

TEK5040 - Project Guide

Tips and suggestions for final projects

Board game detection

Adding a computer into a game can add an extra dimension to the gameplay. The computer can either contribute as a player, with guidance or extra fun elements.

The project “Neural Chessboard” detects chessboard positions from <https://github.com/maciejczykewski/neural-chessboard>

You can try a similar approach to another boardgame, or use the method described in “Neural Chessboard” to do something fun with it.

Alternative user input

Deep neural networks have proved effective for estimating human pose from images. Human pose as an input device can be both fun and useful. You can create games controls, make animation moves or create a personal trainer.

You can find projects online that detect human pose:

<https://medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow-js-7dd0bc881cd5>

<https://github.com/eldar/pose-tensorflow>

or gaze direction:

<https://cpury.github.io/learning-where-you-are-looking-at/>

Create a new model and train on your own data or use the pretrained models as input to computer games or other applications.

Position and depth

Depth and position information from video can be useful for many applications. The vid2depth project by Google Brain, is exciting due to its relative simplicity.

<https://sites.google.com/view/vid2depth>

Train or finetune the vid2depth model, for your own application. Use the result, to e.g. 3D map a room or to find your way back in a unknown building.

ps. you don't necessarily need the ICP loss

Text to Image synthesis

Given a textual description, generate an image. Example, 'The flower has red petals' generates a rose.

Possible dataset: Oxford 102 Flower dataset.

Techniques: word vectors, RNN/LSTM and GAN.

Multimodal classification

For example given an image and a descriptive text classify the object referred in both text and image to a suitable class.

Possible dataset: Oxford 102 Flower dataset.

Techniques: word vectors, RNN/LSTM, CNN

Visual Question answering

Given an image and a question related to the image generate an answer.

Possible dataset: VQA, <http://visualqa.org/>.

Techniques: Word vectors, RNN/LSTM, CNN, Attention.

Question answering

Given a paragraph of text and a question, generate the answer.

Possible dataset: SQUAD (<https://rajpurkar.github.io/SQuAD-explorer/>).

Techniques: Word vectors, RNN/LSTM, Attention

Dialog agent

A system that can converse with a human through text.

Databases: (BABI, <https://research.fb.com/downloads/babi/>, Cornell Movie dialog https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html).

Techniques: Word vectors, RNN/LSTM, Attention, Reinforcement learning.

End-to-end learning for autonomous vehicles

A system predicting the steering angle given the camera and Lidar images.

Database: FFI UGV database.

Techniques: CNN, RNN.

Improve Surround player

Improving the surround player by implementing a (simplified) version of the AlphaGo Zero algorithm.

Paper : Silver, David, et al. "Mastering the game of Go without human knowledge." *Nature* 550.7676 (2017): 354.

Dataset: Self-generated

Techniques: CNN, Reinforcement Learning

Image classification with “sequential/random” sequence of glimpses

Divide each image into several blocks (possibly with some overlap), and create a sequence to give as input to an RNN. You may also e.g. give the RNN as input where in the image it is looking to see if this improves performance or not.

Possible dataset: CIFAR-10

Techniques: (CNN), RNN, (Attention)

Image classification with learned attention policy

Image classification by using several glimpses that only see part of the image. We also learn the prediction policy, i.e. where to look next.

Paper : Mnih, V., Heess, N., & Graves, A. (2014). Recurrent models of visual attention. In *Advances in neural information processing systems* (pp. 2204-2212).

Techniques: (CNN), RNN, Attention, Reinforcement Learning

Possible dataset : Cluttered MNIST