TU WIEN Informatics

Visual Analytics for Convolutional Neural Network Robustness

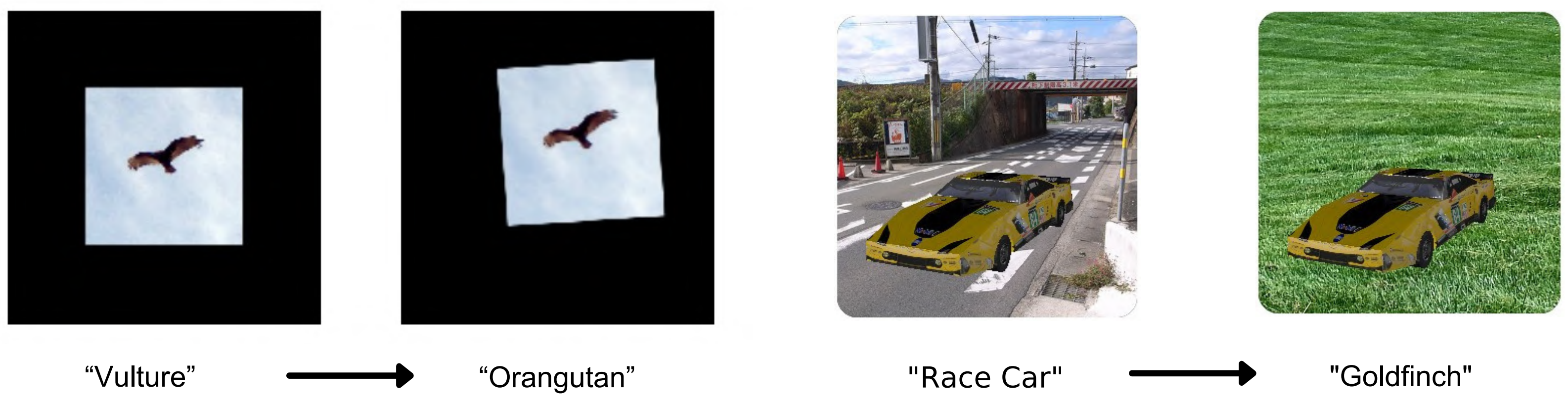
Stefan Sietzen
Visual Computing

TU Wien Informatics
Institute of Visual Computing & Human-Centered Technology
Research Unit of Computer Graphics
Supervisor: Assistant Prof. Dr. Manuela Waldner
Assistance: Dipl.-Ing. Mathias Lechner
Univ. Ass. Dr. Ramin Hasani

Motivation

- CNNs are widely used for computer vision tasks, where they perform exceptionally well (e.g. image classification)
- CNNs are often not robust when trained in a standard way.

E.g. different classification after small rotation, strong reliance on backgrounds, etc.



Contribution

Our contributions are:

1. 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.
2. A case study with five ML researchers, which shows that interactive visual analytics can indeed help with generating new hypotheses about CNN robustness. To support this claim, we present three quantitative evaluations of hypotheses generated by using Perturber.

Scene View

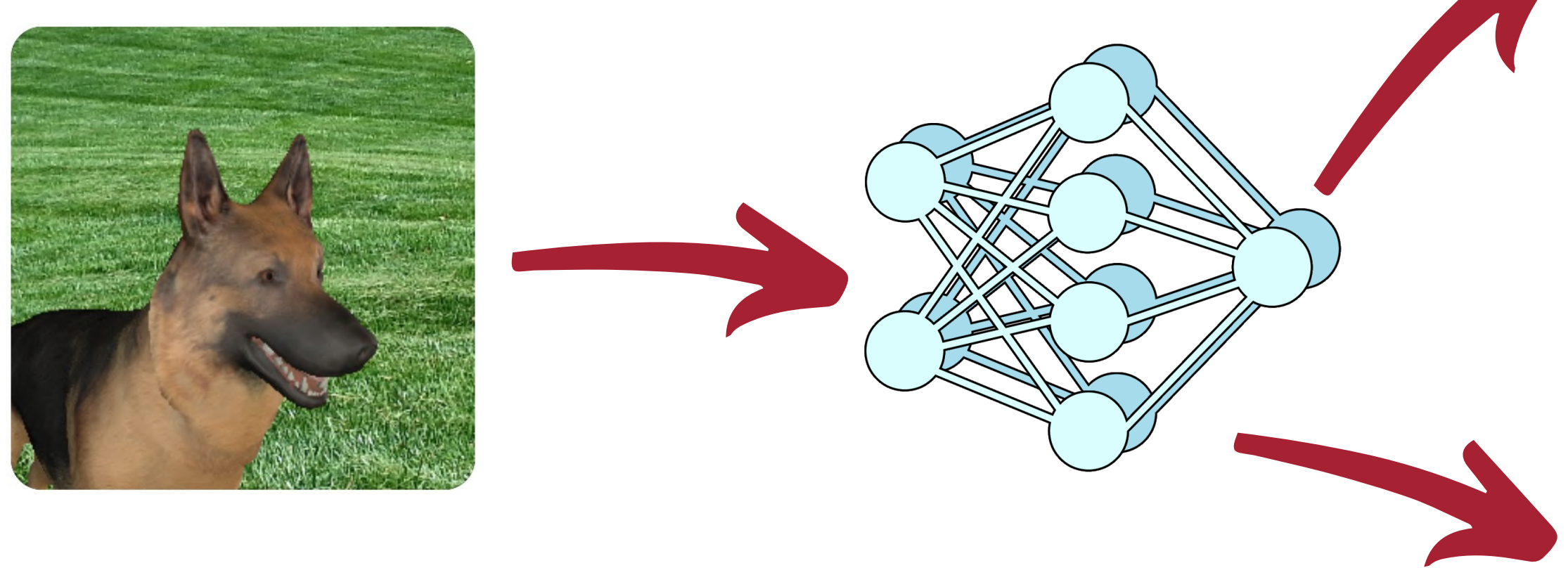
Allows real time manipulation of a 3d scene.

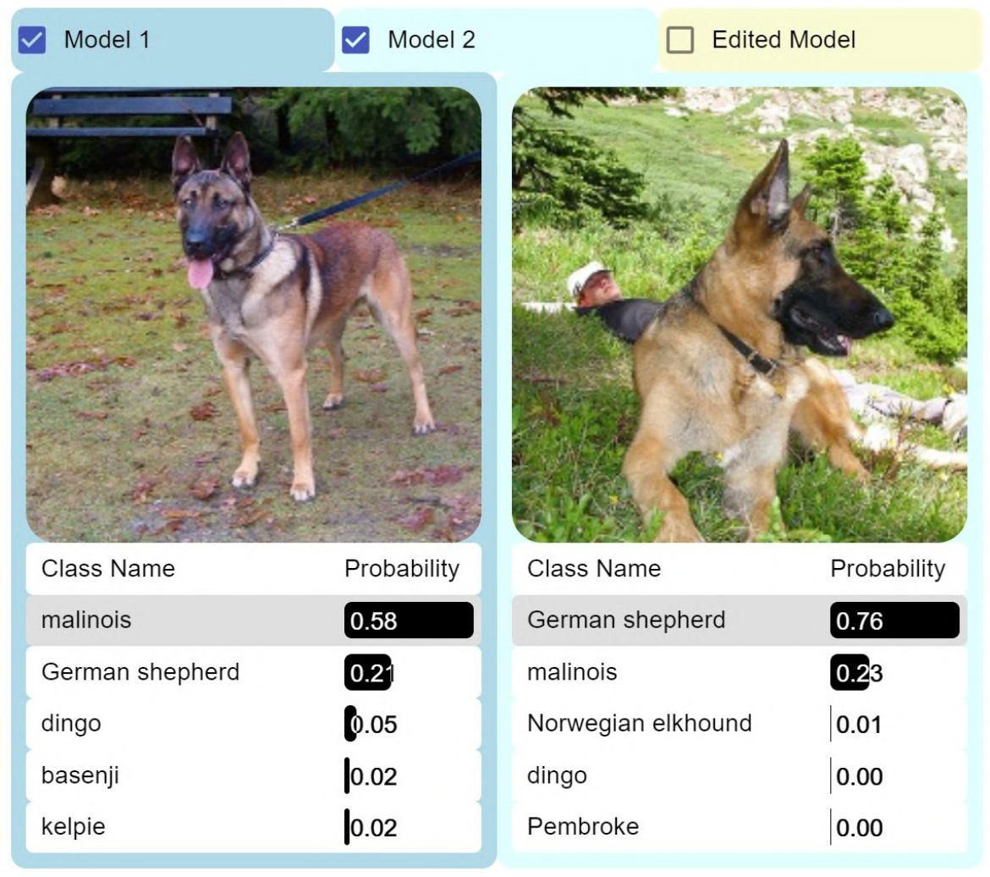
The user can control in real-time:

- Camera view
- Texture & lighting influence
- Texture blur
- Background blur & saturation
- Hue / saturation / lightness
- Gaussian blurring
- Many more.

Manipulate and perturb the input scene

Inference with multiple CNNs





Class Name	Probability
malinois	0.58
German shepherd	0.21
dingo	0.05
basenji	0.02
kelpie	0.02

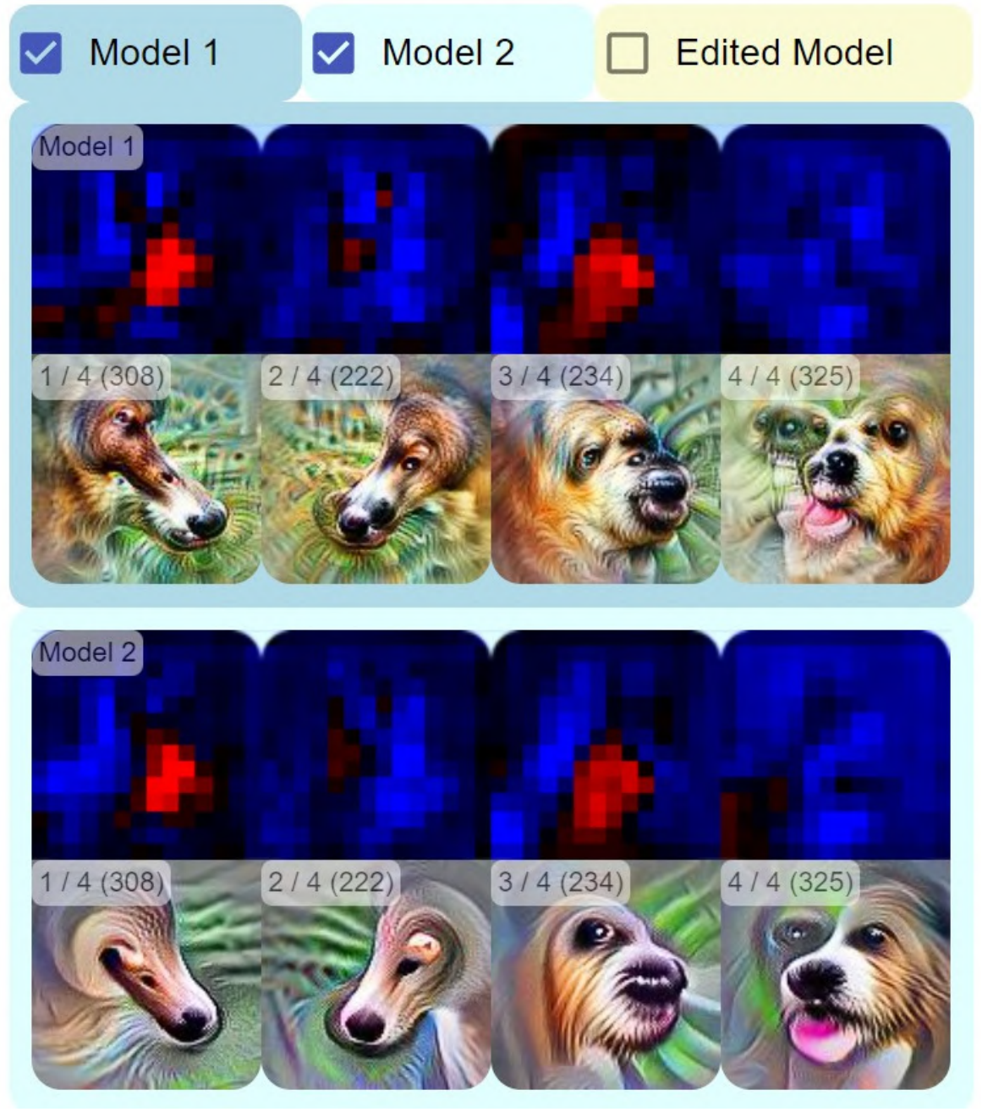
Class Name	Probability
German shepherd	0.76
malinois	0.23
Norwegian elkhound	0.01
dingo	0.00
Pembroke	0.00

Prediction View

Displays top 5 classification scores.

Allows to observe fluctuations and errors in classification while input is perturbed in real time.

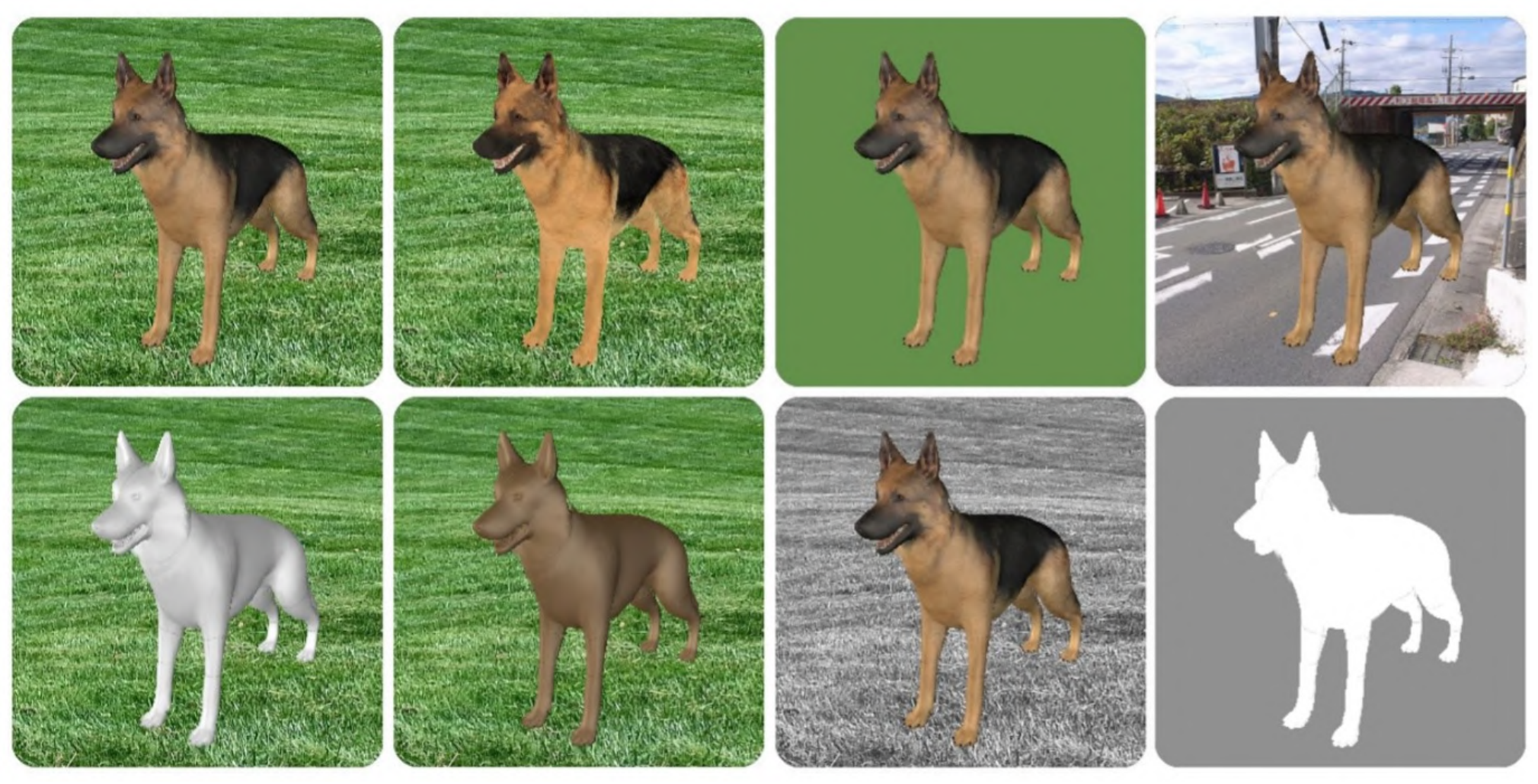
Observing the predictions from multiple models simultaneously lets the user compare their behaviour.



Neuron Activation View

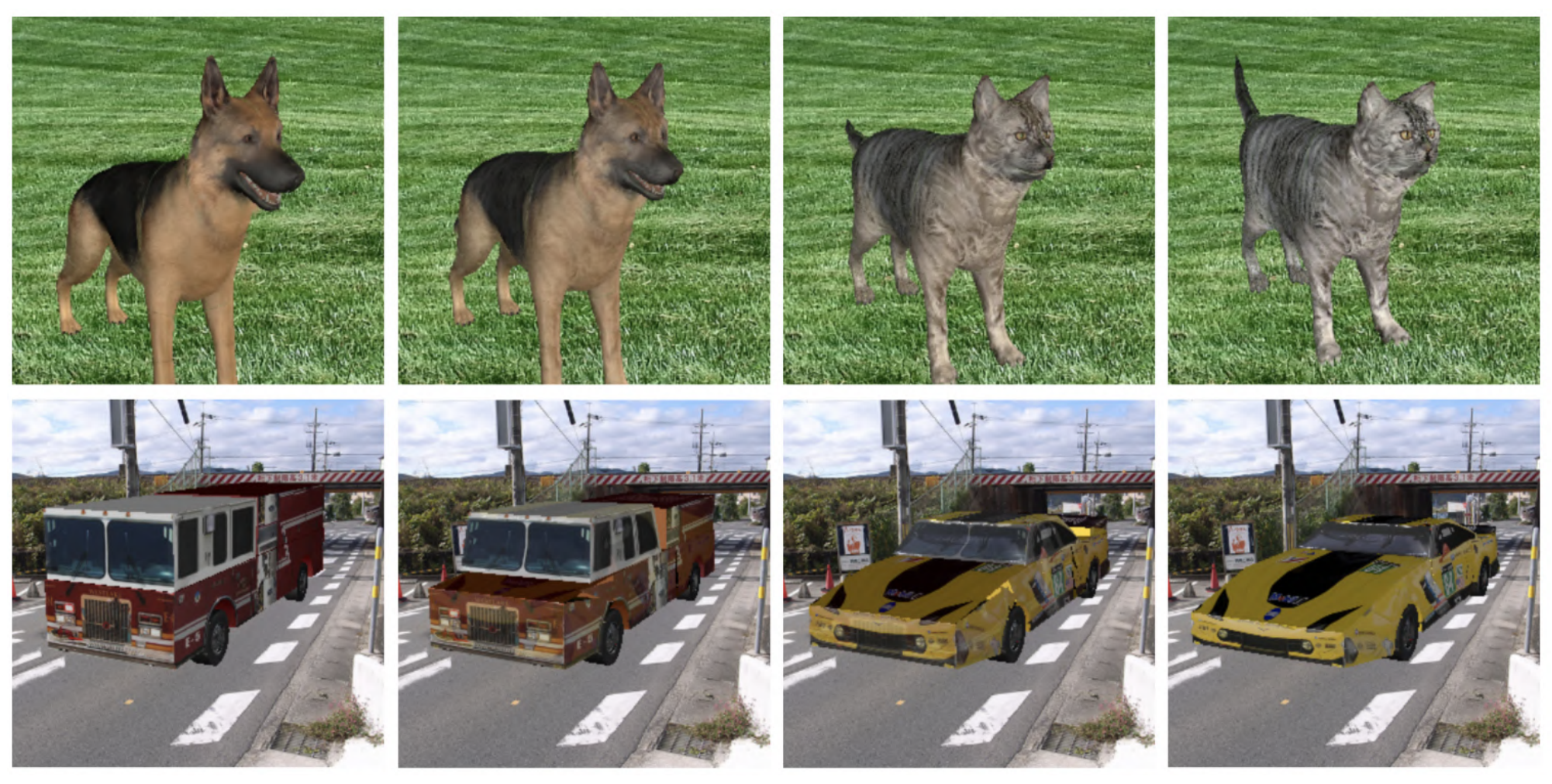
Hidden neuron responses visualized as activation maps and feature visualization.

Reveals responses of neurons in hidden layers of the CNN.

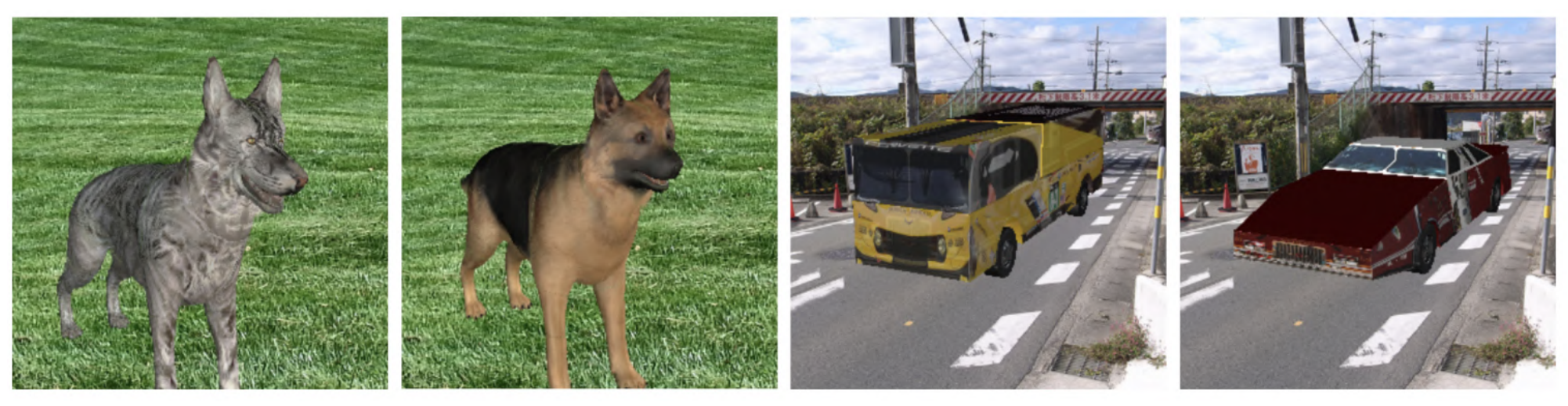


Two 3d object pairs can be morphed into each other:

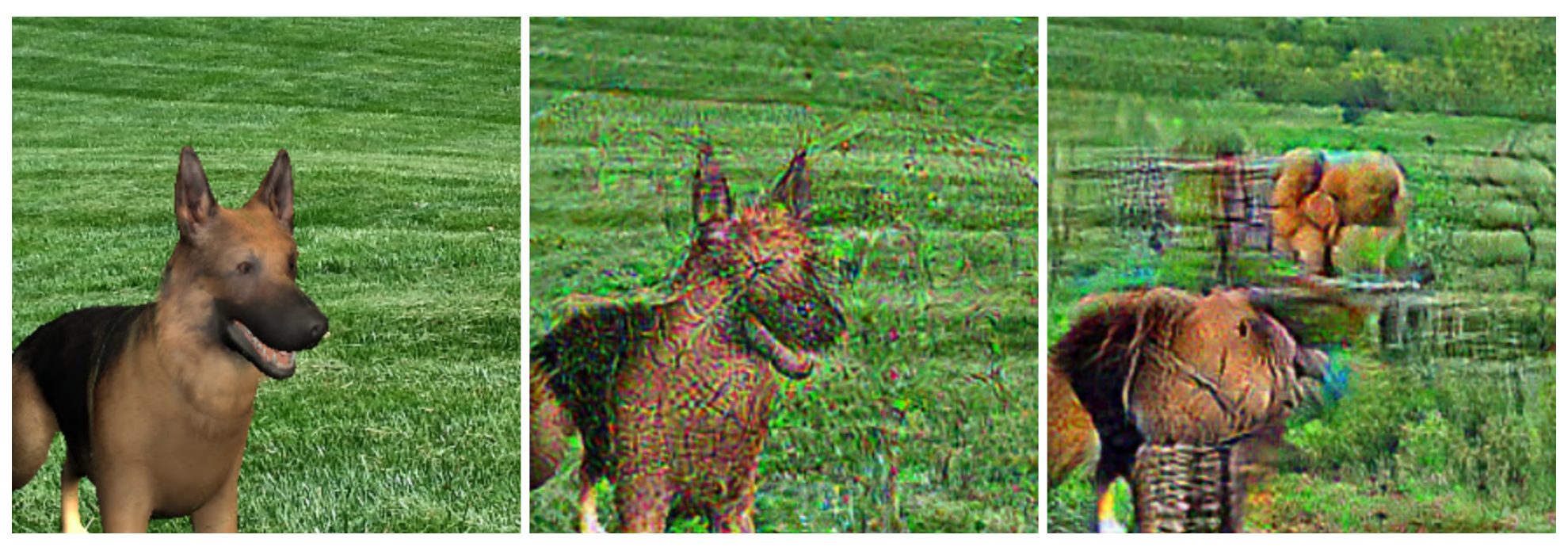
- Dog / cat,
- Fire truck / race car.



Texture and shape can be morphed independently, causing cue conflicts for the CNN.



Adversarial attacks can be generated on the fly.



Implementation

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.

Case Study Results

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:

✓ "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"

Furthermore, by using Perturber we found out that it is possible to increase a CNN's robustness by the following pre-training procedure (tested with a ResNet-18):

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

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