machine learning conference

Deep Learning Solution for the Amazon Robotics Machine Learning Challenge 2017

Speaker: Loris Bazzani

Joint work with Maksim Lapin, Sabine Sternig, Matthieu Guillaumin, Michael Donoser

computer vision berlin

bazzanil@ maksiml@ ssternig@ matthieg@ donoserm@

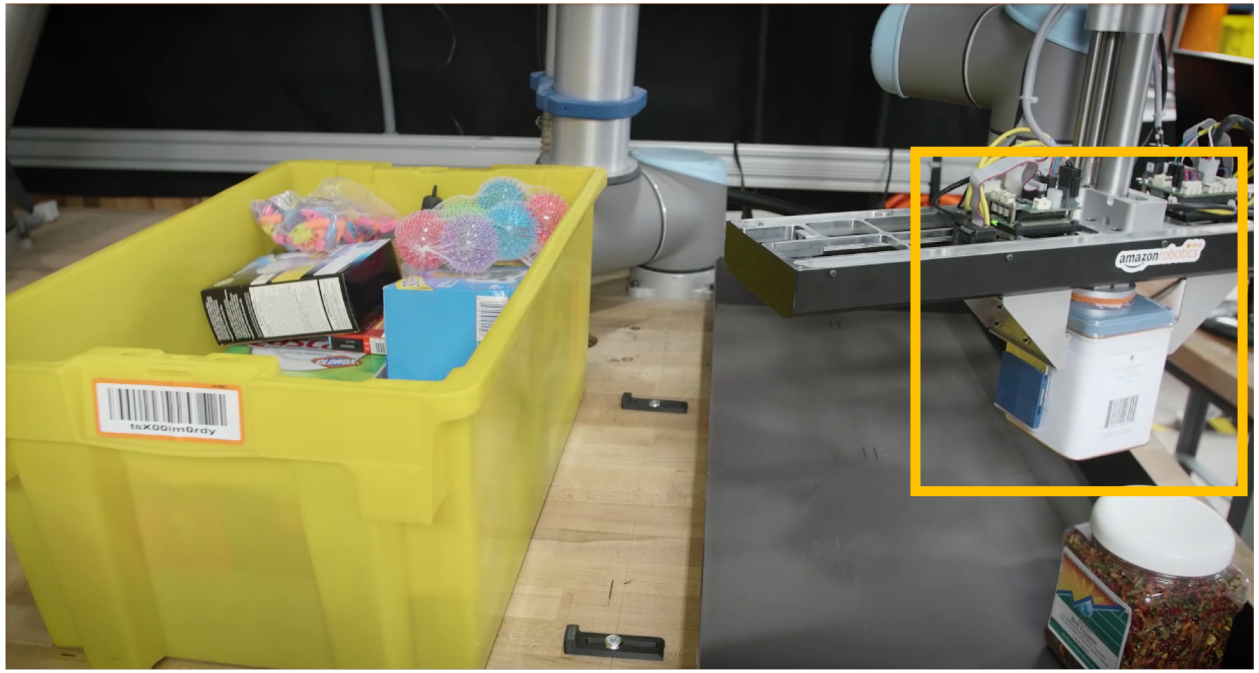
AR MLCHALLENGE WINNERS! 2017

Apr. 27, 2018

Amazon Robotics Machine Learning Challenge (AR-MLC)

- Challenge organized by Amazon Robotics
- Why?
  - Strengthen ML **community** @ Amazon
  - Encourage **sharing** cutting-edge techniques
  - Guide the solution of **real business** problems through the use of ML/CV

Business Value



Amazon Robotics Machine Learning Challenge (AR-MLC)

- **Task**: ASIN identification with manifest

Image of a single ASIN

Manifest:

B00JUAABLY
B071WLMK5Y
B003TTZWZ2
B06Y5GQTJ4
B0000BYC4B
B00000J0JF
...

10-20 candidate ASINs

Database of (Image, ASIN) pairs

34K ASINs

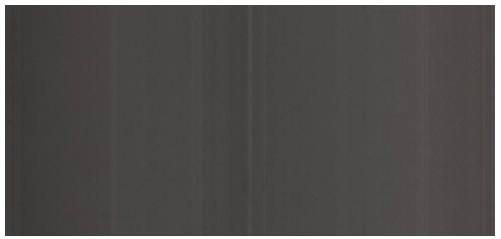
Is the Task Easy?

Mixed color/grey level



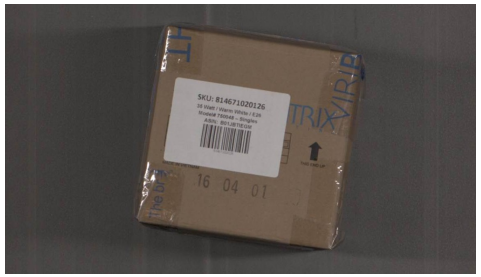

Varying illumination conditions


Bogus images

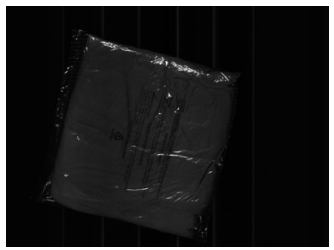
Others





Varying image quality





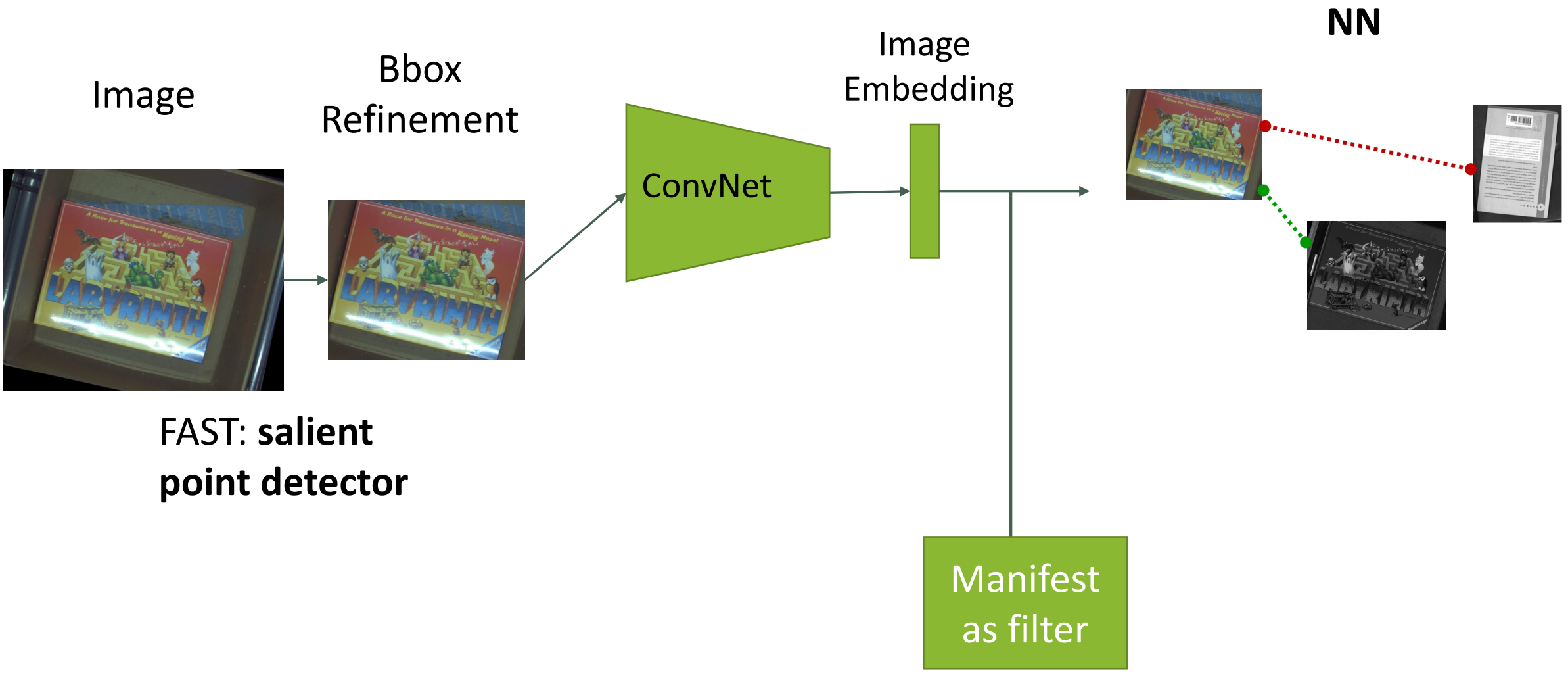

Different views




Outline of the Proposed Method

- Nearest Neighbor (NN) with the triplet loss
- ASIN recognition
- Ensemble: NN+ASIN recognition

NN with Triplet Loss



Triplet Loss Network

Anchor
A

ConvNet

$$L(A, B, C) = log(1 + exp(D(A, B) - D(A, C)))$$

Nam Vo, James Hays. ” Localizing and Orienting Street Views Using Overhead Imagery” ECCV 2016.

Triplet Loss Network

Positive Example B

Anchor A

ConvNet

Shared weights

ConvNet

$D(A, B)$

$$L(A, B, C) = log(1 + exp(D(A, B) - D(A, C)))$$

Nam Vo, James Hays. ” Localizing and Orienting Street Views Using Overhead Imagery” ECCV 2016.

Triplet Loss Network

Positive Example B

Anchor A

Negative Example C

ConvNet

Shared weights

ConvNet

Shared weights

ConvNet

$D(A, B)$

$D(A, C)$

$$L(A, B, C) = log(1 + exp(D(A, B) - D(A, C)))$$

Nam Vo, James Hays. ” Localizing and Orienting Street Views Using Overhead Imagery” ECCV 2016.

Triplet Loss Network

Positive Example B

Anchor A

Negative Example C

ConvNet

Shared weights

ConvNet

Shared weights

ConvNet

$D(A, B)$

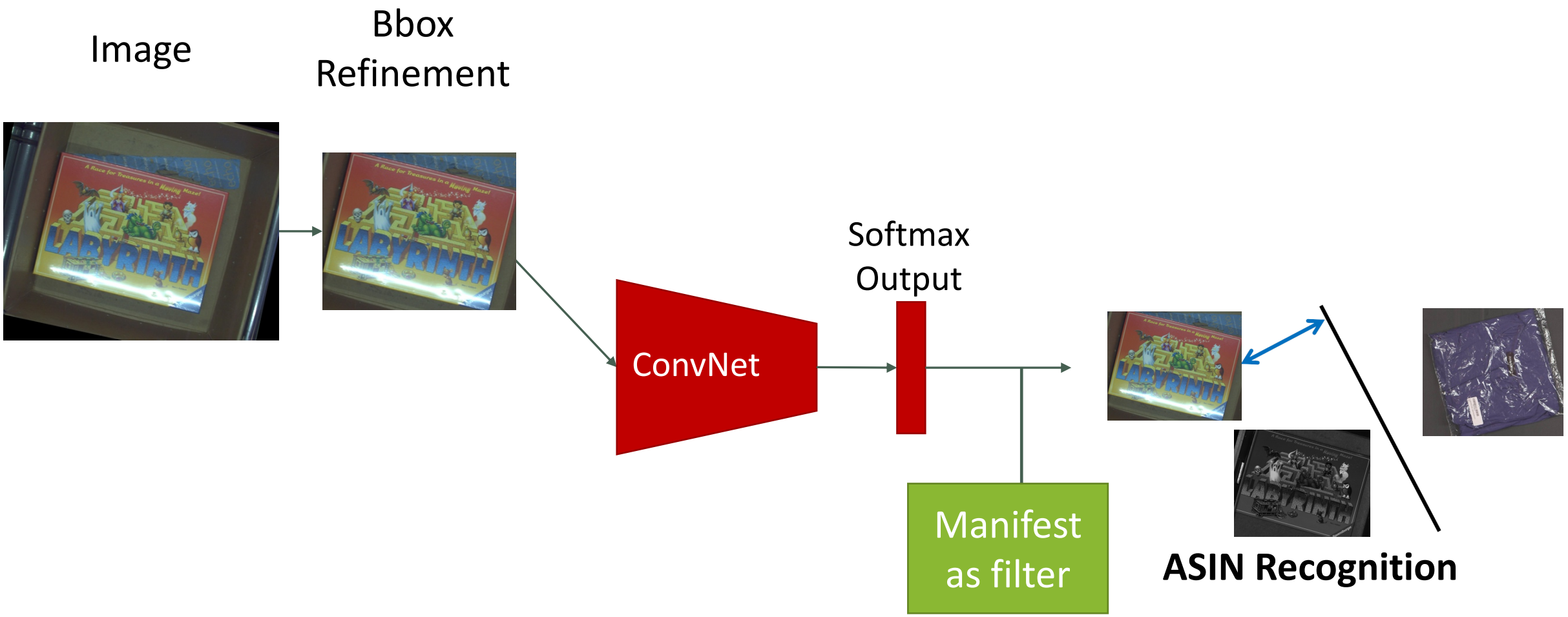
$D(A, C)$

Triplet Loss

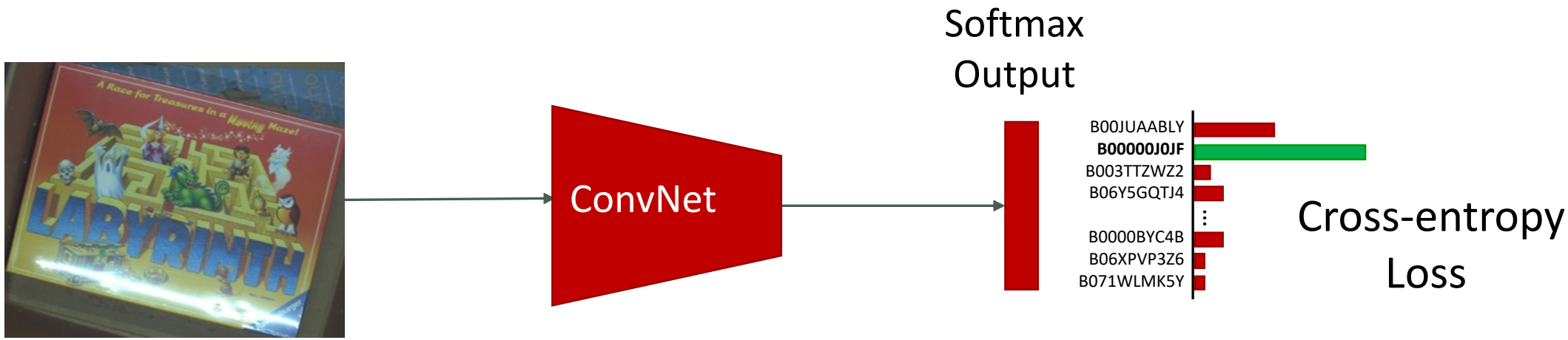
$$L(A, B, C) = log(1 + exp(D(A, B) - D(A, C)))$$

Nam Vo, James Hays. ” Localizing and Orienting Street Views Using Overhead Imagery” ECCV 2016.

ASIN Recognition

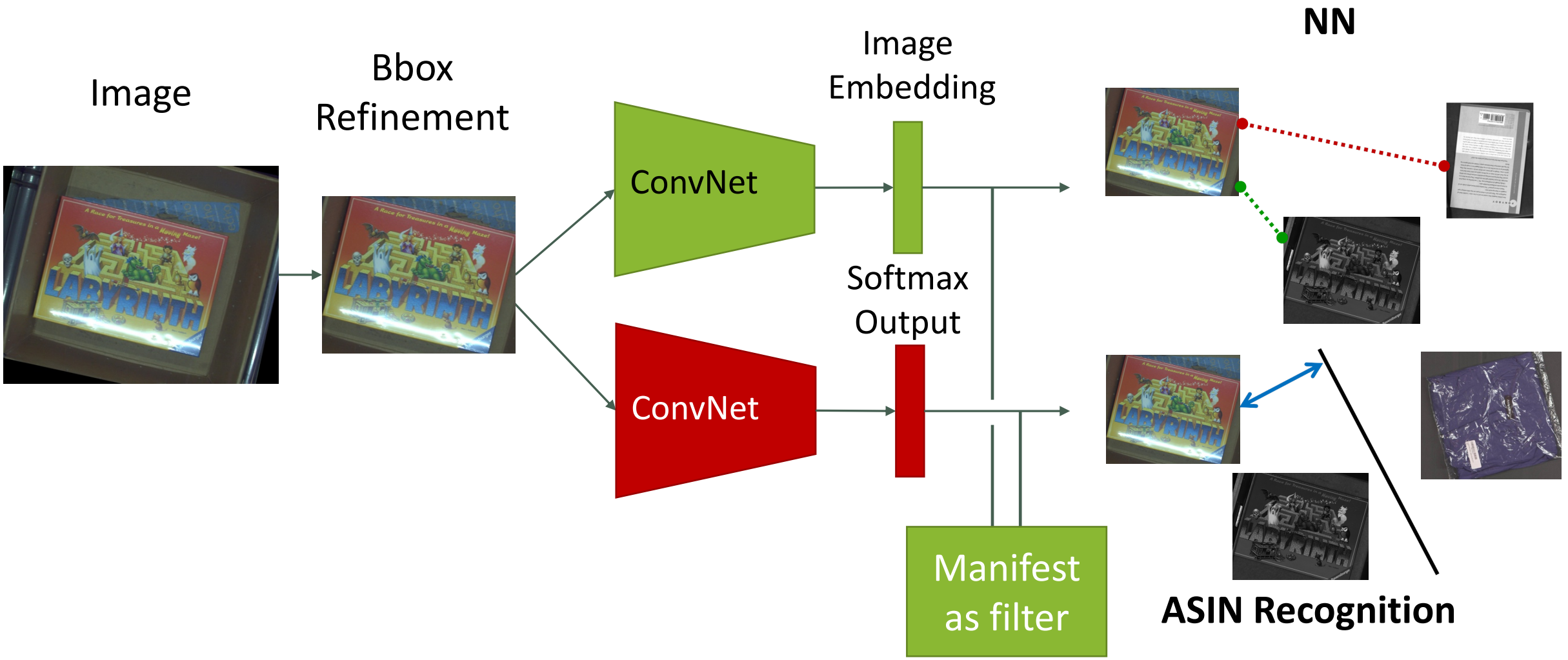


ASIN Recognition Network



- Training challenges @ AR-MLC:
  - 34K classes
  - 3-5 images for each class
- 28% validation accuracy

Ensemble: NN+ASIN Recognition



Dataset and Evaluation

- AR-MLC Phase 1
  - Training data: 133K
  - Number of ASINs: 34K
  - Test data: 38K
- Evaluation

$$\text{score} = \#\text{correct} - \frac{\#\text{wrong}}{4}$$

$$\text{norm score} = \frac{\text{score}}{\#\text{samples}}$$

- AR-MLC Phase 2
  - Training data: 172K (training + testing from phase 1)
  - Number of ASINs: 34K
  - Test data: 21K (new images)

AR-MLC Phase 1

Method	Norm Score	Score
NN Pre-trained	79.15	30344.25

AR-MLC Phase 1

Method	Norm Score	Score
NN Pre-trained	79.15	30344.25
+ BBox Refinement	81.07	31082.25

AR-MLC Phase 1

Method	Norm Score	Score
NN Pre-trained	79.15	30344.25
+ BBox Refinement	81.07	31082.25
NN Triplet Loss	86.12	33018

AR-MLC Phase 1

Method	Norm Score	Score
NN Pre-trained	79.15	30344.25
+ BBox Refinement	81.07	31082.25
NN Triplet Loss	86.12	33018
+ BBox Refinement	**90.31**	**34624.75**

AR-MLC Phase 2

Method	Norm Score	Score
NN Triplet loss + BBox Refinement	90.29	18662

AR-MLC Phase 2

Method	Norm Score	Score
NN Triplet loss + BBox Refinement	90.29	18662
ASIN recognition	94.26	19483.25

AR-MLC Phase 2

Method	Norm Score	Score
NN Triplet loss + BBox Refinement	90.29	18662
ASIN recognition	94.26	19483.25
Ensemble	**94.97**	**19630.5**

AR-MLC Final Leaderboard

Team	Prediction Time	Time per Image (s)	Norm Score	Score
Computer Vision Berlin (ours)	4h 41m 34s	0.82	**94.97**	**19630.5**
DLMafia	52h 39m 49s	9.17	92.07	19030.75
FashionRobo	4h 8m 34s	**0.72**	89.49	18497.0
SJC103	7h 26m 59s	1.30	89.12	18421.75
Alexa Intelligence	6h 58m 21s	1.21	86.79	17940.5
standard-line	86h 44m 43s	15.11	78.60	16245.75
Stack Solo	16h 14m 07s	3.83	74.45	15388.0
drama-queens	41h 05m 57s	7.16	74.41	15380.75
DeepFind	18h 44m 32s	3.26	73.35	15160.75
WolfPack	7h 44m 38s	1.35	69.78	14423.25

https://w.amazon.com/bin/view/AmazonRoboticsMachineLearning/ Docs/ARMLChallenge17/LeaderBoard/

Future Challenges

- Remove the manifest
- Multiple ASINs in the same tray (detection challenge)
- Scaling from 34K ASINs to 10M ASINs
- Integration of the solution with a robotic arm

Thank You!

maksiml@ ssternig@ bazzanil@ matthieg@ donoserm@

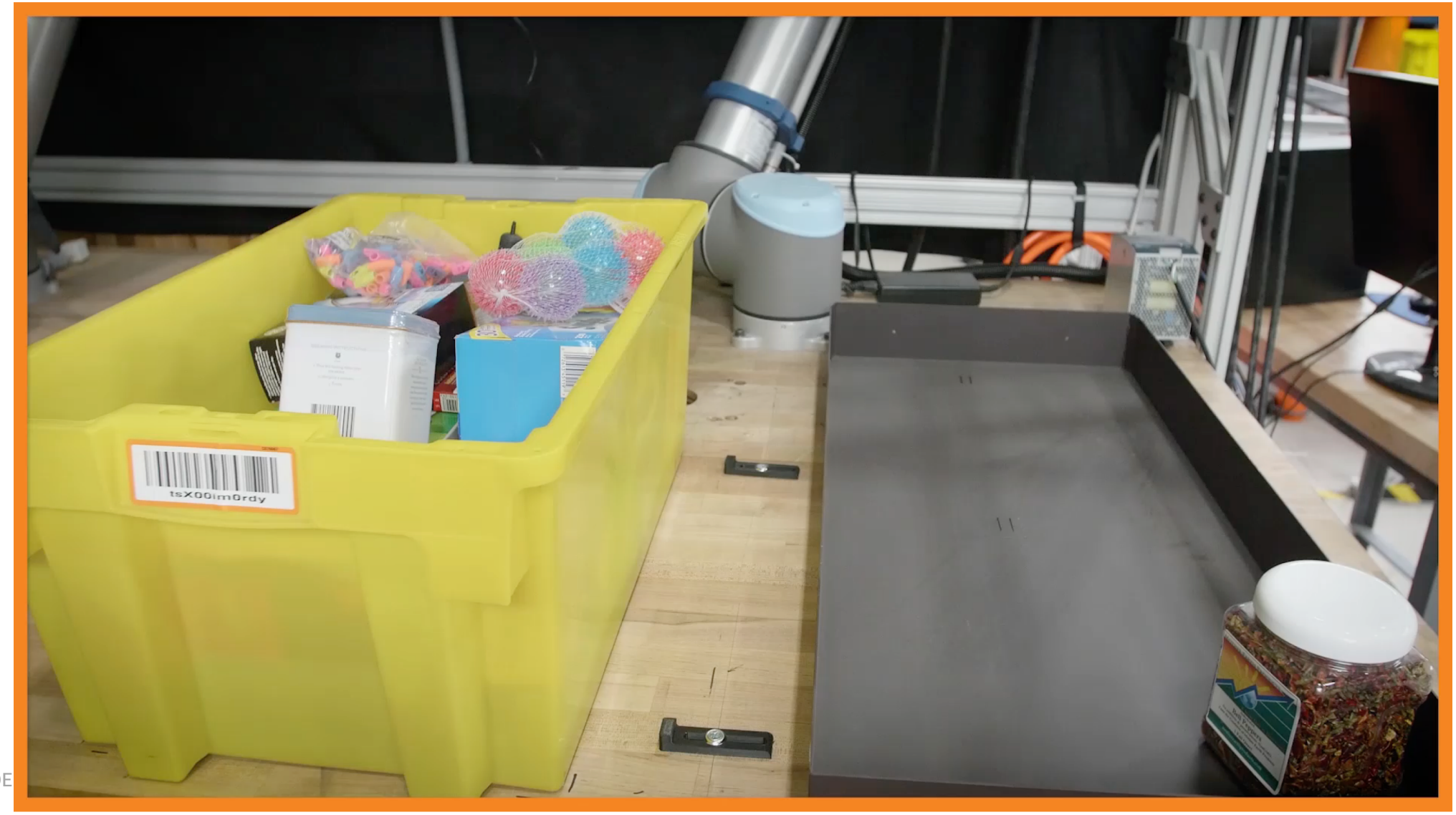
computer vision berlin

Questions?

https://w.amazon.com/bin/view/MLSciences/Berlin/Computer_Vision/Robotics_Challenge/

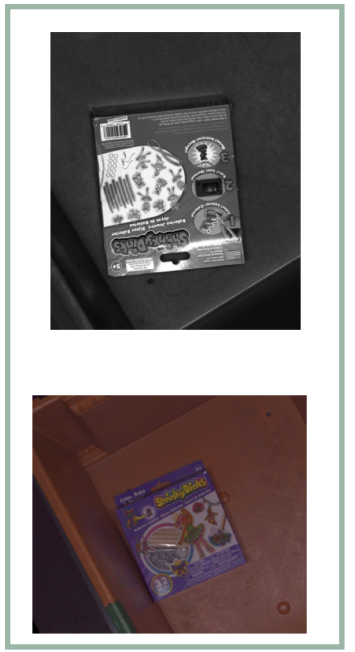
Extra Slides

Business Value



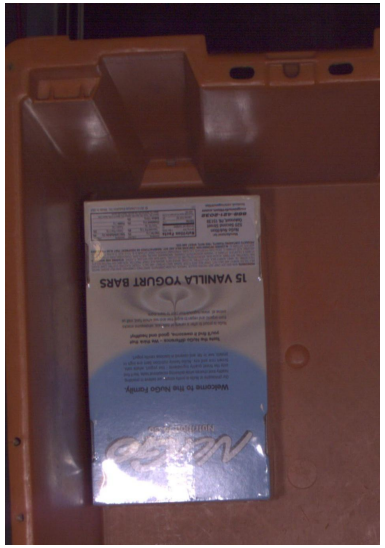
Challenges in the Data

- Mixed color/grey level
- **Varying illumination conditions**
- Different views
- Varying image quality
- Bogus images
- Other things/background



Challenges in the Data

- Mixed color/grey level
- **Varying illumination conditions**
- Different views
- Varying image quality
- Bogus images
- Other things/background



Challenges in the Data

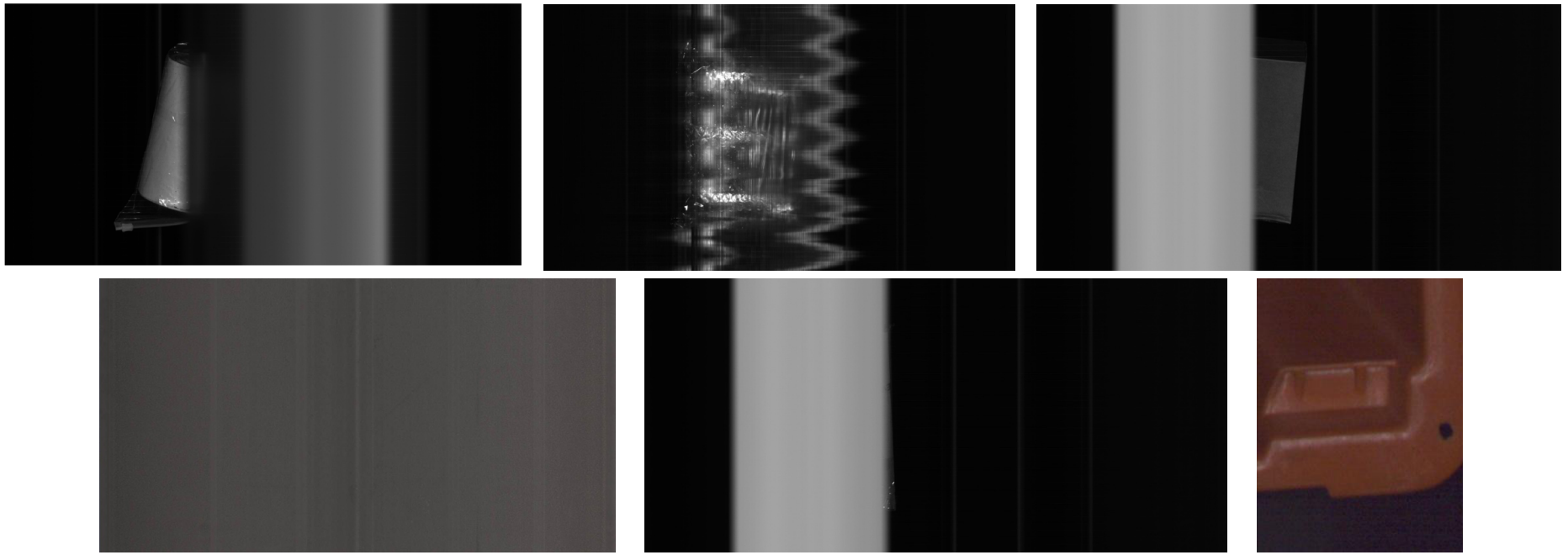
- Mixed color/grey level
- Varying illumination conditions
- **Different views**
- Varying image quality
- Bogus images
- Other things/background

Challenges in the Data

- Mixed color/grey level
- Varying illumination conditions
- Different views
- **Varying image quality**
- Bogus images
- Other things/background

Challenges in the Data

- Mixed color/grey level
- Varying illumination conditions
- Different views
- Varying image quality
- **Bogus images**
- Other things/background



Challenges in the Data

- Mixed color/grey level
- Varying illumination conditions
- Different views
- Varying image quality
- Bogus images
- **Other things/background**

Proposed Approach

Image

Bbox Refinement

FAST for **salient point detector**

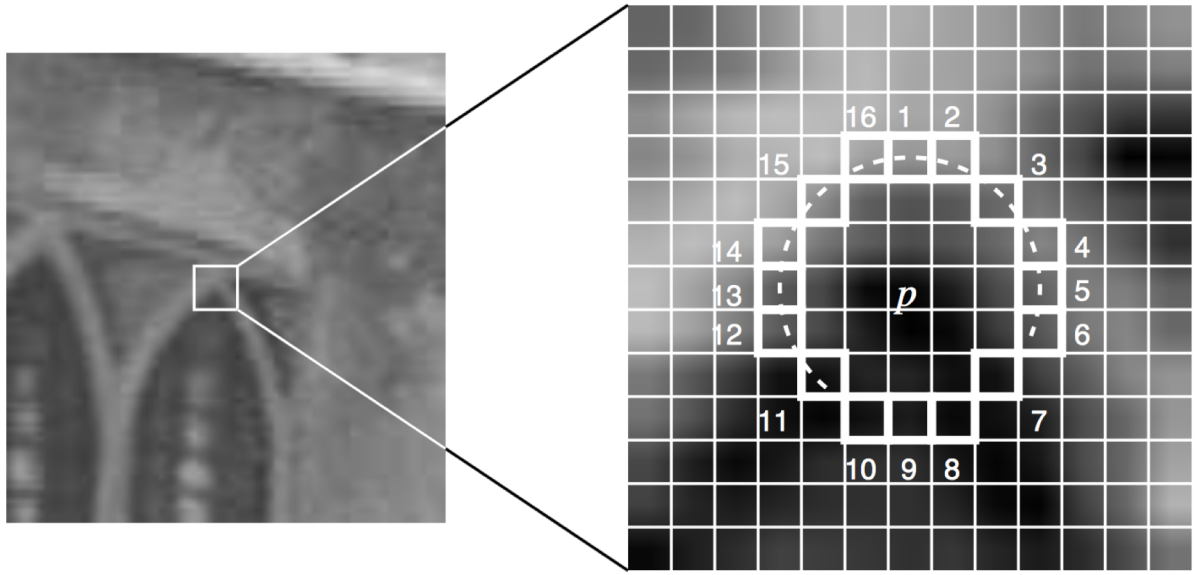
Bounding Box Refinement

- We use FAST to detect **salient points** of the image
- Foreground = bounding box containing all the salient points

E. Rosten and T. Drummond, “Machine learning for high speed corner detection,” ECCV 2006

Image preprocessing

- Interest point detection
  - FAST interest point detector



E. Rosten and T. Drummond, “Machine learning for high speed corner detection,” ECCV 2006

Triplet Logistic Loss

- Original

$$L(A, B, C) = max(0, m + D(A, B) - D(A, C))$$

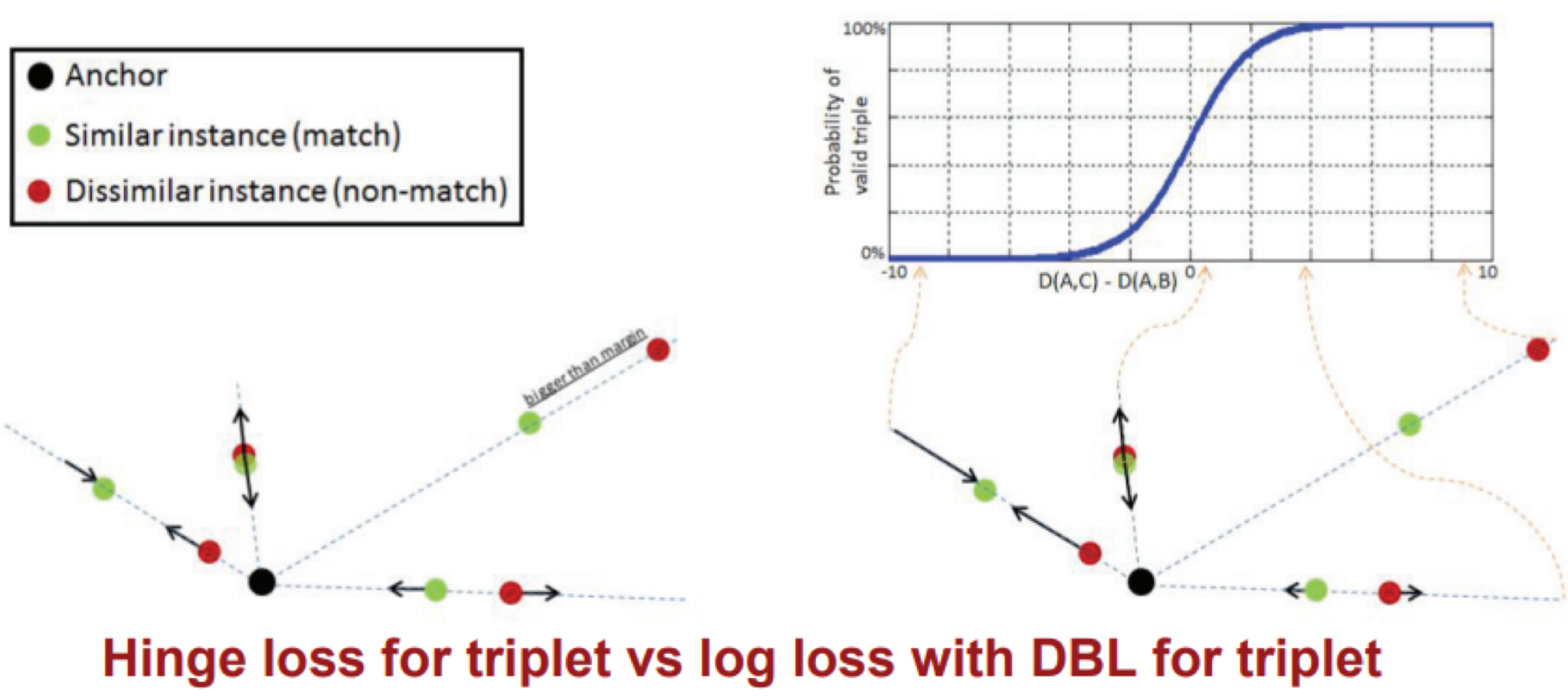
- Loss functions based on Euclidean distance might **not be optimal** for recognition
- Use loss functions **similar** to the standard softmax

$$L(A, B, C) = log(1 + exp(D(A, B) - D(A, C)))$$

Nam Vo, James Hays. ” Localizing and Orienting Street Views Using Overhead Imagery” ECCV 2016.

v2

Intuition behind the Triplet Logistic Loss



Hinge loss for triplet vs log loss with DBL for triplet

Training of Triplet Loss Networks

- VGG-16 pre-trained for object recognition - ImageNet 1K classes
- Each batch contains images from the **same class** and images from **different classes**
- Data augmentation: random rotations + cropping
- Stochastic gradient descent with momentum and weight decay
- Inference: Pool5 feature embedding

Training of ASIN Recognition Networks

- ResNet-152 pre-trained for object recognition - ImageNet 11K classes
- No data augmentation
- Stochastic gradient descent with momentum and weight decay
- Inference: Softmax output scores

Bonus: ResNet Only

- What if we drop the triplet loss network altogether?
- No combination of the rankings
- Use the ResNet prediction scores directly

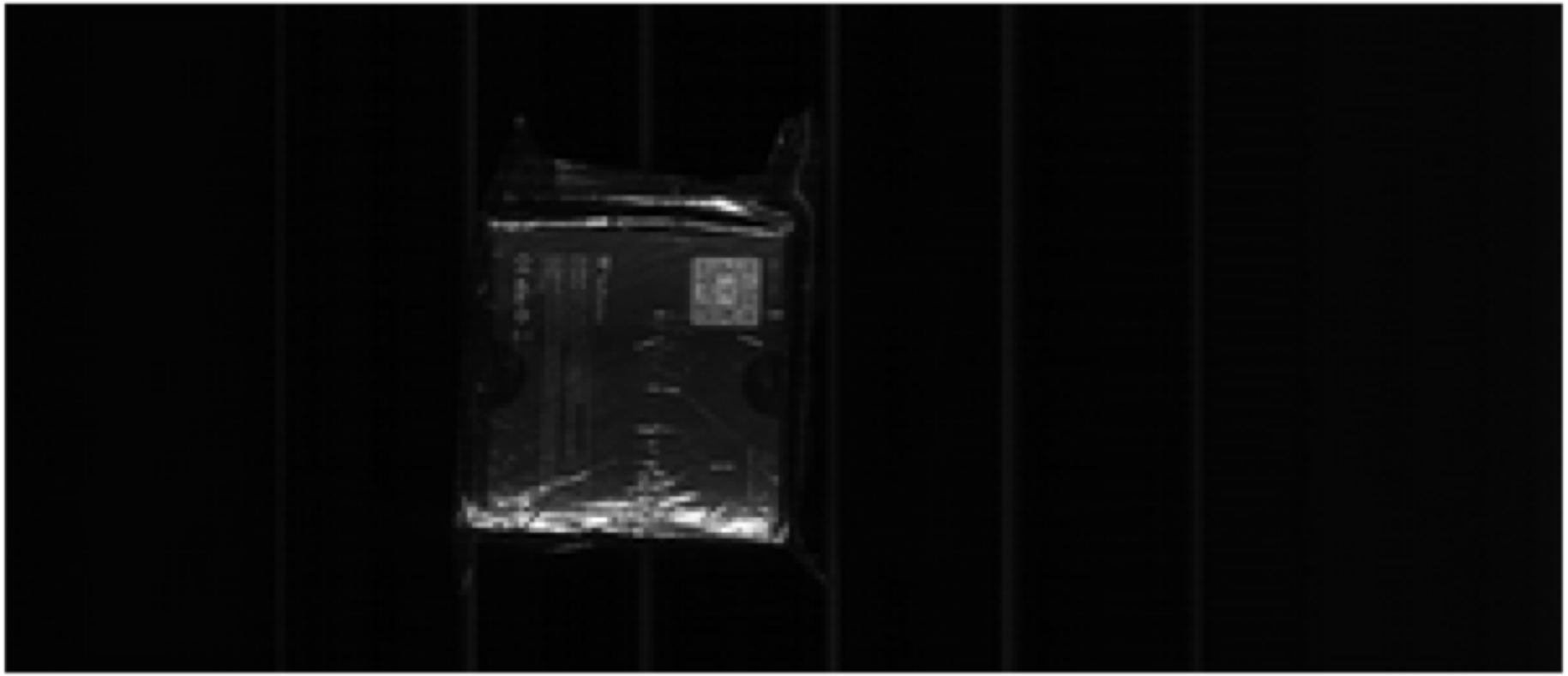
Meta Classifier

- Why we did not bother creating a meta-classifier
  - Mistakes cost too little (1/4 of making a correct guess)

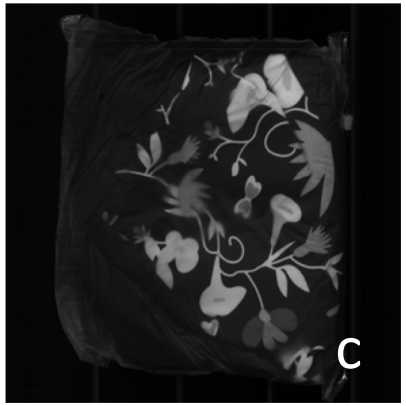
Domain Shift

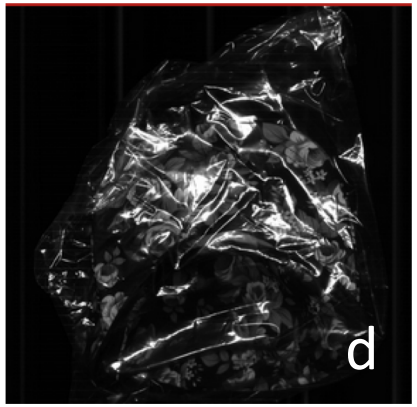
- Why we did not bother with domain adaptation
  - The new Phase 2 images mostly corresponded to new ASINs
  - Not really a domain adaptation scenario, just extra classes

Finding the correct ASIN?



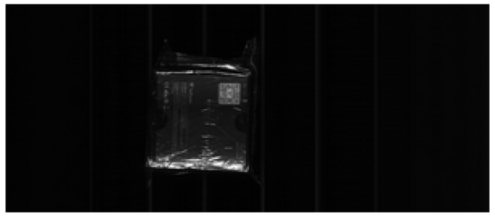
Finding the correct ASIN





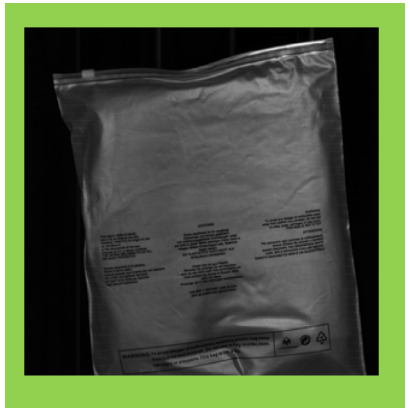
Error cases

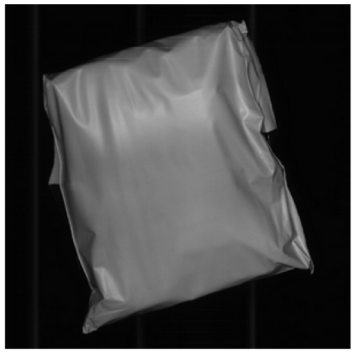
Query image



Error cases – bogus images, similar wrappings

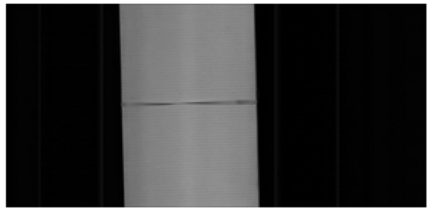
Query image

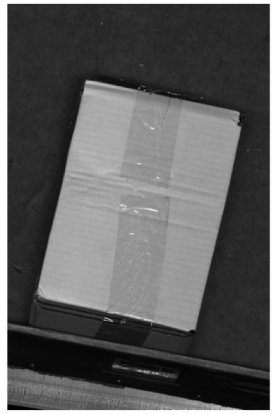


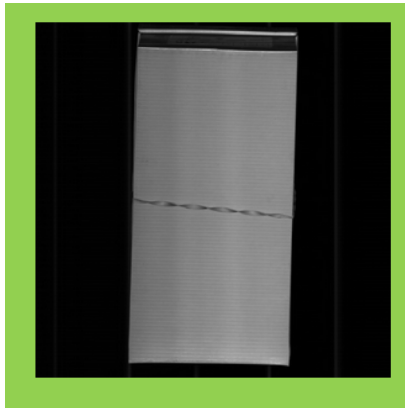


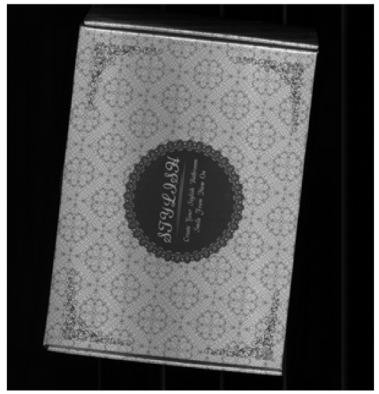
Error cases – boxes

Query image



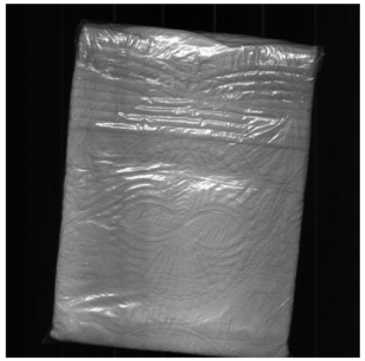


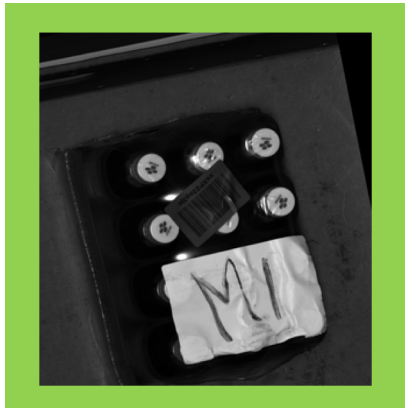


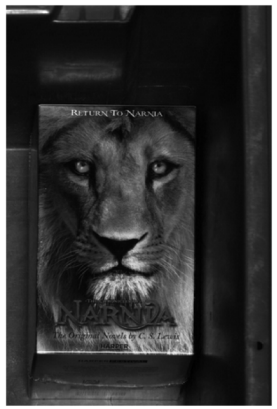


Error cases

Query image







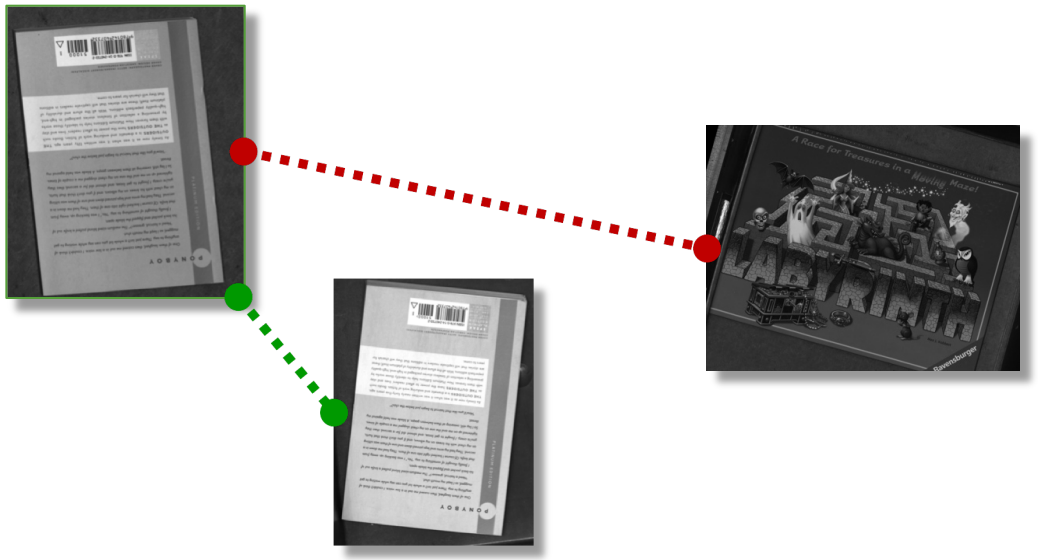
Error cases – bogus images

Query image

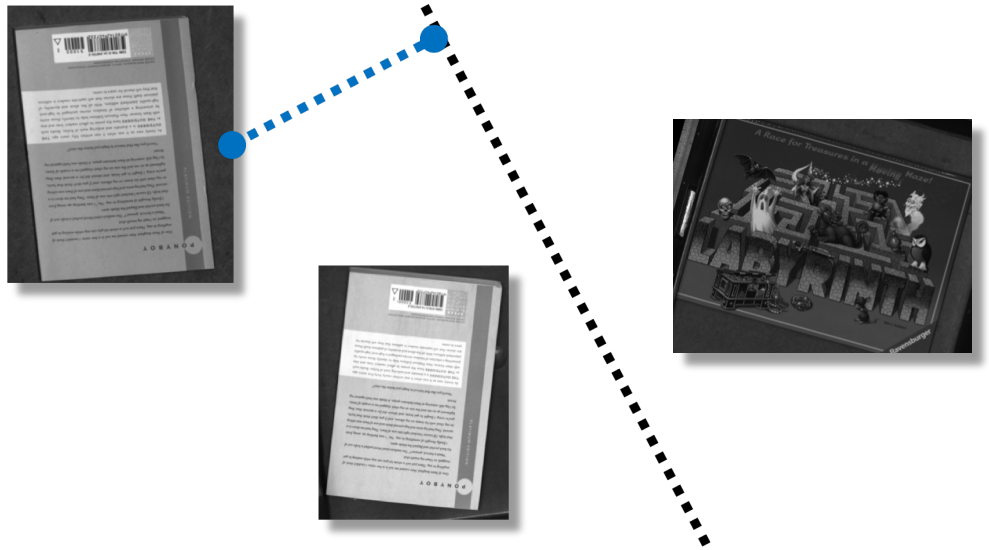
Other Experiments

- Frequency classifier: 19% (phase 1)
- CaffeNet instead of Resnet-101: -2%
- Augmentation of query image: -1%
- Augmentation of dataset: -2%
- Object detection (YOLO) not better than salient feature detection (FAST)
- Hard-negative mining did not help

How to Combine Predictions?



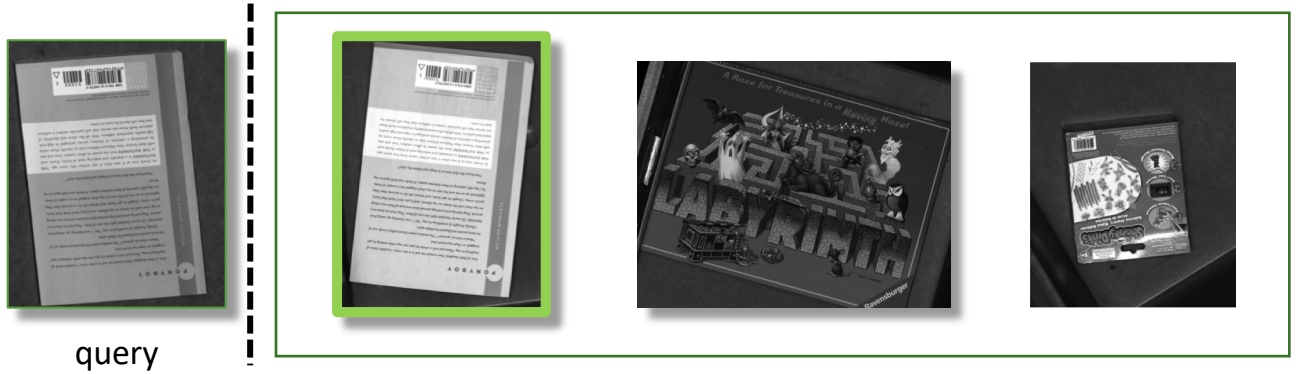
- 1-NN
  - **Rank ASINs** based on the shortest distance



- ASIN Classifier (AC)
  - **Rank ASINs** based on the prediction score ("distance" to the hyperplane)

How to Combine Rankings?

- Normalized rank
  - `0, 1, 2, …, len(manifest)-1`
- Prediction:
  - `argmin(rank)`



1-NN	**1**	**2**	**0**
AC	**0**	**2**	**1**
Total	**1**	**4**	**1**

How to Combine Rankings?

- Weighted sum
  - Weight coefficient
  - Why **2.1**?
    - 2.0: votes count twice
    - +.1: tie-breaker

1-NN	**1**	**2**	**0**
* 1.0	1.0	2.0	0.0
AC	**0**	**2**	**1**
*** 2.1**	0.0	4.2	2.1
Total	**1.0**	**6.2**	**2.1**