GENERATIVE ADVERSARIAL NETWORKS AS A NOVEL APPROACH FOR TECTONIC FAULT AND FRACTURE EXTRACTION IN HIGH-RESOLUTION SATELLITE AND AIRBORNE OPTICAL IMAGES

Bahram JAFRASTEH¹, ², Isabelle MANIGHETTI¹ and Josiane ZERUBIA²

¹ Université Côte d’Azur, Géoazur, Sophia Antipolis, France
² Université Côte d’Azur, Inria, Sophia Antipolis, France
Natural fractures & faults form dense, complex networks of curvilinear traces at ground surface

- Fractures and Faults are ubiquitous on Earth
- Responsible for earthquakes, landslides, reservoir fracturing, etc.
- Form dense complex networks
- Fault planes generally intersect ground surface, forming complex networks of curvilinear traces
- Mapping of fault traces commonly done manually at ground surface or in remote images – **But** very time consuming
- **→ Need to develop a fast, reliable and accurate automatic mapping method**: DEEP LEARNING

Valley of Fire, Nevada

Field photogrammetry *(photo courtesy of I. Manighetti)*

Granite Dells, Arizona

Drone Orthophotography
Ground truth: optical images + manual fault maps

Example of a fault site in Granite Dells, Arizona

Ground truth:
- Red, Green and Blue bands of georeferenced optical image
- Expert manual fault mapping

Approach:
- Actual thickness of fault traces (i.e., several pixels) represented with a Gaussian distribution function
- Binary approach, “fault – not a fault”
- 3840 images used for training

Image resolution: 0.0005 m
Testing Generative Adversarial Network (GAN)
to map faults and fractures

- Generator tries to recognize the fault patterns and reproduces them as images
- Generator minimizes the difference between its synthetic fault images and the expert fault mapping
- Discriminator discriminates the expert and the synthetic mapping
- Based on the Discriminator feedback, the Generator network learns to map the faults more accurately
Testing CNN U-Net to map faults and fractures

- **Training stage:** the model learns from expert manual mapping how faults look like in optical image
- **Validation stage:** verifying model efficiency
- **Test stage:** Calculation of model accuracy

➔ Each stage = different images
Network fed with different images in each training stage
Results: automatic fault extraction with GAN

- **Manual fault traces**
  - Number of fault pixels = 8
- **Network output**
  - Probability of being fault > 0.5
  - Number of pixels correctly identified as fault = 6
  - Probability > 0.5
    - Pixel probability threshold to be a fault ≥ 0.7
    - Recall (%) = (6 / 8)*100 = 75%

- **Probability > 0.5**
  - Pixel probability threshold to be a fault ≥ 0.7

- Scale: 120 cm and 30 cm
Results: automatic fault extraction with U-net

Manual fault traces

<table>
<thead>
<tr>
<th></th>
<th>1.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Number of fault pixels = 8

Network output

<table>
<thead>
<tr>
<th>Probability of being fault &gt; 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
</tr>
<tr>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
</tr>
</tbody>
</table>

Probability > 0.5

<table>
<thead>
<tr>
<th>Probability &gt; 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
</tr>
<tr>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
</tr>
</tbody>
</table>

Number of correctly identified pixels as fault = 6

Recall (%) = (6 / 8) * 100 = 75%
Results: comparison between U-net and GAN

**GAN**
- High GPU memory
- Longer training time (RTX 2080-11 GB, ~48 h)
- High probability values for identified faults
- More appropriate for fault pattern simulation

**U-net**
- Shorter training time (RTX 2080-11 GB, ~24 h)
- Different probability values for identified faults
- More appropriate for fault identification

![Image of GAN results](image)
![Image of U-net results](image)
Results: comparison between U-net and GAN

**GAN**
- High GPU memory
- Longer training time (RTX 2080-11 GB, ~48 h)
- All identified faults with high probability
- More appropriate for fault pattern simulation

**U-net**
- Shorter training time (RTX 2080-11 GB, ~24 h)
- Different probability values for identified faults
- More appropriate for fault identification

---

![Graph showing comparison between U-net and GAN](image)

- **Y-axis**: Percentage of hand-mapped faults correctly identified by model
- **X-axis**: Pixel probability threshold to be a fault
Conclusions

- U-net more appropriate for fault mapping in optical images

- With U-net, more than 60% of the hand-mapped faults are correctly identified

- Although trained with a small dataset, the model has good generalization ability: it well predicts faults in unseen parts of the image

Perspectives

- Examine further the generalization ability of the models (use model to extract faults from different image types and resolutions, including satellite and aerial images)

- Predict the hierarchy of the faults: major faults, minor faults, more uncertain faults

- Convert the predicted probabilities into vector lines for statistical analysis of fault networks: lengths, densities, azimuths, cross-cutting relations, etc.

- Quantitative description of fault networks, useful for rock mechanics and earthquake physics