an inherent property of a face, but rather an external value judgement, the labeling of faces is not included *per se*, but has to be performed by a human. Moreover, while most other supervised learning problems have unambiguous assignment of labels (there is little disagreement among human observers whether a cat or a dog is present in a picture), attractiveness has to be judged on an arbitrary scale. Additionally, even though objective criteria for attractiveness or *beauty* can be found across cultures (Little et al., 2011; Jones & Hill, 1993), unique subjective tastes or preferences has to address. Thus, the concept of a *universal attractiveness learner* seems untenable.

A more flexible and useful, albeit computationally extensive, method is to come up with a general architecture of a model, which can be re-trained for each individual observer (e.g., a user in a dating site) given a reasonably large number of input-output pairs in the form of images as inputs and corresponding judgements of attractiveness as outputs. Consequently, this is the approach we will take in the current work.

## Data preparation

In order to gather a sufficiently large, and, at the same time, realistic data set of faces, we used images from two publicly available databases, namely, the 10k US Adult Faces Database (Bainbridge et al., 2013), and the Labelled Faces in The Wild Database (Huang et al., 2007). From these combined sources, we filtered out 650 male and 650 female faces of people aged between 18 – 30 years old. Filtering was performed using both an automatic procedure, and visual inspection to correct for labeling errors present in the annotations. After this, each of the authors rated the images of opposite sex faces on a scale from 0 to 5 (0 indexing the subjectively felt complete lack of attractiveness, and 5 representing the subjective ideal of attractiveness) with a single decimal place allowed for more subtle differentiation of judgement. 50 images were kept separate as a test set, so a total of 600 images were included in the final training set. Histograms of the ratings of the three authors are given in **Figure 1**.

## Regression versus classification

Before specifying a model to train on the prepared data, one more relevant question had to be resolved, namely, whether to treat the ratings as a continuous or a categorical variable. This decision inevitably determines whether to use a regression or a classification algorithm respectively. Technically, the current operationalization of attractiveness ratings is merely a

## Using a pre-trained network

Since training an entire convolutional networks from scratch is a costly process that requires both a sufficient amount of computational power and a large enough dataset to capture the relationship between images and ratings, we opted for *transfer learning.* Transfer learning refers to the process of “borrowing” the layers of a previously trained (pre-trained) neural network, typically implemented for a similar task or in a similar domain. Since convolutional networks can be conceptualized as hierarchical feature extractors, the spatial hierarchy of features learned by the pre-trained network can be thought to represent a generic model for the visual world (Chollet, 2018; Goodfellow et al., 2016). Needless to say, this transferability is a unique advantage of deep learning models compared to shallow models.

If a pre-trained network is available for a given task, the next question that arises is whether to use the network as a fixed feature extractor or fine-tune its weights along the new application. Unsurprisingly, this decision depends largely on the peculiarities of the task at hand. As already mentioned above, facial attractiveness can be viewed as a value judgement of different facial features and combinations of features (Little et al., 2011), it is reasonable to suggest that a facial attractiveness classifier would utilize these features during training in order to capture the statistical regularities in a person’s judgement. Therefore, we opted for the option of applying a fixed feature extractor to the current task.

## VGG16-Face

As for the pre-trained convolutional network, we utilized the VGG16-Face architecture (Parkhi, Vedaldi, & Zisserman, 2015). The precise details of the network architecture and development context are described in the relevant paper, so in this subsection we merely sketch out its essential properties.

VGG16-Face war originally trained on 2.6 million images depicting more than 2.6K people in order to perform facial recognition (i.e., identifying a unique person across different images). The task was set up as an N-way classification problem. As the name suggests, the structure of VGG16-Face was inspired by the previous work of Symonyan and Zisserman (2015) which demonstrated the utility of deep convolutional networks in solving large-scale image classification problems, focusing, in particular, on the ImageNet dataset. Similarly, VGG16-Face consists of 11 blocks, each starting with a linear operator, followed by a non-linearity (ReLU), and applying a max pooling operation before passing in the activations to subsequent layers. The last layer of VGG16-Face outputs a 4,096-dimensional vector (considered as an *embedding* of an image), which is then passed through a *softmax* function in order to obtain the class probabilities.

For our current application, we removed the *softmax* layer and *froze* all previous layers, i.e., did not update their weights during backpropagation. This was done in order to prevent the algorithm from modifying the representations learned previously by these previous layers. Thus, we used the image embeddings returned by the last layer of the network as a starting point. The implicit assumption is, therefore, that the 4,096-dimensional vectors of extracted features will contain all the information needed for accurately predicting facial attractiveness. Finally, it is important to note, that using the pre-trained VGG16-Face in this manner has one more important advantage. Since the extracted facial features are considered to be invariant to head orientation, background, lighting and other low-level image properties such as brightness or contrast, the contribution of such factors to predicting attractiveness is vastly minimized. Such feature invariance is of great importance to real-world application, where the distribution of facial images exhibit high levels of natural variance.

## Current architecture

Having outlined the base of our model, we now turn to the expansion of the base model necessitated by the current task. The entire network architecture is illustrated in **Figure 2**. As evident from the graphical representation, the features extracted by VGG16-face were fed into a succession of three fully connected (FC) layers aimed at learning a user’s particular judgements of attractiveness. Finally, the output of the last FC layers was passed through a softmax function. Cross-entropy was used as the cost function to be minimized:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

In the above expression, *N* is the number of training examples, *K* is the number of classes, refers to the probability of training example *i* belonging to class *k*, and refers to the predicted probability that example *i* belongs to class *k*.

INSERT DISTRIBUTION OF RATINGS

INSERT FIGURE OF OUR FINAL NETWORK STRUCTURE

Mention the problem that rounding the ratings is problematic for the following reason. Predicting an image rated as 3.4 as belonging to 4-stars class is not that bad as predicting an image rated as 3.4 as belonging to a 5-stars class. However, cross-entropy treats these the same way.