Mimicing Face Pareidolia with Computer Vision

Roman Kessler

June 24, 2020

Abstract

Placeholder

1 Introduction

Face pareidolia is the phenomenon of seeing faces in other patterns, where actually no face is present. A famous example is the face on mars (Fig. 1). Similar examples are known to everyone, when looking at clouds and imagining faces.

However, there has been an interesting series of experiments showing, that when subjects imagined faces in random noise, the neuronal areas of face processing are more active when imagining faces than when not imagining faces in a probe image [1]. However, the nature of the face perception in random noise is of debate. Do images, labelled as face by a subject, contain some physical properties that may drive face perception (*bottom-up paredolia*). Or alternatively, is the task - the instruction to find faces - that strong, that the subject randomly "chooses" some images to contain faces (*top-down pareidolia*)? The answer is probably a combination of both effects, alongside with a variety of other effects.

The aim of this study is to find any physical correlate of a face in random noise. For this we used a deep convolutional neural network with a binary face-classifier.

2 Material & Methods

2.1 Subject and data

One subject was analyzed, which labeled 3295 random noise images with the extend of $43 * 32$px (Fig. 2). The subject was convinced by a cover-story that around 50% of images contain faces, which are barely visible. The appearing face should cover most part of the image. The subject has labeled 29% of the images as face, and the rest as non-face.

Figure 1: The *face on mars* illusion [2].

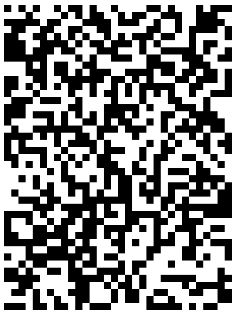
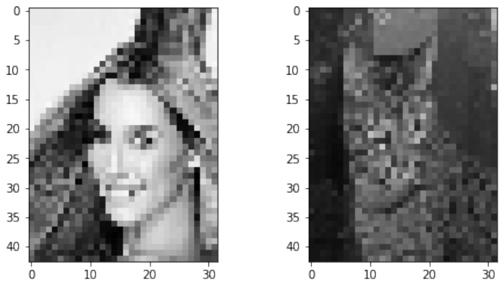


Figure 2: Example stimulus. All stimuli consisted of random black and white pixels.

2.2 Binary face classifier

To disentangle if there may be some physical correlate of a face in the stimuli, the subject labeled as "face", we constructed a binary face classifier. As a core network, we used the MobileNet architecture [3]. We cut the fully connected layers at the end of the network, and replaced them by two fully connected layers, whereas the very last layer produced binary probabilities for two classes. We allowed the full model to be trained (no frozen layers).



(a) (b)

Figure 3: Example training images after rescaling. (a) First category were faces of celebrities. (b) Second category consisted of cat and dog stimuli.

2.2.1 Training categories

To create a binary face classifier we needed face and non-face training images. For the face-class, we randomly chose 8000 images from the *CelebFaces Attributes (CelebA)* dataset [4], which contain faces of celebrities covering most part of the image area. For the non-face class, we chose 8000 cat and dog images from a public repository (`https://www.kaggle.com/chetankv/dogs-cats-image`).

2.2.2 Training image preprocessing

To train a model that shall later be tested on the noise images (Fig. 2), the images were first converted to greyscale, rescaled to $43x32$px, and lastly pseudo-converted to color because MobileNet expects 3 input channels (Fig. **??**).

2.2.3 Training hyperparameters

The training was conducted using Adam optimizer REF and categorical crossentropy as loss function. Only 5 epochs were repeated with batch size of 25 images. Already after 4 epochs, the training accuracy exceeded 0.99%, and cross-validation with a holdout dataset of 2000 images confirmed a less than 1% error rate.

2.3 Face classification in pure noise

We then used the model to predict the classes of the random noise images (Fig. 2). The prediction contains probabilities for both classes for each image, each summing up to 1. To test, if there is a difference (which we expected to be very tiny), we conducted a independent sample t-test on the probability of being categorized as face by the model. Therefore we split the probabilities according to the image labels, which were given by the subject.

3 Results

Data availability statement

All used data, analysis pipelines, as well as further data of more subjects and further experiments can be retrieved from the GitHub repository of the first author `https://github.com/kesslerr/facepareidolia`.

References

[1] K. M. Zimmermann, A.-S. Stratil, I. Thome, J. Sommer, and A. Jansen, “Illusory face detection in pure noise images: The role of interindividual variability in fmri activation patterns,” *Plos one*, vol. 14, no. 1, p. e0209310, 2019.

[2] M. J. Carlotto, “Digital imagery analysis of unusual martian surface features,” *Applied Optics*, vol. 27, no. 10, pp. 1926–1933, 1988.

[3] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[4] S. Yang, P. Luo, C.-C. Loy, and X. Tang, “From facial parts responses to face detection: A deep learning approach,” in *Proceedings of the IEEE international conference on computer vision*, pp. 3676–3684, 2015.

Supplementary Material

MobileNet adaption

```
model = Sequential()
```

```
model.add(MobileNet(weights='imagenet',include_top=False, input_shape = (43, 32, 3) ))
model.add(Flatten())
model.add(Dense(100, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(2, activation='softmax')) # sigmoid
```