Region Based Object Detectors for Recognizing Birds in Aerial Images

M.S. Thesis Defense
By: Pedro Carrion Castagnola
Advisor: Dr. Yi Shang
Outline

1. Introduction
2. Related Work
3. Design and Implementation
4. Experiments
5. Results
6. Conclusions and Future Work
Outline

1. Introduction
   - Problem
   - Motivation
2. Related Work
3. Design and Implementation
4. Experiments
5. Results
6. Conclusions and Future Work
Problem

- Missouri Department of Conservation: Protection and management of birds in different areas in Missouri
- Recognition and counting of birds
- Data: Aerial images
- Difficult problem
  - Birds look small in the images
  - Blurriness
  - Different shapes and colors
  - Complex backgrounds
Motivation

- Try a new type of Neural Network recognize the birds: Capsule Network
- Capsule Network only does classification. First generate region proposals
- Compare the performances of Capsule Network with a simple CNN and a state of the art neural network
- Explore network parameters that can improve the results
Outline

1. Introduction

2. Related Work
   - Dataset
   - Convolutional Neural Network
   - Capsule Network
   - ResNet50 + FPN

3. Design and Implementation

4. Experiments

5. Results

6. Conclusions and Future Work
Previous Work

- Crop images into 512 x 512 px
- Divided dataset into Easy and Hard datasets

Easy image

Hard image
Labels

- One coordinate point approximately at the center of each bird is known
- Labels generation: 20 by 20 and 30 by 30 pixels bounding box size
- 4 Datasets will be used

<table>
<thead>
<tr>
<th>#</th>
<th>Dataset Label Size</th>
<th>Dataset Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20 x 20</td>
<td>Easy</td>
</tr>
<tr>
<td>2</td>
<td>20 x 20</td>
<td>Hard</td>
</tr>
<tr>
<td>3</td>
<td>30 x 30</td>
<td>Easy</td>
</tr>
<tr>
<td>4</td>
<td>30 x 30</td>
<td>Hard</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>20x20 Label</th>
<th>30x30 Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample 1</td>
<td><img src="image1" alt="Sample 1" /></td>
<td><img src="image2" alt="Sample 1" /></td>
</tr>
<tr>
<td>Sample 2</td>
<td><img src="image3" alt="Sample 2" /></td>
<td><img src="image4" alt="Sample 2" /></td>
</tr>
<tr>
<td>Sample 3</td>
<td><img src="image5" alt="Sample 3" /></td>
<td><img src="image6" alt="Sample 3" /></td>
</tr>
<tr>
<td>Sample 4</td>
<td><img src="image7" alt="Sample 4" /></td>
<td><img src="image8" alt="Sample 4" /></td>
</tr>
<tr>
<td>Sample 5</td>
<td><img src="image9" alt="Sample 5" /></td>
<td><img src="image10" alt="Sample 5" /></td>
</tr>
</tbody>
</table>
Convolutional Neural Network

- Convolutional layers - Feature Maps
- Scalar outputs

Capsule Network

- Spatial relationship between simple and complex objects
- Instead of using scalar outputs, it uses vector outputs (capsules)
  - Length represents the probability that an object exists
  - Orientation represents instantiation parameters of an object, as for example position, size, rotation, etc

\[
W_{ij} = [8 \times 16]
\]

Capsule Network

- Reconstruction

<table>
<thead>
<tr>
<th>Scale and thickness</th>
</tr>
</thead>
<tbody>
<tr>
<td>False positive</td>
</tr>
<tr>
<td>True positive</td>
</tr>
<tr>
<td>False negative</td>
</tr>
<tr>
<td>True negative</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Localized part</th>
</tr>
</thead>
<tbody>
<tr>
<td>False positive</td>
</tr>
<tr>
<td>True positive</td>
</tr>
<tr>
<td>False negative</td>
</tr>
<tr>
<td>True negative</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stroke thickness</th>
</tr>
</thead>
<tbody>
<tr>
<td>False positive</td>
</tr>
<tr>
<td>True positive</td>
</tr>
<tr>
<td>False negative</td>
</tr>
<tr>
<td>True negative</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Localized skew</th>
</tr>
</thead>
<tbody>
<tr>
<td>False positive</td>
</tr>
<tr>
<td>True positive</td>
</tr>
<tr>
<td>False negative</td>
</tr>
<tr>
<td>True negative</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Width and translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>False positive</td>
</tr>
<tr>
<td>True positive</td>
</tr>
<tr>
<td>False negative</td>
</tr>
<tr>
<td>True negative</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Localized part</th>
</tr>
</thead>
<tbody>
<tr>
<td>False positive</td>
</tr>
<tr>
<td>True positive</td>
</tr>
<tr>
<td>False negative</td>
</tr>
<tr>
<td>True negative</td>
</tr>
</tbody>
</table>

ResNet50 and Feature Pyramids Network (FPN)

- Residual Network avoid gradient diffusion in deep NN using shortcut connections. Accuracy will not decrease if more layers are added.
- Feature Pyramids Networks allows predictions at different scales (good for small objects). Works with ResNets.


Outline

1. Introduction
2. Related Work
3. Design and Implementation
   - Region Based Object Detectors
   - Step1: Region Proposal Generation
   - Step 2: Proposals Classification
4. Experiments
5. Results
6. Conclusions and Future Work
Region Based Object Detectors

Input image → Region proposals generation → Box proposals (ROIs) → Classification → Predictions
Implementation

Input image -> Region Proposal Network -> Box proposals (ROIs) -> Phase 1

Phase 1:
- Region Proposal Network
- Box proposals

Phase 2:
- ResNet50 + FPN (From Fast RCNN) -> Predictions
- Simple Convolutional Neural Network
- Capsule Network
- Box proposals information plus classification scores

Detectron Pipeline

Region Based Object Detectors for Recognizing Birds in Aerial Images
Step 1: Region Proposals Pipeline

- Sliding window (relative position) + anchors
- ResNet50 + FPN (Backbone architecture)
- Feature map
- Conv. layer
- Fully connected layers
- Coordinate and score for anchors

Region Proposal Network
Step 1: Region Proposals Samples

- Around 800 region proposals per image
- Contains positive and negative samples that will be used for the training the classification methods.
Mini - batches

- No need to use all region proposals. Choose random boxes
- Positive and negative examples according to its Intersection over Union (IoU) with the ground truths

<table>
<thead>
<tr>
<th>Random samples per mini-batch</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoU for positive sample</td>
<td>0.5 or more</td>
</tr>
<tr>
<td>Positive / Negative balance</td>
<td>50:50</td>
</tr>
<tr>
<td>Augmentation</td>
<td>50% prob. flip image</td>
</tr>
</tbody>
</table>

\[
\text{Overlap area} = \frac{\text{Area of union}}{\text{Overlap area}}
\]

\[
\text{IoU} = \frac{\text{Overlap area}}{\text{Area of union}}
\]
Step 2: Classification - ResNet50 + FPN (From Fast R-CNN)
Step 2: Classification - Simple Convolutional Neural Network

```
Box proposals (ROIs)

Input 30x30x3
Conv1 6@24x24
   Kernel size: 7x7
   Stride: 1
MaxPool1 6@12x12
   Kernel size: 2x2
   Stride: 2
Conv2 16@6x6
   Kernel size: 7x7
   Stride: 1
MaxPool2 16@3x3
   Kernel size: 2x2
   Stride: 2
FullyConn1 Nodes: 72
FullyConn2 Nodes: 36
FullyConn3 Nodes: 2
```
Step 2: Classification - Capsule Network
Outline

1. Introduction
2. Related Work
3. Design and Implementation
4. Experiments
   - Step 1: Region Proposal Network Evaluation
   - Step 2: Classification Evaluation
   - Results in the validation dataset
5. Results
6. Conclusions and Future Work
Step 1: Region Proposal Network Evaluation

- Region proposals should contain positive and negative samples
- Region proposals should overlap the maximum number of ground truths.

\[
\text{Recall} = \frac{tp}{tp + fn}
\]

Metric: Average Recall

1. Consider true positive if IoU is higher than threshold
2. Calculate Recall at each IoU threshold
3. Calculate the average recall
Step 1: RPN - Model Training

- 30 by 30 labels obtained better average recall scores
- 30 by 30 labels contains more birds.

<table>
<thead>
<tr>
<th>Dataset Label Size</th>
<th>Dataset Difficulty</th>
<th>Dataset Type</th>
<th>Percentage of birds in boxes</th>
<th>Average recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 x 20</td>
<td>Easy</td>
<td>Validation</td>
<td>0.939</td>
<td>0.539</td>
</tr>
<tr>
<td>20 x 20</td>
<td>Hard</td>
<td>Validation</td>
<td>0.938</td>
<td>0.543</td>
</tr>
<tr>
<td>30 x 30</td>
<td>Easy</td>
<td>Validation</td>
<td>0.978</td>
<td>0.651</td>
</tr>
<tr>
<td>30 x 30</td>
<td>Hard</td>
<td>Validation</td>
<td>0.974</td>
<td>0.630</td>
</tr>
</tbody>
</table>
Step 2: Classification Model Training

- Select model (weights) with the lowest loss during training
- As in RPN training, 30 by 30 labels and easy datasets have better performance

<table>
<thead>
<tr>
<th>Dataset Label Size</th>
<th>Dataset Difficulty</th>
<th>Dataset Type</th>
<th>Classification method</th>
<th>Minimum Validation Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 x 20</td>
<td>Easy</td>
<td>Validation</td>
<td>Simple CNN</td>
<td>0.4830</td>
</tr>
<tr>
<td>20 x 20</td>
<td>Hard</td>
<td>Validation</td>
<td>Simple CNN</td>
<td>0.4535</td>
</tr>
<tr>
<td>30 x 30</td>
<td>Easy</td>
<td>Validation</td>
<td>Simple CNN</td>
<td><strong>0.4323</strong></td>
</tr>
<tr>
<td>30 x 30</td>
<td>Hard</td>
<td>Validation</td>
<td>Simple CNN</td>
<td><strong>0.3923</strong></td>
</tr>
</tbody>
</table>
Step 2: Classification Evaluation

- After making predictions each region proposal will have a “bird score” between 0 and 1
- Remove region proposals overlapped and with a low score
- Precision, Recall and F1 metrics to measure performance

\[
\text{Precision} = \frac{tp}{tp + tn} \quad \text{Recall} = \frac{tp}{tp + fn} \quad F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]
Remove Overlapped Boxes

- Non Maximum Suppression after classification. Keep boxes with higher scores.
  - For this dataset, birds are not highly overlapped.
  - F1 increases if non-maximum suppression removes all overlapped boxes.

RP1: Score: 0.91
RP2: Score: 0.55
Remove Boxes with low score

- For each classifier, find a threshold that maximizes the F1 score in the validation dataset.

- Higher threshold means high Recall (tends to recognize backgrounds as birds).

![Graphs showing F1 score vs. minimum score threshold for different models.](image-url)
Capsule Networks - Reconstruction

- Reconstruct the input image from the output capsule vector using a reconstruction network.
- This network did not produce a significant improvement.
- Difficulty to recognize the shape due to the high variety of bird shapes.

Training samples

Bird images

<table>
<thead>
<tr>
<th>Image 1</th>
<th>Image 2</th>
<th>Image 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reconstructions

<table>
<thead>
<tr>
<th>Reconstruction 1</th>
<th>Reconstruction 2</th>
<th>Reconstruction 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Results in Validation Dataset

- 30 by 30 labels datasets obtained better scores.
- Fast RCNN achieved better scores.

<table>
<thead>
<tr>
<th>Classification method</th>
<th>Validation Precision</th>
<th>Validation Recall</th>
<th>Validation F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast RCNN</td>
<td>0.7799</td>
<td>0.7364</td>
<td>0.7575</td>
</tr>
<tr>
<td>Simple CNN</td>
<td>0.7207</td>
<td>0.6422</td>
<td>0.6792</td>
</tr>
<tr>
<td>CapsNet</td>
<td>0.6885</td>
<td>0.6984</td>
<td>0.6934</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classification method</th>
<th>Validation Precision</th>
<th>Validation Recall</th>
<th>Validation F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast RCNN</td>
<td>0.6461</td>
<td>0.6276</td>
<td>0.6367</td>
</tr>
<tr>
<td>Simple CNN</td>
<td>0.5594</td>
<td>0.6502</td>
<td>0.6014</td>
</tr>
<tr>
<td>CapsNet</td>
<td>0.4590</td>
<td>0.7565</td>
<td>0.5714</td>
</tr>
</tbody>
</table>
Outline

1. Introduction
2. Related Work
3. Design and Implementation
4. Experiments
5. Results
   - Results on test dataset
   - Visualizations
6. Conclusions and Future Work
Results - Easy Test Dataset

- Fast RCNN classification achieved better results
- Using validation threshold helped to improve F1 test score.
- Simple CNN and Capsule Networks tends to have High Recall

<table>
<thead>
<tr>
<th>Classification method</th>
<th>Validation Precision</th>
<th>Validation Recall</th>
<th>Validation F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast RCNN</td>
<td>0.9249</td>
<td>0.8809</td>
<td>0.9024</td>
</tr>
<tr>
<td>Simple CNN</td>
<td>0.9497</td>
<td>0.7335</td>
<td>0.8277</td>
</tr>
<tr>
<td>CapsNet</td>
<td>0.9315</td>
<td>0.7964</td>
<td>0.8587</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classification method</th>
<th>Validation Precision</th>
<th>Validation Recall</th>
<th>Validation F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast RCNN</td>
<td>0.6804</td>
<td>0.6492</td>
<td>0.6644</td>
</tr>
<tr>
<td>Simple CNN</td>
<td>0.4412</td>
<td>0.6033</td>
<td>0.5097</td>
</tr>
<tr>
<td>CapsNet</td>
<td>0.3631</td>
<td>0.6000</td>
<td>0.4524</td>
</tr>
</tbody>
</table>

Easy Dataset - 30x30 labels

Hard Dataset - 30x30 labels
Visualizations - Easy Image 1

Easy image 1 - Fast RCNN classification

Easy image 1 - Simple CNN classification

Easy Image 1 - CapsNet classification
Visualizations - Easy Image 2

Easy image 2 - Fast RCNN classification

Easy image 2 - Simple CNN classification

Easy image 2 - CapsNet classification
Visualizations - Hard Image 1

Hard image 1 - Fast RCNN classification

Hard image 1 - Simple CNN classification

Hard image 1 - CapsNet classification
Visualizations - Hard Image 2

Hard image 2 - Fast RCNN classification  Hard image 2 - Simple CNN classification  Hard image 2 - CapsNet classification
Outline

1. Introduction
2. Related Work
3. Design and Implementation
4. Experiments
5. Results
6. Conclusions and Future Work
   - Conclusions
   - Future Work
Conclusions

- 30 by 30 labels achieved better scores. This bounding box size captures more information of the birds.
- RPN does work for finding the regions of interest, including most of the birds.
- With the proposed pipeline the box proposals are available for any other classifier.
- All overlapped region proposals must be removed for this dataset.
- Classifiers tend to have a high recall. All score metrics increase when increasing the minimum score threshold.
Conclusions

- Fast RCNN classification achieved the best F1 score in all the datasets. Simple CNN and Capsule Networks achieved similar scores.
- Fast RCNN are the results of applying many advanced techniques like Feature Pyramid Network.
- The high variety of birds shapes made capsule network unable to learn instantiation parameters features.
- CapsNet should work better with more classes and no backgrounds.
Future work

- Accurate bounding boxes.
- Add more labels. For example, for flying birds.
- Matrix capsules with EM routing: instead of using vectors to represent the instantiation parameters of an object, use matrices that allows a more complex representation of an object.
Thank You

Questions