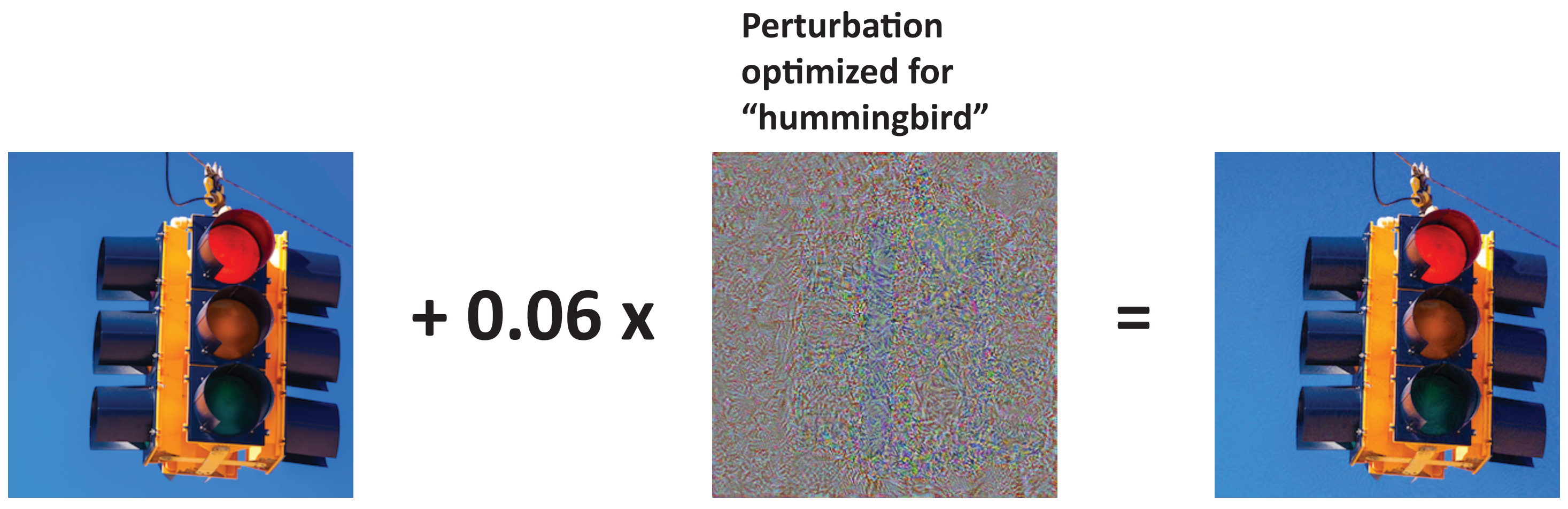
TARGETED ADVERSARIAL ATTACK - 4TH PLACE

Anil Thomas

Artificial Intelligence Products Group, Intel Corporation.

Introduction

We iteratively modify the input images in order to get a classifier's predictions closer to the provided attack labels.



The resulting image shown on the right is classified as **"hummingbird"** by a pre-trained Inception V3 network with 99.9% confidence.

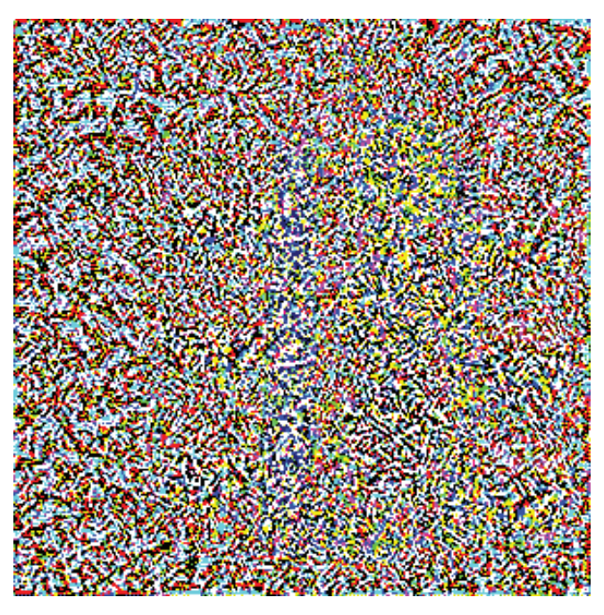
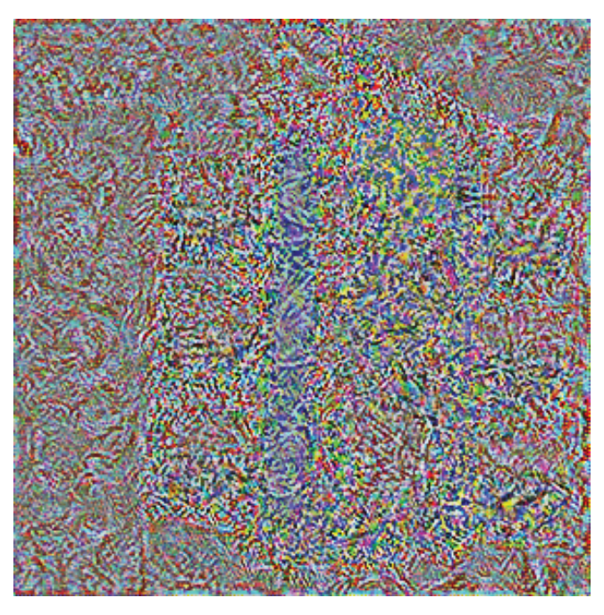
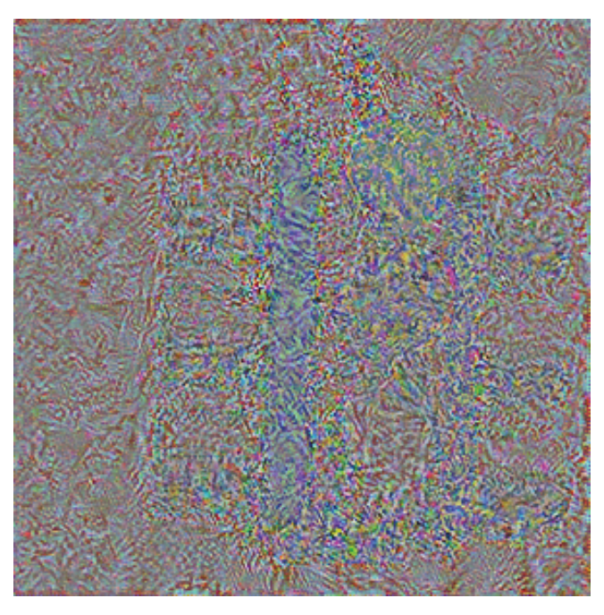
Algorithm

The algorithm used to craft adversarial images is very similar to back-propagation. Instead of adjusting the parameters of the network to minimize the objective, the input images themselves are modified. The set of parameters are held constant.

Adversarial image crafting algorithm: "Adam" is used as the optimizer to minimize the objective. The objective function f is the cross entropy between the provided attack label and the average of predictions from multiple pre-trained classifier networks.

```
$\alpha \leftarrow$ learning rate
$\beta_1, \beta_2 \leftarrow$ decay rates
$t_{max} \leftarrow$ number of iterations
$\delta_{max} \leftarrow$ maximum perturbation allowed
$f \leftarrow$ objective function
$x_0 \leftarrow$ input image
$m_0 \leftarrow 0$   (Initialize 1st moment vector)
$v_0 \leftarrow 0$   (Initialize 2nd moment vector)
$t \leftarrow 0$   (Initialize timestep)
while $t < t_{max}$ do
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_x f_t(x_{t-1})$   (Get gradients of objective w.r.t. $x$at timestep $t$)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$   (Update first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$   (Update second moment estimate)
    $m_t \leftarrow m_t / (1 - \beta_1^t)$   (Compute bias-corrected first moment estimate)
    $v_t \leftarrow v_t / (1 - \beta_2^t)$   (Compute bias-corrected second moment estimate)
    $x_t \leftarrow x_{t-1} - \alpha \cdot m_t / (\sqrt{v_t} + \varepsilon)$   (Update adversarial image)
    $x_t \leftarrow clip(x_t, x_0 - \delta_{max}, x_0 + \delta_{max})$   (Clip adversarial image to allowed range)
```

Example:

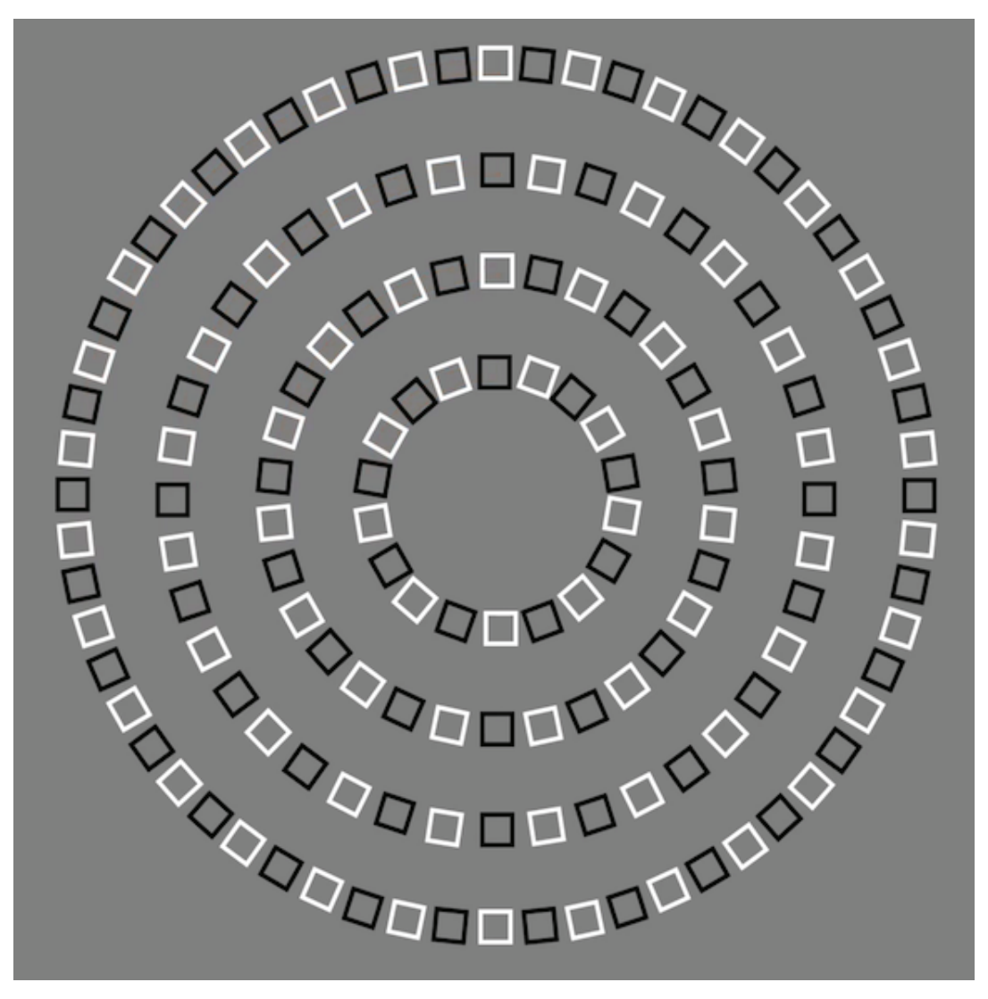
timestep	t = 1	t = 10	t = 40
$x_t - x_0$			
P(hummingbird | x_t)	6.6e-09	0.009	0.998
P(traffic light | x_t)	0.999	0.039	8.3e-05

Note: An α of 0.003 was used to generate the above visualizations, while a value of 100 was used for the contest entry.

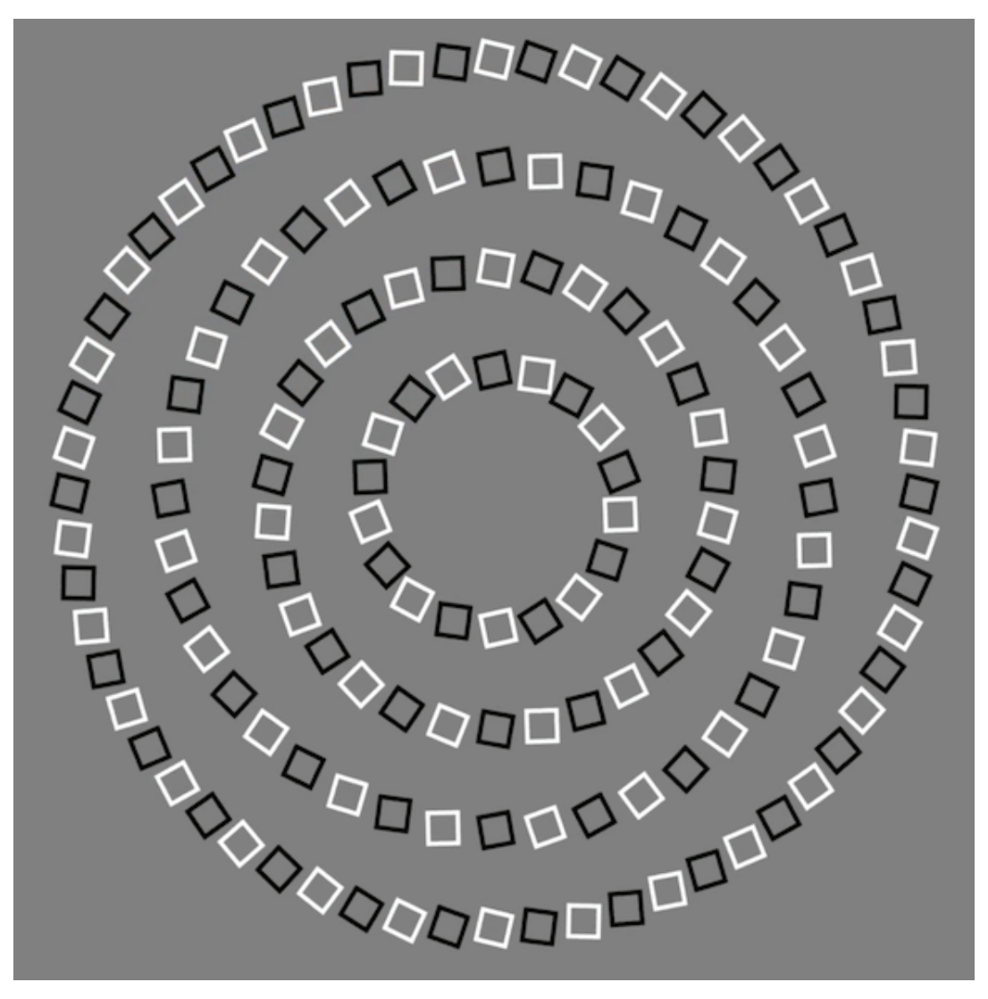
A slight digression

Is it possible to fool the brain using similar techniques?

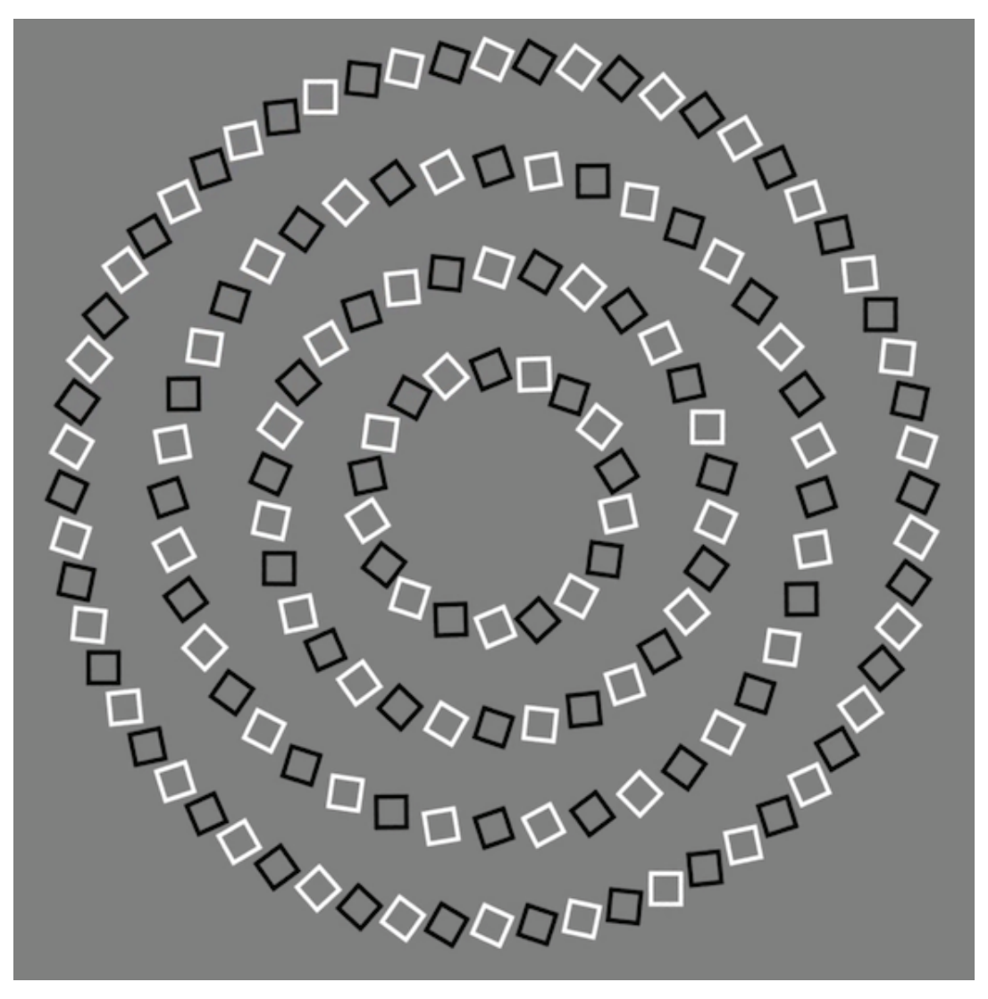
We do not know of any generic perturbations that will work for any image. However it is possible to construct adversarial images for specific cases.



Start with a picture of four circles made up of small squares. The circles appear concentric to the viewer (and they are).



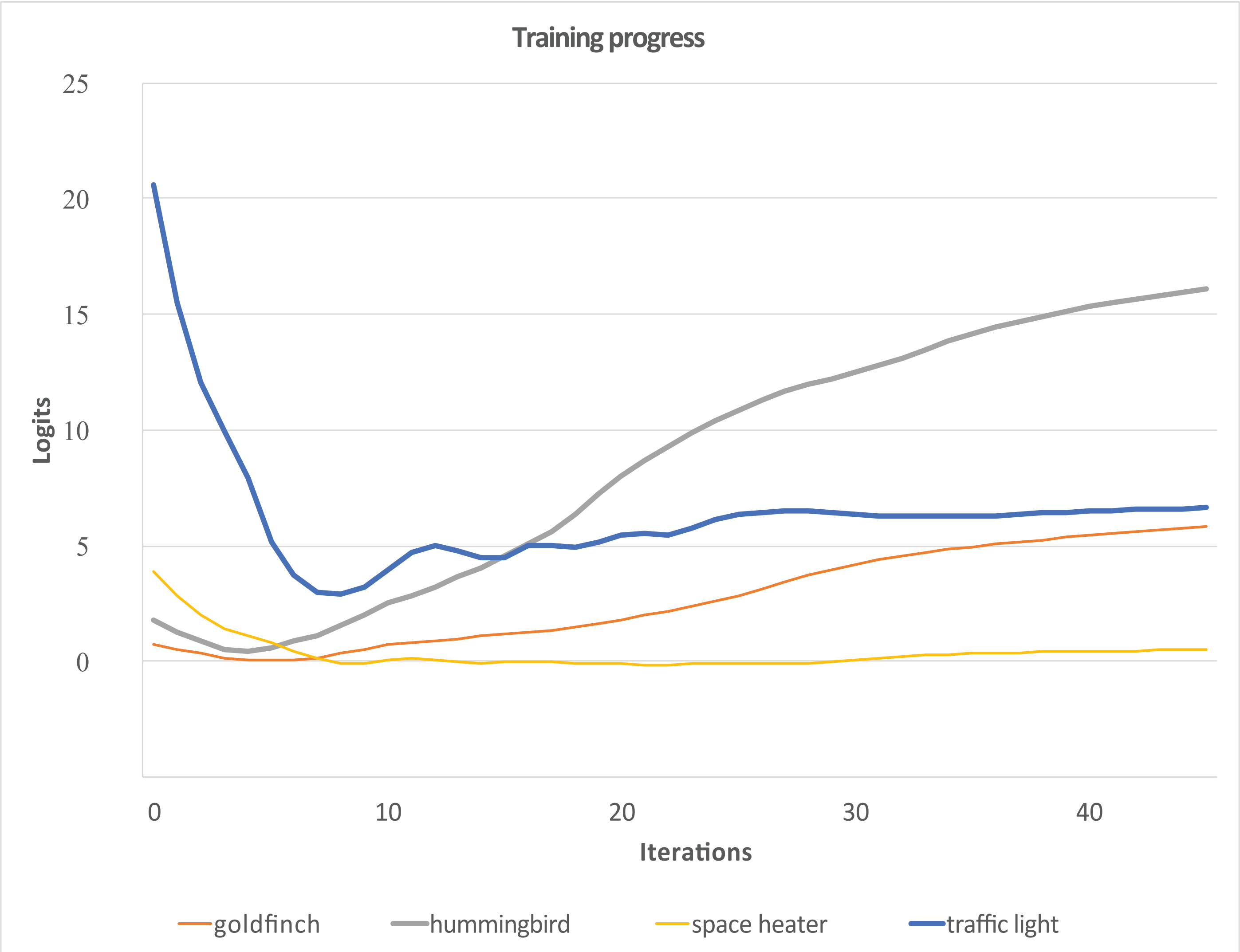
Now rotate each square by a small amount. This begins to confuse the visual system.



Perturbing a bit more makes the visual system perceive the image as intertwining circles, which they are not.*

* Pinna illusion courtesy of Ian Goodfellow and youtuber Heartwood Illusion

Results



In this example, an image of a traffic light is adversarially perturbed to make a classifier misclassify it as "hummingbird". As a side effect, this process causes the classifier to overstate the likelihood of other bird types.