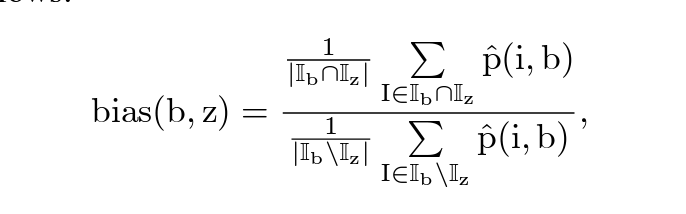
Questions:

* 
* Adapting this definition of bias to the object detection task — p/r/map ratios? What do they mean
* Currently we are using the validation set— can we use another test set (bc it has labels, unlike coco), as long as we focus on the object pairs?
* Score?

Office hours questions and notes

Our approach:

* Validate base model on large val set→ get class that does badly (knife)
* Extract training images of knives, and copy them + add augmentation techniques
* Fine tune base model on this dataset
* Validate fine tuned model on large val set
* Compare results
* Do this for either more classes and/ or different augmentation techniques
* Visualize the failed images, data augmentation, and results

<https://github.com/ultralytics/yolov5/discussions/10469>

* How is object detection with and without co occuring objects?
  + Ex: skateboard without vs with people
  + How to tell when two things are affecting each other? (adapt for object detection vs classification?)
    - Do we end up with different sets of objects biasing each other?

Get numbers for all of them — take pair of images as given

3-5 to go deeper into

* Figure out which data augmentation task we want to do based on baseline model performance on validation set
* Perform data augmentation in the train
* Need: test dataset we can use to compare baseline and regular accuracy
  + Format the coco test dataset OR
  + Choose a different dataset that has our subtask
* If precision is much higher than recall: Model is too conservative, making predictions only when very confident. This often means you can gain performance by adjusting confidence thresholds or focusing on recall-improving techniques
* If recall is much higher than precision: Model is too aggressive, making many predictions. Could benefit from hard negative mining or precision-focused techniques
* If mAP50 is much higher than mAP95: Model is good at rough localization but struggles with precise boundaries. This suggests focusing on techniques like enhanced augmentation for boundary precision
* Check the confusion matrix between classes
* Look for systematic misclassifications where one class is frequently confused with another; These patterns can reveal underlying feature similarities that need better discrimination
* Examine the training examples for classes with poor performance
* Look for issues like:
  + High intra-class variation (objects looking very different within same class)
  + Inter-class similarity (objects looking similar across different classes)
  + Challenging environmental conditions (poor lighting, occlusion, etc.)
* Look for classes where small changes might yield significant improvements
* Analyze false positives and false negatives separately
* Look at confidence scores for errors
* Check if errors occur in specific contexts or conditions
* This can reveal whether the issue is with feature extraction, classification, or localization
* Load model + per class accuracy figure out
* Analyze model performance
* Pick subtask to optimize
* Begin optimizing via hyperparam, augmentation, etc. (ask in OH what is the scope of our messing with the model)

**Human pose estimation?**

**https://calvin-vision.net/wp-content/uploads/Publications/eichner12pami.pdf**

**Potential models to build off of:**

[**https://docs.ultralytics.com/tasks/pose/**](https://docs.ultralytics.com/tasks/pose/)

**https://medium.com/@ravionaldoraffel/create-a-human-pose-detection-using-yolov8-key-point-detection-and-machine-learning-model-f1ebe1f237e1**

[**https://gkioxari.github.io/PersonNet/index.html**](https://gkioxari.github.io/PersonNet/index.html) **(pascal dataset)**

[**https://github.com/leoxiaobin/deep-high-resolution-net.pytorch**](https://github.com/leoxiaobin/deep-high-resolution-net.pytorch) **(coco dataset)**

[**https://cv.gluon.ai/build/examples\_pose/dive\_deep\_simple\_pose.html**](https://cv.gluon.ai/build/examples_pose/dive_deep_simple_pose.html) **(coco dataset)**

**Pascal dataset trained open source models:**

[**https://github.com/chenyicai-0611/yolov5-PASCAL-VOC**](https://github.com/chenyicai-0611/yolov5-PASCAL-VOC)

[**https://github.com/zhang-dut/yolov8-pytorch**](https://github.com/zhang-dut/yolov8-pytorch)

[**https://github.com/VainF/DeepLabV3Plus-Pytorch**](https://github.com/VainF/DeepLabV3Plus-Pytorch)

**Possible, seems too complicated (multiple datasets already used in training):** [**https://github.com/open-mmlab/mmpose**](https://github.com/open-mmlab/mmpose)

**These projects are very flexible and adaptable to your interests/goals:**

* **you are free to focus on the topic(s) that excite you the most (you are even welcome to explore a computer vision topic outside the scope of the class),**
* **you can decide whether you want to collect your own visual data or use one of the existing benchmarks,**
* **you can build off of an existing toolbox or develop an algorithm entirely from scratch,**
* **you can focus your efforts more on analysis or more on building the system (although you should have some of both analysis and system building in your project)**

**Details**: You will fill out text boxes with a description of the problem, an outline of the proposed approach, pointers to related course topics, plans for acquiring the necessary data/computational resources, plans for quantitative and qualitative evaluation, the target outcome (what do you expect to deliver at the end of the project?) and a fallback plan (what are the potential roadblocks? what is the minimum you will be able to deliver if the exploratory parts of the project go wrong?).

* + The more detailed your milestone writeup is, the more the course staff will be able to give concrete and useful feedback. Feel free to include brief questions as well if you have specific concerns about your proposal.
  + Don't worry if you end up changing course after the milestone -- this is expected, and there's no need to re-submit, just incorporate whatever feedback you receive into the final project.
* Project description - Molly
* Approach outline - Molly
* Related course topics/resources - Guanyi
* Plan for data and computational resources - Guanyi
* Evaluation plan (quant/qual) - Tara
* Target outcome - Tara
* Fallback plan - roadblock, MVP - Tara

Improving Human Pose Estimation Performance of YOLOV Models

*Project Milestone Document*

COS429

Guanyi Cao, Tara Shukla, Molly Taylor

## Problem and Approach

Human pose estimation has wide-ranging applications in action classification, human-computer interaction, pedestrian detection, and more. However, models’ performance may be limited for certain poses that are less represented in the training data or for cases of occlusion. Our project aims to evaluate a human pose estimation model and explore methods to improve it. We will analyze the overall accuracy of a baseline model, identify body-part and pose classes with weaker performances, and test various approaches (e.g. supplementing training data, updating the loss function, et. cetera) to improve performance for weaker classes. In particular, we are interested in evaluating the model’s performance on more difficult subtasks, such as classification under occlusion. We will evaluate the success of our interventions qualitatively and quantitatively, gaining insight into strategies for improving model performance and the types of classes and conditions that are more difficult to identify.

## Related Course Topics

As we look to evaluate and improve a model for human pose estimation, we will draw on several course topics:

* **Data augmentation** will be important in increasing model robustness by helping in generalizing the model to new poses and backgrounds.
* **Transfer learning** will be a core focus of the project, as fine-tuning pre-trained models on pose datasets can improve performance while saving time and computational resources compared to training from scratch.
* Understanding **object detection** methods will allow us to localize poses within images, as explained by the **classification and localization** topics covered in class.
* **Convolutional neural networks** (CNNs) will likely be the backbone of most modern pose estimation models, since it is able to capture spatial hierarchies in images and detect body key points effectively.

## Data and Resource Plan

For data acquisition, we will begin with using the Pascal VOC dataset, available on the official website. If additional data is needed, we will consider incorporating datasets such as COCO or MPII Human Pose to increase diversity and capture a broader range of poses. Kaggle can serve as a supplementary source for other datasets if needed.

We’re exploring using one of the following open-source models (and would appreciate any suggestions or considerations that you have):

* <https://docs.ultralytics.com/tasks/pose/#models>
* <https://github.com/chenyicai-0611/yolov5-PASCAL-VOC>
* <https://github.com/princeton-vl/pytorch_stacked_hourglass?tab=readme-ov-file>
* <https://github.com/VainF/DeepLabV3Plus-Pytorch>

If significant computational power is required, we plan to leverage GPUs available on Google Colab or Kaggle, which offer flexible resources that can help with computation. We don’t currently anticipate the computational power needed to exceed these resources.

## Evaluation Plan

We will evaluate the success of our project both quantitatively and qualitatively. Quantitatively, we will use metrics such as Percentage of Correct Keypoints (PCK) and Average Precision (AP), and Object Keypoint Similarity (OKS), to the baseline and the improved models to analyze initial performance and track performance changes. We will evaluate performance on the overall test set and account for subtasks like pose estimation under occlusion or estimation of different poses. For our qualitative evaluation, we will perform visual inspection of images of different poses under varying conditions. We will overlay the baseline and improved models’ detected keypoints and manually compare the alignment with the ground truth. This should give us qualitative insights on specific subtasks in which the models’ performance improves or remains lacking. For instance, we could identify consistent errors in joint detection or alignment issues under certain angles, which would be more difficult to identify with quantitative metrics alone.

## Target Outcome

We will deliver a comprehensive report analyzing the initial YOLOV model’s weaknesses across several metrics and subtasks in the task of human pose estimation. We will document the changes we made to the model architecture or training data, and the reasoning behind these design decisions. The report will include a detailed methodology and intermediate steps results during the iterative model optimization process. Finally, we will deliver the code for an enhanced model that exhibits higher performance accuracy than the initial model in subtasks such as pose accuracy by type or detection under occlusion.

## Fallback Plan

Primarily, computational time and resources could be an issue with our project implementation, considering our usage of Google Colab, which can limit compute usage and take a long time for large models. If the computation poses a problem, we can consider changing to a different model, such as a smaller YOLOV version, or [DeepLav](https://github.com/VainF/DeepLabV3Plus-Pytorch). There is also a high possibility that most of our optimization attempts will yield minimal or zero improvement; in this case, our fallback deliverable will include a thorough analysis of the baseline model’s limitations, documentation of our attempts, and at least one improvement that bolsters model performance in some area.