Machine Learning Algorithms for Automated NIF Capsule Mandrel Selection

By
Kurt Boehm
General Atomics

Presented to
Target Fabrication Meeting
Annapolis, MD

April 2019

Work was performed under General Atomics IR&D Program

- PAMS capsule mandrels are used for production of GDP and beryllium shells
- Each mandrel is inspected for roundness and surface defects

**Automation leads to:**
- **Better shells**
  - Higher throughput
  - Consistent results
- **Less human interaction**
  - Less labor hours
  - Operator independent results
- **Provide statistics**
  - Quantitative feedback mechanism on batch quality
  - How many defects and what type?
- **Increase downstream production yield**
  - More costly to discover defects during downstream operations
Shell selection process with no automation

- Pour
- Sort
- Load

Human Looking through Microscope

Load

30 shells

AFM

1 shell
Some Automaton was added to the process in 2015

Current Demand requires 0.25 FTE to sift and sort through shells

- Inspect ~650 shell mandrels image sets/week
  - 4-6 man hours of loading and placing shells into grids
  - 4-6 man hours /week of looking at images and recording results
    - Optical Yield: 30%
    - AFM Yield: 30%
    - TOTAL Yield <10%
- Identify 24 shell mandrels to go into GDP coater
  - 2-3 batches of 24 shells/ month
  - Metrologize, select best ones
- Capsules are built into CFTA’s and shipped to LLNL
  - 2 CFTA’s/week (2017 and 2018)
Complete Mandrel Selection Process Has Been Automated
Automated Shell Classification Pass/Fail based on AZ-100 images using Deep Learning Algorithm

Deep Learning requires large data set
- 1000’s of classified images are available from past production runs
- Algorithm looks at top and bottom image of shell and classifies as pass or fail by comparing it to historical data
Example: Recognizing MNIST Handwritten Digits using a Two-Layer Network

<table>
<thead>
<tr>
<th></th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>53%</td>
</tr>
<tr>
<td>1</td>
<td>2%</td>
</tr>
<tr>
<td>2</td>
<td>1%</td>
</tr>
<tr>
<td>3</td>
<td>4%</td>
</tr>
<tr>
<td>4</td>
<td>1%</td>
</tr>
<tr>
<td>5</td>
<td>7%</td>
</tr>
<tr>
<td>6</td>
<td>4%</td>
</tr>
<tr>
<td>7</td>
<td>1%</td>
</tr>
<tr>
<td>8</td>
<td>15%</td>
</tr>
<tr>
<td>9</td>
<td>12%</td>
</tr>
</tbody>
</table>

Input Layer

Hidden Layer

Output Layer
Convolutional Neural Networks add more and more Network Layers and use a “Sliding Filter”
Transfer Learning Allows Application of Existing Architectures and their Trained Weights to new Problems

Fine-tune a pre-trained model (Transfer learning)

Pre-trained CNN (AlexNet – 1000 Classes)

VGG19 Network Structure

Modify these two layers and re-train on new problem data set
Machine Learning for Image Classification can Automate Microscope Image Inspection

BAD

GOOD

1.5-2 mm
Classification Suffers from Ambiguity of Data Set

Operator dependency makes it difficult to set success criterion
Reasonable results could be achieved using pre-trained networks on commercially available software packages

- Only small modification needed in the last couple of layers to apply these networks to our problem
- Ran four network architectures to compare
- Used increasing size of input data (500, 1000, 2000 images) of past production data
- Test Data set of 206 images, 103 good ones 103 bad ones

<table>
<thead>
<tr>
<th></th>
<th>TOTAL</th>
<th>% age accuracy</th>
<th>BAD</th>
<th>% age</th>
<th>GOOD</th>
<th>% - age</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>176</td>
<td>0.85</td>
<td>84</td>
<td>0.82</td>
<td>92</td>
<td>0.89</td>
</tr>
<tr>
<td>VGG16</td>
<td>179</td>
<td>0.87</td>
<td>86</td>
<td>0.83</td>
<td>93</td>
<td>0.90</td>
</tr>
<tr>
<td>GoogleNet</td>
<td>177</td>
<td>0.86</td>
<td>87</td>
<td>0.84</td>
<td>90</td>
<td>0.87</td>
</tr>
<tr>
<td>VGG19</td>
<td>177</td>
<td>0.86</td>
<td>76</td>
<td>0.74</td>
<td>99</td>
<td>0.96</td>
</tr>
<tr>
<td>Operator Reclassify</td>
<td>155</td>
<td>0.75</td>
<td>70</td>
<td>0.68</td>
<td>85</td>
<td>0.83</td>
</tr>
</tbody>
</table>

VGG19 recognizes good shells, but can’t find all the bad ones
Complimentary inspection was implemented while loading shells into inspection tray.

Data Processing

Load

300 shells

Az-100 Inspection

30 shells

Bad shells

30 shells

Auto AFM

1 shell
Concept: Using Machine Learning Algorithms to sort shells in pre-inspection step
Optical system on culling robot yields higher contrast images showing defects with higher clarity.
We looked at what kind of defect is on the shell to determine pass / fail

- Implosion
- Vacuoles
- Bubbles
- Star Crack
- Crack
- Residue
- Debris
- Ring Cracks
- Dents

Courtesy of Wendy Sweet
For Culling Images: Find things that are interesting in an image and then find out what they are

More common commercial application of this problem:
- Identify multiple objects in an image

Process:
- Draw a box around an object of interest
- Query a different CNN to figure out what the object is
Defect recognition reaches >95% accuracy

Samples from Training Dataset:

- Crack
- Vacuole
- Points

Spider Cracks
Water Spots

Example results from running the code
Defect Analysis in Action

4 x Speed
Defect type statistics will be used to improve shell production processes.

Defect statistics from 1000 shells in 10 batches

(100 shells/batch)

Getting this level of detailed statistics from a manual process is prohibitively expensive.
Combining both Machine Learning-algorithms shows 91% accuracy for good shells with a 10-20% drop in yield*.

*Compared to manual process
Summary: Applying Computer Vision, Machine Learning and Automation Fulfills its Promise for Capsule Selection

- Better shells
  - Higher throughput
  - Consistent results
- Less human interaction
  - Operator independent results
  - Less labor hours
- Provide statistics
  - Quantitative feedback mechanism on batch quality
- Increase downstream production yield
  - Additional and complimentary inspection of shells

Reduce operator involvement from 8-12 hours / week to handling trays

For free!!!