Three-stage Visual Relationship Detector

Yichao Lu *
Layer6 AI
yichao@layer6.ai

Cheng Chang *
Layer6 AI
jason@layer6.ai

Himanshu Rai *
Layer6 AI
himanshu@layer6.ai

Guangwei Yu
Layer6 AI
guang@layer6.ai

Maksims Volkovs
Layer6 AI
maks@layer6.ai

Abstract

We present our solution to the Open Images 2019 - Visual Relationship challenge, the largest challenge of its kind to date with nearly 9 million training images. The task requires detecting relationships connecting two objects. These relationships can be categorized into three broad categories, human-object, object-object and object-attribute. Our solution consists of two models - a detection model and a relationship model. The two models are trained separately on distinct tasks and assimilated into a pipeline later during inference. We present two ideas crucial to our approach, partial weight transfer for object detection, and multi-feature aggregation for relationship detection. The proposed approach outperforms the second place team by over 5% relatively on the held-out private leaderboard.

1. Introduction

Visual relationship detection is a core computer vision task that has gained a lot of attention recently. The task comprises of object detection followed by visual relationship detection to detect relationship predicate over a pair of objects. Successfully solving visual relationship involves understanding useful semantic information in a scene, which is considered a challenging task. Object detection alone fails to capture useful semantic cues present in the relationships, as it ignores the connection between objects in the scene. The Open Images 2019 Visual Relationship challenge introduces a uniquely large and diverse dataset of annotated images designed to tackled this challenge, and provides a public benchmark for visual relationship detection.

In this paper we present our solution which won over the second place team by over 5% on the held out private dataset. Specifically, we present our partial weight transfer technique, henceforth referred to as PWT approach in object detection, and a two-stage visual relation detection pipeline.

1.1. Challenge Framework

The Open Images 2019 Visual Relationship challenge is based on the Open Images V5 dataset [3], which contains 9 million images annotated with image-level labels, object bounding boxes, object segmentation masks, and visual relationships.

The task is to detect pairs of objects and the associated relationships. The relationships include human-object relationships (e.g. "man holding camera"), object-object relationships (e.g. "spoon on table"), and object-attribute relationships (e.g. "handbag is made of leather"). Each of the relationships can be expressed as a triplet, written as a pair of objects connected by a relationship predicate (e.g. "beer on table"). Visual attributes are also triplets, where an object is connected with an attribute using the relationship "is" (e.g. "table is wooden"). The challenge involves 329 unique triplets, which spans 57 different object classes, 5 attributes, and 10 predicates.

2. Approach

Our pipeline consists of three stages. In the first stage, given a scene, an object detection model finetuned for the visual relation challenge generates bounding boxes along with their associated confidences. In the second stage, two separate models (GBM and CNN) are used to model spatial, semantic and visual features. Finally, a third stage model, which takes the predictions from the first two stages as its input, learns to generate the final prediction. Figure 2 summarizes our approach depicting the three stages.

In this section, we describe our approach in each of the three stages.

*Authors contributed equally to this work, and their order is determined randomly.

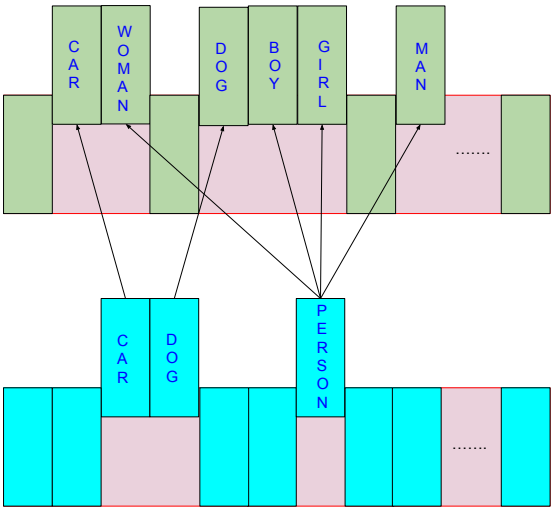

Figure 1. Partial Weight Transfer on classifier heads. Top box represents the classifier head trained on open images and lower box represents classifier head of a coco pretrained model

2.1. Object Detection with Partial Weight Transfer(PWT)

Object Detection Results (mAP)		
Model	Validation	LB
Cascade RCNN(baseline)	0.43	0.048
baseline + PWT	0.53	0.065
Faster RCNN-HRNet+PWT	0.51	0.062
Blend and TTA	0.56	0.068

Table 1. Validation and Leaderboard (LB) performance for detection models evaluated on the Open Image Object Detection challenge. LB score is obtained by submitting to the 2019 Open Image Object Detection challenge using only overlapping classes from visual relationship. The mAP score is the detection mAP. TTA incidates test time augmentation.

Training an object detection model in a reasonable amount of time on a large dataset requires significant resources. Our goal is to train a competitive object detection model to support downstream visual relation task with limited resource. In order to achieve this, we propose a partial weight transfer (pwt) strategy. The main idea is to transfer as much information from models trained on well-established datasets for general object detection problem, where models can achieve very impressive performance, for instance coco pretrained models. Minimal fine-tuning is then performed on the task specific dataset.

Transfer Learning is a popular and economic approach for improving generalization on specific tasks. However, choosing which weights to keep or discard can have a significant impact on the performance. The common approach is to take existing faster/cascade RCNN [1] models trained on large general datasets where a high performance is observed. Then the classification and regression heads are replaced with random initialized weights to train task-specific detectors. However, we observed that finetuning the network this way still takes significant amount of time to converge, and could not achieve very high performance.

We define two datasets $\mathcal{S}_{\text{gen}}$ and $\mathcal{S}_{\text{task}}$. $\mathcal{S}_{\text{gen}}$ denotes a general dataset where a model f_{gen} was trained with high performance for some task. Our goal is to train another model f_{task} on a related task over the dataset $\mathcal{S}_{\text{task}}$. The related task in our case is object detection over a different but overlapping set of classes compared to the original object detection task solved by f_{gen}. There should be some correlation between the weights of the common or related classes of the classifier heads of f_{gen} and f_{task}. Following this intuition, it should be possible to partially transfer the common class weights from one model's classification head to the other model's classification head and recover the performance. Figure 1 demonstrates this process. In this example, the vectors in the fully connected layer of f_{gen} associated with car and dog are copied to their matching position in f_{task}. The weights for a more general class, person, is copied from f_{gen} to the corresponding weights in f_{task} for four corresponding fine-grained classes: woman, boy, girl, and man.

To train f_{task}, we re-write the multi-label cross-entropy loss function as

$$\mathcal{L}_{cls} = \sum_{n=1}^{N}\sum_{k=1}^{K} y_k^{(n)} log\left[\frac{z^k}{\sum_{j=1}^{K} z^j}\right] \tag{1}$$

$$z^k = \begin{cases} e^{(\omega_k^{\text{gen}})^\top x^n + \omega_{k0}^{\text{gen}}} & \text{if} \quad k \in \mathcal{S}_{\text{gen}} \\ e^{(\omega_k^{\text{task}})^\top x^n + \omega_{k0}^{\text{task}}} & \text{if} \quad k \in \mathcal{S}_{\text{task}} \end{cases}$$

Consider the case where we are trying to plug in the shared class weights from a coco pretrained model to an open images trained model. In the above equation, ω^{gen} represents the weights of the common classes taken from the coco pretrained models and ω^{task} are the weights from the model trained on open-images with 57 classes. K is the total number of classes and N is the total number of training examples. Apart from transferring the shared fully connected weights, it is equally important to transfer the backbone weights from the coco model to the open model. This ensures the compatibility of the classifier weights with the backbone weights.

The detection model can be independently evaluated following the evaluation protocol [1] from the Open Images

[1] https://www.kaggle.com/c/open-images-2019-object-detection/overview/evaluation

2019 Object Detection challenge held in parallel to the Visual Relationship challenge on Kaggle. Table 1 shows the detection performance by submitting only the classes from the visual relationship challenge. In Table 1,we clearly observe the gain we achieve over the baseline by using our PWT strategy, which transfers across the private dataset.

2.2. Relationship Model with Spatial-semantic and Visual Features

Given the bounding boxes predicted by the object detection model, the goal of the relationship model is (1) to detect ordered pairs of objects that are related to each other, and (2) to determine the types of relationships (predicates) between these pairs of objects. The second stage visual relationship model uses a combination of tree-based gradient boosting models and convolutional neural networks.

The task of visual relationship detection requires models to simultaneously learn spatial, semantic, and visual features to make accurate predictions. In practice, we find that tree-based gradient boosting models (GBMs) are effective at learning spatial and semantic features, and that convolutional neural network (CNN) based models are superior at capturing visual features. Thus our second stage visual relationship model comprise both tree-based and CNN-based models to take advantage of spatial, semantic and visual features.

2.2.1 Learning spatial and semantic features

Spatial information refers to position-related features such as the relative position of the object in the image and the relative position of a pair of objects with each other. Spatial information can be very important in determining visual relationships in the sense that (1) objects that are far away from each other are unlikely to have a relationship, and (2) the relative position of the two objects can be very informative when determining relationships such as "on" or "under". On the other hand, semantic information can capture the probability of co-occurrence of two objects or the probability of two objects having a certain type of relationship.

The input features to our GBMs include:

1. Object spatial features: We encode features such as the (relative and absolute) position of the object in the image and the size of the object.

2. Object semantic features: We include features such as the probability that an object is involved in a relationship.

3. Pairwise spatial features: This group of features are the most important ones. We encode the relative position of two objects with each other, the iou of the two objects, and the euclidean distance of their centers, etc.

4. Pairwise semantic features: Similar to pairwise spatial features, in pairwise semantic features, we summarize the frequency that two objects appear together, and the frequency that a certain type of relationship is associated with the object pair.

We consider two alternatives for defining the GBM training objective. The first one is to train a single GBM for multiclass classification over all the possible relationships (including an additional class "None", which denotes that there is no relationship between two objects). The second one is to train separate GBM models for every relationship type with a binary classification objective. For example, for the relationship type "under", we find all pairs of objects that can possibly form an "under" relationship. Then we label the ones that are present in the groundtruth visual relationships as positive samples, and the other ones as negative samples. The advantage of the first option is that it is more computationally efficient as it only requires one model for all types of relationships. However, in practice, we find the second option to perform consistently better than the first one. We presume that this is because allowing the model to focus on a particular type of relationship eases the optimization.

2.2.2 Learning visual features

Models that rely solely on spatial and semantic features can be accurate in most cases. However, there exist certain circumstances where these features can lead to wrong results. For example, models without visual features cannot distinguish a man sitting on a motorcycle from a man standing in front of a motorcycle; see Figure 3.

We use CNN-based architecture for learning visual features. We prepare the input to our proposed model as follows. Given two objects of interest in an image, in order to minimize the distraction by the background and other objects in the same image, we first crop the image so that the cropped frame only contains the union of the two objects of interest. Then for each pixel in the cropped image that does not belong to the bounding box of either object, we turn it into background by making it black. In this way, we expect the model to be informed of which two objects it should focus on.

Similar to the GBM model, for our CNN-based model for learning visual features, training separate models for dealing with individual relationship types yields better results than training with a multiclass objective.

2.3. Multi-feature Aggregation

The final stage takes the predictions from the second stage visual relationship model as features, and combines them to make the final prediction. A straightforward approach to combining the predictions from the visual rela-

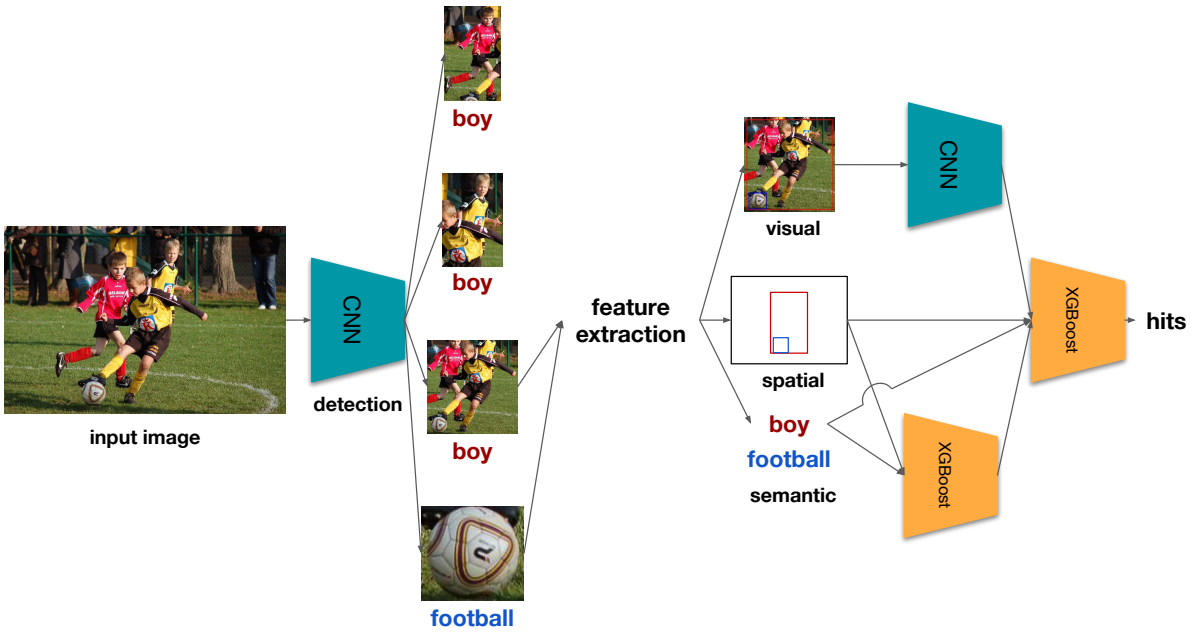

Figure 2. The overall architecture for our proposed relationship detection model.

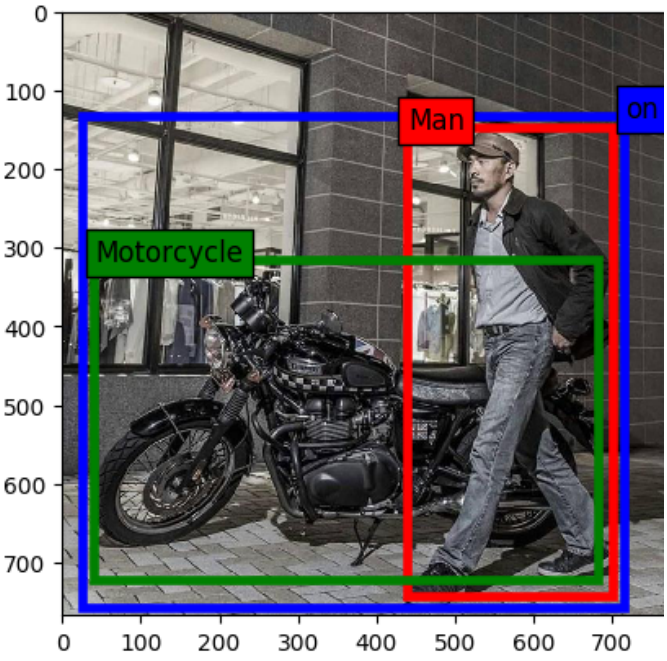

Figure 3. A failure case without visual features.

tionship model is to average the predictions from the first stage. However, such approach is suboptimal since the model needs to know when it is appropriate to trust predictions made by certain kind of models. Therefore in the last stage we train another GBM that not only takes spatial and semantic features as input, but also includes the predicted probabilities from the second stage as its features. The training objective for the last stage GBM is exactly the same as that of the second stage GBM. We begin by splitting the official training set into two parts. We train the first stage models on the first split of the official training set, and use the official validation set for validation. The second stage model is trained on the second split of the official training set, and again validated on the official validation set.

3. Experiments

We present our experiment setting and results on the Open Images V5 dataset, which is part of the Open Images 2019 Visual Relationship challenge hosted on Kaggle.

3.1. Balancing Classes

The class distribution in this dataset is very imbalanced, See Figure 4. This implies that the model does not see sufficient instances of the infrequent classes and biased towards the popular classes. In order to fix this problem, we partitioned the dataset into several subsets. For each subset we sample a maximum of 10k instances from each class. If a class had less than 10k instances, we retained all the instances in the subset. Finetuning the network this way gave us a substantial boost in the performance. The gains begin to diminish after training with 2 subsets.

3.2. Implementation detail

In this section we describe the 3 stages in our pipeline in detail: object detection, visual relation, and relationship

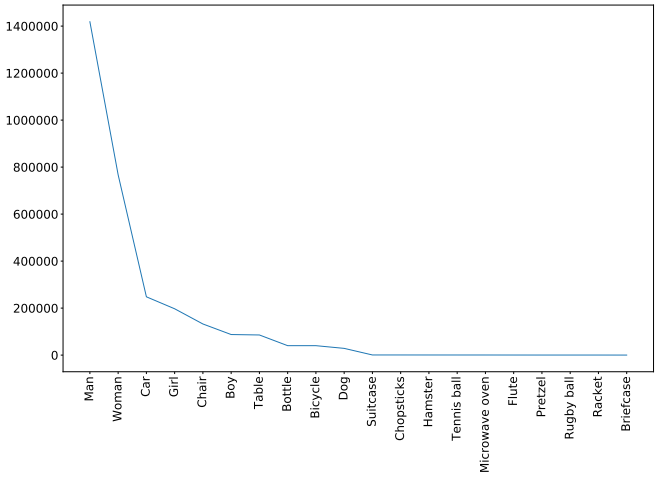

Figure 4. Sampled Imbalanced Class Distribution

fine-tuning.

Detection model For the first stage object detection, we trained an ensemble of cascade RCNN and HRNet[5][6] detection networks using our PWT approach. We chose COCO[4] as the general dataset $\mathcal{S}_{\text{gen}}$, while the task specific dataset $\mathcal{S}_{\text{task}}$ is the challenge dataset which consist of 57 classes. By adopting the PWT approach, our model not only trained to competitive detection performance quickly, but we also observed improved generalization for the 57 classes relevant for relationship detection task.

On inspection, we found that out of the 57 classes is $\mathcal{S}_{\text{task}}$, 44 are a direct/indirect subset of the 80 from $\mathcal{S}_{\text{gen}}$. In order to leverage this, we retain the weights of these 44 classes with the following steps. First we finetune a pretrained network with a 57 way classifier head, which is initialized from random. Henceforth we refer to this network as the open pretrained net. After few iterations of training, we apply our PWT approach, by copying the classification head weights for these 44 classes from the COCO pretrained network and replaced the corresponding positions of classification head of the open pretrained net with the COCO weights. We also switched the backbone of the open pretrained net with the backbone from the COCO pretrained model. This ensures the quick recovery of the performance for the 44 common classes. Forcing the network to reuse the partial COCO-weights in this way lead to an accelerated convergence as well as higher mAP.

During training, we feed the groundtruth bounding boxes as the input to our visual relationship model. We have also tried feeding the predicted bounding boxes by our object detection model as input. We expected the latter approach to perform better. The rationale is that, since the model is exposed to the same type of bias (e.g., shifted bounding boxes or mislabeled classes) during training and inference, the model should learn to correct the errors made by the object detection model, and also yield more robust predictions. However, such approach does not perform well in our experiments. We presume that this is because using the predicted boxes during training introduced bias in the data that the model consumes, which eventually misled the model.

In addition, we apply a few heuristic-based post-processing steps for the purpose of improving the mAP metric. For example, a wine glass can usually only be held by a single person. However, the model tends to predict that the glass is held by every single person that is close to it. This can potentially hurt the mAP since it introduces false positives to the predictions. In the light of this, our apply rule-based post-processing, e.g., setting confidence thresholds to discard false predictions, to refine the predictions given by our model. We find that these rule-based tricks consistently improved our model's performance.

Special "is" relationship For the Open Images 2019 challenge, the problem contains a special "is" relationship where the predicate is unary with respect to the object and indicate a specific attribute (e.g. "wooden", "plastic" etc). A separate single stage pipeline consisting only of the object detection approach using the PWT approach is applied to generate prediction for the "is" model. The most intuitive approach is to use the trained detection model for object classes that can have an "is" relationship (e.g. chair, bottle etc). We can then train an CNN classifier to detect the correct attributes (e.g. "wooden" chair). We found this model to be a decent baseline giving us about 0.06 score on the leaderboard. The performance of this model indicates that further improvements can be achieved by co-learning the detection and attribute classification tasks. In order to achieve that, one can form all possible pairs of object-attributes and use them as distinct classes in a detection model. The concern with this approach is the lack of sufficient data to train a strong detection model. This problem can be easily solved by using our PWT strategy. We used this strategy to transfer the weights from one of our base detection models and then finetuned on the available data. For instance, both "wooden" and "plastic" pianos get the piano classifier weight from our base detection model as well as its backbone. Finetuning this way, we immediately observed a much better performance of 0.08 on the leaderboard. The detector model we used here was a single cascade RCNN detector model with 43 classes. The first 42 classes are all the possible object-attribute pairs and the last class is used for negative samples.

Relationship model To capture spatial and semantic features, we use the XGBoost library [2] due to its excellent performance in our experiments. We begin by mapping the groundtruth bounding boxes to the bounding boxes in the

Relationship	XGBoost	CNN	Second Stage
at	0.37	0.35	**0.42**
on	0.35	0.31	**0.36**
holds	0.49	0.52	**0.54**
plays	0.49	0.58	**0.59**
interacts_with	0.42	0.42	**0.44**
wears	0.38	0.41	**0.42**
inside_of	0.31	0.35	**0.37**
under	0.20	0.20	**0.25**
hits	0.58	0.47	**0.61**

Table 2. The performance of different models in terms of mean Average Precision (mAP) on relationships detection [3] on the validation set.

groundtruth visual relationships. Then we iterate through every possible ordered pairs of objects. Pairs of objects that are not present in the groundtruth visual relationships are labelled as "negative samples".

The network architecture of our CNN model is relatively simple. We take the backbone network of the pretrained object detection model, and add a few fully-connected layers with the ReLU activation function. We additionally include spatial and semantic features such as the position of the objects and the predicted classes of the objects in the penultimate layer of the network. The activation for the final layer is sigmoid, which squashes the predicted value to the interval of (0, 1) so as to represent the probability that a particular relationship type exists.

4. Experiments and Results

Ensemble detector We tried several strategies to ensemble the detectors. The one that worked the best was the weighted NMScite or explain strategy that we describe here. We took the prediction files from different detection models and weighted the confidences of the boxes using the leaderboard performances of these models. We then performed the traditional NMS using these boxes with one minor difference. At the time of performing NMS, we summed up the weighted confidences from all the boxes instead of choosing the most confident box. Apart from summing, we also tried taking the maximum of all confidences, but found out that the former works best. The results for the single models as well as the ensembles is show in Table 1.

Relationship detection The result for the first stage and second stage models are presented in Table 2. We can have several observations from the validation results. Firstly, in the first stage, XGBoost is able to yield better results on relationships such as "on" and "hits", while CNN is able to get better performance on relationships such as "wears" and "plays". This suggests that under most circumstances relationships such as "on" and "hits" can be inferred using spatial and semantic clues, while relationships such as "wears" and "plays" requires more visual reasoning. Secondly, the second stage model is consistently better than the first stage models, across all types of relationships. This indicates that the second stage model is able to learn to aggregate the predictions from the first stage models, and to combine spatial, semantic, and visual features to make final predictions.

Team	Public LB	Private LB
Layer6 AI	0.46382	0.40801
tito	0.44079	0.38818
Very Random team	0.42894	0.37853

Table 3. Final results on the private leaderboard.

5. Conclusion

We present our first place solution to the Open Images 2019 - Visual Relationship challenge. Our approach consists of two parts: object detection and visual relationship. In object detection, we proposed a novel partial weight transfer approach, which enabled us to achieve high mAP with low computational cost in a small amount of time. In visual relationship, we introduce a two-stage approach which combines spatial, semantic and visual features. The resulting model is over 5% better than the second team's model in terms of performance on the private leaderboard as shown in Table 3 and our gains generalize between the public and the private leaderboard.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

[2] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[3] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018.

[4] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[5] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[6] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui

Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *CoRR*, abs/1908.07919, 2019.