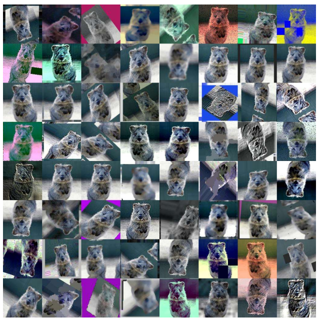
# Deep Learning - Augmentation

Bernhard Kainz

# Input augmentation

- Artificially inflate training data size through applying expected transformations during training
- https://github.com/aleju/imgaug
- https://pytorch.org/docs/stable/torchvision/transforms.html
- Excellent regularizer against overfitting





Input augmentation is used to Artificially inflate training data size through applying expected transformations during training

# Transformations

- Random
  - flipping
  - scaling
  - rotations
  - intensity/contrast variations
  - cropping/padding
  - noise
  - affine transformations
  - perspective transformations

# Input augmentation

- Don’t just use all of them blindly. Carefully select expected transformations

# Anomaly detection

- Predict continuation
- Measure distance in a latent space
- Reconstruct the input
- Classify artificial, subtle variations
- Also known as out-of-distribution detection

# With Deep Networks

- Learns well from lots of data
- Own feature representation: Robust to noise and allows for
- learning cross domain patterns
- Already applied in ads: Google itself invests lots in this same
- kind of pattern recognition (targeting/relevance)

# approaches

- Unsupervised - Use autoencoder reconstruction error and use moving averages, use dropout with a set time window
- Supervised - RNNs Learn from a set of yes/nos in a time series. RNNs can learn from a series of time steps and predict when an anomaly is about to occur.
- Use streaming/minibatches (all neural nets can learn like this)

# Anomalies in images

- Encode, find outliers in latent space
- Reconstruct and build difference to input (AnoGAN)
- Interpolate sample patches into image and learn interpolation factor (Foreign patch interpolation)
- Example medical image out-of-distribution channelling

-> [https://youtu.be/0-JYFxY3zfw](https://youtu.be/0-JYFxY3zfw)