*Questions we want to answer:*

- **Where to start?**
  - I.e. What's the easiest size ratio? (Basically, **define an easy-to-hard curriculum through empirical evaluations**)
  - We hypothesize that the easiest size is somewhere in the middle:
    - Small sizes will offer the most context, but lowest object resolution. May be overwhelmed by confuser objects in the foreground.
    - Large objects will have the highest object resolution, but will have minimal context and may only present part of the object.
    - A somewhat equal balance between object resolution and scene context may make for the easiest task and thus the optimal curricular starting point.
- **Does curriculum learning work?**
  - How will our size-based curriculum effect the performance of deep neural networks on visual recognition tasks?
    - Can we confirm the work of Bengio et al. that CL improves generalization and convergence?
  - Once we define a curriculum, we test whether it improves generalization and convergence
  - What about low-shot learning? Do the features learned from CL perform better for classifying a new object from only a few examples?
    - Perhaps we could incorporate SVM of kNN here.
      - Use the network as a feature extractor to embed examples
- **How gradual of a curriculum?**
  - What are the effects of varying the granularity of the curriculum?
- **What can be done to improve the effects of CL? (Apply what we've learned)**
  - *Architecure choices*
    - Spatial transformer layers – transforms big/small objects into more canonical sizes
    - Dilated convolutions – preserves weak signals (e.g. from low res objects)
  - *Gradient methods*
    - Larger learning rate when starting new curriculum?
  - *Cost function*
    - How can we constrain model complexity at the start, and decrease complexity constraints as curriculum progresses?
      - Incorporate a discussion about the bias-variance tradeoff and model complexity
      - L1 (LASSO) regularization
        - Decrease regularization as model progresses through curriculum?
- Models
  - LeNet (Low complexity - designed for MNIST, maybe underkill)
  - AlexNet (High complexity - designed for ImageNet, maybe overkill)
  - Something in between? (Can we design an architecture that rivals ImageNet accuracy with less parameters?

*Experiments outline:*

1. **Define an easy-to-hard curriculum**
    - Split train/val sets 'n' ways by size ratio
    - Train/eval LeNet on each split
    - Use validation accuracy for each split to define a curriculum
2. **Study the effects of curriculum learning (baseline approaches)**
    - Apply the following curricula for LeNet and AlexNet:
        - No curriculum
        - Random curriculum
        - Easy-to-hard
        - Hard-to-easy
        - Big-to-small
        - Small-to-big
    - Report for each:
        - Generalization
        - Convergence
        - Few-shot learning
            - **incorporate SVM or kNN here.**
                - Use the network as a feature extractor to embed examples

3. **Improving the effects of CL (Apply what we've learned)**
    - *Curriculum choices*
        - Vary the curriculum granularity

    - *Architecure choices*
        - Spatial transformer layers – transforms big/small objects into more canonical sizes
        - Dilated convolutions – preserves weak signals (e.g. from low res objects)
    - *Gradient methods*
        - Larger learning rate when starting new curriculum?
    - *Cost function*
        - How can we constrain model complexity at the start, and decrease complexity constraints as curriculum progresses?
            - Incorporate a discussion about the bias-variance tradeoff and model complexity
            - L1 (LASSO) regularization
                - Decrease regularization as model progresses through curriculum?

- Models
    - LeNet (Low complexity - designed for MNIST, maybe underkill)
    - AlexNet (High complexity - designed for ImageNet, maybe overkill)
    - Something in between? (Can we design an architecture that rivals ImageNet accuracy with less parameters?

TODO: *Write some notes...*