Stacking With Auxiliary Features

Nazneen Rajani and Ray Mooney

nrajani@cs.utexas.edu and mooney@cs.utexas.edu

University of Texas at Austin

Introduction

Introduction

- Ensembling algorithms cannot effectively discriminate across:

Introduction

- Ensembling algorithms cannot effectively discriminate across:
  - component systems

Introduction

- Ensembling algorithms cannot effectively discriminate across:
  - component systems
  - input instances

Introduction

- Ensembling algorithms cannot effectively discriminate across:
  - component systems
  - input instances
- We propose Stacking With Auxiliary Features (SWAF) as a general ML algorithm

Introduction

- Ensembling algorithms cannot effectively discriminate across:
  - component systems
  - input instances
- We propose Stacking With Auxiliary Features (SWAF) as a general ML algorithm
- Demonstrate SWAF on various challenging structured prediction tasks:

Introduction

- Ensembling algorithms cannot effectively discriminate across:
  - component systems
  - input instances
- We propose Stacking With Auxiliary Features (SWAF) as a general ML algorithm
- Demonstrate SWAF on various challenging structured prediction tasks:
  - Slot Filling (SF)

Introduction

- Ensembling algorithms cannot effectively discriminate across:
  - component systems
  - input instances
- We propose Stacking With Auxiliary Features (SWAF) as a general ML algorithm
- Demonstrate SWAF on various challenging structured prediction tasks:
  - Slot Filling (SF)
  - Entity Discovery and Linking (EDL)

Introduction

- Ensembling algorithms cannot effectively discriminate across:
  - component systems
  - input instances
- We propose Stacking With Auxiliary Features (SWAF) as a general ML algorithm
- Demonstrate SWAF on various challenging structured prediction tasks:
  - Slot Filling (SF)
  - Entity Discovery and Linking (EDL)
  - ImageNet object detection

Introduction

- Ensembling algorithms cannot effectively discriminate across:
  - component systems
  - input instances
- We propose Stacking With Auxiliary Features (SWAF) as a general ML algorithm
- Demonstrate SWAF on various challenging structured prediction tasks:
  - Slot Filling (SF)
  - Entity Discovery and Linking (EDL)
  - ImageNet object detection

Introduction

- Ensembling algorithms cannot effectively discriminate across:
  - component systems
  - input instances
- We propose Stacking With Auxiliary Features (SWAF) as a general ML algorithm
- Demonstrate SWAF on various challenging structured prediction tasks:
  - Slot Filling (SF)
  - Entity Discovery and Linking (EDL)
  - ImageNet object detection

} NLP

Introduction

- Ensembling algorithms cannot effectively discriminate across:
    - component systems
    - input instances
- We propose Stacking With Auxiliary Features (SWAF) as a general ML algorithm
- Demonstrate SWAF on various challenging structured prediction tasks:
    - Slot Filling (SF)
    - Entity Discovery and Linking (EDL)
    - ImageNet object detection

$\left.\begin{array}{l} \\ \\ \end{array}\right\}$ NLP

Introduction

- Ensembling algorithms cannot effectively discriminate across:
  - component systems
  - input instances
- We propose Stacking With Auxiliary Features (SWAF) as a general ML algorithm
- Demonstrate SWAF on various challenging structured prediction tasks:
  - Slot Filling (SF)
  - Entity Discovery and Linking (EDL)
  - ImageNet object detection

} NLP

→ Vision

Slot Filling

org: Microsoft
1. city_of_headquarters: 2. website: 3. subsidiaries: 4. employees: 5. shareholders: ⋮

Microsoft is a technology company, headquartered in Redmond, Washington that develops ...
city_of_headquarters: Redmond **provenance:** **confidence score:** 1.0

Entity Discovery and Linking (EDL)

Source Corpus Document:
Hillary Clinton Not Talking About '92 *Clinton*-Gore Confederate Campaign Button..

FreeBase entry:

Hillary Diane Rodham Clinton is a US Secretary of State, U.S. Senator, and First Lady of the United States. From 2009 to 2013, she was the 67th Secretary of State, serving under President Barack Obama. She previously represented New York in the U.S. Senate.

FreeBase entry:

William Jefferson "Bill" Clinton is an American politician who served as the 42nd President of the United States from 1993 to 2001. Clinton was Governor of Arkansas from 1979 to 1981 and 1983 to 1992, and Arkansas Attorney General from 1977 to 1979.

Entity Discovery and Linking (EDL)

Source Corpus Document:
Hillary Clinton Not Talking About ’92 *Clinton*-Gore Confederate Campaign Button..

FreeBase entry:

Hillary Diane Rodham Clinton is a US Secretary of State, U.S. Senator, and First Lady of the United States. From 2009 to 2013, she was the 67th Secretary of State, serving under President Barack Obama. She previously represented New York in the U.S. Senate.

FreeBase entry:

William Jefferson "Bill" Clinton is an American politician who served as the 42nd President of the United States from 1993 to 2001. Clinton was Governor of Arkansas from 1979 to 1981 and 1983 to 1992, and Arkansas Attorney General from 1977 to 1979.

Entity Discovery and Linking (EDL)

Source Corpus Document:
Hillary Clinton Not Talking About '92 *Clinton*-Gore Confederate Campaign Button..

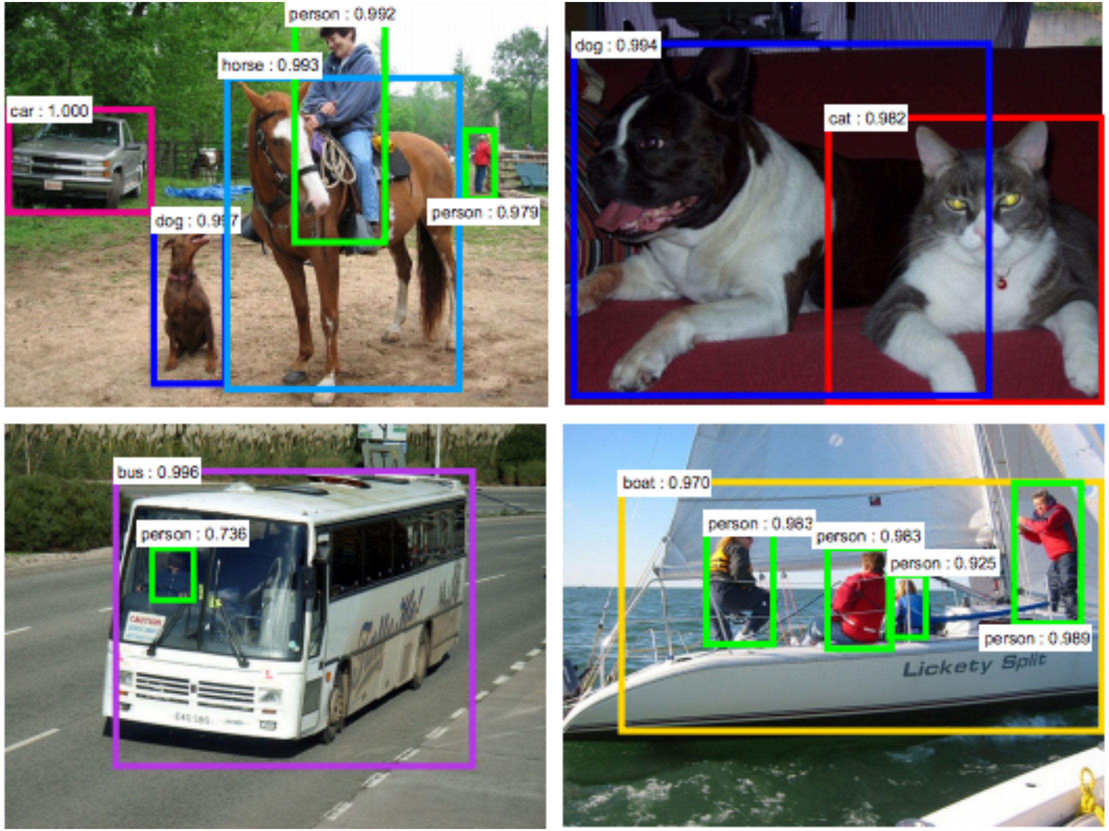
FreeBase entry:

Hillary Diane Rodham Clinton is a US Secretary of State, U.S. Senator, and First Lady of the United States. From 2009 to 2013, she was the 67th Secretary of State, serving under President Barack Obama. She previously represented New York in the U.S. Senate.

FreeBase entry:

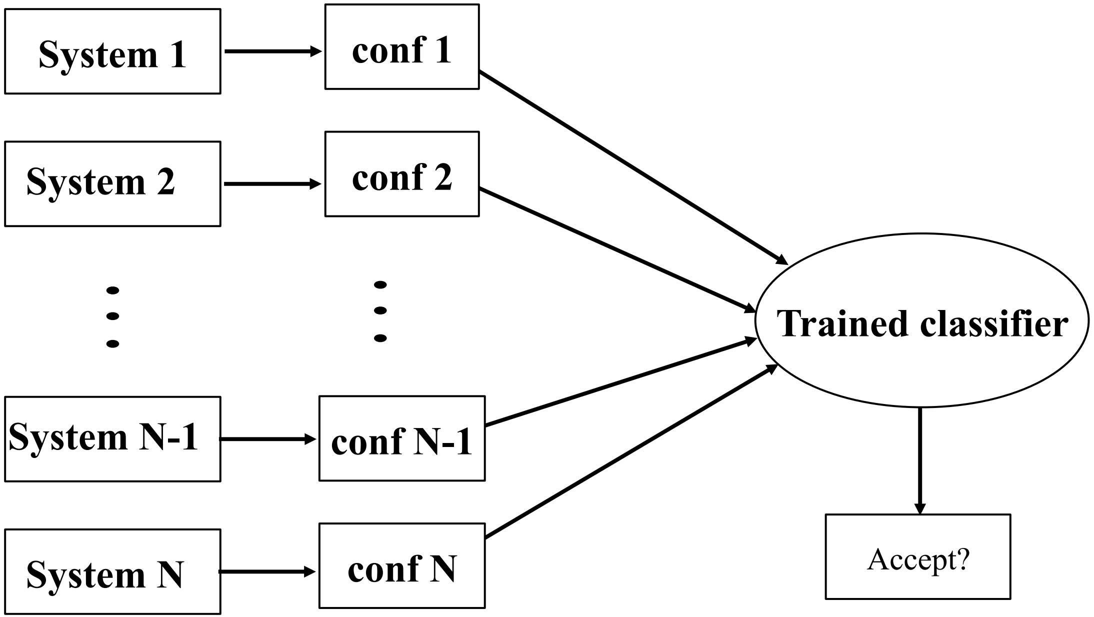
William Jefferson "Bill" Clinton is an American politician who served as the 42nd President of the United States from 1993 to 2001. Clinton was Governor of Arkansas from 1979 to 1981 and 1983 to 1992, and Arkansas Attorney General from 1977 to 1979.

ImageNet Object Detection



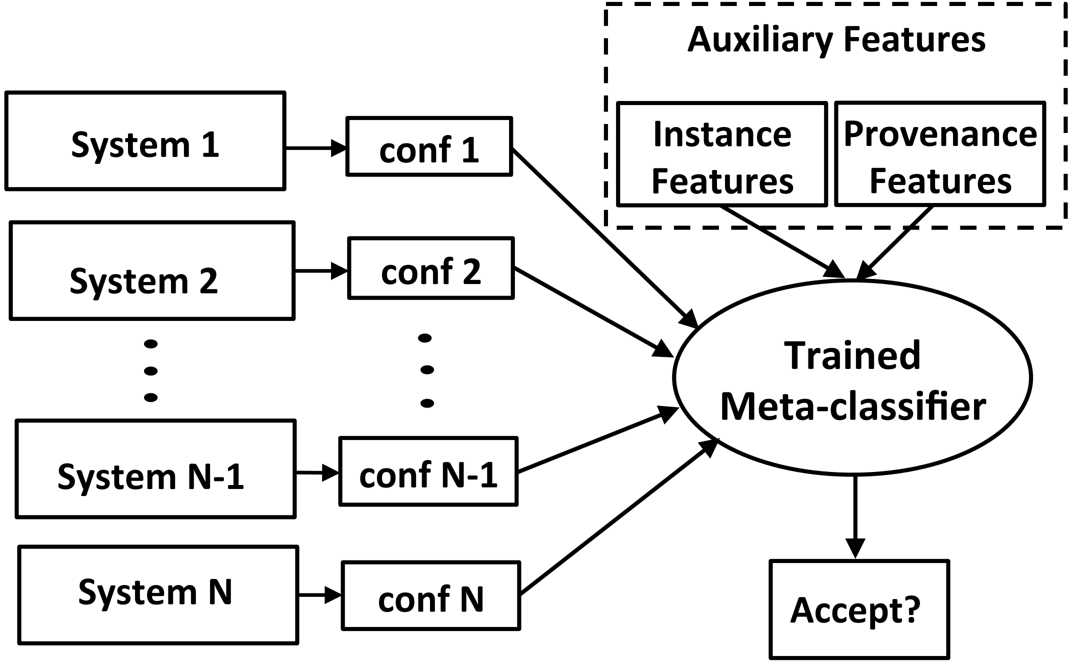
Ensemble Algorithms

- Stacking (Wolpert, 1992)



Stacking With Auxiliary Features (SWAF)

- Stacking using two types of auxiliary features:



Instance Features

Instance Features

- Enables stacker to discriminate between input instance types

Instance Features

- Enables stacker to discriminate between input instance types
- Some systems are better at certain input types

Instance Features

- Enables stacker to discriminate between input instance types
- Some systems are better at certain input types
- SF — slot type (per: age)

Instance Features

org: Microsoft
1. city_of_headquarters: 2. website: 3. subsidiaries: 4. employees: 5. shareholders: ⋮

Microsoft is a technology company, headquartered in Redmond, Washington that develops …
city_of_headquarters: Redmond **provenance:** **confidence score:** 1.0

Instance Features

org: Microsoft

1. city_of_headquarters:
2. website:
3. subsidiaries:
4. employees:
5. shareholders:

⋮

Microsoft is a technology company, headquartered in Redmond, Washington that develops …

city_of_headquarters:

Redmond

provenance:

confidence score:

1.0

Instance Features

org: Microsoft

1. city_of_headquarters:
2. website:
3. subsidiaries:
4. employees:
5. shareholders:

⋮

Microsoft is a technology company, headquartered in Redmond, Washington that develops …

city_of_headquarters:
Redmond
provenance:

confidence score:
1.0

Instance Features

- Enables stacker to discriminate between input instance types
- Some systems are better at certain input types
- SF — slot type (per: age)

Instance Features

- Enables stacker to discriminate between input instance types
- Some systems are better at certain input types
- SF — slot type (per: age)
- EDL — entity type (PER/ORG/GPE/FAC/LOC)

Instance Features

- Enables stacker to discriminate between input instance types
- Some systems are better at certain input types
- SF — slot type (per: age)
- EDL — entity type (PER/ORG/GPE/FAC/LOC)
- Object detection — object category and VGGNet's fc7 features

Provenance Features

Provenance Features

- Enables the stacker to discriminate between systems

Provenance Features

- Enables the stacker to discriminate between systems
- Output is reliable if systems agree on source

Provenance Features

- Enables the stacker to discriminate between systems
- Output is reliable if systems agree on source
- SF and EDL — document and offset provenance

Provenance Features

org: Microsoft

1. city_of_headquarters:
2. website:
3. subsidiaries:
4. employees:
5. shareholders:

⋮

Microsoft is a technology company, headquartered in Redmond, Washington that develops …

city_of_headquarters:
Redmond
provenance:

confidence score:
1.0

Provenance Features

org: Microsoft

1. city_of_headquarters:
2. website:
3. subsidiaries:
4. employees:
5. shareholders:

⋮

Microsoft is a technology company, headquartered in Redmond, Washington that develops ...

city_of_headquarters:
Redmond

provenance:

confidence score:
1.0

Provenance Features

org: Microsoft

1. city_of_headquarters:
2. website:
3. subsidiaries:
4. employees:
5. shareholders:

⋮

Microsoft is a technology company, headquartered in Redmond, Washington that develops ...

city_of_headquarters:
Redmond
provenance:

confidence score:
1.0

Provenance Features

- Enables the stacker to discriminate between systems
- Output is reliable if systems agree on source
- SF and EDL — document and offset provenance

Provenance Features

- Enables the stacker to discriminate between systems
- Output is reliable if systems agree on source
- SF and EDL — document and offset provenance
- Object detection — bounding box provenance

Document Provenance Feature

- For a given query and slot, for each system, *i*, there is a feature DP_i:
  - *N* systems provide a fill for the slot.
  - Of these, *n* give same provenance *docid* as *i*.
  - $DP_i = n/N$ is the document provenance score.
- Measures extent to which systems agree on document provenance of the slot fill.

Offset Provenance Feature

- Degree of overlap between systems' provenance strings.
- Uses Jaccard similarity coefficient.

$$OP(n) = \frac{1}{|N|} \times \sum_{i \in N, i \neq n} \frac{|substring(i) \cap substring(n)|}{|substring(i) \cup substring(n)|}$$

- Systems with different *docid* have zero OP

Offset Provenance Feature

Offsets	System 1	System 2	System 3
Start	8	1	18
End	29	16	29

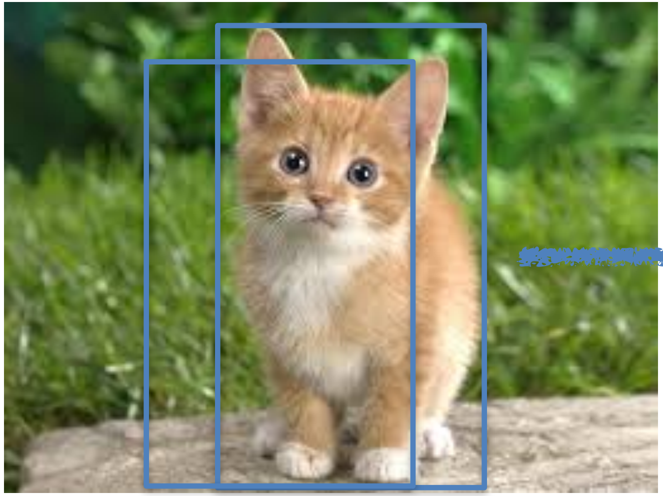
$$\overbrace{\text{Former President}}^{\text{System 2}}\ \underbrace{\text{Barack Obama}}_{\text{System 3}}$$

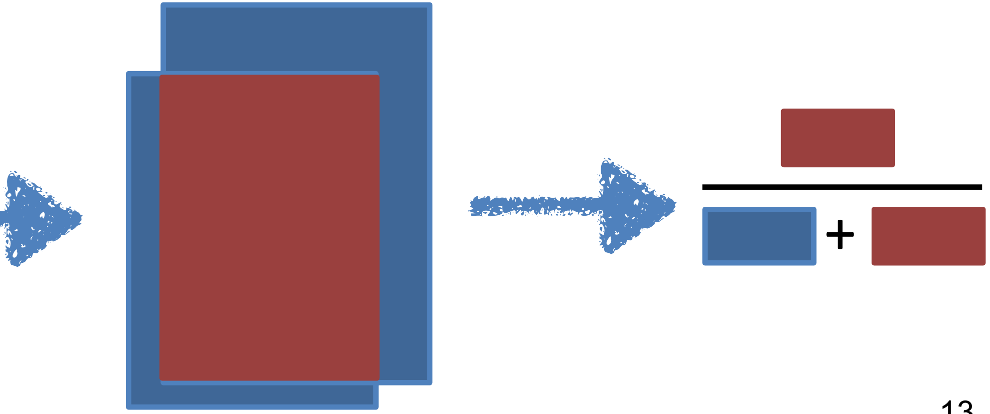
$$OP_1 = \frac{1}{2} \times \left(\frac{9}{29} + \frac{12}{21} \right)$$

Provenance Features

- Object detection — measure BB overlap

$$BBO(n) = \frac{1}{|N|} \times \sum_{i \in N, i \neq n} \frac{|\text{Area(i)} \cap \text{Area(n)}|}{|\text{Area(i)} \cup \text{Area(n)}|}$$





Baselines

- Mixtures of Experts (MoE) (Jacobs et al., 1991)
  - same intuition as instance auxiliary features
  - partition the problem into sub-spaces
  - learn to switch experts based on input using a gating network
- Oracle Voting
  - Vary the number of systems from 1 to n and use the one that results in best performance
  - Upper-bound on voting

Results

- 2016 SF — 8 component systems

Results

- 2016 SF — 8 component systems

Approach	Precision	Recall	F1

Results

- 2016 SF — 8 component systems

Approach	Precision	Recall	F1
Mixtures of Experts (Jacobs et al., 1991)	0.168	0.321	0.180

Results

- 2016 SF — 8 component systems

Approach	Precision	Recall	F1
Mixtures of Experts (Jacobs et al., 1991)	0.168	0.321	0.180
Oracle voting (>=4)	0.191	0.379	0.206

Results

- 2016 SF — 8 component systems

Approach	Precision	Recall	F1
Mixtures of Experts (Jacobs et al., 1991)	0.168	0.321	0.180
Oracle voting (>=4)	0.191	0.379	0.206
Top ranked system (Zhang et al., 2016)	0.265	0.302	0.260

Results

- 2016 SF — 8 component systems

Approach	Precision	Recall	F1
Mixtures of Experts (Jacobs et al., 1991)	0.168	0.321	0.180
Oracle voting (>=4)	0.191	0.379	0.206
Top ranked system (Zhang et al., 2016)	0.265	0.302	0.260
Stacking	**0.311**	0.253	0.279

Results

- 2016 SF — 8 component systems

Approach	Precision	Recall	F1
Mixtures of Experts (Jacobs et al., 1991)	0.168	0.321	0.180
Oracle voting (>=4)	0.191	0.379	0.206
Top ranked system (Zhang et al., 2016)	0.265	0.302	0.260
Stacking	**0.311**	0.253	0.279
Stacking + instance features	0.257	0.346	0.295

Results

- 2016 SF — 8 component systems

Approach	Precision	Recall	F1
Mixtures of Experts (Jacobs et al., 1991)	0.168	0.321	0.180
Oracle voting (>=4)	0.191	0.379	0.206
Top ranked system (Zhang et al., 2016)	0.265	0.302	0.260
Stacking	**0.311**	0.253	0.279
Stacking + instance features	0.257	0.346	0.295
Stacking + provenance features	0.252	0.377	0.302

Results

- 2016 SF — 8 component systems

Approach	Precision	Recall	F1
Mixtures of Experts (Jacobs et al., 1991)	0.168	0.321	0.180
Oracle voting (>=4)	0.191	0.379	0.206
Top ranked system (Zhang et al., 2016)	0.265	0.302	0.260
Stacking	**0.311**	0.253	0.279
Stacking + instance features	0.257	0.346	0.295
Stacking + provenance features	0.252	0.377	0.302
SWAF	0.258	**0.439**	**0.324**

Results

- 2016 SF — 8 component systems

Approach	Precision	Recall	F1
Mixtures of Experts (Jacobs et al., 1991)	0.168	0.321	0.180
Oracle voting (>=4)	0.191	0.379	0.206
Top ranked system (Zhang et al., 2016)	0.265	0.302	0.260
Stacking	**0.311**	0.253	0.279
Stacking + instance features	0.257	0.346	0.295
Stacking + provenance features	0.252	0.377	0.302
SWAF	0.258	**0.439**	**0.324**

Results

- 2016 EDL — 6 component systems

Approach	Precision	Recall	F1
Oracle voting (>=4)	0.588	0.412	0.485
Mixtures of Experts (Jacobs et al., 1991)	0.721	0.494	0.587
Top ranked system (Sil et al., 2016)	0.717	0.517	0.601
Stacking	0.723	0.537	0.616
Stacking + instance features	0.752	0.542	0.630
Stacking + provenance features	**0.767**	0.544	0.637
SWAF	0.739	**0.600**	**0.662**

Results

- 2016 EDL — 6 component systems

Approach	Precision	Recall	F1
Oracle voting (>=4)	0.588	0.412	0.485
Mixtures of Experts (Jacobs et al., 1991)	0.721	0.494	0.587
Top ranked system (Sil et al., 2016)	0.717	0.517	0.601
Stacking	0.723	0.537	0.616
Stacking + instance features	0.752	0.542	0.630
Stacking + provenance features	**0.767**	0.544	0.637
SWAF	0.739	**0.600**	**0.662**

Results

- 2015 ImageNet object detection— 3 component systems

Approach	Mean AP	Median AP
Oracle voting (>=1)	0.366	0.368
Best standalone system (VGG + selective search)	0.434	0.430
Stacking	0.451	0.441
Stacking + instance features	0.461	0.45
Mixtures of Experts (Jacobs et al., 1991)	0.494	0.489
Stacking + provenance features	0.502	0.494
SWAF	**0.506**	**0.497**

Takeaways

- SWAF produced SOTA on SF and EDL
- Significant improvements on ImageNet object detection
- Our approach is more robust than MoE in terms of number of component systems
- For object detection — works well for images with multiple instances of the same object