Motivation

- Adversarial examples have recently been shown to successfully trick object detection networks trained on satellite imagery \([1]\).
- Dangerous implications for national defense
- Increasingly important as systems move to real-time: by using hand-selected features surrounding a detected object, can create a lightweight algorithm to boost prediction accuracy.

Data

- Data comes from (Large Dataset for Object Detection in Aerial Images) \([2]\)
- 4000 “macro-images” with bounding box labels
- Classes: 'ship', 'large-vehicle', 'storage-tank', 'airplane'
- Training/Test: 3134/385/534 “sub-images”
- Limited data = need complex models to get insight from data

Features

- Features for YOLO = CNN
  - Per class for context-gen: 1/(avg_dist to objects)
  - Counts
  - Average angle
  - Made features for sub-images and macro-images
- Context Algorithms
  - K-NN
  - Object classified based on k nearest points
  - Black Box Classifier
  - Black out object and train CNN on surrounding pixels
  - Linear SVM
  \[ \min \left( \sum_{i=1}^{n} \max \left( 0, 1 - y_i (a_i - b_i) \right) + \lambda \sum_{w \in W} w^2 \right) \]
  - RBF SVM:
  \[ K(x, x') = \exp \left( - \frac{|x - x'|^2}{2\sigma^2} \right) \]
- Decision Tree
- Random Forest: Constructs multiple decision trees
- Simple NN (MLP): 3 layers, \( \alpha = 1 \)
- Naïve Bayes: uses counts to determine probabilities
- QDA: GDA but each class has its own covariance
- AdaBoost:
  - Fitting generic weak classifiers
  \[ F(x) = \text{sign} \left( \sum_{i=1}^{n} \alpha_i u_i(x) \right) \]

Models

- YOLOv3 \([3]\)
  - CNN for Object Detection and Classification
  - Outputs predicted class probabilities and bounding box corners for detected objects

Results

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Description (size)</th>
<th>Metric</th>
<th>Sub_1</th>
<th>Sub_2</th>
<th>Sub_3</th>
<th>Sub_4</th>
<th>Macro_1</th>
<th>Macro_2</th>
<th>Macro_3</th>
<th>Macro_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub_1</td>
<td>Counts of surrounding classes in sub-image [4]</td>
<td>K-NN</td>
<td>0.952</td>
<td>0.711</td>
<td>0.861</td>
<td>0.975</td>
<td>0.896</td>
<td>0.905</td>
<td>0.998</td>
<td>0.467</td>
</tr>
<tr>
<td>Sub_2</td>
<td>Sub_1 + avg distances to classes in sub-image (8)</td>
<td>LinSVM</td>
<td>0.903</td>
<td>0.824</td>
<td>0.904</td>
<td>0.950</td>
<td>0.892</td>
<td>0.909</td>
<td>0.879</td>
<td>0.852</td>
</tr>
<tr>
<td>Macro_1</td>
<td>Counts of surrounding classes in macro-image [4]</td>
<td>RBF</td>
<td>0.927</td>
<td>0.883</td>
<td>0.878</td>
<td>0.963</td>
<td>0.887</td>
<td>0.851</td>
<td>0.897</td>
<td>0.879</td>
</tr>
<tr>
<td>Macro_2</td>
<td>Macro_1 + avg distances to classes in macro-image (8)</td>
<td>DT</td>
<td>0.911</td>
<td>0.889</td>
<td>0.926</td>
<td>0.943</td>
<td>0.866</td>
<td>0.881</td>
<td>0.913</td>
<td>0.772</td>
</tr>
<tr>
<td>Macro_3</td>
<td>Macro_2 + avg angles to classes in macro-image (8)</td>
<td>RF</td>
<td>0.932</td>
<td>0.883</td>
<td>0.890</td>
<td>0.942</td>
<td>0.902</td>
<td>0.960</td>
<td>0.914</td>
<td>0.824</td>
</tr>
<tr>
<td>Macro_4</td>
<td>Macro_2 + Macro_4 (12)</td>
<td>MLP</td>
<td>0.915</td>
<td>0.860</td>
<td>0.902</td>
<td>0.949</td>
<td>0.895</td>
<td>0.918</td>
<td>0.885</td>
<td>0.849</td>
</tr>
<tr>
<td>Ada</td>
<td></td>
<td>NB</td>
<td>0.844</td>
<td>0.819</td>
<td>0.840</td>
<td>0.879</td>
<td>0.882</td>
<td>0.895</td>
<td>0.811</td>
<td>0.886</td>
</tr>
<tr>
<td>QDA</td>
<td></td>
<td>GDA</td>
<td>0.846</td>
<td>0.820</td>
<td>0.838</td>
<td>0.882</td>
<td>0.884</td>
<td>0.870</td>
<td>0.809</td>
<td>0.885</td>
</tr>
<tr>
<td>Ada</td>
<td></td>
<td>YOLOv3</td>
<td>0.918</td>
<td>0.897</td>
<td>0.869</td>
<td>0.930</td>
<td>0.884</td>
<td>0.804</td>
<td>0.849</td>
<td>0.818</td>
</tr>
</tbody>
</table>

References

4. github: https://github.com/yashc95/context4sats

Future Work

- Compute final boost to YOLO classification
- Potentially put context-algorithm in the loop with YOLO
- Assess time-complexity in addition