Adversarially trained classification models are more dependent on backgrounds to make their predictions.

Adversarially trained classification models are more sensitive to yaw axis rotations of the main subject.

Train a model adversarially once:
1. Initialize early layers (e.g. first two ResNet blocks) with adversarially pre-trained weights.
2. Freeze initialized layers and do standard training for the rest.

We conducted case studies with five ML researchers. Each session, conducted as an online meeting with screen sharing, lasted approximately one hour.

During these sessions our participants generated several hypotheses, two of which we investigated and confirmed by quantitative experiments:

- The resulting model had increased robustness against adversarial examples at a negligible decrease of accuracy. For details please refer to the thesis.

The interactive visual analytics application "Perturber", which lets users manipulate and perturb a 3d input scene while simultaneously showing the responses of a CNN in real-time.

Observing the predictions from multiple models simultaneously lets the user compare their behaviour.

The implementation uses WebGL and performs all computations on the client’s GPU. We use React as the main GUI library, Three.js for rendering the 3d scene, and TensorFlow.js for neural network inference.

The frame rate varies between 5 Hz on a 2018 MacBook Pro when comparing predictions of two models to more than 40 Hz when inspecting intermediate activations of a single model on a gaming notebook.

We used React as the main GUI library, Three.js for rendering the 3d scene, and TensorFlow.js for neural network inference.