

# Winning Solution- Team Layer6 AI Open Images - Visual Relationship

Presented by

*Himanshu Rai, Jason Chang*

# Layer6 Challenge Team



Yichao Lu



Jason Chang



Himanshu Rai

# Three Stage Model

## First Stage



- Object detection with partial weight transfer

## Second Stage



- CNN models for visual and GBM models for spatial and semantic feature extraction.

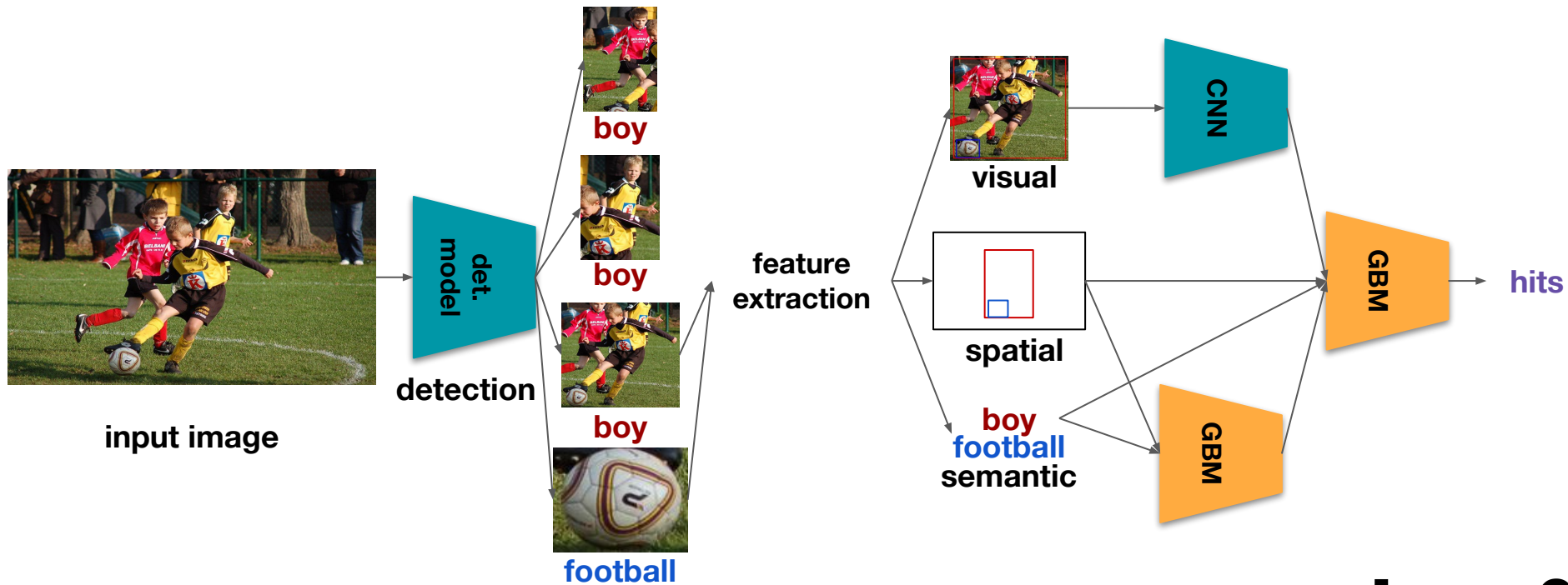
## Third Stage



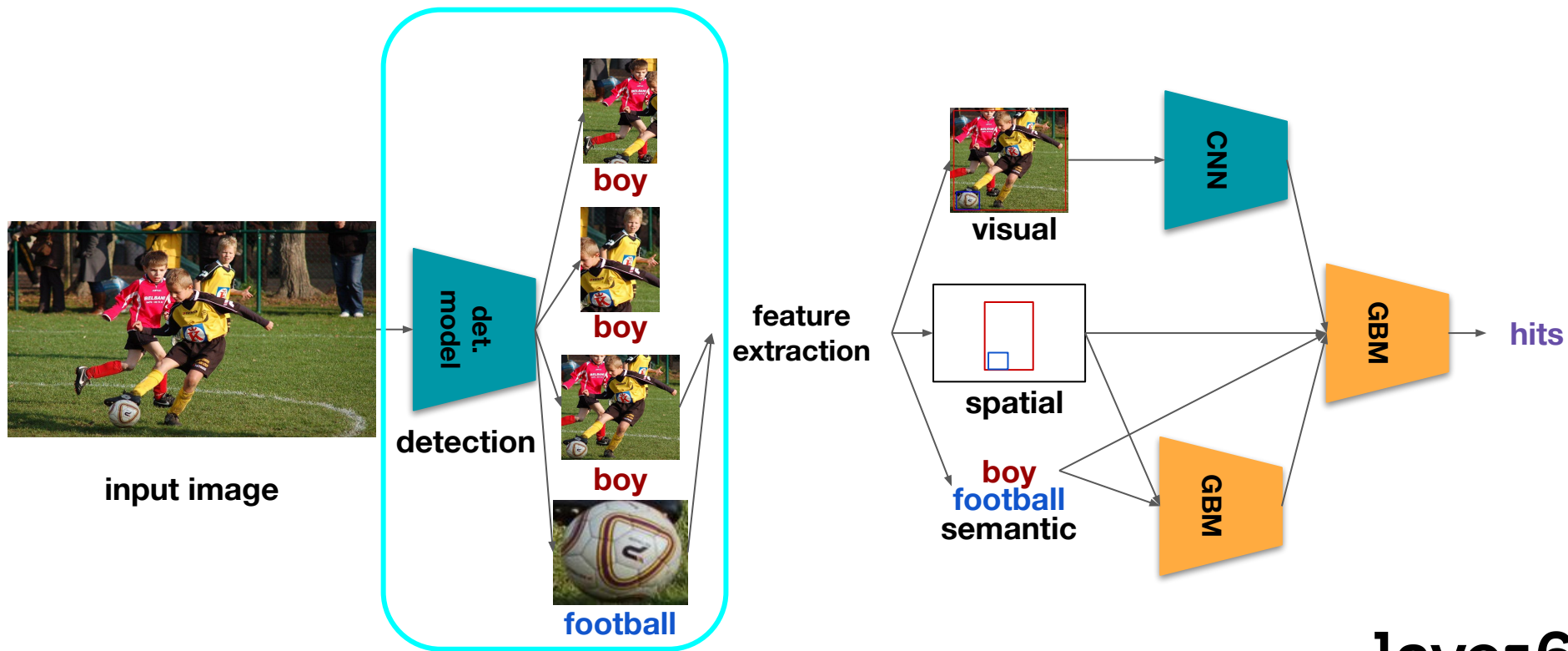
- GBM combining outputs from first two stages for final prediction.

**layer 6**

# Three Stage Model



# First Stage



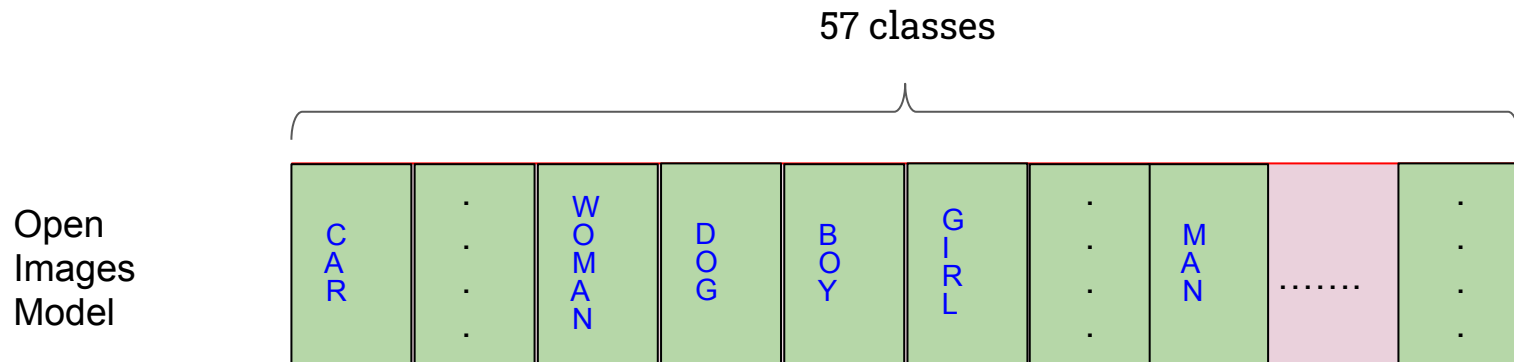
# First Stage - Object Detection

- Trained on **57** classes of the Visual Recognition Challenge
- Trained several SOTA models - Cascade RCNN, HRNet etc(COCO pretrained)
- Convergence observed to be slow
- Hard to obtain a high mAP (probably due to less instances of a class)

# Partial Weight Transfer

- We propose a very effective and economic way of Transfer Learning
- Want to leverage well trained high performance COCO models
- Map all possible classes between our dataset(57) and COCO(80)
- Two kind of matches - exact and approximate. Found 44 matches.

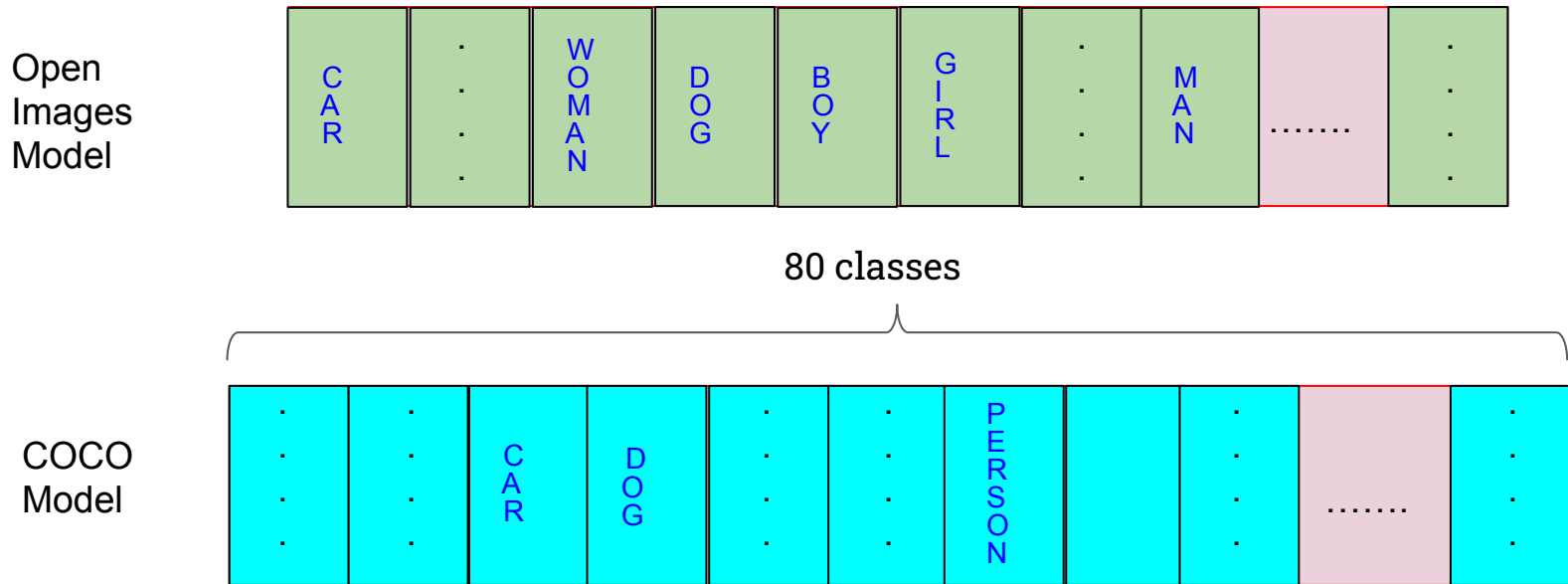
# Partial Weight Transfer



- Train a model on open-images dataset until convergence/ reasonable score

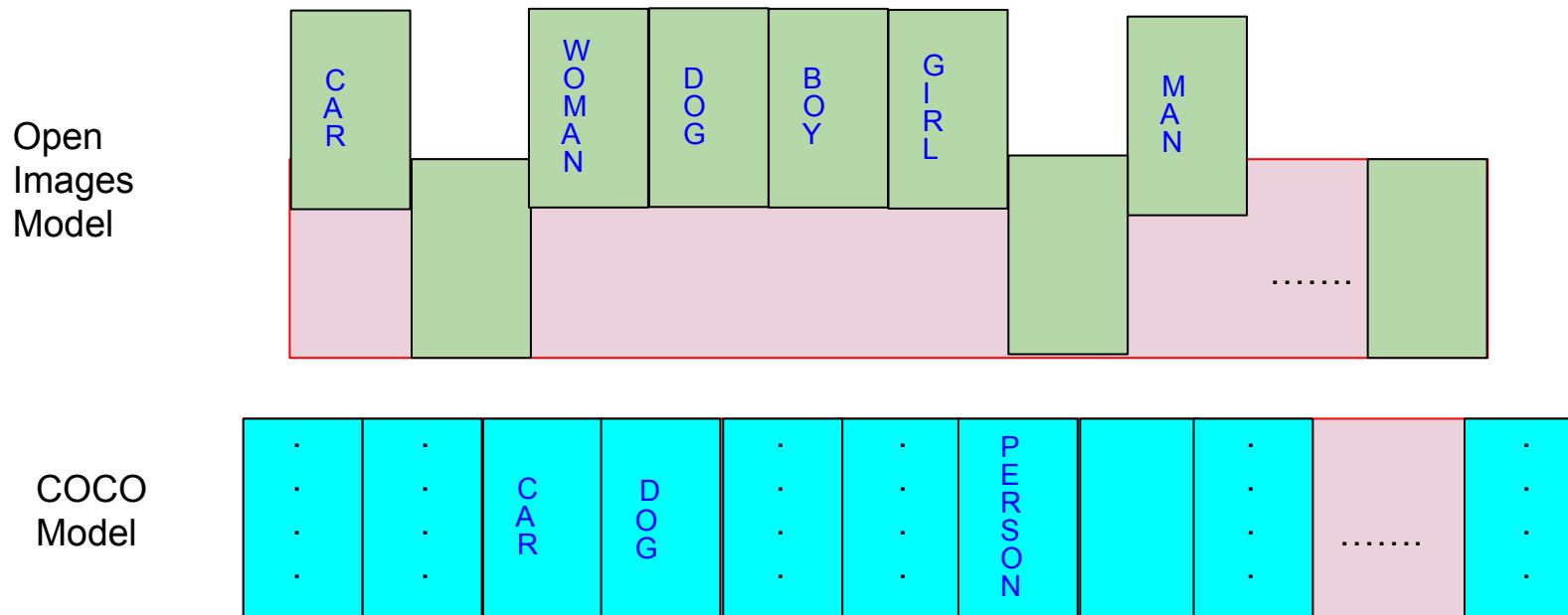


# Partial Weight Transfer



- Pick a high performance coco-pretrained model. Find all mappable classes.

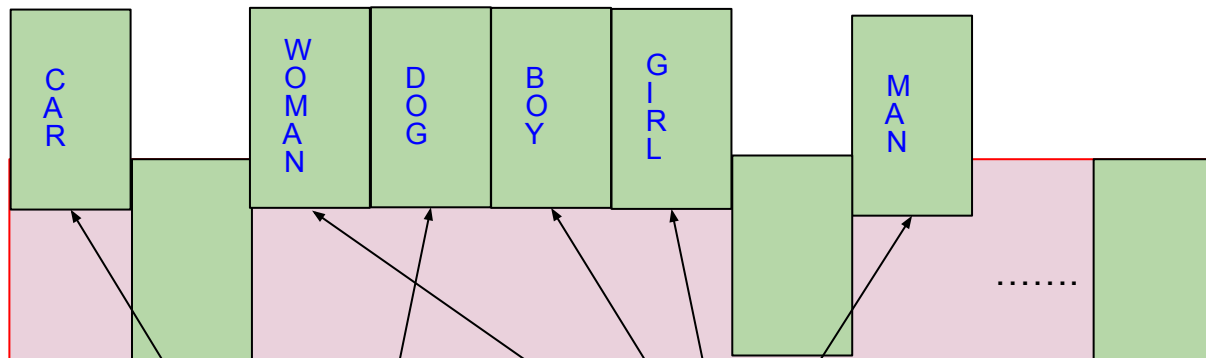
# Partial Weight Transfer



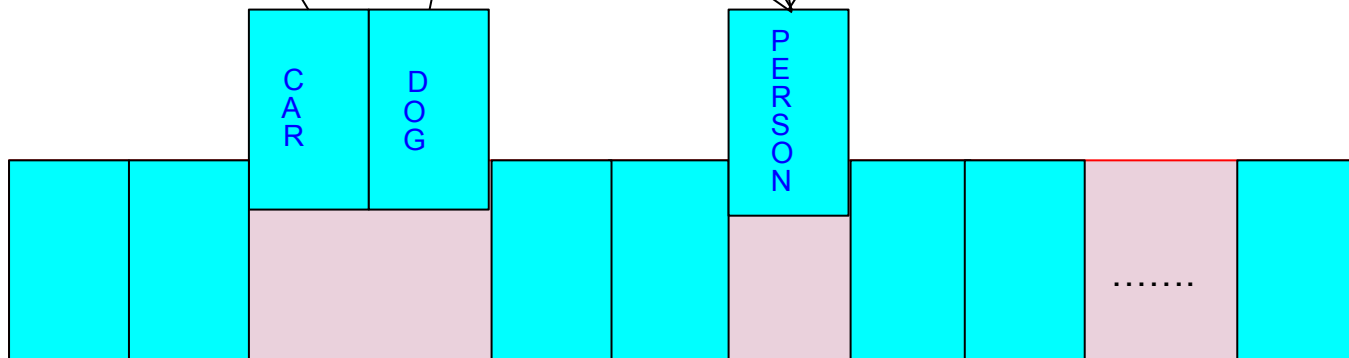
- Throw away classifier weights of all mapped classes

# Partial Weight Transfer

Open  
Images  
Model



COCO  
Model



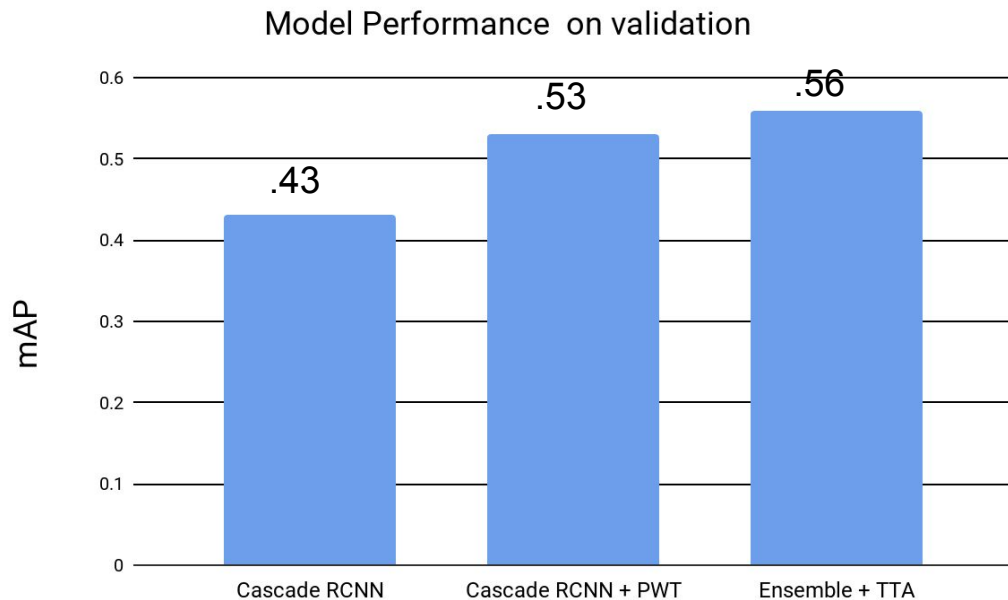
**layer 6**

# Partial Weight Transfer

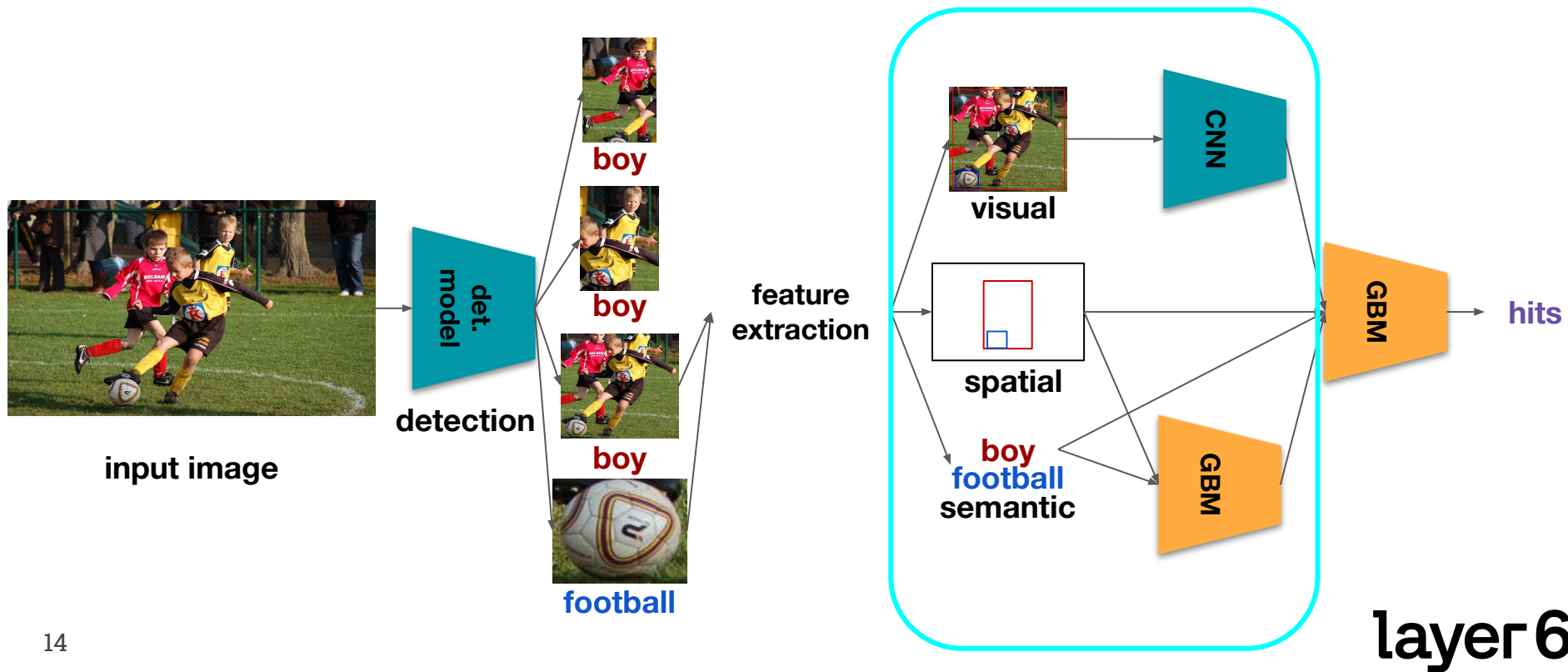
- Instead of throwing away classifier and regression heads, apply PWT
- After transferring classifier weights, transfer the regression as well as as well as the backbone weights.
- Fine tuning this way improved mAP by  $\sim .1$  in one day (over two TITAN V GPUs).

# Partial Weight Transfer

- Ensembling is done by NMS and combining scores weighted by individual mAP performances



# Second Stage



# Second Stage

- Some of the relations like **on**, **at** are highly spatial



# Second Stage

- Other relations like *holds*, *wears* etc have a strong visual dependence





# Second Stage

- Second stage consists of two models dealing with spatial, semantic and visual features
- The semantic features are highly relevant due to the unbalanced dataset
- Spatial and semantic relations - GBMs, Visual - CNNs

# Second Stage - Spatial and Semantic

- Extract four categories of spatial and semantic features:
  - **Object Spatial Features** - size of bounding boxes, absolute position etc.
  - **Object Semantic Features** - count and probability of class appearing in different relationships etc.
  - **Pairwise Spatial Features** - relative positions, IOU, distance between boxes etc.
  - **Pairwise Semantic Features** - probability and count of co occurrence etc.

# Second Stage - Spatial and Semantic

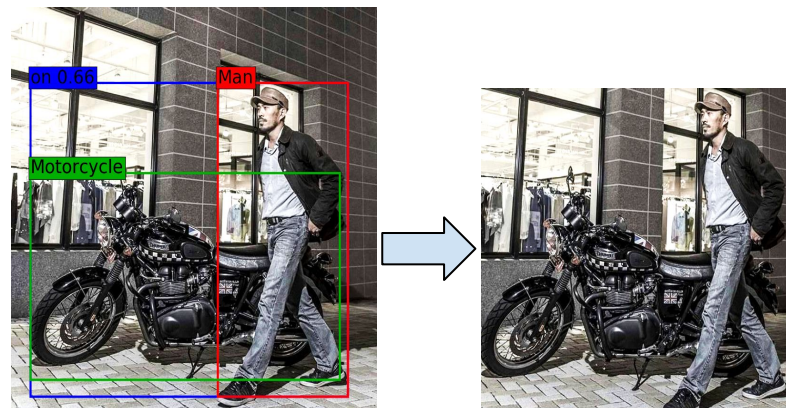
- Trained separate GBM models for each relationship using binary classification objective
- We found it to perform better than training a single model with multiclass classification objective
- We form positive and negative pairs and train the models on the groundtruth

# Second Stage - Visual

- Filter out detections from first stage using empirically determined threshold
- Form all possible valid pairs from the detections
- For all such pairs, prepare input and pass through the CNN model to get predictions.

# Pixel Filtering Process Flow

Spatio-Semantic model

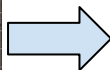
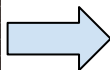
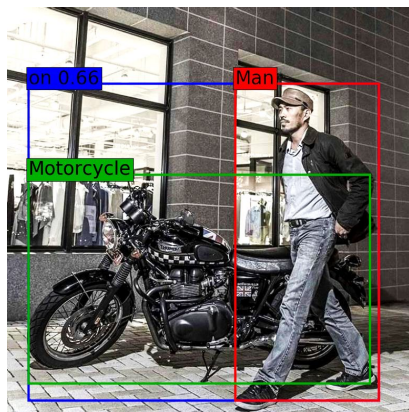


- For a pair, crop out the relation box(enclosing box)

Man 'on' Motorcycle  
Model probability = 0.66

# Pixel Filtering Process Flow

Spatio-Semantic model

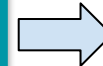
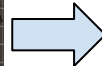
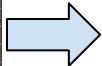
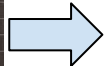
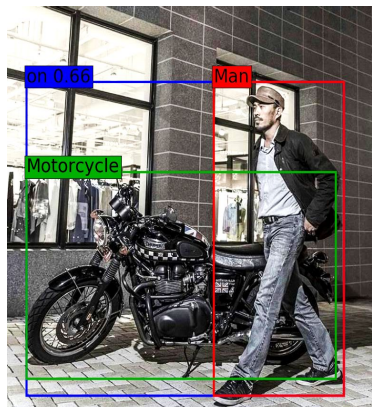


Man 'on' Motorcycle  
Model probability = 0.66

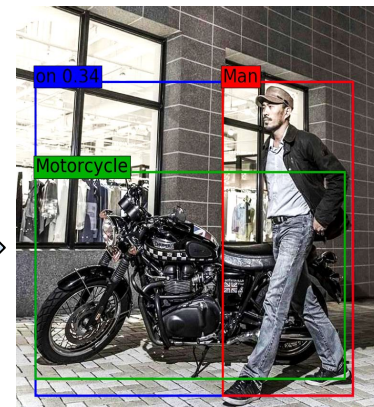
- Remove pixels not belonging to either of the two boxes.

# Pixel Filtering Process Flow

Spatio-Semantic model



Visual model



Man 'on' Motorcycle  
Model probability = 0.66

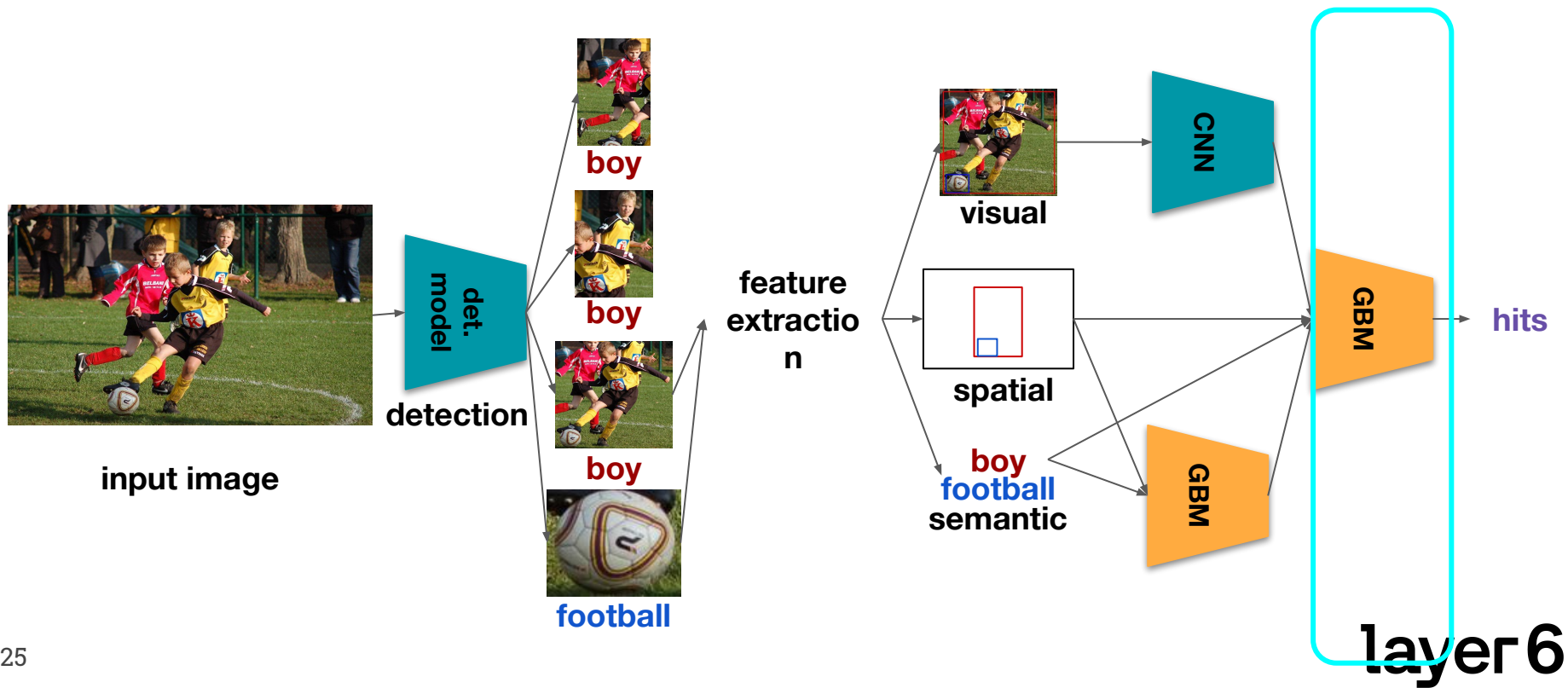
Man 'on' Motorcycle  
Model probability = 0.34

# Second Stage - Visual

- Reuse the object detector backbone.
- Predict relationship classes using the extracted visual features.
- Each relationship gets its own CNN model (binary prediction model).



# Third Stage



# Third Stage - Aggregation Model

- Used GBMs for aggregating predictions from the second stage
- Apart from the predictions from spatio-semantic and visual models, we also use the spatial and semantic features
- Used different splits for training the second and third stages






# Third Stage - Aggregation Model

| <b>Relationship</b> | <b>Spatial</b> | <b>Visual</b> | <b>Averaging</b> | <b>Aggregation</b> |
|---------------------|----------------|---------------|------------------|--------------------|
| at                  | 0.37           | 0.35          | 0.35             | 0.42               |
| plays               | 0.49           | 0.58          | 0.55             | 0.59               |
| Interacts_with      | 0.42           | 0.42          | 0.41             | 0.44               |
| inside_of           | 0.31           | 0.35          | 0.32             | 0.37               |
| hits                | 0.58           | 0.47          | 0.58             | 0.61               |

# The “is” model

- “*Is*” : a special case where subject is a class and object is an attribute
- Use a pure detection model for this class
- Forming all possible combinations of *object-is-attributes* gives 42 classes
- Lack of sufficient training instances necessitated use of PWT strategy to obtain good performance in reasonable time.

# Leaderboard Summary

| # | △pub | Team Name        | Notebook | Team Members   | Score ⓘ |
|---|------|------------------|----------|--|---------|
| 1 | —    | Layer6 AI        |          | <br>★★★★★ | 0.40801 |
| 2 | —    | tito             |          | <br>★★★★★ | 0.38818 |
| 3 | —    | Very Random team |          | <br>★★★★★ | 0.37853 |
| 4 | ▲1   | [ods.ai] n01z3   |          | <br>★★★★★ | 0.36597 |
| 5 | ▼1   | Ode to the Goose |          | <br>★★★★★ | 0.34779 |

# Thank you!

<https://layer6.ai>

