Deep Learning Object Detection

for Counting Fish

…

Jamie Shaffer
July 2020

Agenda

- Background
- Challenge
- Concept
  - Data
  - Model
  - Evaluation
- Conclusion

Acknowledgements and information sources are listed at the end of the presentation.

Background

Pacific Northwest Salmon

Salmon hatch in freshwater streams

Migrate to the saltwater ocean

Return upstream to their natal streams

Spawn and die

Salmon hatch in freshwater streams

Migrate to the saltwater ocean

Background

Pacific Northwest Hydroelectric Power

Columbia & Snake River Dams

Background

Power *and* Salmon

Built fish ladders

Adjust dam outflow

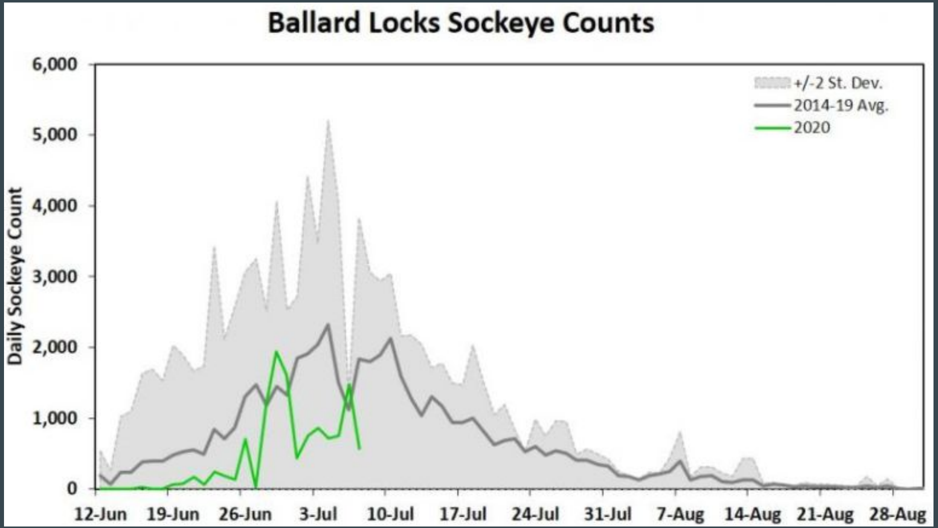
Fishing guidelines (daily limits, season duration)

Monitor fish migrating upstream through the ladders

Challenge

Accurate & Timely Fish Counts

1. Estimate from a daily sample (Ballard Locks)
2. Real-time count (Bonneville Dam)
3. Video count (Rock Island Dam)



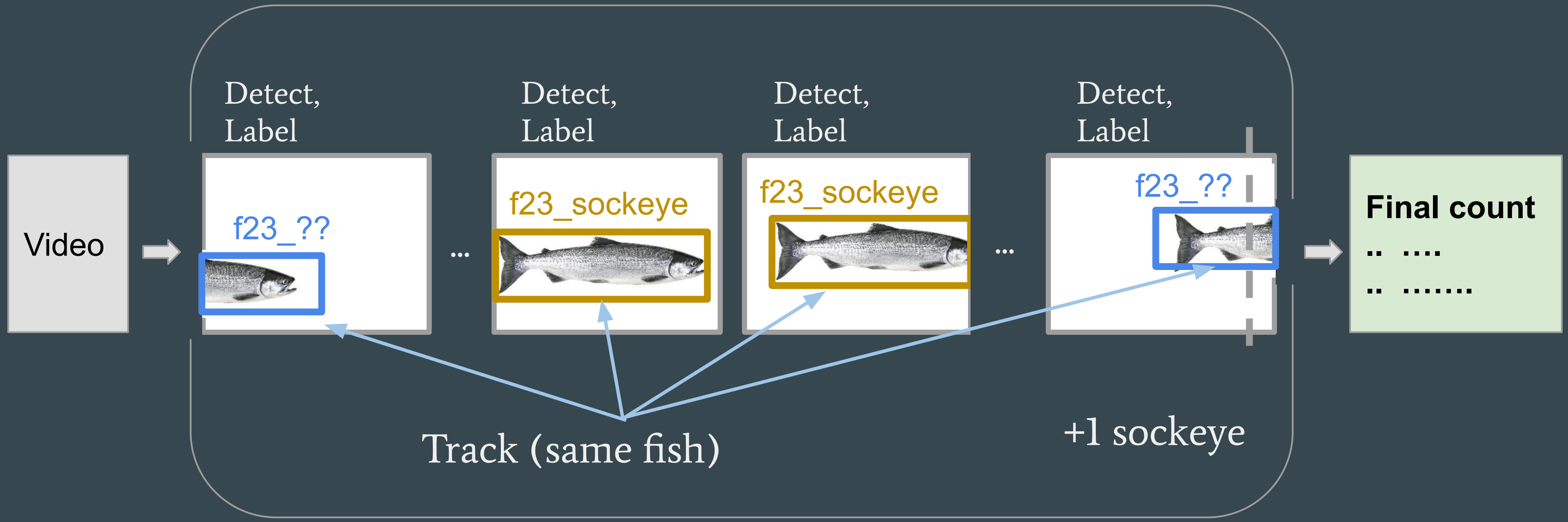
Challenges:

- Sampling error
- Human expertise
- Fish counts affect
  - Policy decisions
  - Recreational fishing
  - Commerce

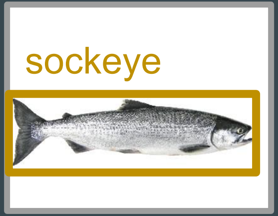
Is machine learning for object detection ready to assist in this challenge?

Concept

Detect → Label → Track → Record Count



Concept



Collect & Prepare Data (Images)

Select & Train Model

Test & Evaluate

Data

Public Viewing Windows

- Lighting
- Depth of tank
- Obstructions

Detect:

- Fish or no fish (count)
- Fish details (count species)

Web scraped images

Fish Count Window

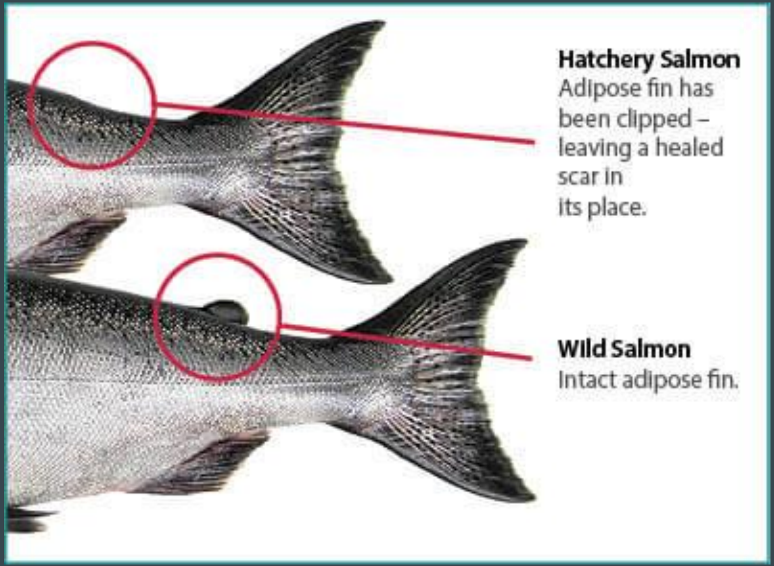
Snapshots from video

Data

Label & Prepare Images

3 classes:

- adipose
- no_adipose
- unknown (out of view)



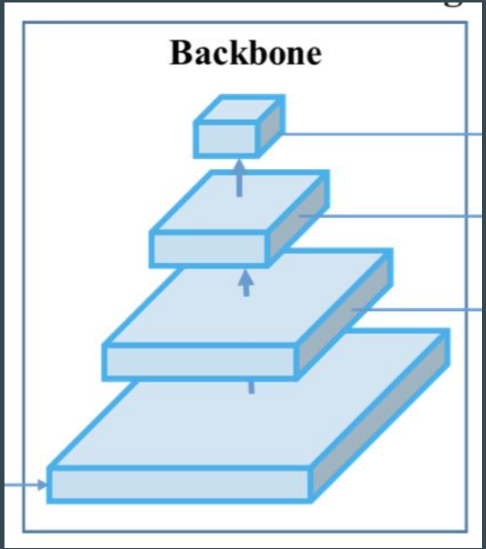
Final:

- **317** snapshots from video
- 1025 annotations (~3.2 per image)
  - Bounding box + Label
- 15 null examples (no fish)
- Augmented by flip, blur, brightness (roboflow.ai) → **951 images**

Model

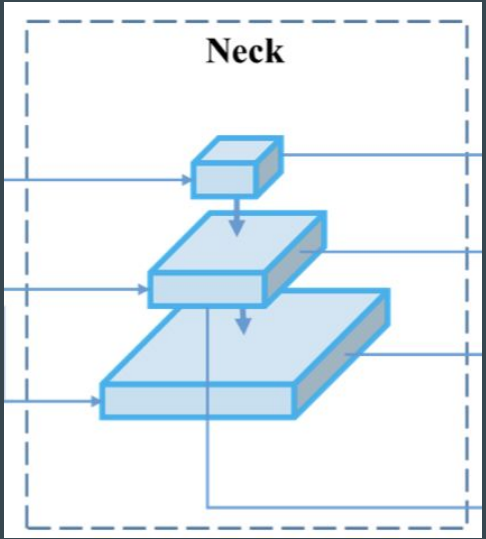
Classification + Object Detection

Image

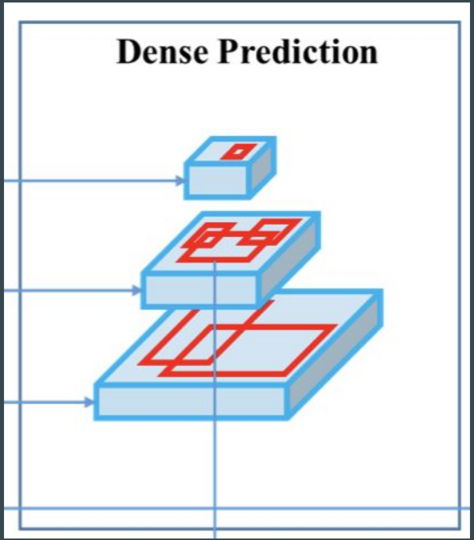


Classification cNN to extract features

- VGG16
- Darknet



- Collect feature maps from different stages
- More ‘pieces’ for the final layer to use

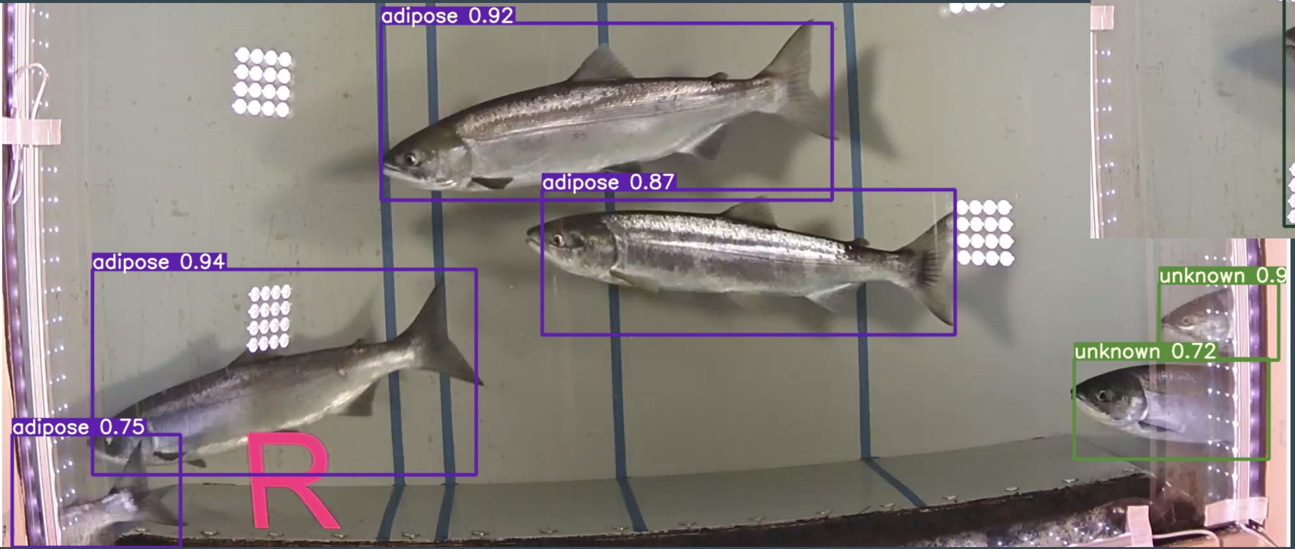


- One shot final step:
- Predict class
- Predict bounding box

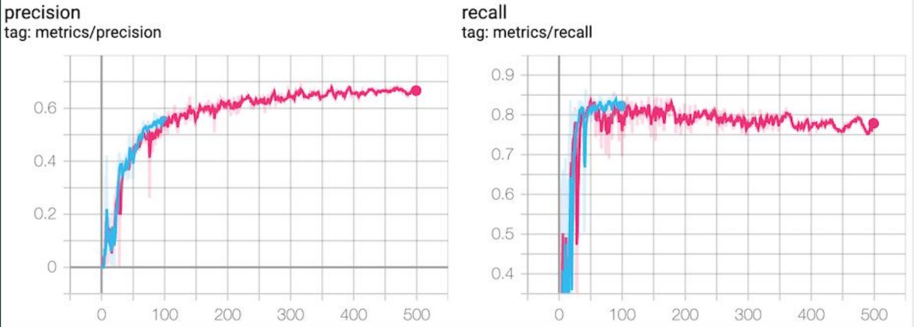
Model selected: **YOLOv5**

Evaluation


adipose 0.96
no_adipose 0.86

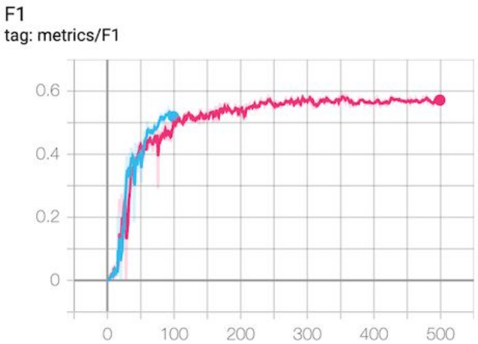

adipose 0.92
adipose 0.87
adipose 0.94
adipose 0.75
R
unknown 0.9
unknown 0.72

Evaluation



Precision: "If you say it's a fish, what percentage of the time is it really a fish?" ~70%

Recall: “If there’s a fish in there, what percentage of the time do you find it?” ~80%



F1 score: ~60%

2*Precision*Recall / (Precision + Recall)

Evaluation

mAP@0.5

mAP

Average Precision for 1 class:

- Calculate the BEST Precision at each level of Recall, then AVERAGE these

$$\frac{\#TP(c)}{\#TP(c)+\#FP(c)}$$

mean Average Precision for 3 classes:

- Find the Average Precision for each class, take the mean of those

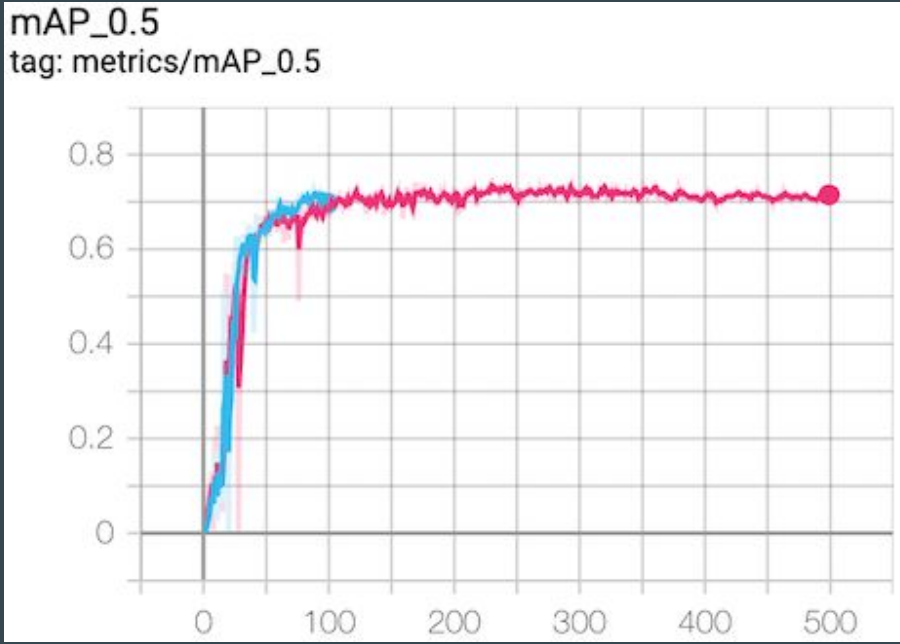
@0.5

IoU = intersection over union

Common definition of “@0.5”:

IoU > 0.5, it’s a valid detection.

Evaluation



mAP@0.5 ~ 70%

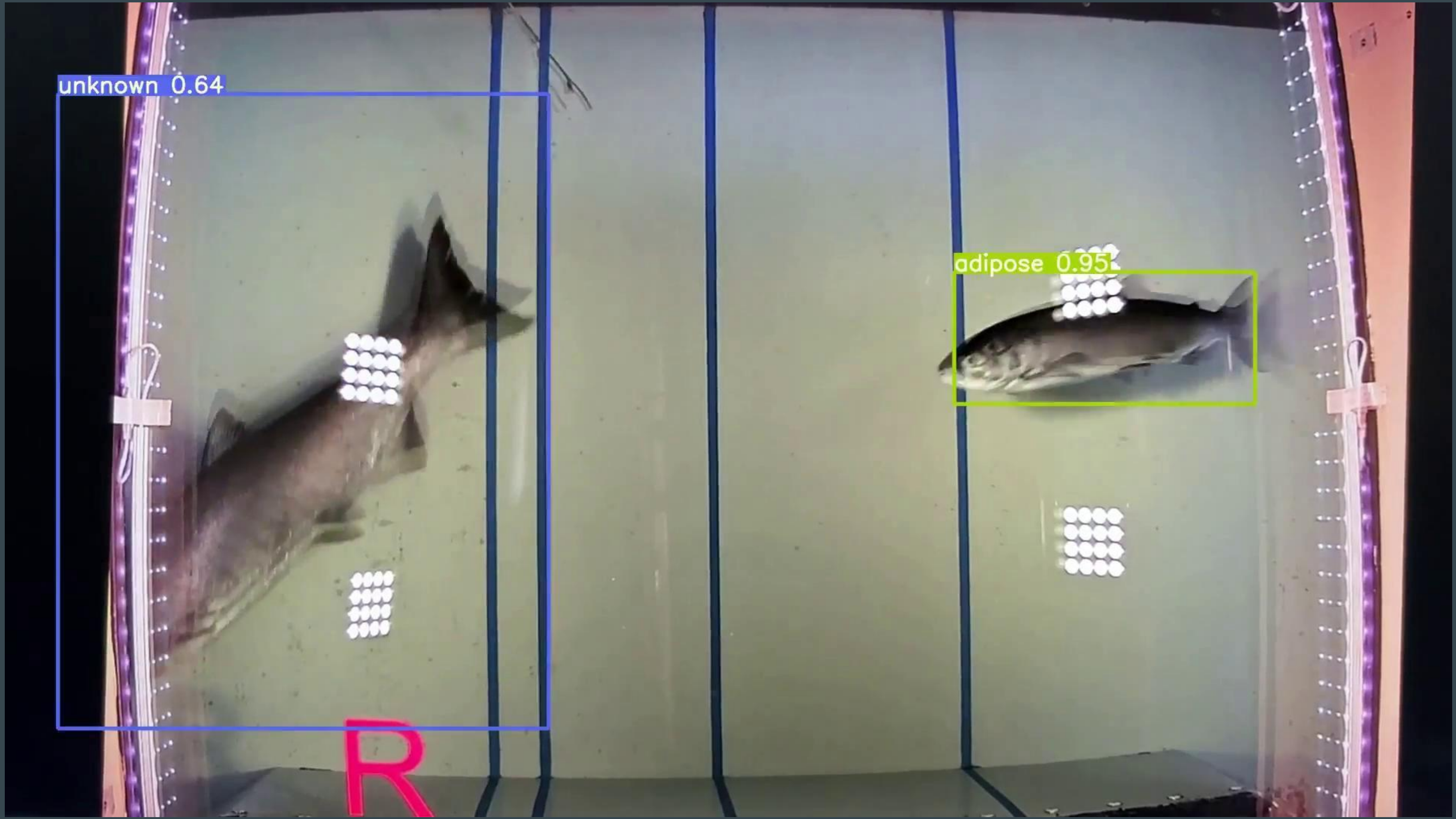
- Average precision (correct label)
- over all classes,
- if we set the valid detection threshold to 50% bounding box overlap

Conclusion:

The model presented today is not ready for production. It needs:

- Expert labeled images (more classes, fewer errors)
- More images (trained only on images from sockeye run, no chinook or coho)
- Object counting (e.g. implement the SORT method)
- Handling of difficult cases (salmon exit the frame and return, crowded conditions)

Technology and expertise exist for all of these, indicating that modern machine learning object detection methods -- e.g. Deep Learning networks like YOLO -- may soon be ready to assist in this challenge.



Questions?

Acknowledgements

Special thank you to the staff at the **Chelan County PUD** for providing information on salmon counting methods and for a 1-hour video clip.

D. Patterson, T. Mosey, T. West

https://www.chelanpud.org/environment/fish-and-wildlife/fish-counts

Team Roboflow.ai

J. Nelson

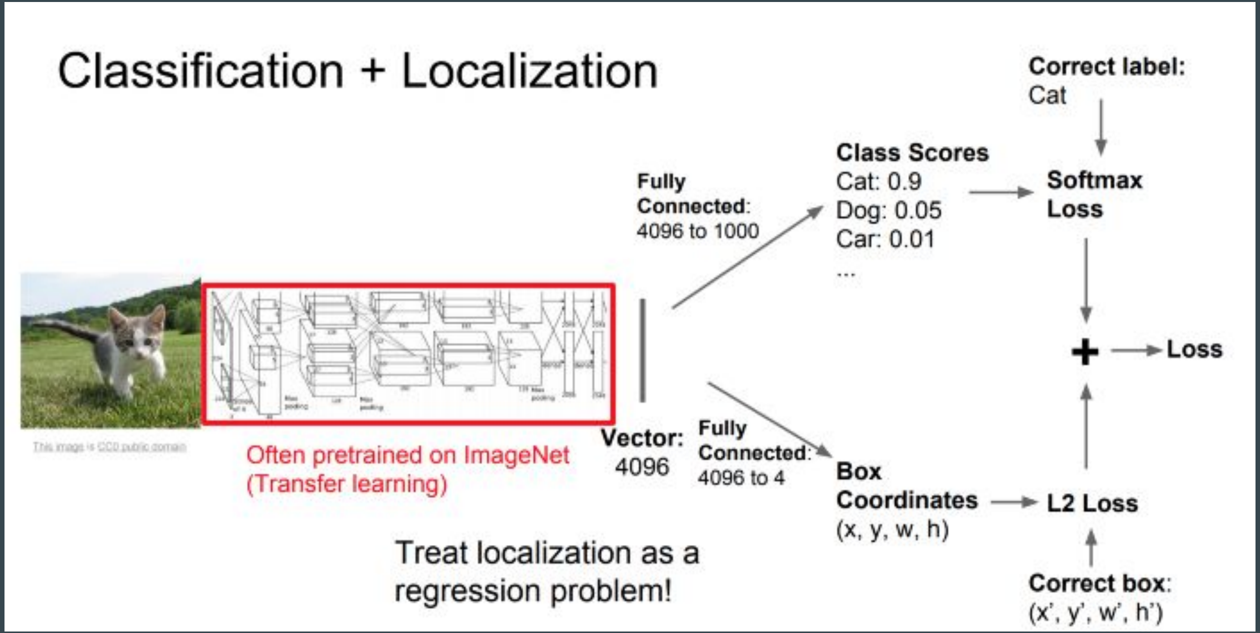
Image augmentation tools & template notebooks for YOLO models

Sources

- https://pixabay.com/photos/animal-river-water-stone-fish-4623023/
- https://pixabay.com/photos/salmon-fish-run-jump-upstream-1107404/
- https://pixabay.com/photos/fisherman-salmon-wading-release-937027/
- https://pixabay.com/photos/sockeye-salmon-run-adams-river-50303/
- https://pixabay.com/photos/natural-landscape-river-water-fish-4620642/
- https://pixabay.com/photos/animal-river-water-4623019/
- https://pixabay.com/photos/fish-salmon-chinook-bay-landscapes-386853/
- https://static.seattletimes.com/wp-content/uploads/2018/12/Columbia-and-Snake-rivers-dams-W-780x520.jpg
- https://www.ncwlife.com/wp-content/uploads/2018/03/IMG-Rock-Island-Dam-1920x1
- http://www.eregulations.com/wp-content/uploads/2020/06/191.jpg
- https://static.seattletimes.com/wp-content/uploads/2016/10/b224c62e-9403-11e6-b59c-0a10cc50aad4-375x349.jpg
-
- http://mediad.publicbroadcasting.net/p/northwestnews/files/styles/medium/public/201403/fish_ladder.jpg
-
- https://thefisheriesblog.files.wordpress.com/2013/05/de1de-two_fins2.jpg
-
- https://static.seattletimes.com/wp-content/uploads/2018/10/Lower-Snake-River-Dams
- http://www.eregulations.com/wp-content/uploads/2020/06/Marine_-_Sockeye_Ocean_1.3_-_brightness.jpg
- CNN diagram is from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
- Architecture figures are from the YOLOv4 paper
- https://arxiv.org/pdf/2004.10934.pdf
-
- Kitten classification + detection drawing from
- https://towardsdatascience.com/object-detection-using-deep-learning-approaches-an-end-to-end-theoretical-perspective-4ca27eee8a9a

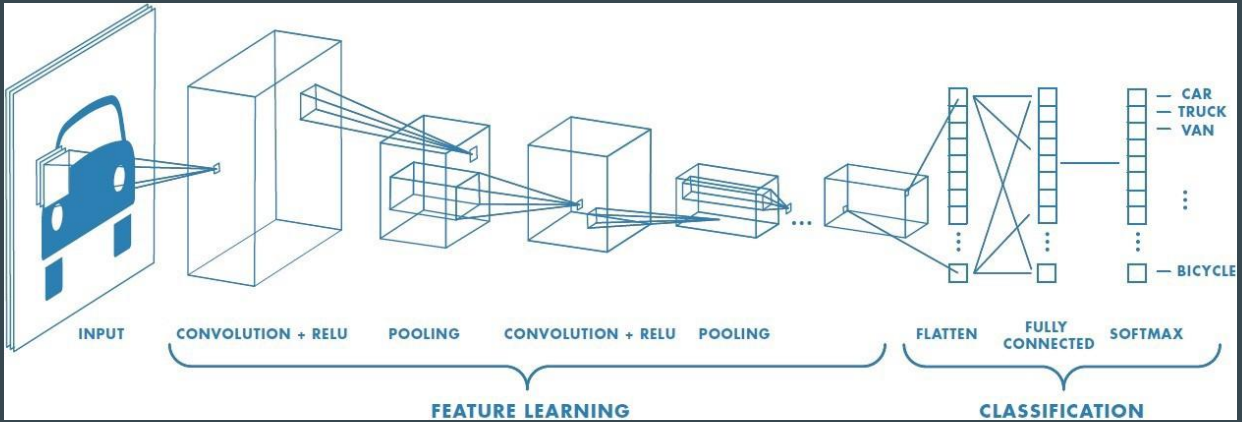
Addendum

Model: Classification + Localization



Model: Step 1: Object Classification

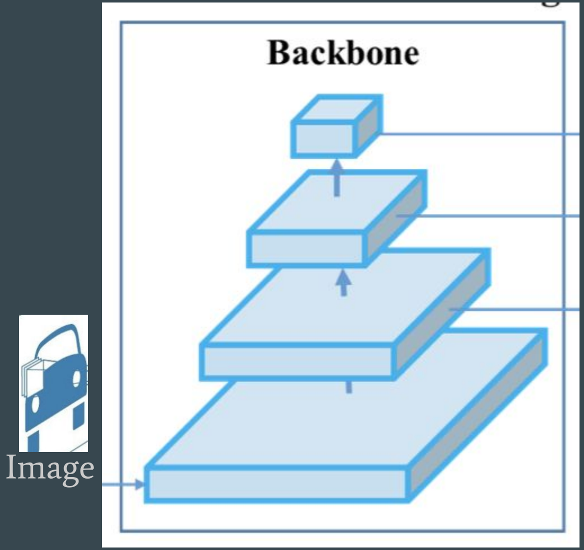
Convolutional Neural Networks (CNN), also called "Deep Learning"



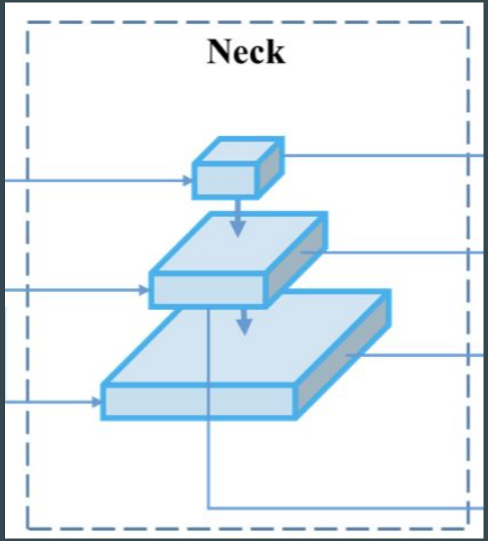
- Break down an image into pieces
- Predict what it is based on pieces

'Guitar and Violin', Picasso, c. 1912 →

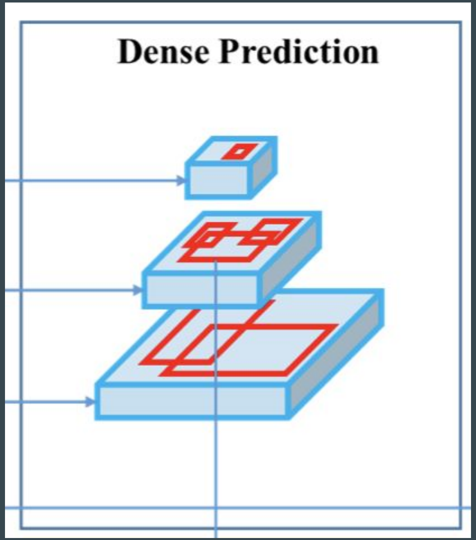
Model: Step 2: Object Detection



- Classification CNN:
- VGG16
- Darknet



- Collect feature maps from different stages
- More ‘pieces’ for the final layer to use



- One shot final step:
- Predict class
- Predict bounding box

Evaluation: Training and Inference

1 hour 12 mins (fast!) - Training time for 500 sessions with all 951 images (epochs)

13 MB (small!) - Size of model weight file

12 msec (fast!) - Time required for inference (object detection) on a simple image

This translates to a nearly 1 - 1 ratio for video processing -- a 1 minute video can be run through the model in about 1 minute.

*Times are from Google Colab with GPU enabled.

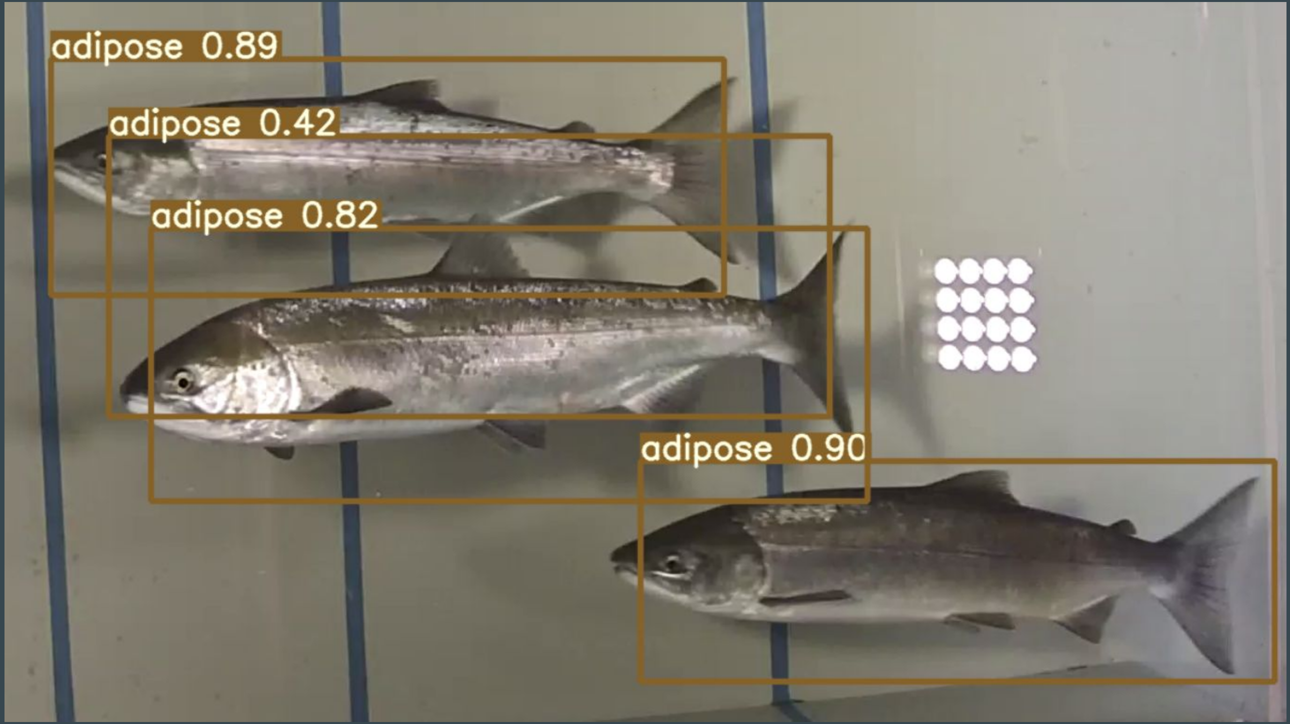
Gallery of Difficult Images

Difficult images are great for providing to clues to poor predictions!

Difficult Images

Shadows
Object overlap
Crowded conditions

-- there's an extra box in this image @0.42



Difficult Images

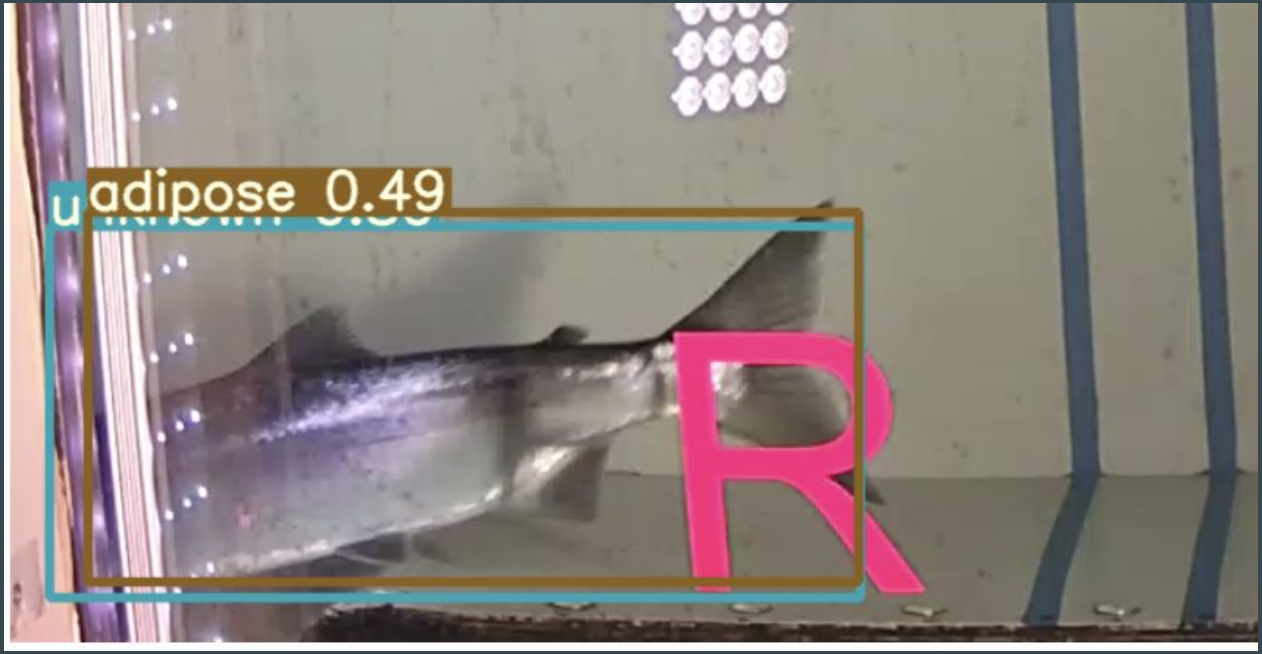
Crowded conditions

2 or 3 fish were not found

Difficult Images

0.4 threshold

Model may be linking large fish with no_adipose



Difficult Images

Obstructed view

Phantom floor fish -- reminiscent of the Viola-Jones algorithm, suggesting that the model may be using lighting rather than other features