



ASi

DATA SCIENCE
STRATA LONDON 2017

@ASIDataScience

Gaining Additional labels

Yingsong Zhang
yingsong.z@asidatascience.com

23/05/2017



This presentation
reflects my subjective
opinions

Supervised learning

- Given
 - Labelled data $\{(features, label)\}$, as $\{(x^l, y^l)\}$
 - Implicitly, a lot of unlabeled data $\{(x, .)\}$
- From the labelled data, inferring a function $f(x)$ for predicting the unknown labels for $\{(x, .)\}$

Labelled data



Classifier

Semi-supervised learning

- Given
 - Labelled data $\{(features, label)\}$, as $\{(x^l, y^l)\}$
 - Explicitly, a lot of unlabeled data $\{(x, .)\}$
- Inferring a function $y = f(x)$ for predicting the unknown labels for new instances of features, from
 - the labelled data set
 - AND the unlabeled data set



Why bother?

- The performance of your classifier almost always improves with more labelled data.
- Labelled data can sometimes be expensive, difficult, or time consuming to obtain
- Unlabelled data bear a lot of additional information, for example, the distribution
- Can we improve our performance for free?
- Make use of the unlabelled data – semi-supervised learning

Common types of semi-supervised learning

- Self-training
- Co-training and multi-view learning
- Generative Models

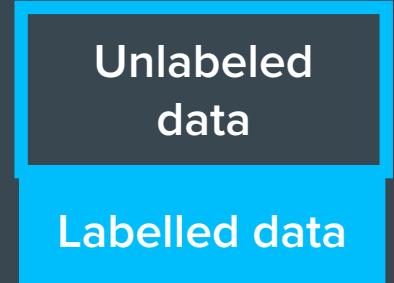
Common types of semi-supervised learning

- Self-training
- Co-training and multi-view learning
- Generative Models
 - strong assumption on the distribution of the data

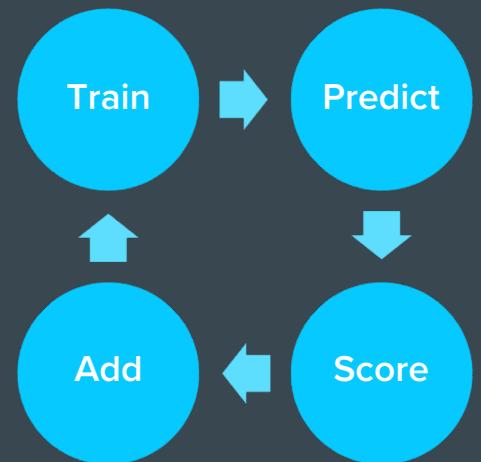
Self-training

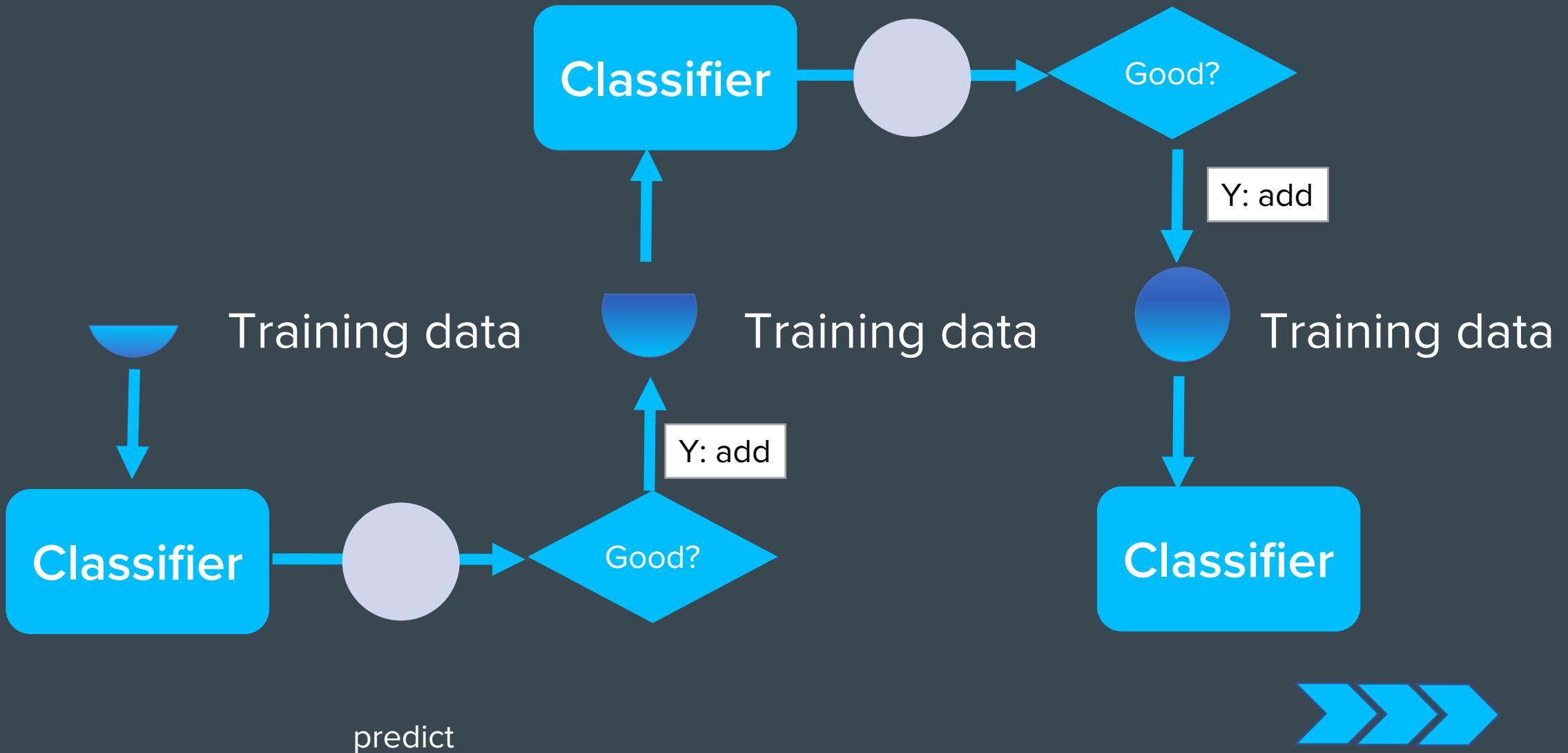
- Wrapper:

1. Initialize the training set with only the labelled data
2. Train a classifier $f(x)$ on the training set
3. Predict on the unlabeled data with $f(x)$
4. Score $\{(x, f(x))\}$ with the selection metric
5. Add the m best new instances to the training set
6. Repeat 2-5 with the enlarged training set



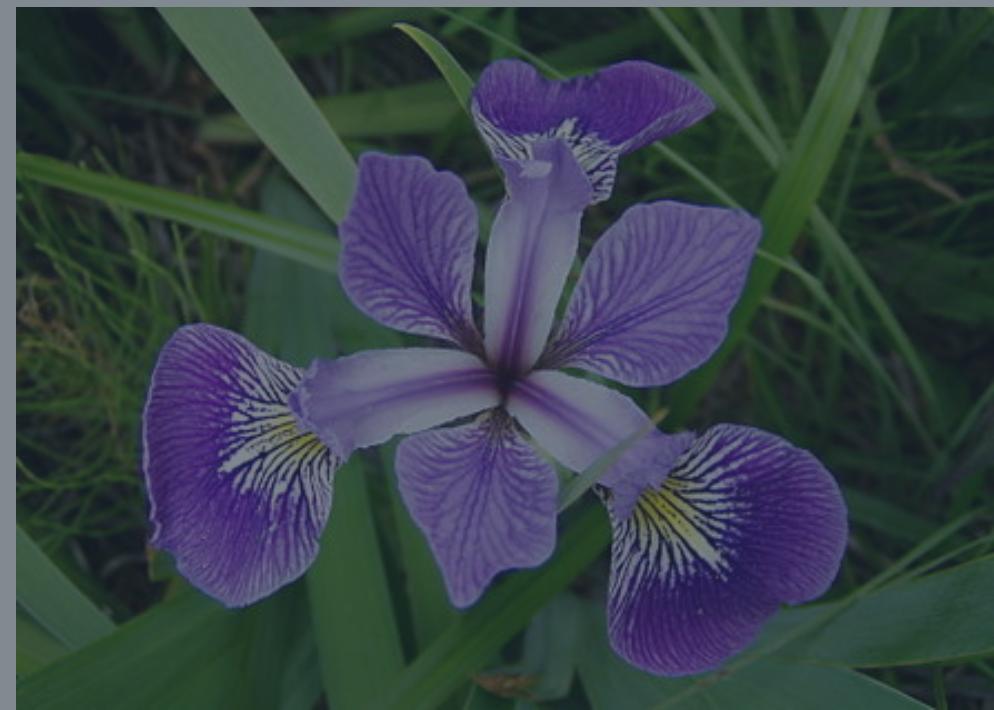
Train





Toy problem demo: Iris data set

- This well-known data set consists of 3 different types of irises of 150 samples
- Each sample has 4 features
- We will only use the first 2 features for 2d visualization purpose



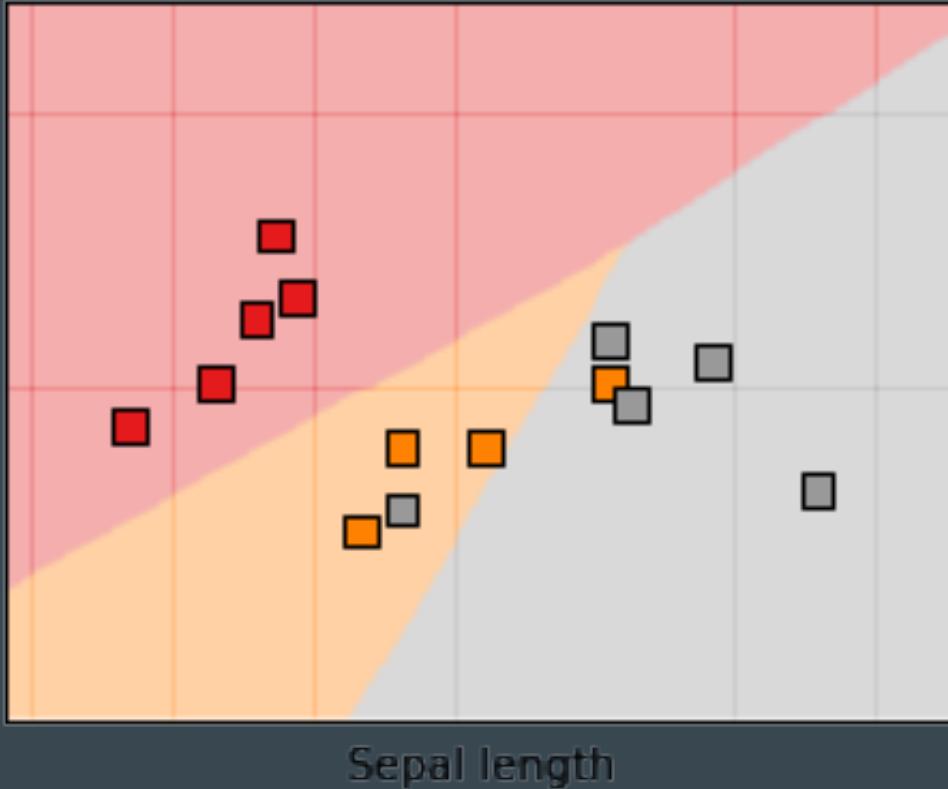
Settings

- Randomly select 15 samples as the labelled data
- The rest, 135 samples, treated as unlabeled data
- Logistic regression
- Code is available at Git Hub
- https://github.com/zysalice/strata_2017_selftraining/blob/master/ToyProblem_iris.ipynb

■ (x, y)

Trained with 15 labelled samples

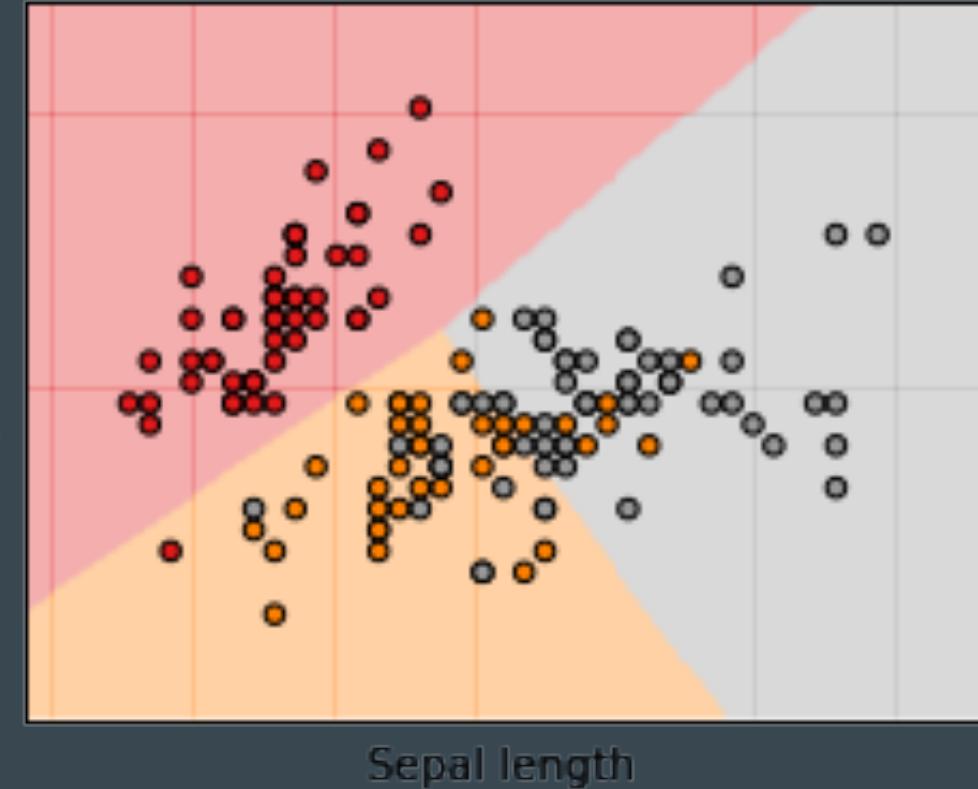
Sepal width



Sepal length

Trained with 150 labelled samples

Sepal width



Sepal length

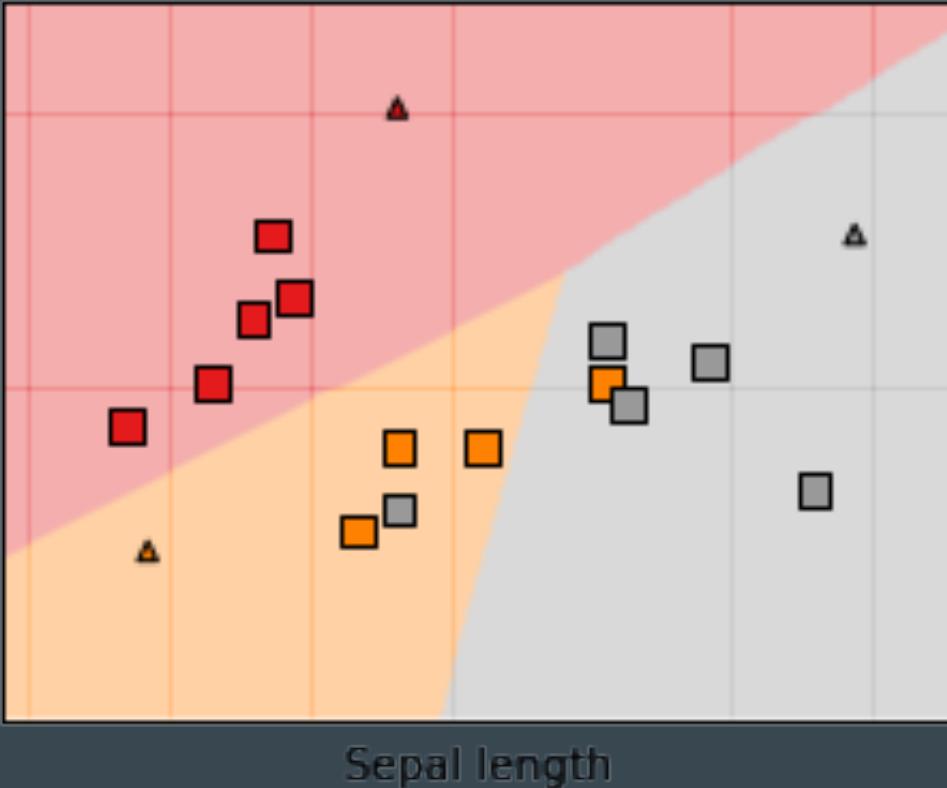
Comparison of Decision Boundaries

■ (x, y)

▲ $(x, f(x))$

With 15 labelled + 3 unlabeled samples

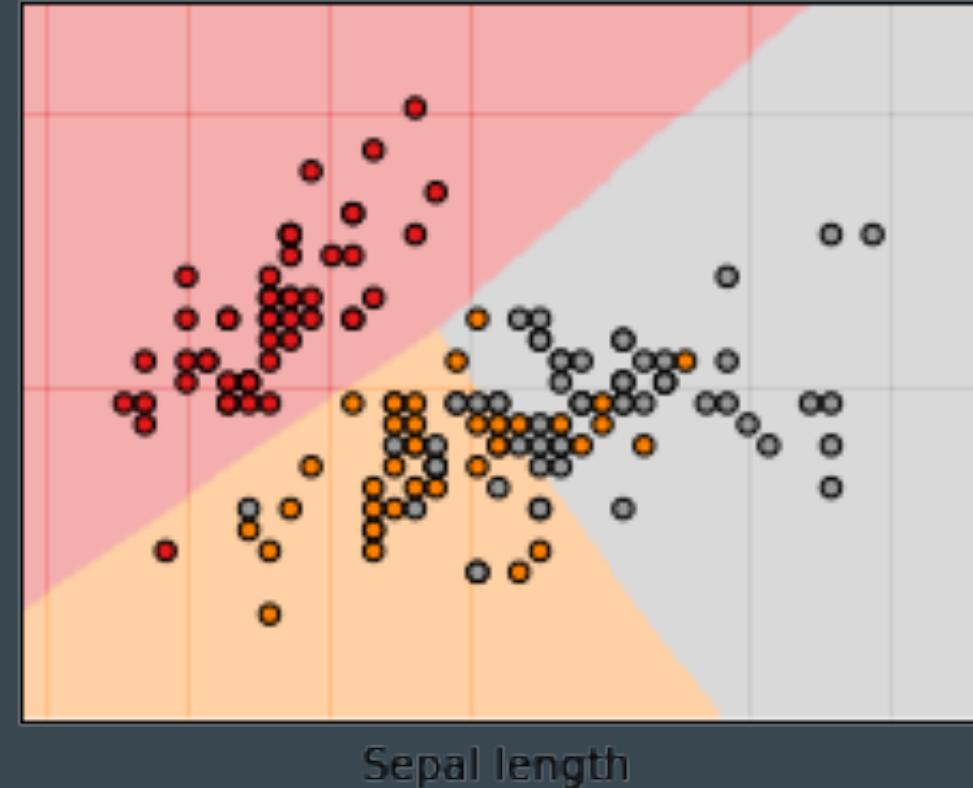
Sepal width



Sepal length

Trained with 150 labelled samples

Sepal width



Sepal length

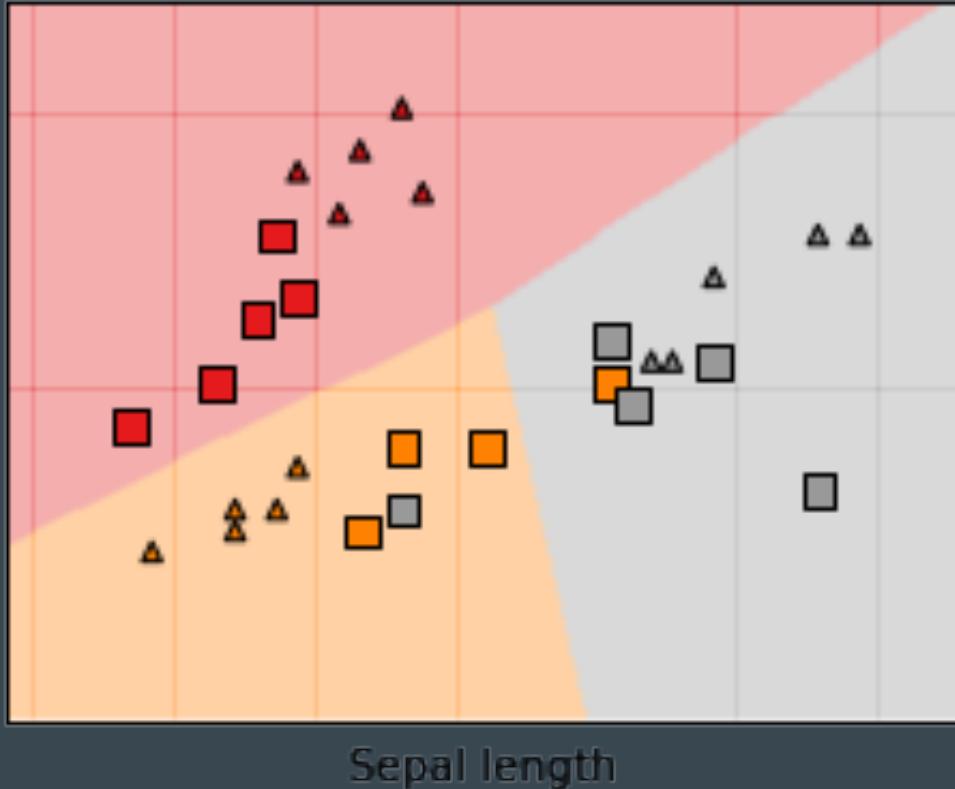
Comparison of Decision Boundaries

■ (x, y)

▲ $(x, f(x))$

With 15 labelled + 15 unlabeled samples

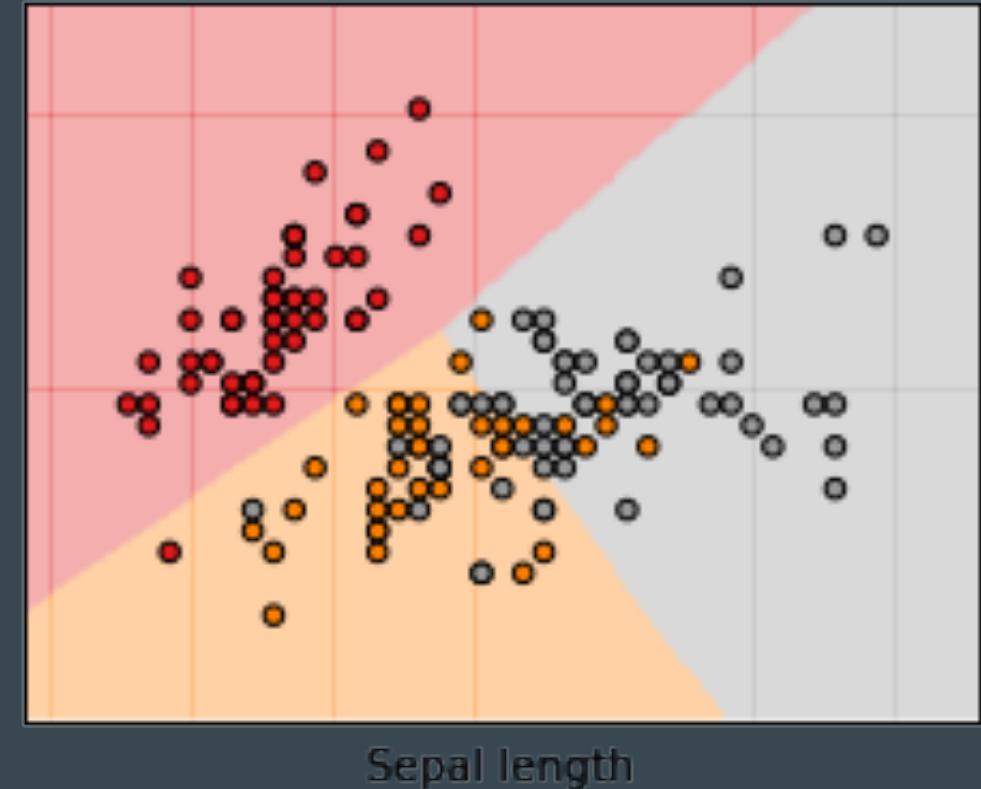
Sepal width



Sepal length

Trained with 150 labelled samples

Sepal width



Sepal length

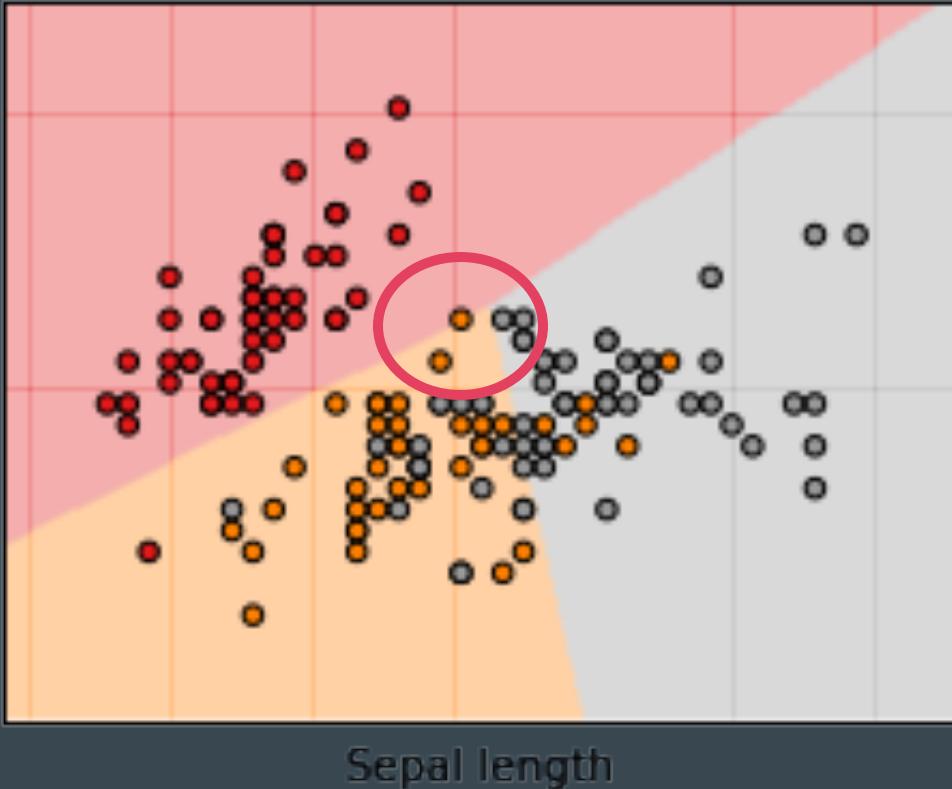
Comparison of Decision Boundaries

■ (x, y)

▲ $(x, f(x))$

With 15 labelled + 15 unlabeled samples

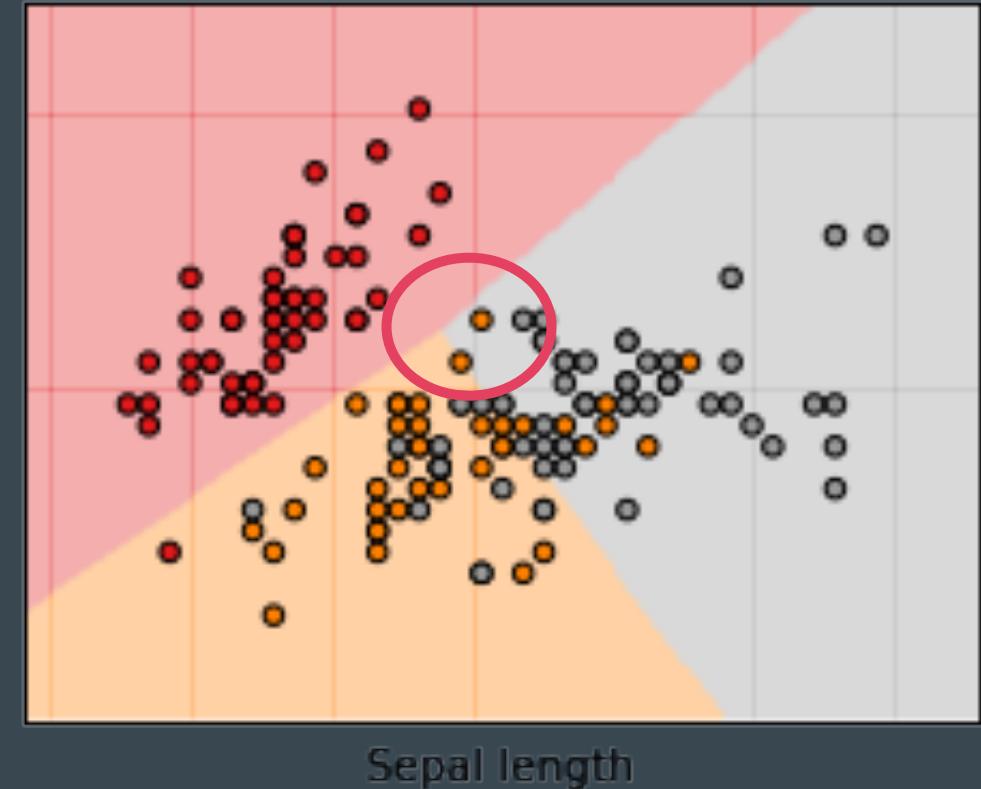
Sepal width



Sepal length

Trained with 150 labelled samples

Sepal width



Sepal length

Comparison of Decision Boundaries

Process of unlabeled data selection has a big influence

- Only add good instances
 - include only those points in which the model has high confidence
 - Use a metric that is independent from the classifier if possible
- Allow bad instances to be deleted
 - the new classifier doesn't agree with the old classifier
- Adding and deleting incrementally
 - the damage of adding/deleting wrong instance is limited

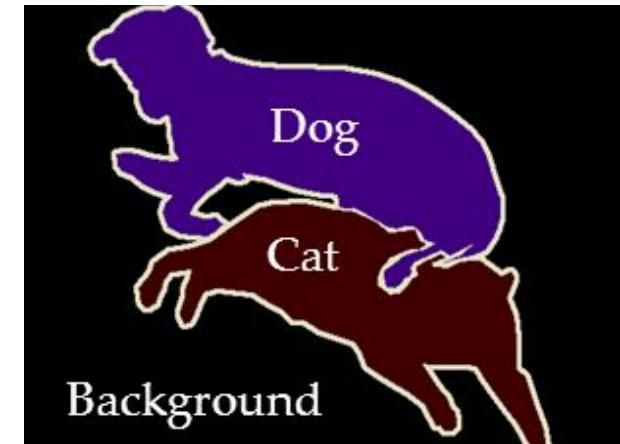
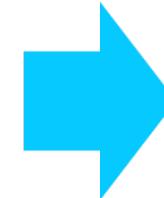
No free lunch

- More effort required to design selection process
- Unlabeled data does not always help
 - adding incorrect labels can corrupt the model
 - there are situations that self-training performs badly no matter what
 - No theoretical guarantee of convergence to better result

Self-training in CNN semantic image segmentation

“Fully convolutional networks for semantic segmentation”

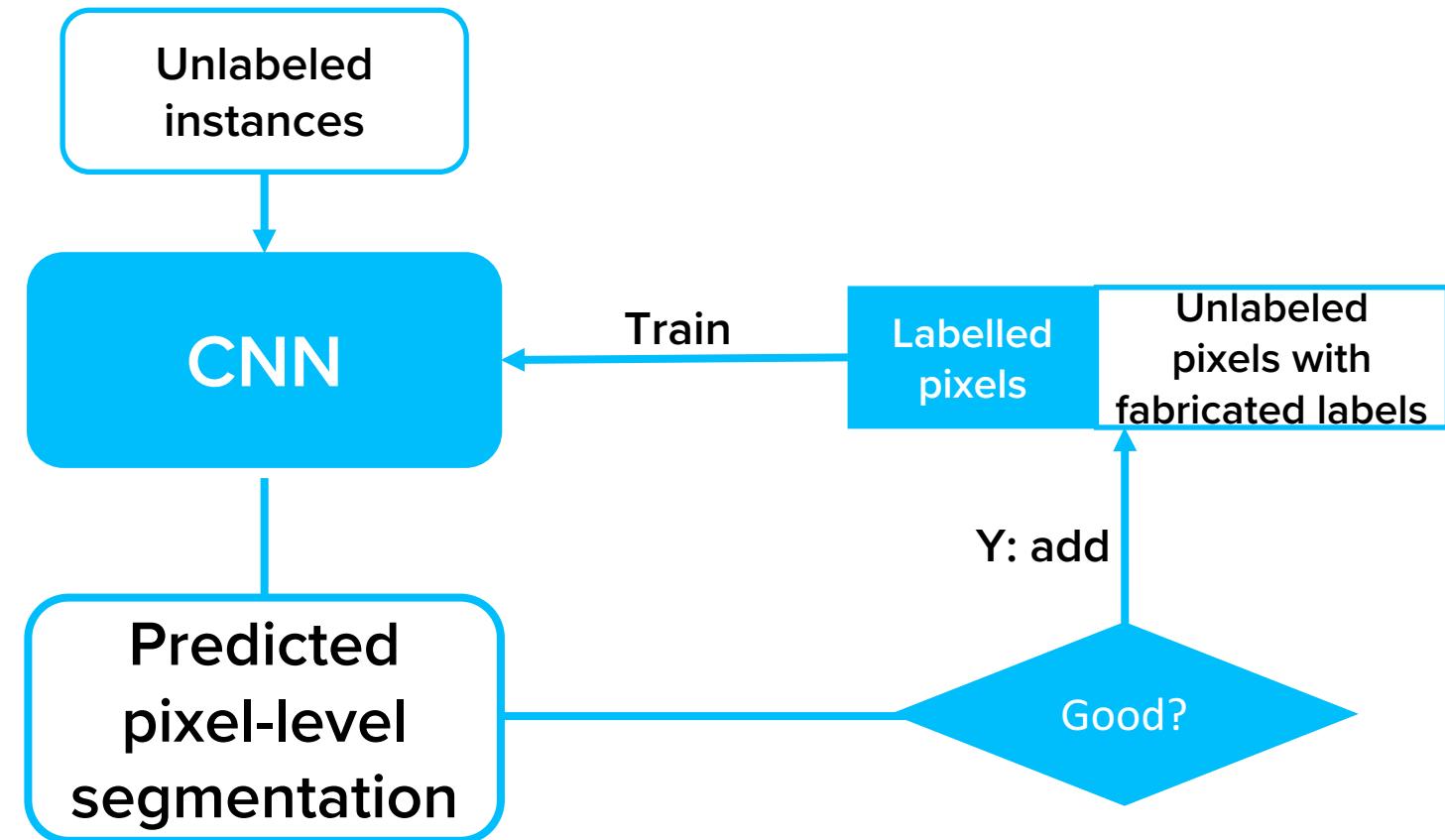
Long et al,
CVPR2015



- Image goes in
- Labels for each pixel come out
- Training set: images with pixel-level annotation - expensive

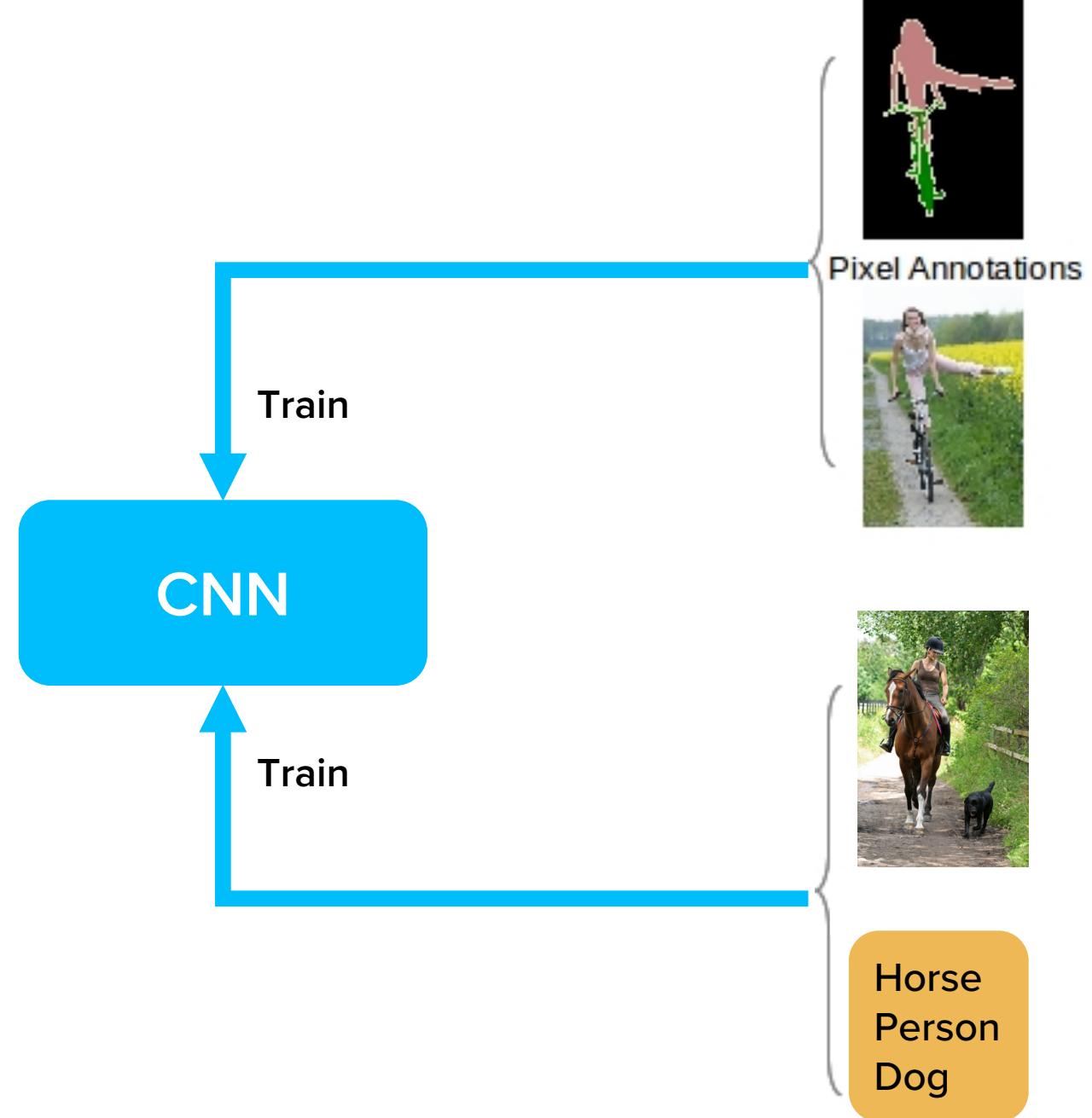
Self-training with CNN

- Start from a basic CNN for semantic segmentation



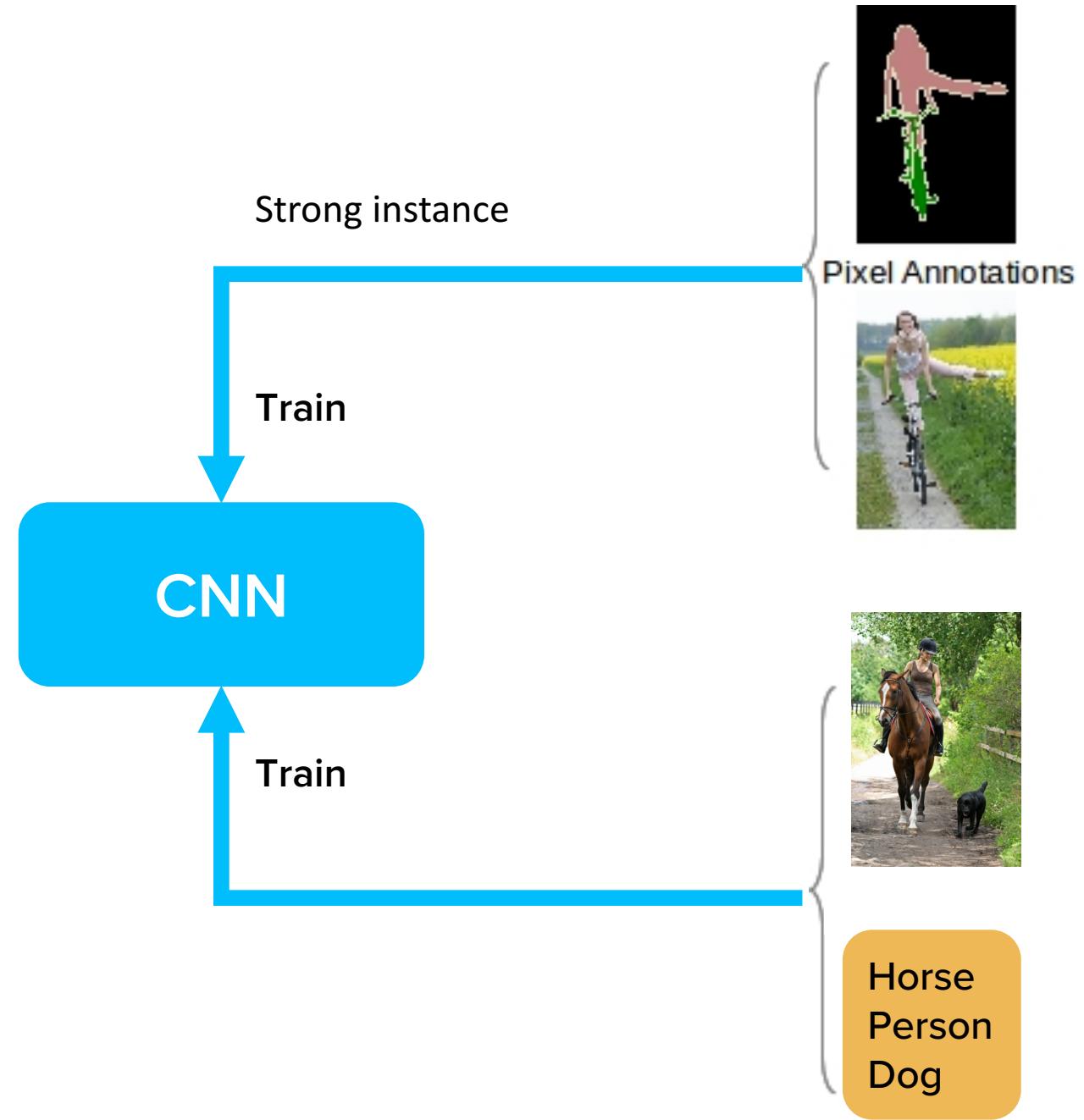
Semantic Image segmentation (common scene)

- Papandreou, George, et al.
"Weakly-and semi-supervised learning of a DCNN for semantic image segmentation." arXiv preprint arXiv:1502.02734 (2015).



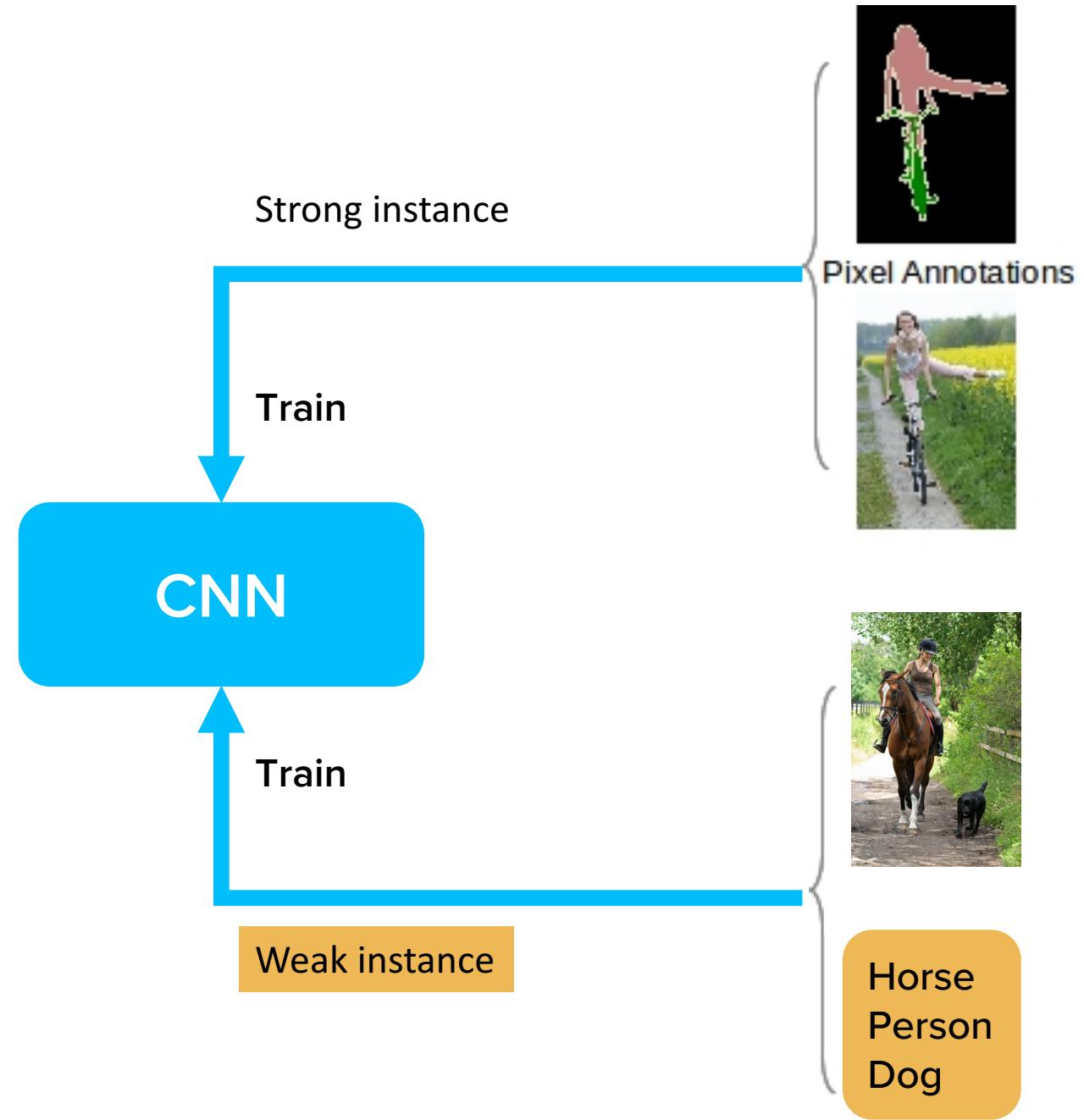
Semantic Image segmentation (common scene)

- Papandreou, George, et al.
"Weakly-and semi-supervised learning of a DCNN for semantic image segmentation." arXiv preprint arXiv:1502.02734 (2015).



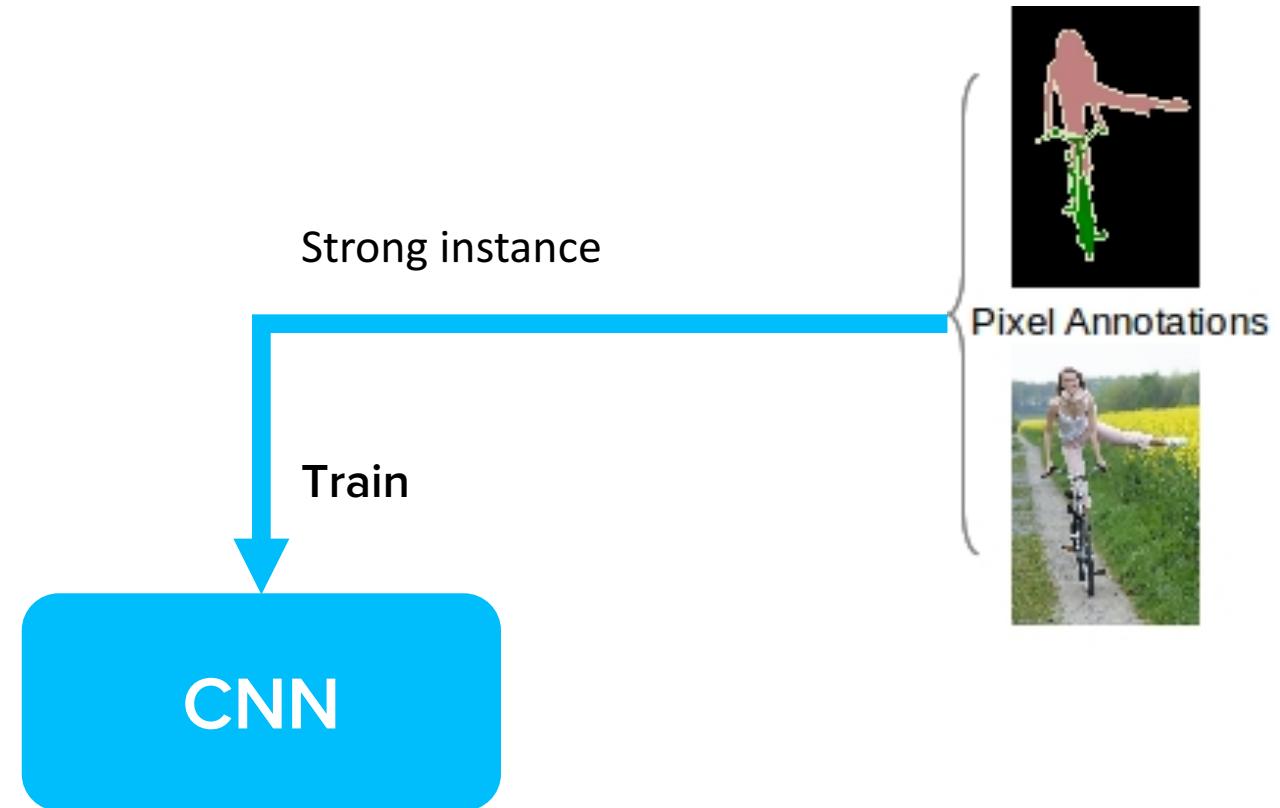
Semantic Image segmentation (common scene)

- Papandreou, George, et al.
"Weakly-and semi-supervised learning of a DCNN for semantic image segmentation." arXiv preprint arXiv:1502.02734 (2015).



Semantic Image segmentation (common scene)

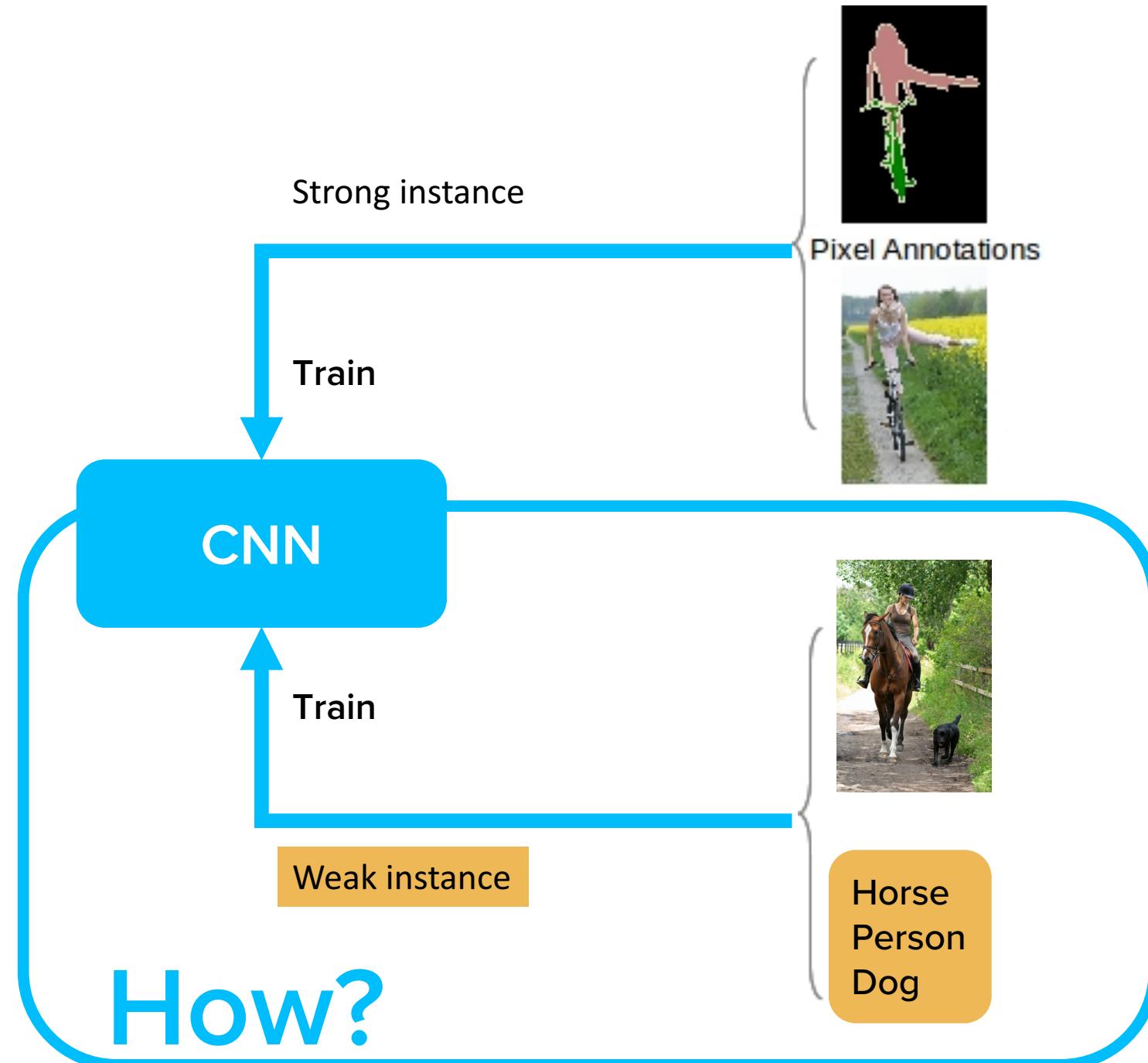
- Papandreou, George, et al.
"Weakly-and semi-supervised learning of a DCNN for semantic image segmentation." arXiv preprint arXiv:1502.02734 (2015).



- Train an CNN with the strong instance
- The cost function is the cross-entropy function

Semantic Image segmentation (common scene)

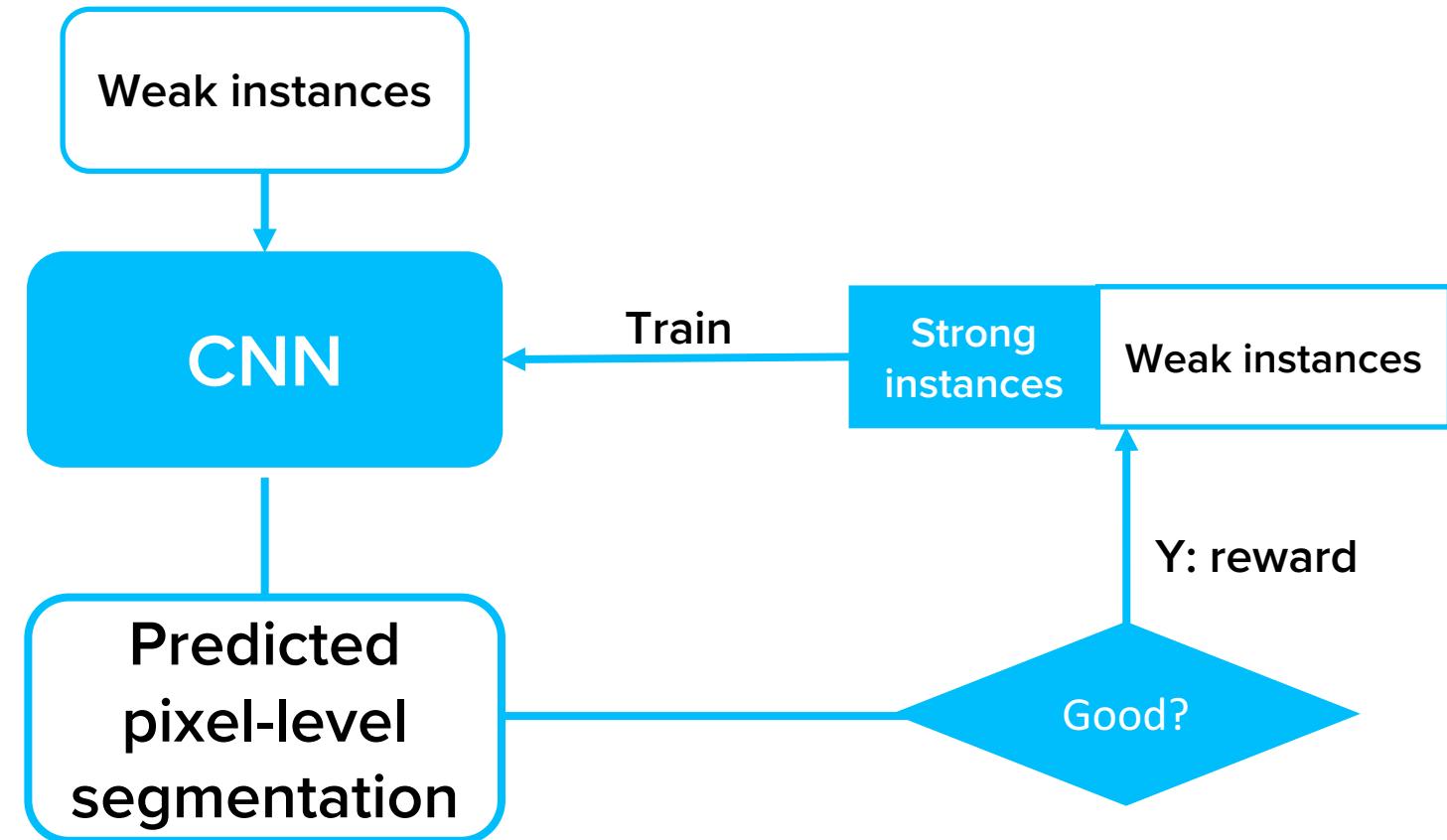
- Papandreou, George, et al.
"Weakly-and semi-supervised learning of a DCNN for semantic image segmentation." arXiv preprint arXiv:1502.02734 (2015).



Semantic Image segmentation (common scene)

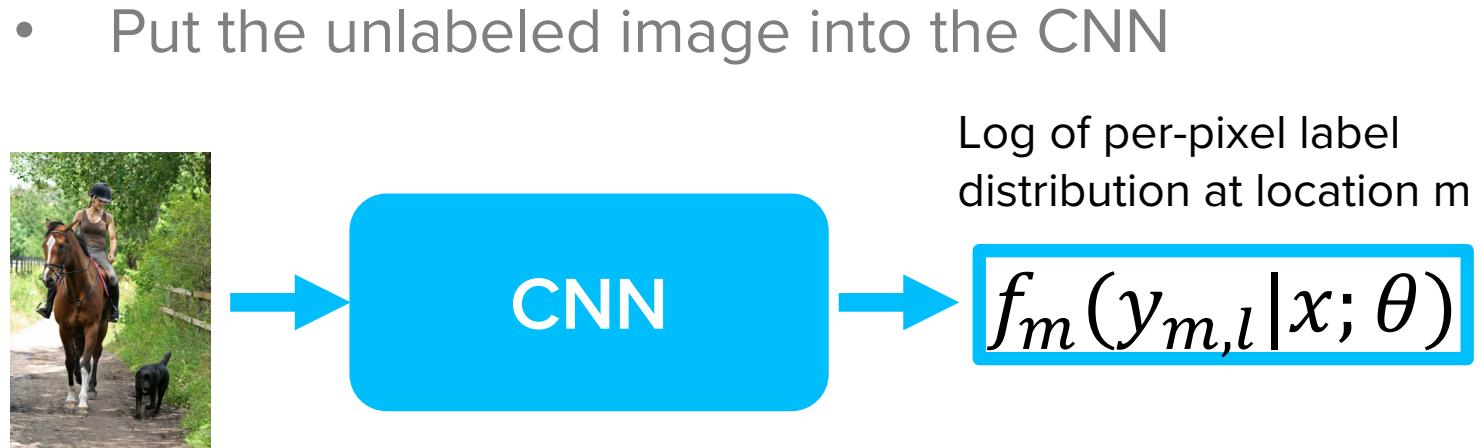
- Papandreou, George, et al.
"Weakly-and semi-supervised learning of a DCNN for semantic image segmentation." arXiv preprint arXiv:1502.02734 (2015).

- Use the self-training wrapper



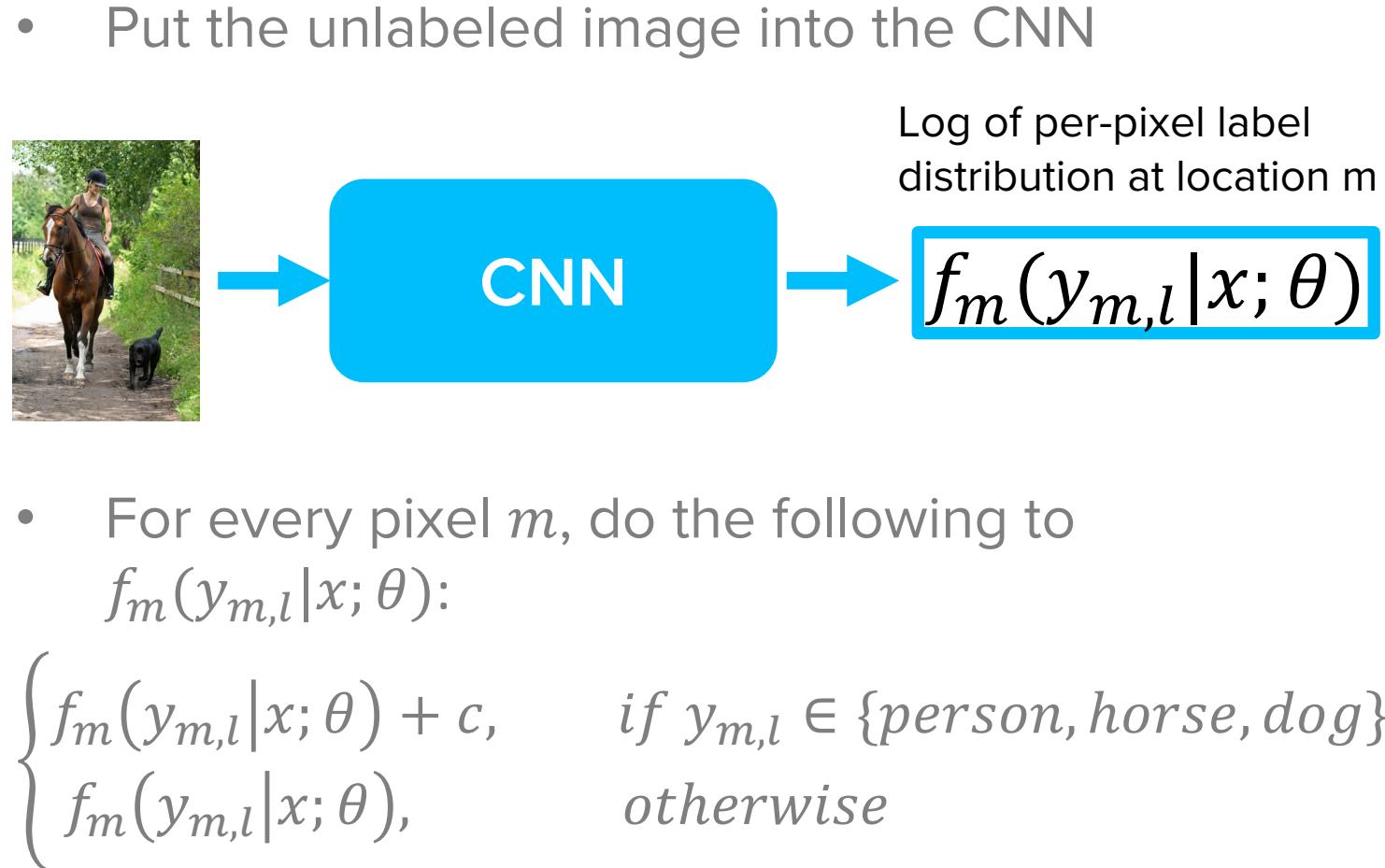
Semantic Image segmentation (common scene)

- Papandreou, George, et al.
"Weakly-and semi-supervised learning of a DCNN for semantic image segmentation." arXiv preprint arXiv:1502.02734 (2015).



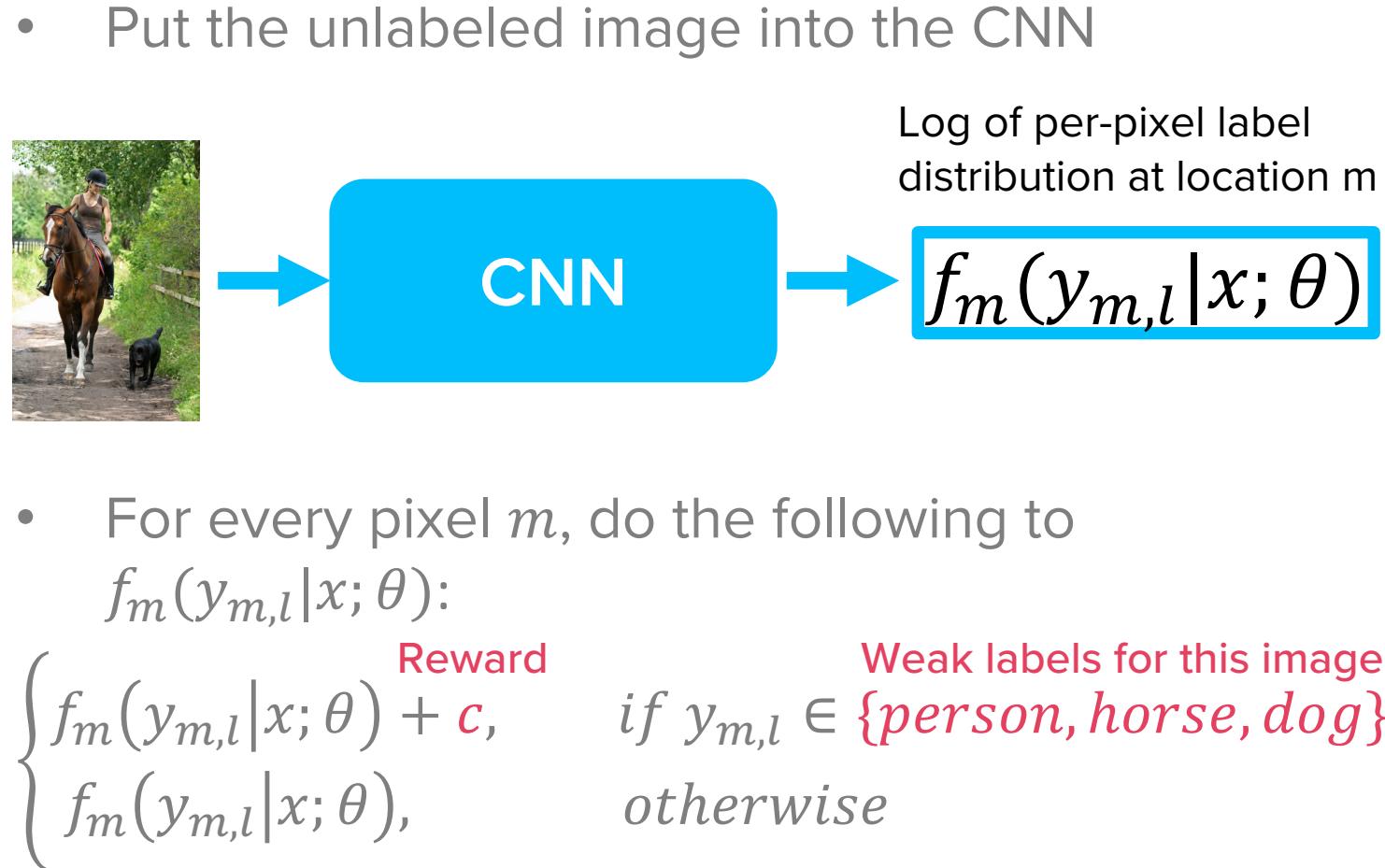
Semantic Image segmentation (common scene)

- Papandreou, George, et al.
"Weakly-and semi-supervised learning of a DCNN for semantic image segmentation." arXiv preprint arXiv:1502.02734 (2015).



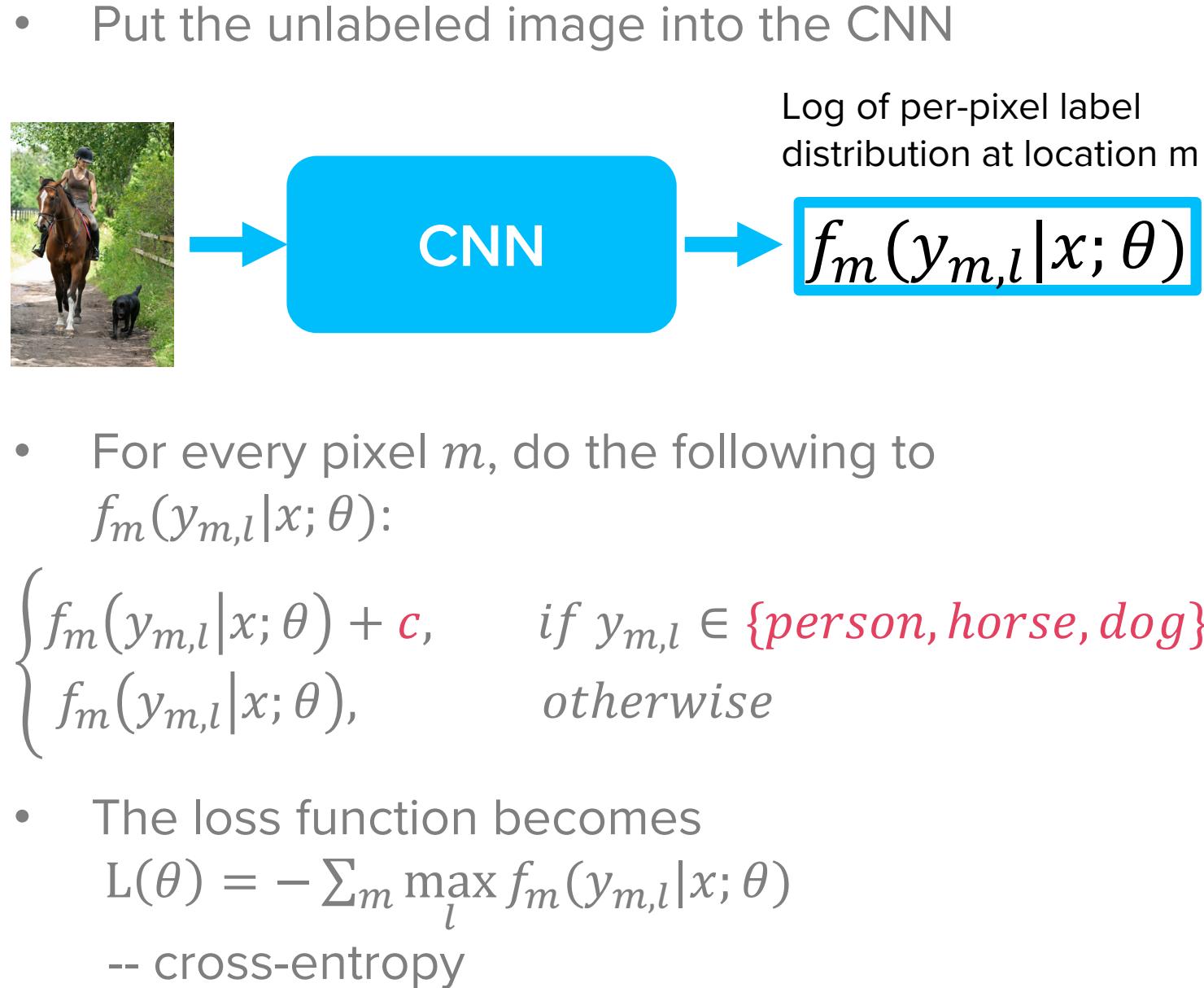
Semantic Image segmentation (common scene)

- Papandreou, George, et al.
"Weakly-and semi-supervised learning of a DCNN for semantic image segmentation." arXiv preprint arXiv:1502.02734 (2015).



Semantic Image segmentation (common scene)

- Papandreou, George, et al.
"Weakly-and semi-supervised learning of a DCNN for semantic image segmentation." arXiv preprint arXiv:1502.02734 (2015).



Improved performance by semi-supervised learning

	# Strong	#weak	Dice measure
supervised	15582	-	0.687
	10582	-	0.676
	1464	-	0.625
semi-super	1464	9118	0.646
	10582	123287	0.677
	15582	118287	0.700

Self-training in feature engineering

Filtering twitter message

- Twitter is a great source for information
- After scraping, we filtered twitter message using keywords
- Human experts labelled a small portion of the data with 'Spam' (68%) and 'Good' (32%)
- Purpose:
 - develop an algorithm for filtering out the spams.

Naïve Bayesian spam filter

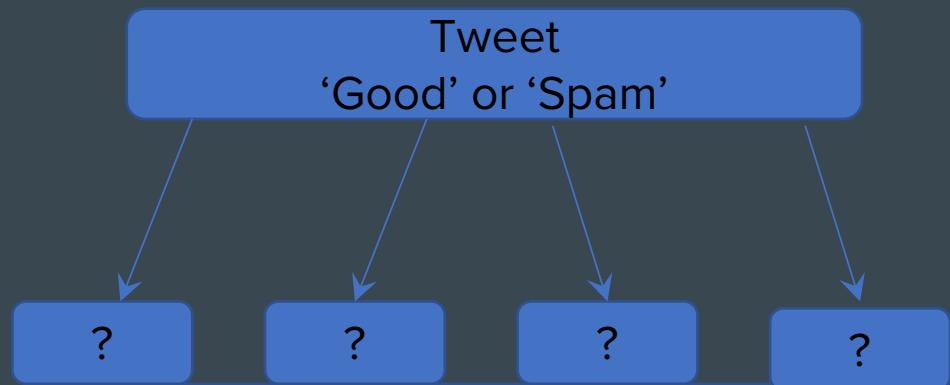
- A simple Naïve Bayesian spam filter seems to perform reasonably well

Confusion Matrix (percentage)

		Predicted label	
		S	G
True label	S	80.24%	19.76%
	G	39.32%	60.68%

How can we improve

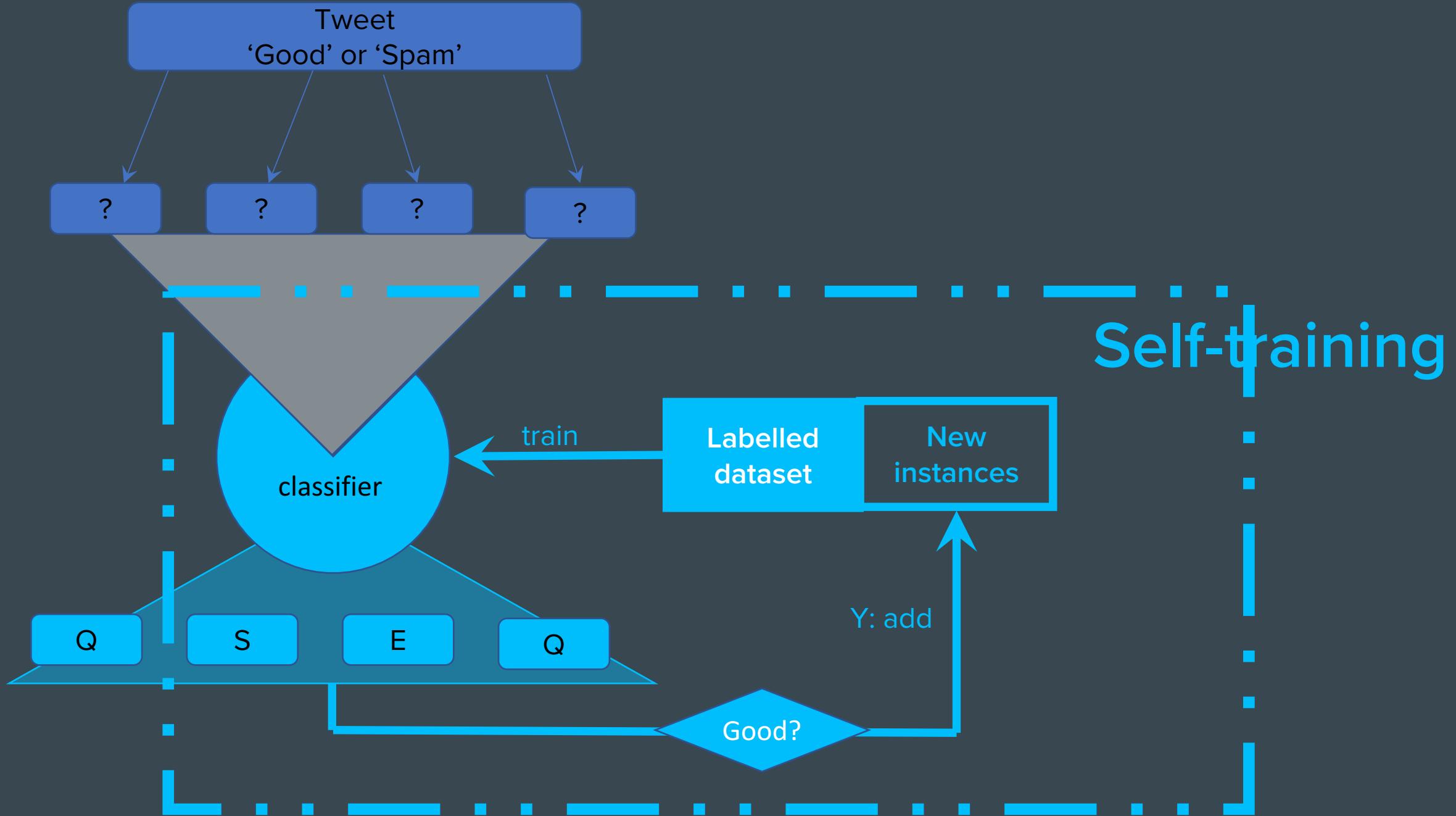
- More features than just ‘words’
 - A lot of ‘Good’ tweets contain questions and statements
 - The new features:
 - the proportion of question, statement, emotion or others
 - Semantic level features – we don’t have labels



question, statement, emotion or other?

We need to train this classifier.





Prediction with augmented features from semi-supervised learning

- Reduced False positive rate from 19.76% → 18.63%

Confusion Matrix (percentage)

		Predicted label	
		S	G
True label	S	81.37%	18.63%
	G	39.32%	60.68%

A few more words...

- If possible, get more labelled data
- **Beware of overfitting when using complicated models**
- A lot of effort required
- It won't always help

SHERLOCK ML

<https://sherlockml.com>

Invite code: *Strata2017*