UNIVERSITY OF BIRMINGHAM

Deep image classifier using simulated STEM images

By Ziyue Wang MSci 4th Year

Contents

- Introduction
- Aims of project
- Simulation
- Machine learning (transfer learning)
- What to do next

Introduction

- Automated image classification is needed for mass amounts of produced STEM images, which can be done through deep learning
- Hasn’t been done explicitly for STEM, with fully 3D rotational objects with deformations
- Possible eventual use: classification of chiral catalysts in commercial production

Aims

- Write a simulation program capable of producing simulated and experimental-like simulated STEM images
- Use transfer learning or build own deep network to classify images into categories dependent on deformation e.t.c, using a single image as input
- Create neural network which takes in multiple experimental-like images to find the ground truth

Simulation

- Written in Python (OOP)
- Capable of producing triangular based pyramids, cuboids and spheres
- Deformation (at corners) of cuboid implemented
- Objects can be rotated during imaging/between images (using quaternions)
- Gaussian blur filter, background noise (carbon layer) and Poisson noise (shot noise) added

Rotations about all axis with single deformation

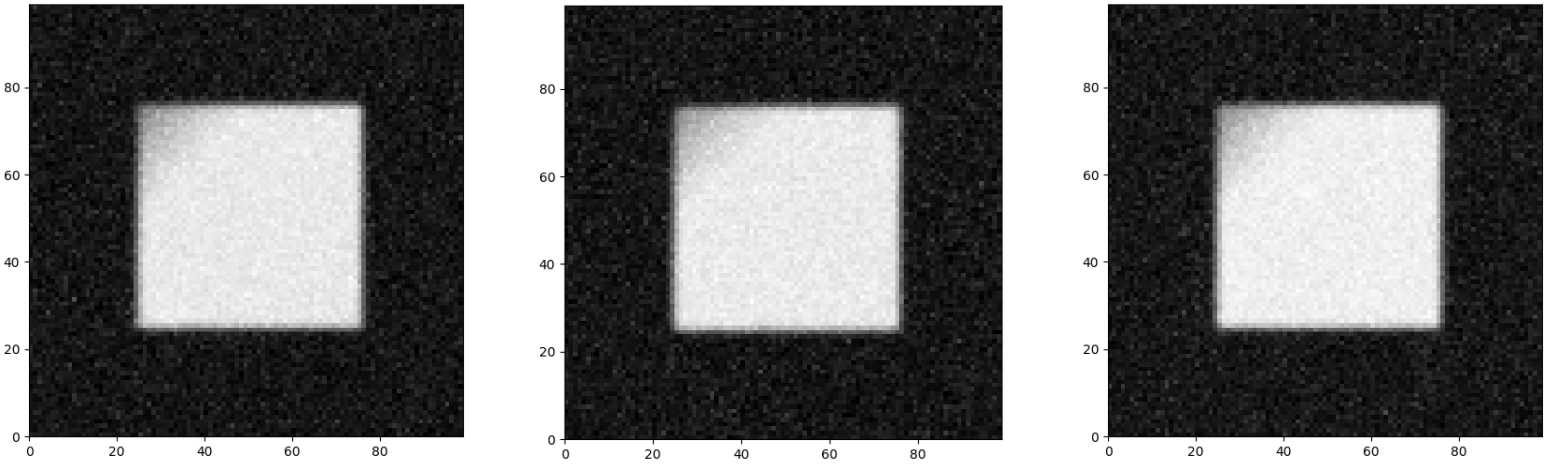

Fig 1: Cube with singular deformation rotated around z,y,x axis from left to right respectively. All plots are 100x100 pixels with depth of 100, in A.U. Multiple images compiled together for these gifs.

Example images

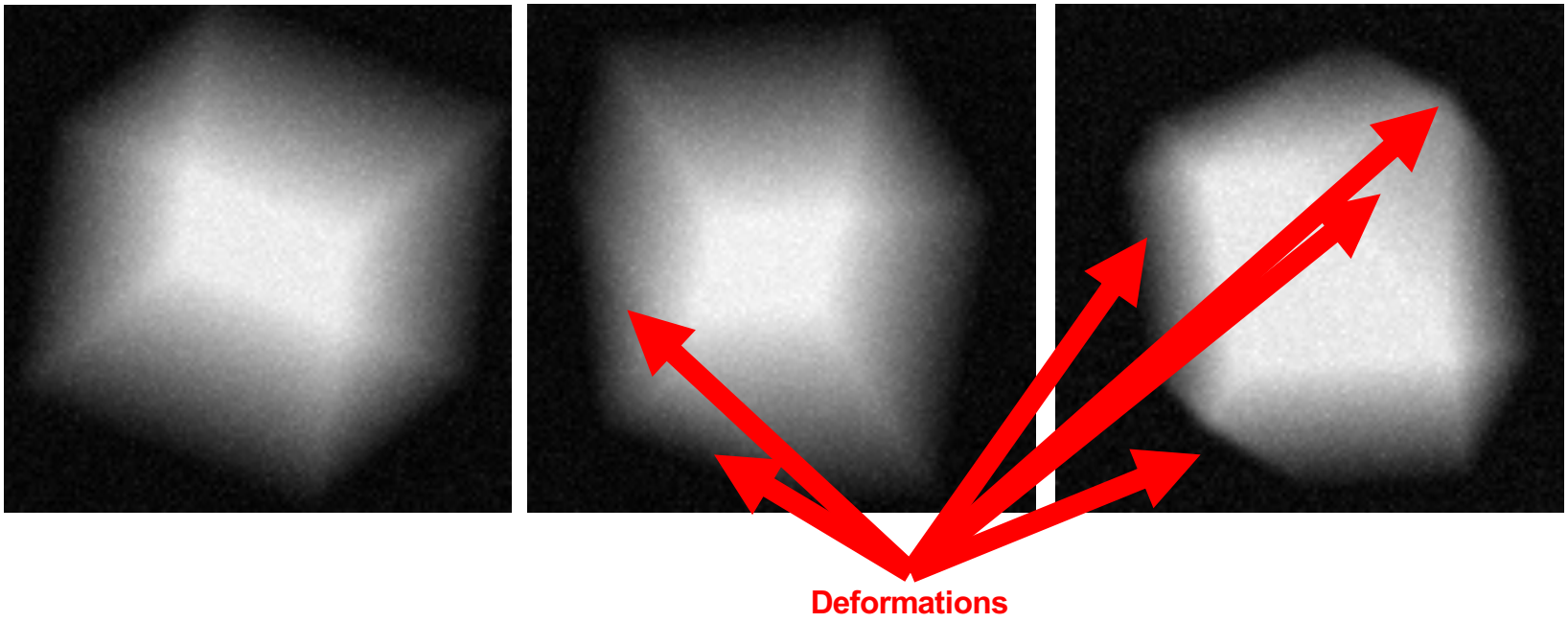

Fig 2: Cuboid objects (varying dimensions and rotations) of 0, 2 and 4 deformations (left to right respectively, 128 x 128 pixels).

Higher number deformations

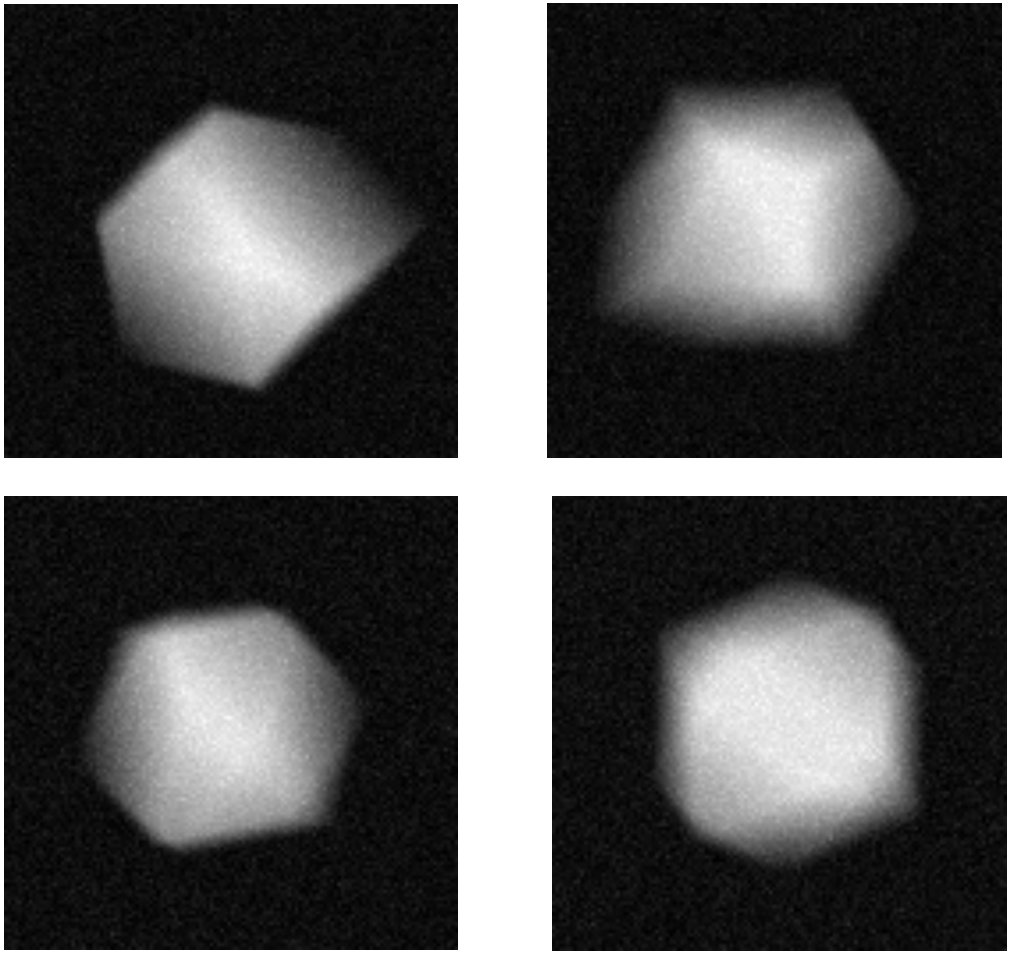

Fig 2: Cuboid objects (varying dimensions and rotations) of 5, 6, 7 and 8 deformations (left to right respectively top to bottom, 128x128 pixels).

Rotation during imaging

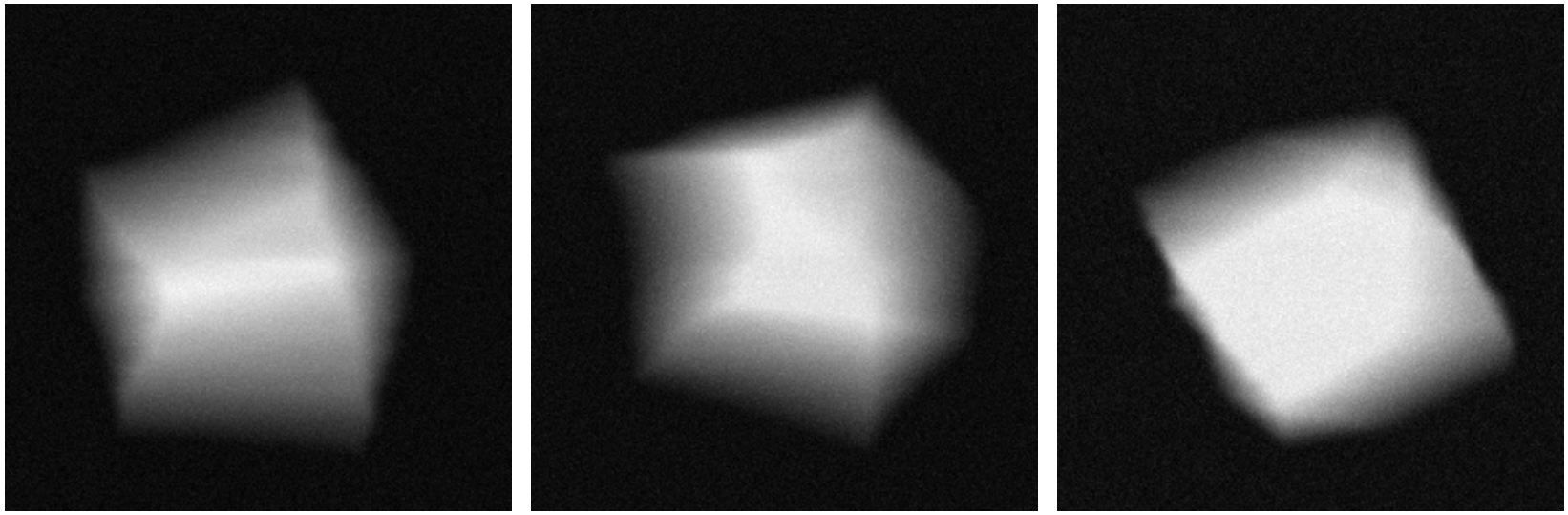

Fig 3: Cuboid objects (varying dimensions and rotations) of 0, 2 and 4 deformations (left to right respectively, 128x128 pixels) under rotation throughout imaging process.

Progress of simulation – production of images for training

- Code has been optimized (sped up by ~3x)
- Can run script on multiple threads simultaneously (only on individual cores) – lower overall processing time
- For simplicity, rotation during imaging has been turned off for initial machine learning

Machine learning

- Begin with transfer learning, only retraining last layer of Convolutional neural network (CNN).
- Using retrain.py provided in tensorflow package (uses inception-v3 2015) (https://github.com/tensorflow/hub/blob/master/examples/image_retraining/retrain.py)
- Classifier of no deformation vs any deformation – test accuracy 94.4% (trained with 1000 images of each category)

Classifying various deformations

- Classifier of varying number of deformations – low accuracy ~ 55% test accuracy (used 128x128 pixels)
- Attempt to classify same number of deformations based on deformation permutations.
- 100x100 pixels with 2 deformations provide low success (~50% test accuracy)

Proposed feature detectors

- Number of vertices (2D) – more deformations -> more vertices. This can be also done for corners (3D) using edge detection.
- Total volume of object
- Distance between corners

Future plans – to do list

- Write own specific layer(s), targeting the deformations
- Increase pixel density to 256x256 and retry deformation location classifier.
- Increase data set size
- Develop the ground truth classifier

Summary

- Simulation capable of producing simulated STEM images of various objects with deformations.
- Initial transfer learnt neural network can identify no deformations vs 1-8 deformations very well
- Need to find a way for classifier to identify varying deformations (and permutations of deformations)