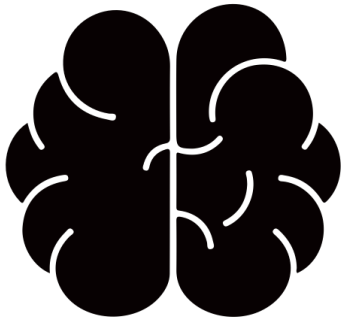
Adversarial Images Steering Human Attention

Lynn Le
s4596390



Master Thesis Medical Biology
at the department of Artificial Cognitive Systems
Radboud University, August 2019

Daily Supervisor: Dr. Umut Güçlü
Supervisor: Prof. Dr. Marcel van Gerven

Abstract

Adversarial examples are inputs that are slightly modified with the objective to mislead computer algorithms. Adversarial images, perceived as a great threat for computer vision models, has not yet been proven to be dangerous for the human visual system. Some slight effects of adversarial images on humans have recently been shown, however these results apply only in very limited experimental settings. It remains a question whether adversarial images designed to attack computer vision models can influence visual processes such as overt attention. In this study, we address this question by making use of state-of-the-art approaches to generate adversarial images with the intent to alter human attention. We train a CNN-ensemble to predict saliency maps and take the learned parameters to generate adversarial images with the pursue to influence human eye movements in the same way as the CNN-ensemble. We find that our adversarial images has fool the saliency prediction of well-performing convolutional network models and also has succeeded in steering human eye-movements to target locations.

Contents

List of Figures

1 Introduction

Artificial neural networks (ANNs) is a major branch in the field of artificial intelligence (AI), and is inspired by the human brain. One of the most dominant sort of ANN is the Convolutional Neural Network (CNN) (LeCun, Bengio, & Hinton, 2015; Krizhevsky, Sutskever, & Hinton, 2012). CNNs are predominantly used to solve complex image-driven pattern recognition tasks. The success of the CNNs is not only inspiring for the development of new technologies, but also aids in a better insight regarding the human brain. Research has been done using brain data obtained from humans and other primates to predict their behavior (O'Connell & Chun, 2018). CNNs have also been used to predict brain activity, from large-scale neural activations of specific regions (Güçlü & van Gerven, 2015, 2017), to the action potentials of individual neurons (Greene & Hansen, 2018). The fact that CNNs are capable of predicting biological activity indicates that the mechanisms and computational principles underlying CNNs may resemble those of biological brains.

Just as visual illusions take advantage of the shortcomings of the human visual system, adversarial images takes advantage of the shortcomings of CNNs. Adversarial images however, can have severe impact on the performance of machines, causing them to misclassify images, which is a fundamental challenge for CNNs. The existence of adversarial images is believed to be a threat against machine-vision systems in applied settings, and some researchers believe it reduces their advantage for understanding the human mind (Zhou & Firestone, 2019). However, if we are able to make adversarial images that have an effect on both CNNs and the human visual system, the effects of adversarial optimization becomes a powerful tool to understand both CNNs and the brain.

There are striking similarities between the human visual system and deep learning models as underlined by studies such as those of Güçlü and van Gerven (Güçlü & van Gerven,

2015, 2017). Recently, experimental evidence has indicated that specific types of adversarial attacks can be constructed that also have an impact on the judgement of time-limited humans when forced to categorize particular images (Elsayed et al., 2018). They were able to fool time-limited humans, but non-time-limited humans did not misclassify it. However, because humans are the ones labeling classifications, it is difficult to decide whether a sample is adversarial as opposed to a sample's class being changed. Additionally, it would be more of an adversarial image if it was able to affect humans which are not necessarily time-limited. Therefore, in this study, we aim to generate adversarial images to deceive computer models and humans using neural networks, but we do not enforce categorization tasks. Instead, we aim to make adversarial images based on adversarial saliency targets and CNNs that are trained to predict visual saliency.

Visual saliency has been of great interest in both neurobiology and computer vision and is a great way to test human attention. Attention and visual saliency are the product of a complex set of processes involving diverse neural mechanisms working simultaneously. Understanding the nature of these processes and their interplay is challenging. Even considering decades of research involving visual psychophysics and accumulating neurophysiological data, a consensus on the specific mechanisms involved remains elusive (Carrasco, 2011). We do know that certain visual patterns tend to draw overt attention causing eye-movements towards that area (Torralba, Oliva, Castelhano, & Henderson, 2006). In this study we want to see whether deep neural networks can learn these patterns that influence eye-fixation and use the learned information to manipulate eye movements. If we reach the level of attacking human eye fixations in such settings, it can be can be safely assumed that the model makes use of the correct information representing the mechanisms of the human brain to predict saliency on an image.

Here, we make use of well-known pretrained models to characterize visual salience. We use the saliency models to generate adversarial images. The performance of the saliency models will then be tested based on the generated adversarial images and we want to see whether these adversarial images will change human eye-fixation when compared with ground truth images.

Outline

Chapter 2 explains the full architecture and functions of the models that are used in our experiments and the techniques that were used to make the adversarial images. Chapter 3 briefly explains the human visual system and how we use our adversarial images to test the transferability from models to the human brain. The last chapter brings everything together in a discussion and conclusion.

2 Artificial Neural Networks

2.1 Introduction

AI models are characterized by their flexibility and capability to integrate different methodologies emulating biological systems behavior. Machine learning (ML) is a branch in AI which makes use of algorithms that aims to recognize patterns, classify and predict based on large amounts of data (Solomonoff, 1957). Despite having studied the brain for centuries, it remains an unclear consensus on how it works for scientists. There are two main theories on how the brain represents and computes information. Some theories propose that each individual neuron can retain a high amount of information and represent complex ideas (Gross, 2002). A popular belief though, is that the brain represent information via population, or distributed, coding and that individual neurons do not carry dense information (Thorpe, 1998). Artificial neural networks (ANNs) are models built to detect patterns, based on the second theory. ANNs are self-adaptive, non-linear, data-driven models that do not require specific assumptions regarding the underlying model.

2.1.1 The perceptron

The perceptron is a form of a linear discriminant model and is introduced by Frank Rosenblatt in 1962. The model consists of two classes in which the input vector x is first transformed nonlinearly with a fixed function to give a feature vector $\phi(\mathbf{x})$, which is used to construct a generalized linear model in the second class. The perceptron model is in the form

$$y(x) = f\left(\mathbf{w}^T \phi(\mathbf{x})\right) \tag{2.1}$$

where the activation function $f(\cdot)$ is given by a step function in the form

$$f(a) = \begin{cases} 1, \text{ a} \geq 0 \\ \text{-1, a} < 0 \end{cases} \tag{2.2}$$

The vector $\phi(\mathbf{x})$ typically includes a bias component $\phi_0(\mathbf{x}) = 1$. Because the perceptron is not a probabilistic model, it uses target values of $t = 1$ for class A and $t = -1$ for class B which matches the choice of activation function (Rojas, 2013). The perceptron is the simplest form of artificial neural network and can only be applied to linearly separable data.

2.1.2 Multilayer perceptron

A feedforward network, also called a multilayer perceptron, is a complex form of perceptrons that is able to deal with nonlinear outputs and are essentially classic deep learning models. Feedforward models learn in order to approximate a function f. For instance, for a classifying model, $y = f(x)$ predicts an output y based on the input x. The network predicts the relationship between x and y, $y = f(x;\theta)$ by learning the parameters θ that produces the best function approximation (Schmidhuber, 2015). Ultimately, information passes through the function being evaluated from x, then through the intermediate computations used to define f, and finally to the output y, resulting in a feedforward mechanism (Schmidhuber, 2015).

2.1.3 Deep Neural Network

A deep neural network is a collection of artificial neurons organized in a sequence of multiple layers, where each of these neurons receive activation signals from other neurons in a previous layer and which the neuron uses to perform computations. The neurons cooperatively carry out a complex mapping from the input to output. This mapping is learned from the data by adapting the parameters of each neuron using backpropagation (Rumelhart, Hinton, Williams, et al., 1988).

Convolutional neural networks

CNNs are primarily used in the field of pattern recognition within images. Just like other DNNs, CNN's neurons self-optimise through learning. For instance, an image classifier CNN model takes in raw image vectors as input and uses a chosen loss function to obtain the final

output of the class prediction score. Generally, CNNs are comprised of three types of layers. These are convolutional layers, pooling layers and fully connected (FC) layers. Convolutional layers contain neurons organized into three dimensions: the size of the input (height x width) and the depth of the input. The depth refers to the different channels of the training images (e.g. red, green blue channels). Pooling layers perform downsampling along the spatial dimensions of the input, reducing the number of parameters within that activation. The FC layers produce class scores from the activations. Overall through this process of transformation, CNNs transform the original input, layer by layer, using convolutional and downsampling techniques to produce predictions for image recognition tasks. The architecture, learnable kernels, and features of CNNs and their hyperparameters such as depth, stride and padding to optimize output, have been explained and used widely and can be found in various papers (Rastegari, Ordonez, Redmon, & Farhadi, 2016; Long, Shelhamer, & Darrell, 2015; O'Shea & Nash, 2015; Sermanet et al., 2013; Zeiler & Fergus, 2014; Dong, Loy, He, & Tang, 2015; Simonyan & Zisserman, 2014).

2.1.4 Loss functions

To denote how far off the model's prediction **Y** is from the target vector **T**, an objective function is defined that typically computes a scalar loss $\mathscr{L}$, for instance the mean-squared-error (MSE) loss ($\mathscr{L}_{MSE}$). The chosen loss function measures the costs for incorrect predictions, and as a reaction thereof, the parameters are modified. Different algorithms mostly differ from each other by the loss function that is minimized.

2.1.5 Adversarial Images

Adversarial images contain perturbations that lead the prediction model to predict differently than they were trained to be. As demonstrated for the first time in 2013 (Szegedy et al., 2013), changing pixels on an image with small perturbations can cause a model to misclassify that image. Adverarial examples are defined as slightly modified samples of input data intended to cause a machine learning classifier to misclassify it (Kurakin, Goodfellow, & Bengio, 2016). While Luo et al. have proposed a method to generate attacks with high noise tolerance and taking into account the human perceptual system (Luo, Liu, Wei, & Xu, 2018), two recent papers suggested that humans can to some degree be affected by adversarial perturbations. Zhou and

Firesone have shown that humans are able to intuitively predict how a machine will classify adversarial images (Zhou & Firestone, 2019), and Elsayed et. al showed some effects of adversarial images on human classification (Elsayed et al., 2018). However, these results apply only in very limited experimental settings (e.g. in Elsayed's study a very short viewing times) and require relatively large and transferable perturbations, which often tend to yield meaningful features resembling the target class.

In our case, we will not have large perturbations that will aim to change a class of an image, instead, we make subtle perturbations with the goal to change subconscious eye movements. We propose a method that next to the pixel loss, makes use of the total variation loss $\mathscr{L}_{TV}$, spatial blurring at each image location. This limits the model to the same perception capabilities as the human visual range. And because the $\mathscr{L}_{TV}$ is fully differentiable, it allows gradients to backpropagate through the network when generating the perturbations. More detailed on how we generated the adversarial images are explained in the following section.

2.2 Materials and Methods

2.2.1 Data

We used the SALICON dataset that is introduced in 2015 (Jiang, Huang, Duan, & Zhao, 2015) and is the largest dataset available for saliency prediction. The data set contains 10,000 training images, and 5,000 validation images with saliency targets. Instead of using eye-tracking devices, SALICON uses a method involving a mouse (Jiang et al., 2015). Although this data collection method may affect the accuracy and quality of the dataset, it is still an acceptable and scalable proxy for the collection of ground-truth data. The testing data is not provided with ground truth targets, so we split the 5000 validation images into two, resulting in 2500 test data, and 2500 validation data. The input images (**X**) are jpg format and were RBG with a shape of (3, 480, 640) pixels and the corresponding saliency target heat maps (**T**) are png format with a shape of (1, 480, 640).

Data preprocessing

Prior to using the images to train the models, each **X** is normalized and is kept the original size (3, 480, 640). Every heatmaps **T** is resized to (1, 48, 64). Both **X** and **T** are then normalized in

order to match the input requirements of the pretrained models in PyTorch.

2.2.2 Training

CNN-ensemble

We constructed an ensemble of four CNN models trained on ImageNet, and one CNN model trained on the SALICON training set. The model trained on SALICON dataset is a self-made architecture, and each of the other models is an instance of one of these architectures: Inception V3, DenseNet, AlexNet, Sueezenet. The CNN-ensemle is trained on saliency maps representing eye fixations on a particular image. Diverse visual features can be learned to predict saliency maps, and then the features can be used to generate adversarial images. In our experiment, we do that by making the CNN-ensemble that is trained on the SALICON dataset and then use the features of the model to perturb the original input in a way that the CNN-ensemble starts predicting a false saliency map for that image.

Modifying pretrained models

Because the pretrained models are classifiers, the output layers are FC layers outputting class scores. In order to use the pretrained models as saliency models, the last layers had to be replaced with various layers in order to obtain the desired output dimension. To avoid loosing extreme spatial information that the pre-trained model has learned, the layers of each model are carefully inspected and selected such that no layer that caused excessive dimensionality reduction is included in the final ensemble. A schematic overview of the selected layers can be found in figure 2.1 and table 2.1.

MSE Loss Function

For training the last layers of each network, we utilized the mean squared error (MSE) loss function that considers the Euclidian distance of the salient areas between the ground truth saliency map and the predicted output. It is defined as:

$$\mathscr{L}_{MSE}(t,y) = \frac{1}{N}\sum_{j=1}^{N}(t_j - y_j)^2 \tag{2.3}$$

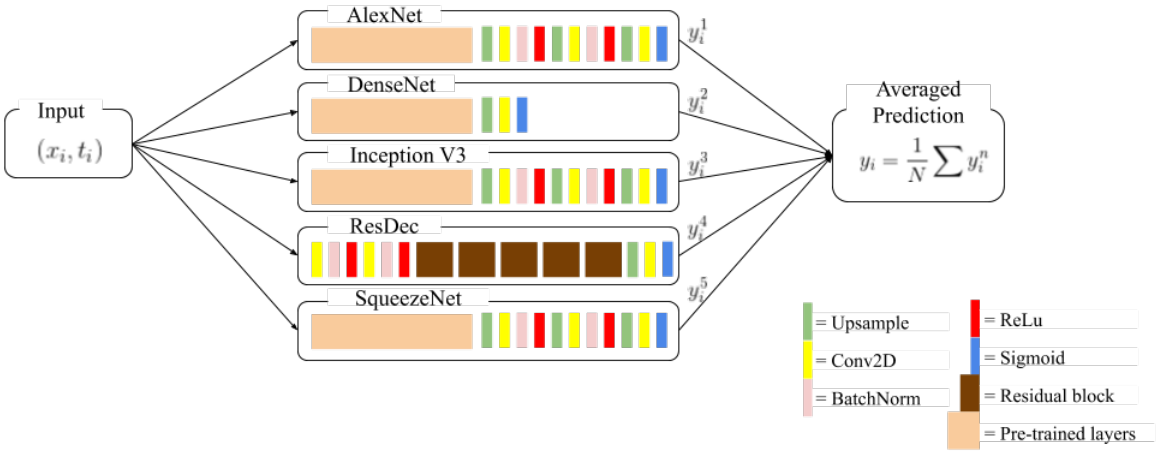


Figure 2.1: Architecture of the CNN-ensemble Using different pretrained networks to extract their pretrained layers and then combining them with upsampling and convolutional layers to obtain the correct dimentions for the output.

	AlexNet	**DenseNet**	**InceptionV3**	**SqueezeNet**
Pretrained	features[:5]	- features[:7] - ReLu	- until .Mixed_5d(x)	- features[:8] - ReLu
Upsample	48	(48+3, 64+3)	48	48

Conv2D	in=192; out=192;
	k=3; stride=1;
	pad=1
		in=512; out=1
		k=6; stride=1;
		pad=1
			in=288; out=288;
			k=3; stride=1;
			pad=1
				in=256; out=128;
				k=3; stride=1;
				pad=1

BachNorm	192		288	128
Upsample	48		48	48

Conv2D	in=192; out=192;
	k=3; stride=1;
	pad=1
			in=288; out=144
			k=3; stride=1;
			pad=1
				in=128; out=64;
				k=3; stride=1;
				pad=1

BatchNorm	96		144	64
Upsample	(48+3, 64+3)		(48+3, 64+3)	(48+3, 64+3)

Conv2D	in=96; out=2;
	k=6; stride=1;
	pad=1
			in=144; out=1;
			k=6; stride=1;
			pad=1
				in=64; out=1;
				k=6; stride=1;
				pad=1

Table 2.1: Modification details of the pretrained models The first row 'Pretrained' show which layers were selected from the pretrained model to use. Then the rest of the rows indicate what values were used for which layer. The full architecture of the model is shown in Figure 2.1.

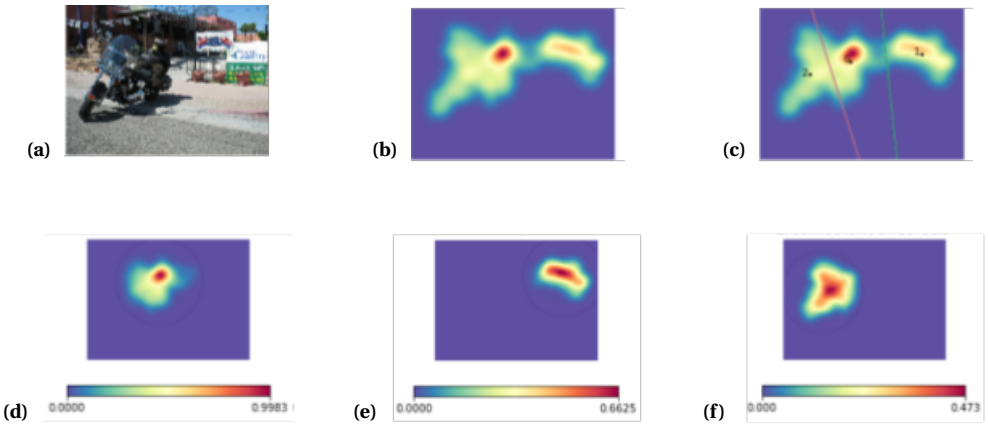


Figure 2.2: Selecting adversarial clusters An algorithm is written to detect and select the clusters that are not the largest and make them into new targets.

MSE loss generally has been chosen either directly or with some variations in other studies for visual saliency prediction (Pan et al., 2017; Jetley, Murray, & Vig, 2016).

2.2.3 Generating adversarial images

For each image, we want to generate perturbation to add to the original image, giving adversarial attacks that transfer across models. This means, for an image input image **X** (fig. 2.3a) with a target **T** (fig. 2.3b), we generate adversarial perturbation such that models will predict adversarial heatmaps $\mathbf{T}_{adv}$ (as fig. 2.3e). This way, a different perturbation is constructed for each image, however each perturbation contains a fixed ratio of weight parameter for pixel loss ($\mathscr{L}_{MSE} * \alpha$) and total variation loss ($\mathscr{L}_{TV} * \beta$).

A k-nearest neighbor algorithm is written to generate $\mathbf{T}_{adv}$. For each **X**, the algorithm locates random k_i centroids, and for each pixel it determines the cluster that it will be in which corresponds to the nearest k_i centroid. Then it calculates the new centroid of the formed k_i cluster, using the weighted average of the centroid where the pixel values on the heatmap are the weights. A cluster centroid is randomly selected that reach above a speciofic threshold, while eliminating the largest cluster.

A gaussian blur is then placed on the location of the selected centroid, smoothening out the cluster. The size of the gaussian blur depends on the pixel density of the cluster. The clusters are shown in figure 2.2.

We have a model that predicts a saliency heatmap $\mathbf{Y}$ containing two clusters at position k_1 and k_2 based on an input image $\mathbf{X}$. The k-nearest neighbor method is used to make adversarial targets resulting in $\mathbf{Y}_{adv}$ containing only one cluster k_1. We want to modify the image $\mathbf{X}_{adv}$ by minimizing the loss between $\mathbf{Y}_{adv}$ and $\mathbf{X}_{adv}$. With the parameters of the trained CNN-ensemble, we can perform iterated gradient descent on $\mathbf{X}$ in order to generate our $\mathbf{X}_{adv}$ (see algorithm 1.). This iterative approach is often used to generate adversarial images.

Loss functions for perturbations

In order to make our adversarial images, we need to use a pixel loss and total variation loss.

Pixel loss: The pixel loss is computed in a per-pixel basis, where each value of the predicted saliency map $\mathbf{Y}$ is compared with its corresponding target $\mathbf{T}$. For the pixel loss we used the MSE, which is explained in equation (2.3).

Total-variation loss: Total variation loss acts as a spatial smoother to regularize image and prevent its denoising. Because we wanted to also fool humans and not only computers we decided to use the total variation loss method as follows:

$$\mathscr{L}_{TV}(\mathbf{P}) = \big(\sum_{i=1}^{M}\sum_{j=1}^{N-1}(\mathbf{P}_{i,j} - \mathbf{P}_{i,j+1})^2\big) - \big(\sum_{i=1}^{M-1}\sum_{j=1}^{N}(\mathbf{P}_{i,j} - \mathbf{P}_{i+1,j})^2\big) \tag{2.4}$$

where $\mathbf{P}$ is the perturbation matrix and NxM are the dimension of $\mathbf{P}$. The total variation (TV) loss encourages spatial smoothness in the generated image. By reducing the total variation of the signal subject to it being a close match to the original signal, removes unwanted detail whilst preserving important details such as edges.

Adversarial loss: We use the combination of both pixel and total variation loss for the calculation of the perturbation.

$$\mathscr{L}_{adv}(\mathbf{Y},\mathbf{T},\mathbf{P},\alpha,\beta) = \mathscr{L}_{MSE}(\mathbf{Y},\mathbf{T}) * \alpha + \mathscr{L}_{TV}(\mathbf{P}) * \beta \tag{2.5}$$

Perturbations

The perturbations were then made by first initializing a tensor of random numbers between 1 and 0, then using the ADAM optimizer (Kingma & Ba, 2014) to optimize the adversarial loss using the following algorithm:

Algorithm 1. Adversarial Perturbation

Input: $\mathbf{X}, \mathbf{T}, \alpha, \beta, \epsilon$

Output: $\mathbf{X}_{adv}$

Initialize $\mathbf{P} \leftarrow 0$

for $i = 1, ..., I$ **do**

$\mathbf{X}_{adv} \leftarrow \mathbf{X} + \mathbf{P}$

$y \leftarrow M(\mathbf{X}_{adv})$

$l \leftarrow \mathscr{L}_{adv}(\mathbf{Y}, \mathbf{T}, \mathbf{P}, \alpha, \beta)$

$\mathbf{P} \leftarrow \mathbf{P} + \nabla l \cdot \epsilon$

end for

return $\mathbf{X} + \mathbf{P}$

2.3 Results

2.3.1 Making adversarial images and testing the CNN-ensemble

While training the model's, the losses of the training and test set are monitored in order to stop the training in case of overfitting. In figure A of the appendices, the plots of the losses over epochs are shown and the plots show until which epoch each model is trained. After training the model, We first assess the transfer of our constructed images to the CNN-ensemble and showed that the models perform well on both images images. Attacks using adversarial images always succeeded in changing the predictions of the test models (see fig. 2.3 for example predictions).

2.3.2 Testing the adversarial images on separate models

So far, only the CNN-ensemble has been tested, which does not show any transferability of adversarial attack amongst different models. We therefore also tested the performance of different models: The 5 individual models that make up the ensemble shown in figure 2.1 and two well-performing models (ResNet152 and VGG19) that were not used in the generation of the adversarial images (He, Zhang, Ren, & Sun, 2016; Simonyan & Zisserman, 2014). Just as the other 5 models, we took the pre-trained ResNet152 and VGG19 model and then replaced last

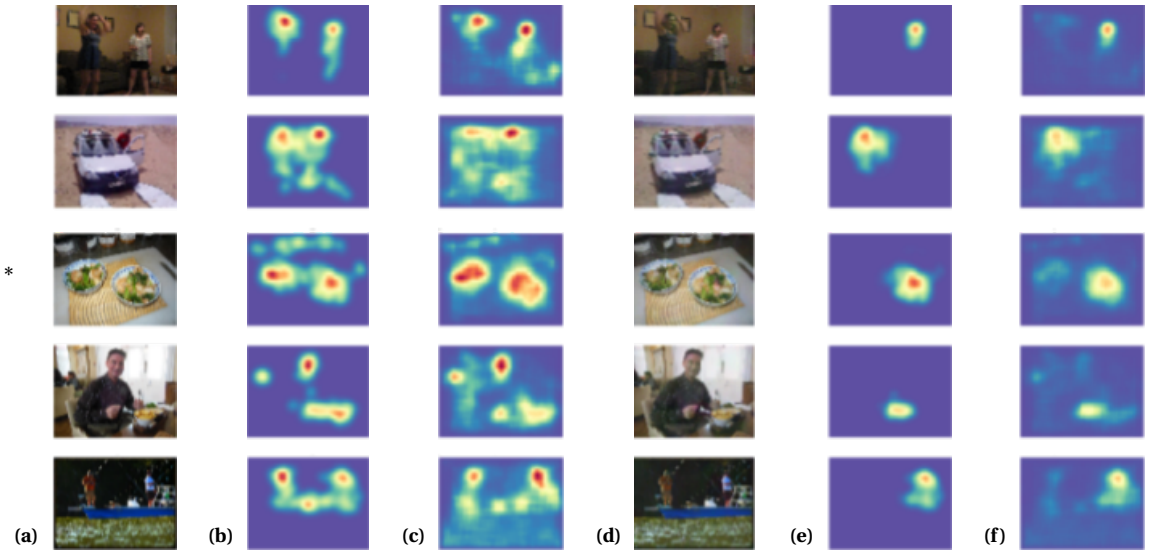


Figure 2.3: Model's prediction based on ground truth and adversarial images. The images in the first three columns (a - c) are the ground truth images (a) from the validation test set along with each corresponding ground truth target (b) and model's prediction. The ground truth image (a) is used to generate adversarial images shown in the fourth column (d), which are based on the adversarial target (e), and finally used to test the model's prediction (f).

few layers with our own layers so that the model becomes a salience model. The source code of the modifications can be found in the GitHub repository *Adverliency* of account *lelynn*. Some example prediction of the adversarial images are visualized in figure 2.4. We can see that for the models that were an element of the CNN-ensemble (fig. 2.4a), the predictions for the adversarial images appear to be accurate and so do the ground truth predictions. For the models that were not part of the CNN-ensemble (Fig. 2.4b): the prediction for the adversarial input by the ResNet, seems accurate, however the prediction by the VGG seem to be similar to the ground truth. This tells us that for this particular image, the adversarial attacks is transferable to all the models that make up the CNN-ensemble and also the ResNet152 model, but not to the VGG19 model.

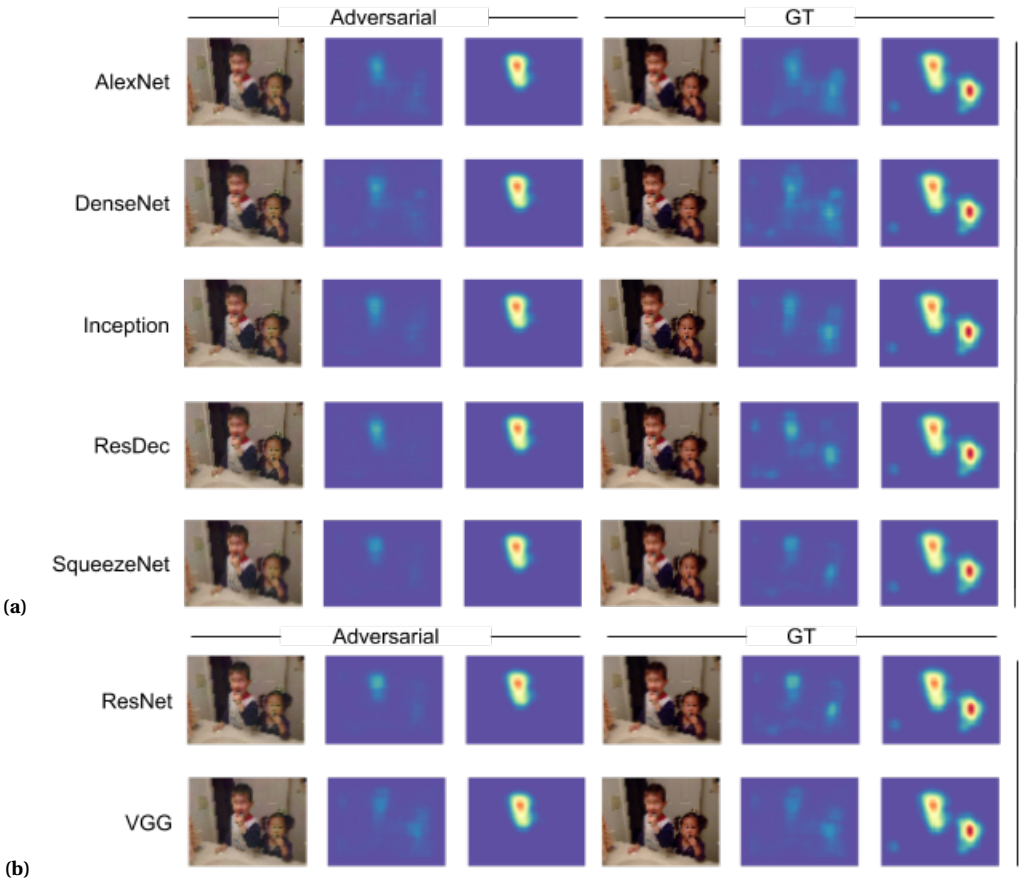


Figure 2.4: performances of the semi-pretrained individual models. The first three columns belong to the adversarial images and the last three belong to ground truth images. Column 1 and 3 are the input, column 2 and 5 are predictions, and column3 and 6 are the targets. Top 5 rows (a) are predictions from models that have been used to make an enseble and the ResNet an VGG were not present in the architecture of the CNN-ensemble.

3 Adversarial images for humans

3.1 Introduction

Since humans and other primates have retinas that limit visual details, it is a natural process to make eye-movements continuously in order to observe the surrounding environment. When presented free-viewing image, visual cues will rival for our attention based on bottom-up signals, affecting which directions our eyes will move. As shown in the previous chapter, CNNs built to predict visual attention exploit particular features useful for predicting how human observers will explore a given visual scene. These models predict eye movements very accurately solely on visual features such as color. This supports the notion that changes of these images based on these learned features might guide viewing behavior.

3.1.1 Adversarial images for humans

One can say that adversarial images are visual illusions to machines. Visual illusions are images that deceive a human into thinking it is something that it is not. As mentioned before, researchers have been able to show similarities between CNNs and the human visual system (Güçlü & van Gerven, 2017, 2015). Activity in deeper CNN layers has been observed to be predictive of activity recorded in the visual pathway of humans and primates (Güçlü & van Gerven, 2017; Cadieu et al., 2014; Yamins & DiCarlo, 2016). These similarities encourages the idea of the possibility that adversarial examples made for computers vision models, may be able to affect the human visual system. Elsayed et al. have shown that adversarial images can affect classification of images for time-limited humans (Elsayed et al., 2018). Though it was shown in a very limited experimental setting, it is a further suggestion of resemblance between modern CNNs and human cortex to some extend.

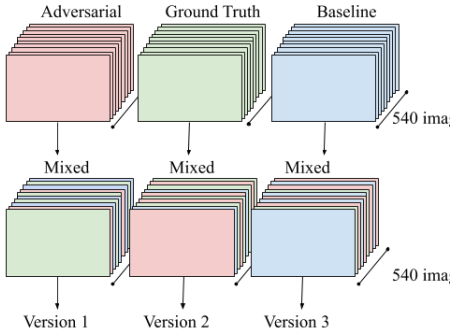


Figure 3.1: Different versions of experimental runs. The experiment contains 3 versions. Each subject is assigned to one version. For one subject, 540 images are presented, with each image being either adversarial ground truth, or original.

There are still major differences in computer and human vision. For example, the eccentricity and sensitivity to spatial information is very different. This might cause perturbations on adversarial example that are out of human's visual range, and consequently would have no impact on human perception. To count for these differences, we induced blurring on the perturbation using the total variation loss as described in the previous chapter. This chapter explains how we investigated whether the constructed adversarial images from chapter 2 can change human eye movements compared to ground truth images.

3.2 Material and methods

Participants

16 naive subjects between 18 and 30 years old participated in the complete experiment, 5 of which were male and 11 female. All participants had (corrected-to-)normal vision. And for each participant, informed consent was obtained before the study in accordance with the guidelines of the Donder's Centere for Cognition provided by the Donder's Intitute, Nijmegen.

Participants were seated comfortably 60 cm from the screen, with their head stabilized on a head rest. Eye movements were monitored by measuring each participant's both eyes using an infrared video-based eye tracker (Eyelink 1000 Plus; SR Research), operating at 1,000 Hz.

Stimuli and procedure

The images for the experiment were 224 x 224 pixels with 3 color dimensions RGB and saved as JPG format. each image was shown for 3.4 seconds and then have a 0.7 seconds fixation point in between the presented images, making it approximately 40 minutes screen time including answering catch trials. The experiment consisted of three different versions where each participant is assigned to one specific version. The only difference in the versions is that the image type per image. Each version contains the same amount of adversarial, ground truth and baseline images, which are pseudo-randomly distributed within that version, a representation is shown in figure 3.1.

3.2.1 Experimental Conditions

There are 540 n experimental images with each image having a number ranging from 0 to 540. Additionally 9 catch images are presented throughout the experiment for each subject ensuring that the subject is paying attention to the images. Each of the experimental images are selected from the validation dataset, containing of three image types: ground truth, baseline, and adversarial.

- **Ground truth image:** Images obtained from the SALICON training set are used and presented in its original size for the experiment (480x640 pixels). These ground truth images have no perturbations.
- **Adversarial image:** Images with added adversarial perturbation **P** to ground truth images, crafted to cause machine learning models to predict different heatmaps than original target (e.g., if the image was originally a scene with four salient areas, we perturbed the image to be predicted to have one salient area). We used the α and β to adjust the strength of the perturbation. For instance, we chose β to be large (to improve the chances of adversarial examples transfer to human) but kept it small enough that the perturbations are subtle that humans are not aware of the perturbations.
- **Baseline image:** similar to adversarial images but the adversarial perturbation is not targeted at one cluster. The perturbation is made using a target heatmap representing the average of all heatmaps. This is to have nearly identical perturbation statistics to the adversarial image as control.

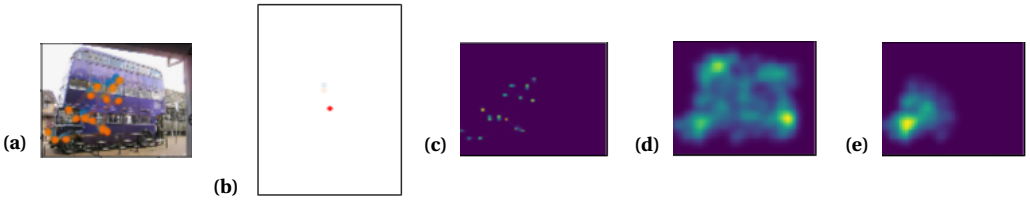


Figure 3.2: Visualizing raw eyetracking data and transforming to 2D histogram. The x-y coordinates of each eye during an image are plotted over the image that the subject sees for 3.3 seconds (a). Between the images, a fixation point in the center of the screen is shown and here eye data is also plotted to see whether they fixated properly (b). Then a 2D histogram is made based on the raw data over time (c). The histogram will then be compared with the original (d) and adversarial (e) target heat maps.

- **Catch image image:** This image is shown once in 30 images along with the question "have you seen this image before?" to see whether the subject is actually paying attention. Half of the time the correct answer is yes, and the other half is no. The catch images are not used for data analysis, only to test whether subject's are paying attention.

3.2.2 Data analysis

For obtaining the eyetracking data, the Eyelink program is used with the program Pylink on python. I tracked both eyes and calibrated it before each trial. During recordings, there are three files being saved. 1) the log files where all the images shown and keypresses and their corresponding times that they occurred during the experiment are logged, this is called the log file. 2) the raw eye-tracking data is saved in the edf file. This contained the necessary data to see where the subject was looking at on the image. The file contains (x, y) coordinates for both left and right eyes, and the pupil diameter for each eye. Each timestamp is 1ms, since the framerate was set to 1000Hz. The pupil diameter and the rest of not mentioned data saved in the file were not used in the experiment. The third file 3) is a csv file that contains all the images that were shown and whether they were catch images or not and what the correct answer is for the catch trial is.

The data is analyzed based on the three files resulting from eye recording. The eye-coordinates with time-stamps data was saved in an *.edf* file with the Eyelink program which is then manually converted to *.asc* type. The eye-coordinates are used to plot onto the images that are shown (as shown in Figure 3.2a). The data is then converted to a 2D histogram (χ) (shown in Figure 3.2c) counting the number of datapoints during an image over x and y buckets. There

are two different scoring systems to compare the eye fixations on different images.

- The first one is based on this formula:

$$log\_score = \frac{1}{n} \sum (log \frac{p_{gt}(\chi)}{p_{adv}(\chi)}) \tag{3.1}$$

  where $P_{gt}(\chi)$ is the distribution of the subject's 2D histogram χ for one image multiplied by its corresponding ground truth target ($\mathbf{T}_{gt}$) and $P_{adv}(\chi)$ is the distribution of the histogram χ times adversarial target ($\mathbf{T}_{adv}$).

- The second scoring system is the similar to the first, but then uses negative plots to calculate, as follows:

$$neg\_log\_score = \frac{1}{n} \sum (log \frac{p_{neg\_gt}(\chi)}{p_{adv}(\chi)}) \tag{3.2}$$

  Where $P_{neg\_gt}(\chi)$ is χ histogram multiplied by the negative target ($\mathbf{T}_{gt} - \mathbf{T}_{adv}$).

3.3 Results

3.3.1 The Eye fixation maps

For each image, we took the raw x - y coordinates and made a 2D histogram (χ) which counts the number of datapoints over x and y buckets. The buckets are chosen based on the amount of pixels on the target heatmaps (48 x 64). The χ histogram is used to compare with the two targets to test for any effects.

3.3.2 The Divergence scores

As mentioned before, to test the effects of the adversarial images, we compute divergence scores using the formula in equation 3.2. What we expect to see in the divergence score distribution plots is that the adversarial curve is relatively more to the left compared to the baseline and ground truth images, whereas curves of the baseline and ground truth should be relatively similar, as shown in figure 4.1.

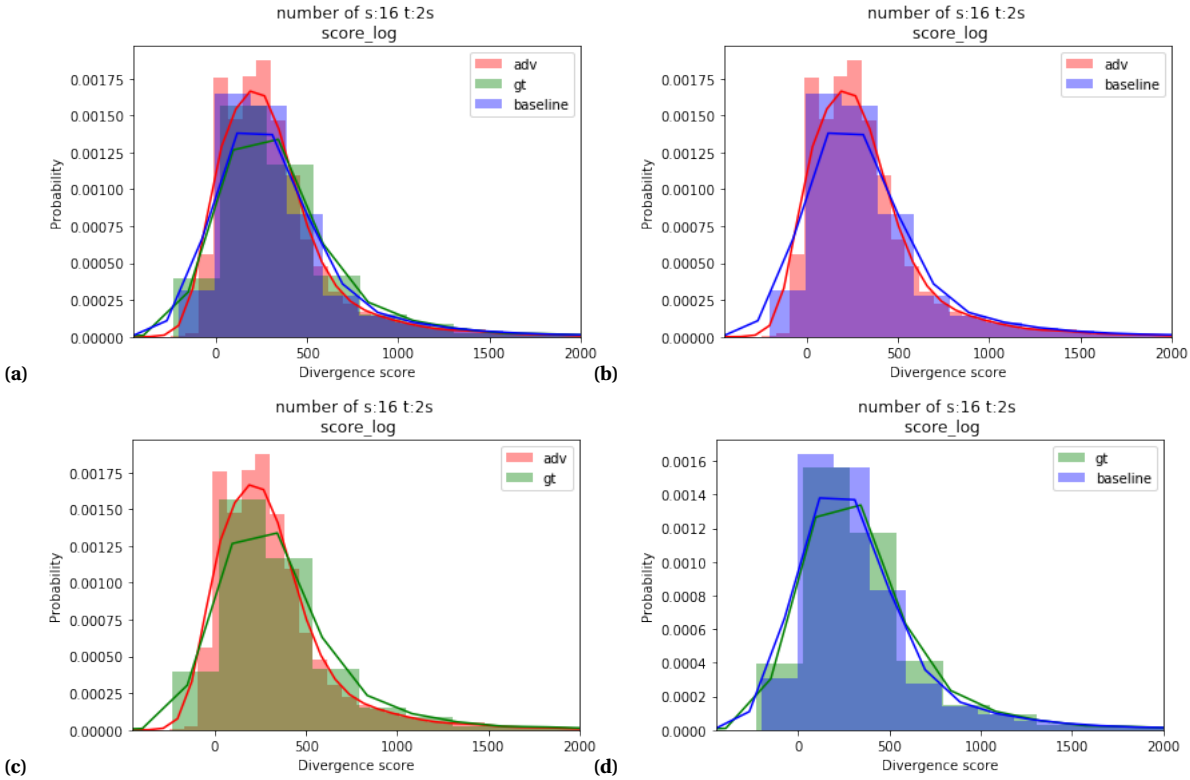


Figure 3.3: Visualizing the results of all 16 subjects. The log_scores of all 16 subjects plotted in a 1D histogram. a) compares all three score kinds together. b) compares adversarial and baseline. c) compares adversarial and ground truth scores, and d) compares the ground truth and baseline images. The average of all the scores are: baseline_scores: 375.013 +- 13.914, adv_scores: 339.380 +- 8.291, gt_scores: 393.662 +- 12.669. Error values are standard error of the mean.

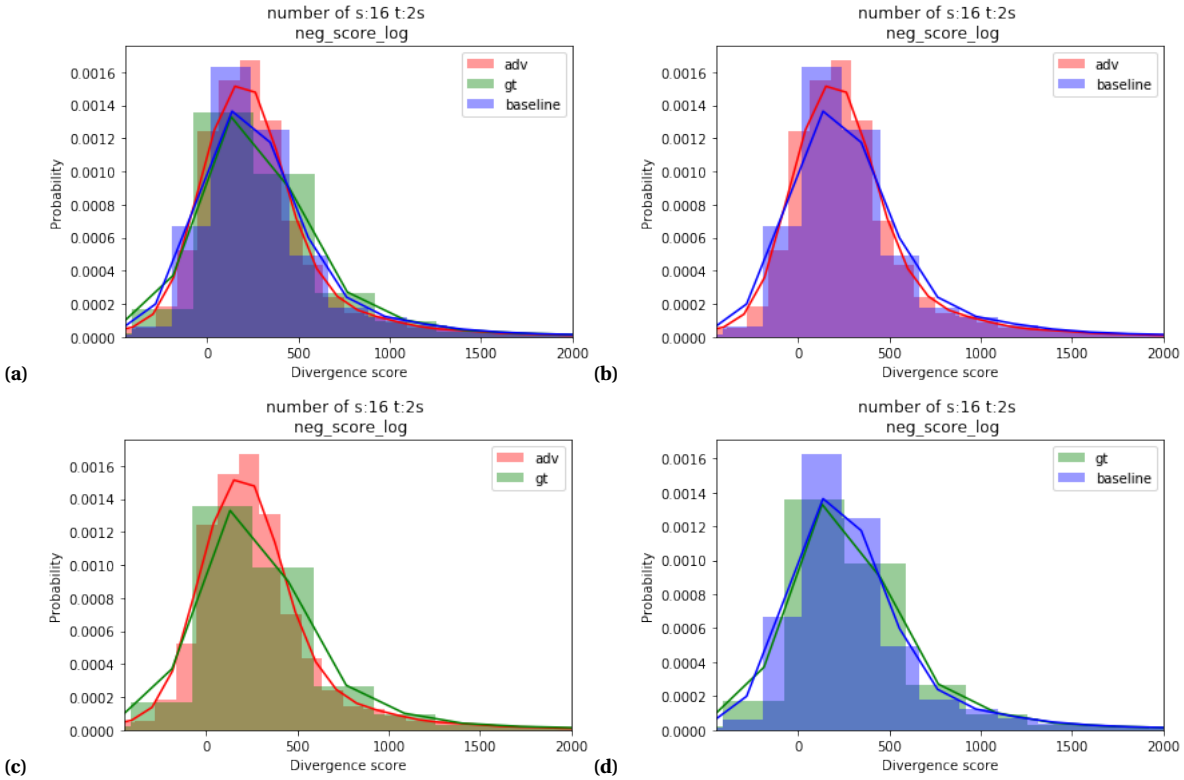


Figure 3.4: Scores of 16 subjects using negative targets] The neg_log_scores of all 16 subjects plotted in a 1D histogram. a) compares all three score kinds together. b) compares adversarial and baseline. c) compares adversarial and ground truth scores, and d) compares the ground truth and baseline images. baseline_scores: 336.021 +- 14.341, adv_scores: 290.868 +- 9.024, gt_scores: 339.114 +- 13.424. Error values are standard error of the mean.

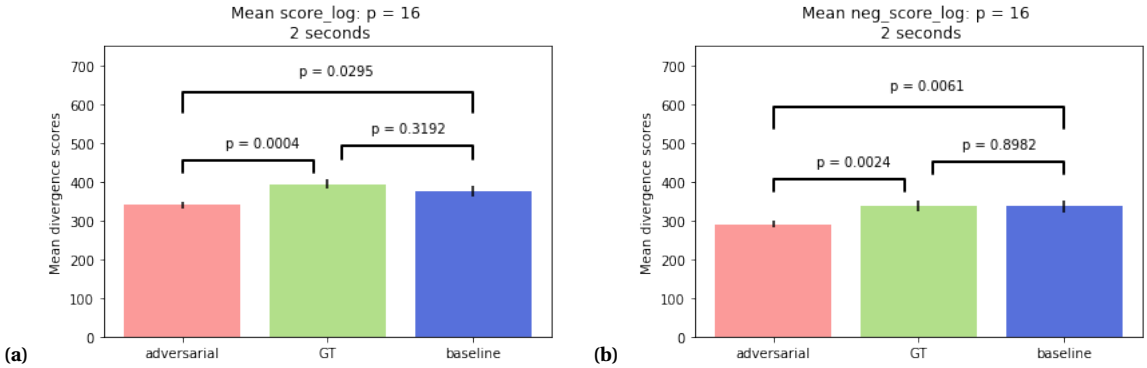


Figure 3.5: Testing significance by comparing the P-values Paired t-test of the averaged scores were calculated.

The distribution of log_scores obtained from 16 subjects are plotted and compared in figure 3.3. The neg_log_scores are shown in figure 4.1. Both group of plots show different distribution of scores with fitted curves for each score type. The red curve are fitted over adversarial scores, green is fitted over the ground truth and blue is fitted over baseline scores. In the distribution plots, we see the slight difference which appears to be in the direction of what we expect. Additionally, the average score of the adversarial images are the lowest when compared to the other two averages.

With the scores for all images for all subjects, we perform a paired t-test to see whether the effects shown on the histogram are significant. What we expect is to have significant difference between the adversarial and ground truth scores, and adversarial and baseline scores whereas there should not be a big difference between the ground truth and baseline. We also expect is the average of the adversarial score to be the lowest value and the baseline and ground truth values not to differ from each other too much. This is also what we observe. The results are significant based on the paired t-test on all of the scores, as shown in figure 3.5. When comparing adversarial and ground truth log_scores, the difference was significant with $P < 0.0005$. When comparing adversarial with baseline, there is also a significant difference with $P < 0.05$. When comparing neg_log_scores, the differences were also significant with adversarial vs. ground truth $P < 0.005$. When comparing adversarial with baseline, $P = 0.006$. And finally, there was no significant difference between ground truth and baseline, which is what we expect. This tells us that the adversarial perturbations generated using the CNN-ensemble, has influenced human perception towards the targeted eye movements.

4 Discussion and Conclusion

4.1 Discussion

Our results suggest that human eye movements can be influenced by adversarial images that have been designed to fool machines. This suggests that there is some degree of similarity between the information of an image that humans and machines extract and uses when deciding areas to pay attention to. Elsayed et al. suggested that perhaps they were able to fool time-limited humans but not no-limit humans because the 1) *"lateral and top-down connections used by the no-limit human are relevant to human robustness to adversarial examples"* or 2) *"adversarial examples do not transfer from feed-forward networks to no-limit humans because of these architectural differences"* (Elsayed et al., 2018). There is a major flaw in their suggestion however, because their task was specifically designed for time-limited humans, to ensure that they did not change the true class of the image, therefore it is logical that they only fooled time-limited humans. In order to make such claims, they should have executed an experiment designed for non-limited humans which imaginably does not involve a categorization task.

Our chosen method and task for participants was an beneficial choice for testing transferability of adversarial images across machines and humans, because the ground truth signals were not labeled based on human judgment. This allowed us to show that humans could be affected by adversarial attacks, even when allowed to look at an image for multiple seconds. We suggest here that humans are vulnerable for adversarial examples and that it is possible to transfer from feedforward networks to no-limit humans.

Adversarial images are mostly portrayed as a threat to the machine the learning industry and we showed how it can also hold possible threats for humans. The beauty and the danger is that the human observer is unaware of the manipulation: when asked after experiment, all

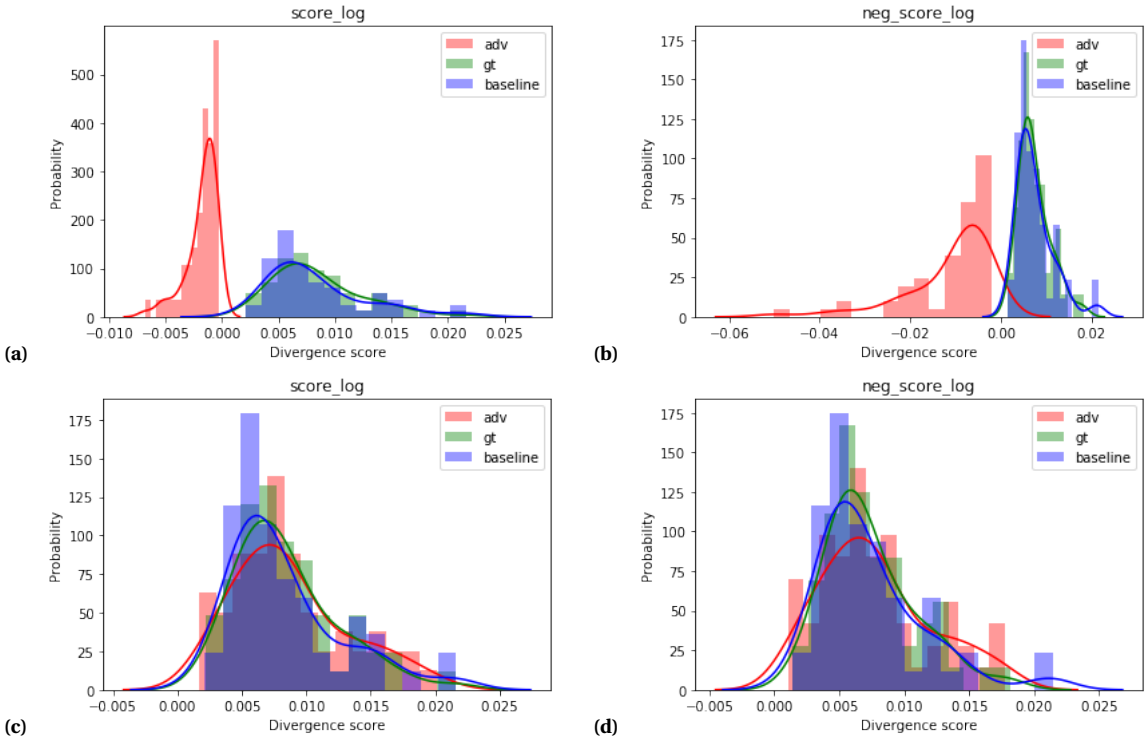


Figure 4.1: Plots of data if it were to be of maximal or minimal effect. The scores at extreme cases

subjects were not aware of any image transformations. A risky applications can be for example images on billboards and online ads that contain adversarial perturbations to guide eye movements. Obviously, we do not suggest that all adversarial images that affect CNNs, can be used to fool humans. We simply introduce examples that do affect both CNNs and humans.

Although our results did not show an immense amount of alteration when compared to the dissimilaty of curves in the plot with maximum effect in figure 4.1, the results were certainly significant as shown in the t-test 3.5. Regardless of normalizing the targets, the scoring system still has a slight bias for ground truth targets, because when looking at the cluster on the adversarial image it will always overlap with a cluster on the ground truth cluster. So the scoring favors ground truth targets, and nonetheless, the results are significant. We also computed the scores with the negative maps, however this method is expected to have a bias towards the adversarial target since we remove a cluster from the ground truth target for scoring. By plotting the maximal effect (using the corresponding targets as data), we see how it would look like if the adversarial images had maximum effect on observers or no effect at all. It also shows that the biases of the scoring method are excusable.

4.1.1 Future experiments

It is also possible that the adversarial images could cause even larger effects if the parameters were adjusted. For example, one can test whether adversarial images would be more effective when using different β and α for the adversarial loss function, or utilize a different loss function completely. Another suggestion could be a change in targets, perhaps the SALICON ground truth targets were not accurate enough for the models to generate proper adversarial images, since they did not use eye-trackers. It could also be interesting to improve the trained models by implementing category-specific saliency predictions as done in the Dodge & Karam study (Dodge & Karam, 2018). Or perhaps, we could also test whether probabilistic models that are shown to be well performing saliency models (Jetley et al., 2016), permits better construction of adversarial images.

4.2 Conclusion

So far it has never been shown whether adversarial images could influence a human observer's attention. Our results suggest that it is possible for adversarial images, that are manipulated in a specific way, to cause human observers to change their eye-movements without them being aware. Our findings further supports the notion that there are striking similarities between CNNs and the human visual system. The evidence of this particular connection gives us a powerful tool to bridge the gap between neuroscience and AI.

4.3 Acknowledgements

I would like to thank everyone who has helped in in some way during my master internship. Thank you to my daily supervisor Dr. Umut Güçlü for giving me this great opportunity to do this project and guiding me throughout and making it a very fun project. Thank you to my supervisor Prof. Dr. Marcel van Gerven for allowing me to be part of his great department and also support. Also a big thank you goes to Dr. Luca Ambrogioni for all the advice and input and his great contribution to this project. I am very grateful for Dr. Yağmur Güçlütürk who introduced me to this department and has always given me constructive and useful advice. Thank you to the rest of the colleagues in the department for not only showing me how to run experiments (Nadine) and providing me instructions on the eye-tracker and Eyelink codes (Jordy) but also for great conversations, laughs and lunch (Gabi). Finally, I am very grateful to my friends and family, especially Louis Le, Grandma, and Stan van Lier, for your support and help with everything I do.

References

Cadieu, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., ... DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS computational biology*, *10*(12), e1003963.

Carrasco, M. (2011). Visual attention: The past 25 years. *Vision research*, *51*(13), 1484–1525.

Dodge, S. F., & Karam, L. J. (2018). Visual saliency prediction using a mixture of deep neural networks. *IEEE Transactions on Image Processing*, *27*(8), 4080–4090.

Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, *38*(2), 295–307.

Elsayed, G., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., & Sohl-Dickstein, J. (2018). Adversarial examples that fool both computer vision and time-limited humans. In *Advances in neural information processing systems* (pp. 3910–3920).

Greene, M. R., & Hansen, B. C. (2018). Shared spatiotemporal category representations in biological and artificial deep neural networks. *PLoS computational biology*, *14*(7), e1006327.

Gross, C. G. (2002). Genealogy of the "grandmother cell". *The Neuroscientist*, *8*(5), 512–518.

Güçlü, U., & van Gerven, M. A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, *35*(27), 10005–10014.

Güçlü, U., & van Gerven, M. A. (2017). Increasingly complex representations of natural movies across the dorsal stream are shared between subjects. *NeuroImage*, *145*, 329–336.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Jetley, S., Murray, N., & Vig, E. (2016). End-to-end saliency mapping via probability distribution prediction. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 5753–5761).

Jiang, M., Huang, S., Duan, J., & Zhao, Q. (2015). Salicon: Saliency in context. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1072–1080).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3431–3440).

Luo, B., Liu, Y., Wei, L., & Xu, Q. (2018). Towards imperceptible and robust adversarial example attacks against neural networks. In *Thirty-second aaai conference on artificial intelligence*.

O'Connell, T. P., & Chun, M. M. (2018). Predicting eye movement patterns from fmri responses to natural scenes. *Nature communications*, *9*(1), 5159.

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Pan, J., Sayrol, E., Nieto, X. G.-i., Ferrer, C. C., Torres, J., McGuinness, K., & O'Connor, N. E. (2017). Salgan: Visual saliency prediction with adversarial networks. In *Cvpr scene understanding workshop (sunw)*.

Rastegari, M., Ordonez, V., Redmon, J., & Farhadi, A. (2016). Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision* (pp. 525–542).

Rojas, R. (2013). *Neural networks: a systematic introduction*. Springer Science & Business Media.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, *5*(3), 1.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat:

Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Solomonoff, R. J. (1957). An inductive inference machine. In *Ire convention record, section on information theory* (Vol. 2, pp. 56–62).

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Thorpe, S. (1998). Localized versus distributed representations. In *The handbook of brain theory and neural networks* (pp. 549–552).

Torralba, A., Oliva, A., Castelhano, M. S., & Henderson, J. M. (2006). Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, *113*(4), 766.

Yamins, D. L., & DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, *19*(3), 356.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833).

Zhou, Z., & Firestone, C. (2019). Humans can decipher adversarial images. *Nature communications*, *10*(1), 1334.

A Appendix

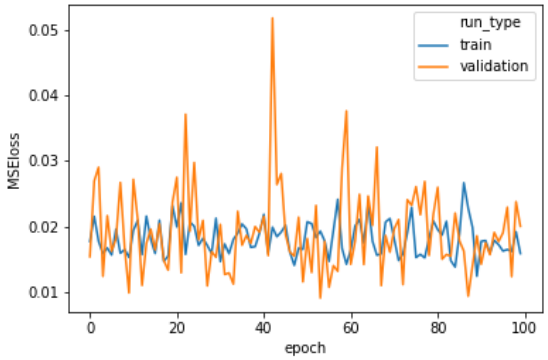


Figure A.1: Plots of the mean squared error (MSE) loss during training the last layers of the **AlexNet** model.

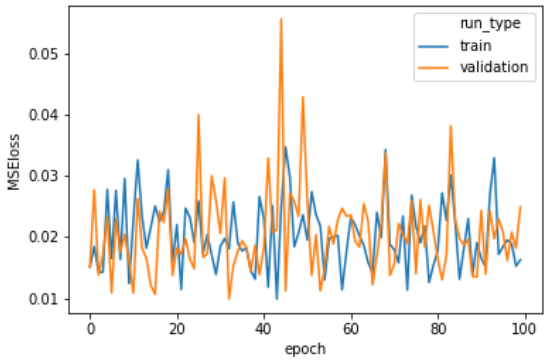


Figure A.2: Plots of the mean squared error (MSE) loss during training the last layers of the **DenseNet** model.

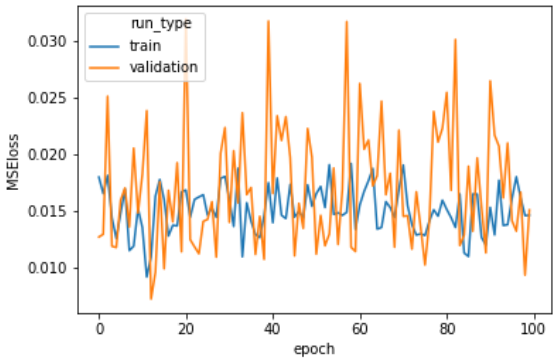


Figure A.3: Plots of the mean squared error (MSE) loss during training the last layers of the **Inception V3** model.

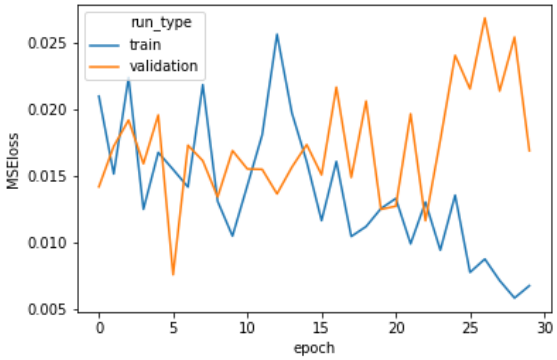


Figure A.4: Plots of the mean squared error (MSE) loss during training of the **ResDec** model.

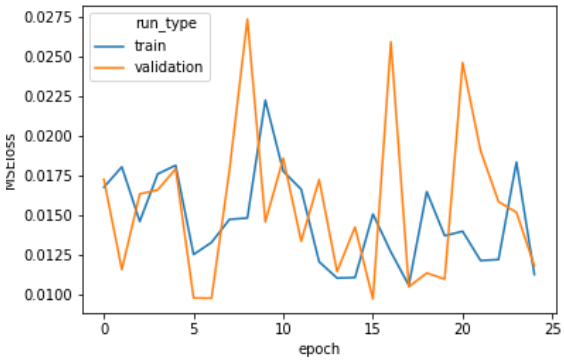


Figure A.5: Plots of the mean squared error (MSE) loss during training the last layers of the **SqueezeNet** model.

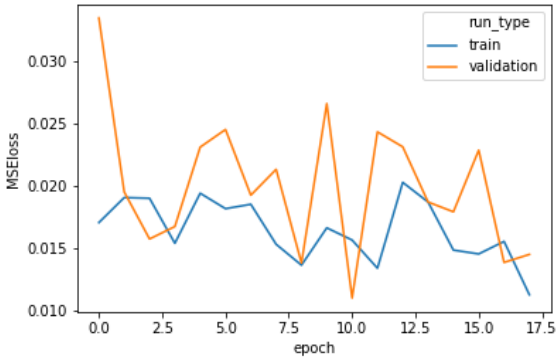


Figure A.6: Plots of the mean squared error (MSE) loss during training the last layers of the **VGG19** model.

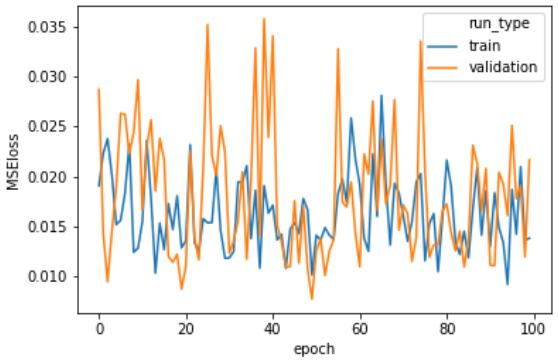


Figure A.7: Plots of the mean squared error (MSE) loss during training the last layers of the **ResNetM-SEplot** model.