

Portfolio

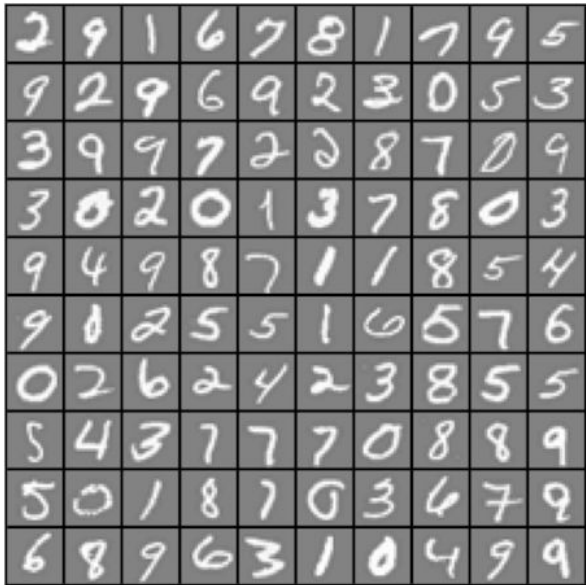
<https://ino-jeong.github.io/>

정인오

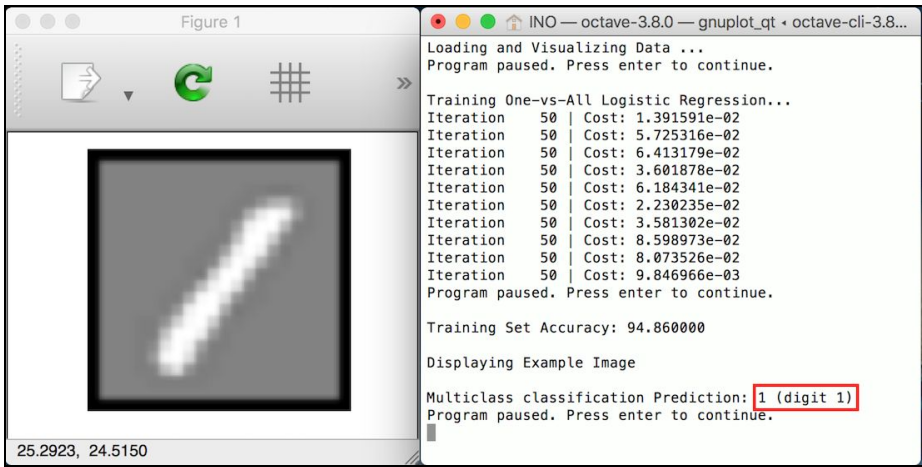
ino.h.jung@gmail.com

010-9907-9386

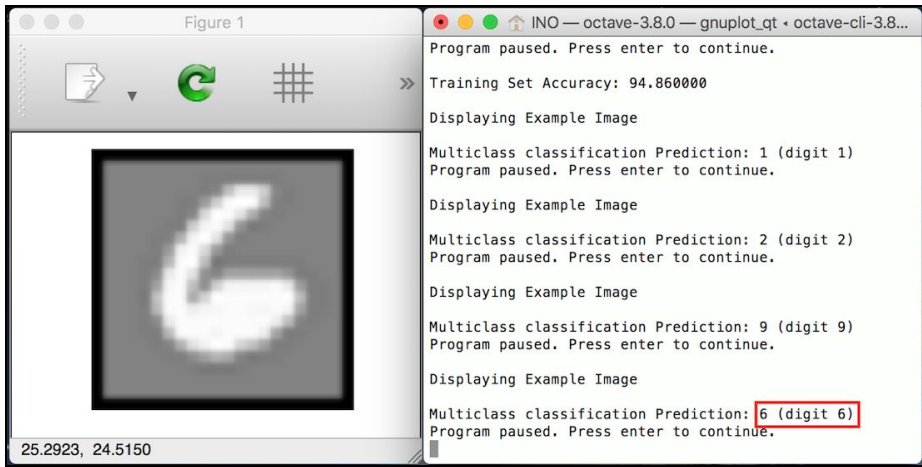
1. OCR implementation, multi-class classification (Coursera)



data set sample



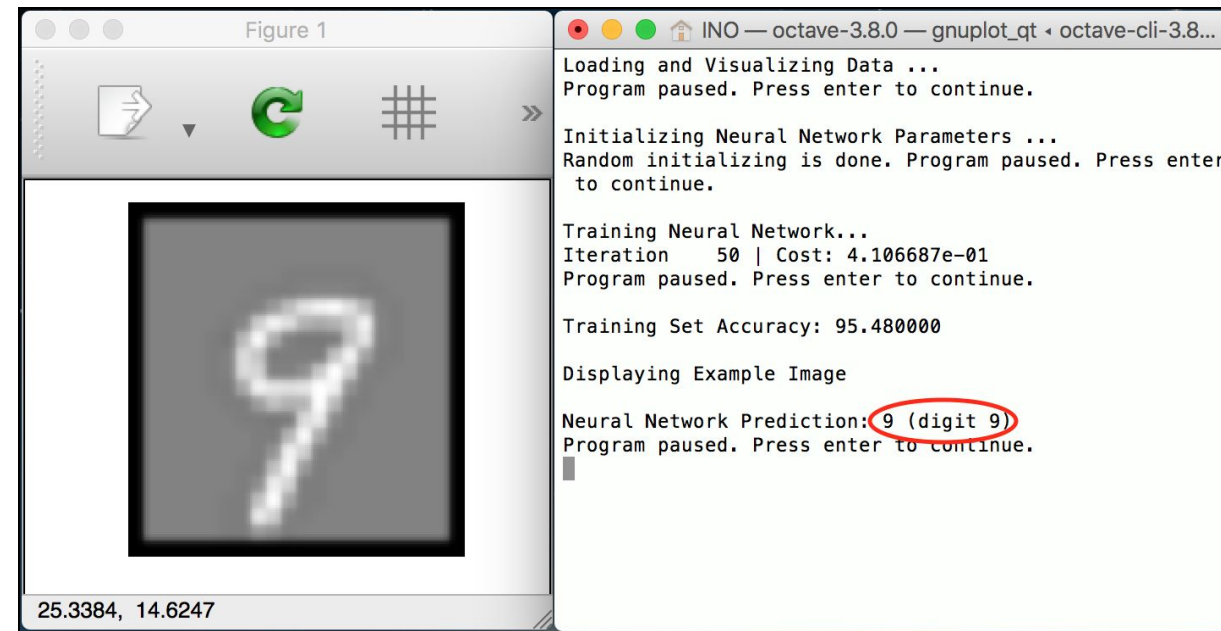
training set prediction (1)



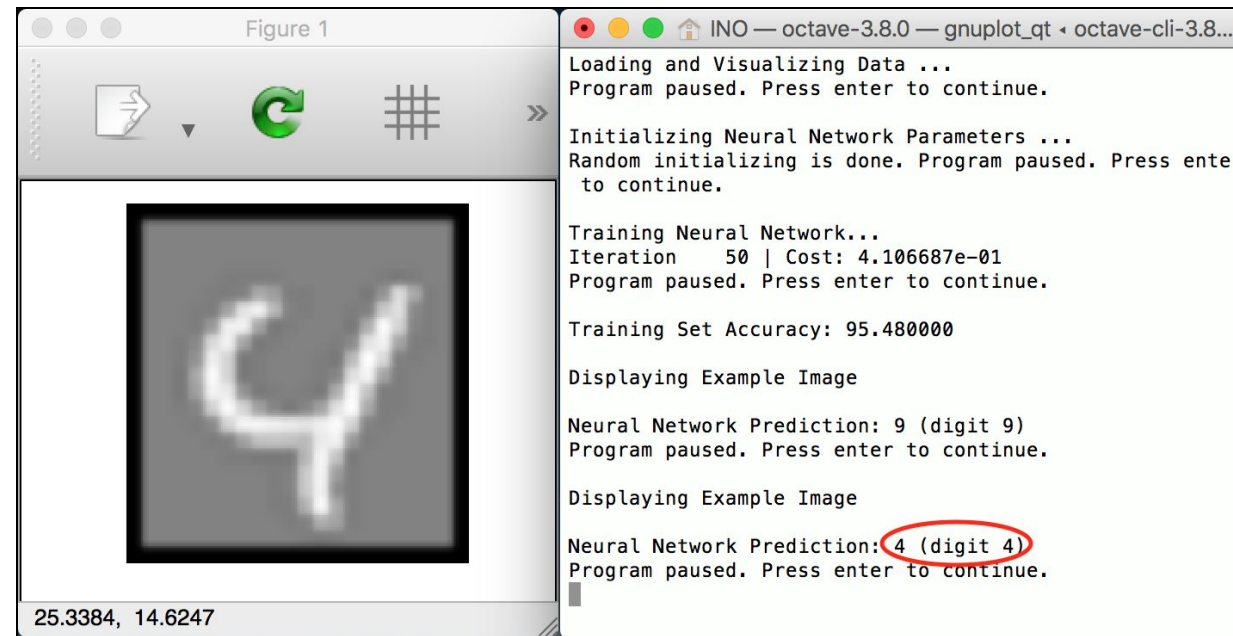
training set prediction (2)

- [https://github.com/ino-jeong/Portfolio/tree/master/OCR\(multiclass_classification\)](https://github.com/ino-jeong/Portfolio/tree/master/OCR(multiclass_classification))
- Training set accuracy : 94.86%
- Octave(추천) 또는 Matlab에서 main.m 실행
- 구현환경 : GNU Octave 3.8, Mac OS
- Coursera Machine Learning 과정 구현 과제
- Training set : 20 X 20 pixel, grayscale, 5000 examples of handwritten digits
- Model : Multi-class classification
- Cost function 및 Training / Prediction 과정 구현 :
 - lrCostFunction.m
 - oneVsAll.m
 - predictOneVsAll.m

2. OCR implementation, neural-net (Coursera)



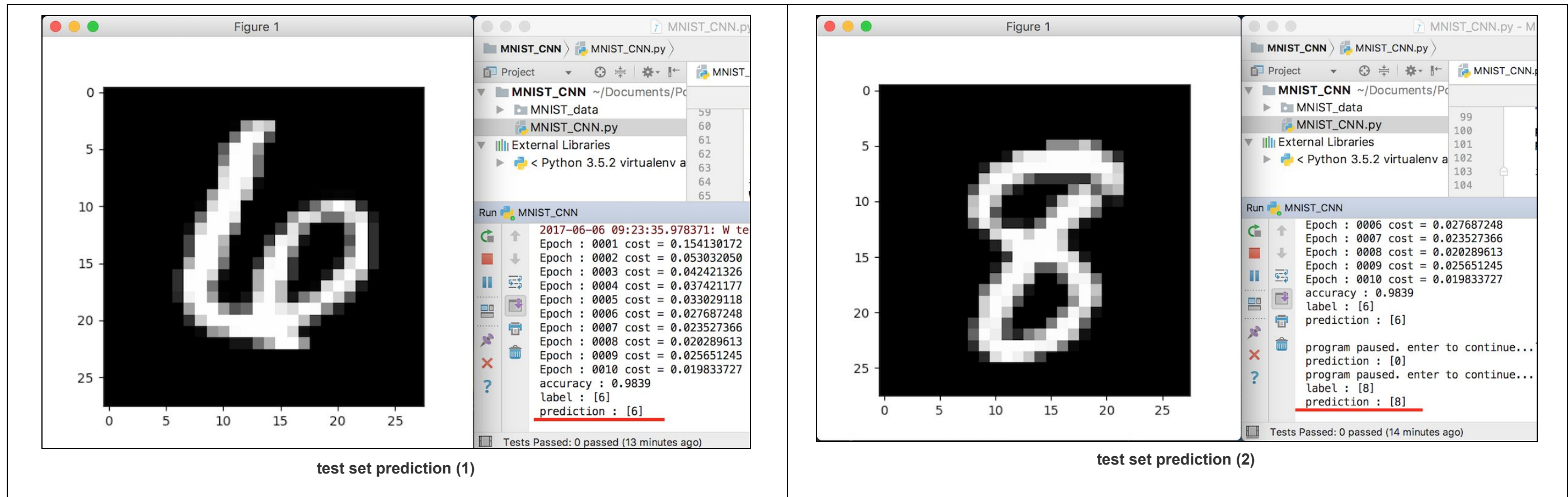
training set prediction (1)



training set prediction (2)

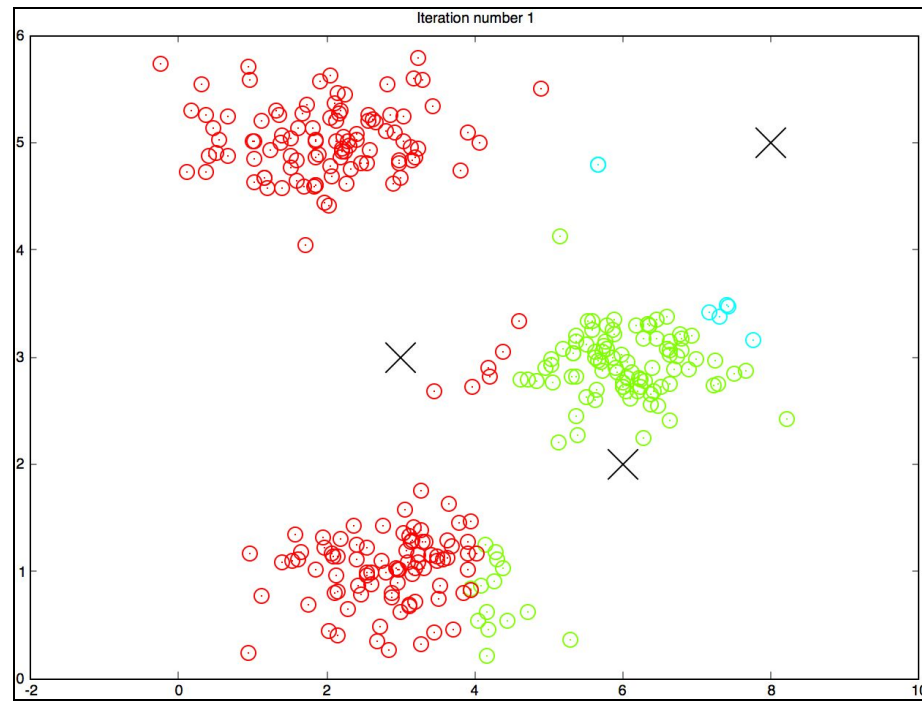
- [https://github.com/ino-jeong/Portfolio/tree/master/OCR\(neural_net\)](https://github.com/ino-jeong/Portfolio/tree/master/OCR(neural_net))
- Training set accuracy : 95~96% (up to random initialization)
- Octave(추천) 또는 Matlab에서 main.m 실행
- 구현환경 : GNU Octave 3.8, Mac OS
- Coursera Machine Learning 과정 구현 과제
- Training set : 20 X 20 pixel, grayscale, 5000 examples of handwritten digits (1번과 동일 set)
- Model : Neural Net, 3 layer (1 hidden layer)
- Layer 구성 및 backpropagation 구현 :
 - sigmoidGradient.m
 - nnCostFunction.m

3. MNIST with CNN implementation



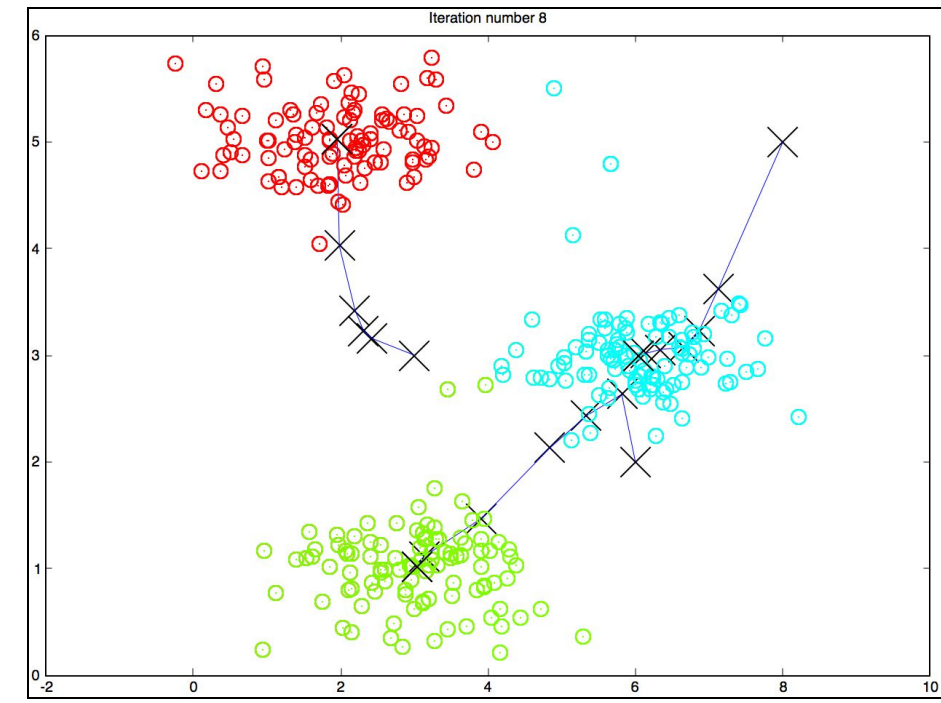
- https://github.com/ino-jeong/Portfolio/tree/master/MNIST_CNN
- Test set accuracy : 98.39% ~ 98.67% (if number of epoch is increased)
- 구현환경 : Python 3.5 with Tensorflow 1.1, Mac OS
- CNN을 통한 MNIST classifier 구현 (하기 reference 참조) :
 - Tensorflow official tutorial https://www.tensorflow.org/get_started/mnist/pros
 - 'DeepLearningZeroToAll' lecture by prof. Sunghun-Kim <http://hunkim.github.io/ml/>
- Training set : as per MNIST specification (28 X 28 pixel, grayscale)
- Model : Convolution Neural Network :
 - 1st layer :
convolution with 3x3 filter, 1 channel in / 32 channel out → ReLu → Max-Pooling with 2x2 filter
 - 2nd layer :
convolution with 3x3 filter, 32 channel in / 64 channel out → ReLu → Max-Pooling with 2x2 filter
 - 3rd later :
Fully connected layer

4. K-means clustering (Coursera)



before k-means clustering

(same color means they treated as same group)

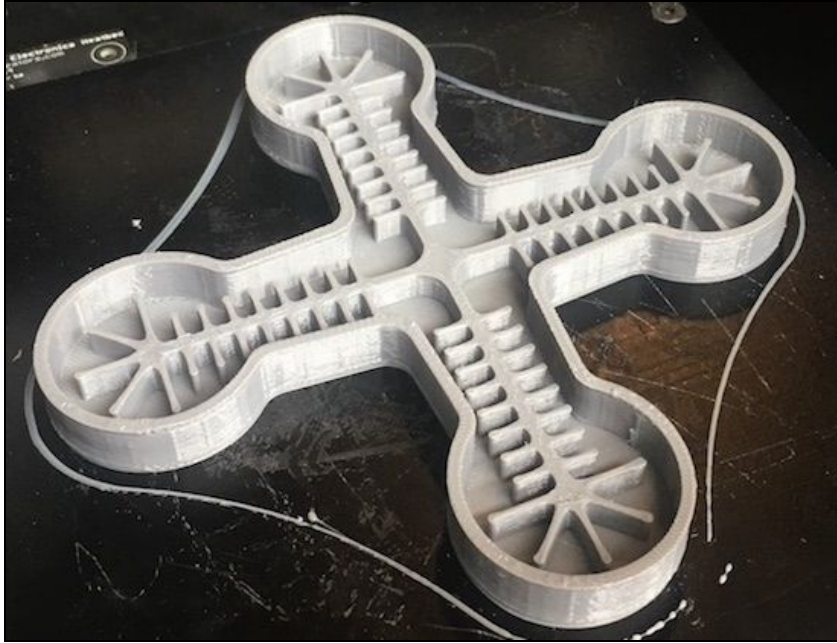


after 8-iteration of k-means

(same color means they treated as same group)

- https://github.com/ino-jeong/Portfolio/tree/master/k_means
- Octave(추천) 또는 Matlab에서 main.m 실행
- Basic k-means clustering implementation
- 구현환경 : GNU Octave 3.8, Mac OS
- Coursera Machine Learning 과정 구현 과제
- Model : K-means
- K-means clustering algorithm 구현(finding 3 clusters in examples) :
 - computeCentroids.m
 - findClosestCentroids.m
 - kMeansInitCentroids.m

5. Soft Robotic Gripper Fabrication



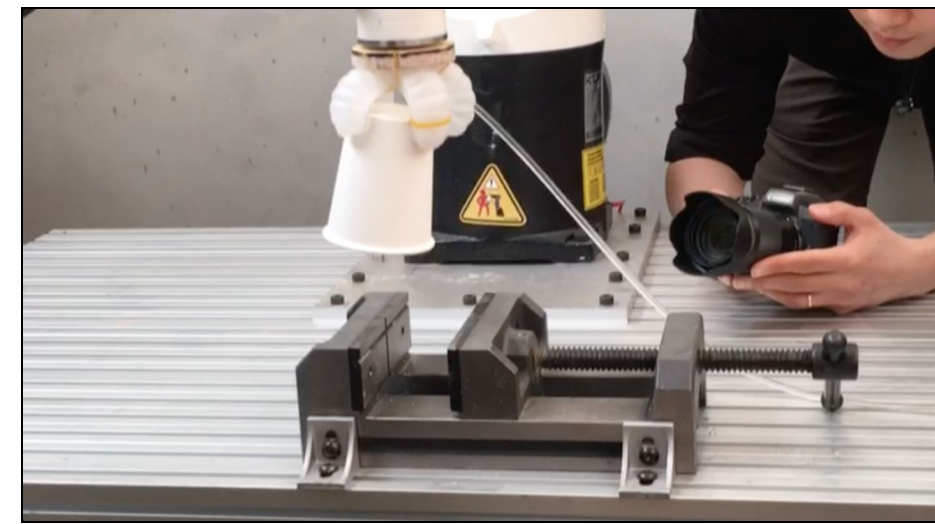
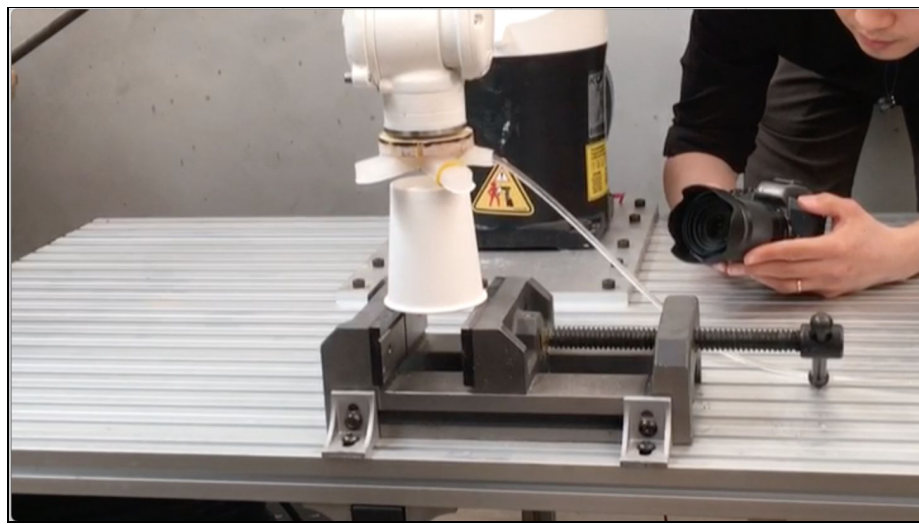
3d printed gripper mold



casted silicone gripper



attached on robotic arm



robot grip test

- Robot arm :
 - (Kuka) KR 6 R900 sixx KR AGILUS
- Soft robotic gripper :
 - Material : Ecoflex 00-30 silicone
 - Dimension : W 135mm x L 135mm x H 10 mm
 - Air powered
 - Mold : 3d printed (by ultimaker), pla
 - Office paper used for inelastic side (inner side) constrainer