# Pallas Starter Projects

For computation libraries, you may choose between PyTorch and Tensorflow.

## 1 Predicting Flow

- **Dataset**: KITTI
- **Objective 1**: Given a pair of successive images $I_t, I_{t+1}$ and their associated, label bounding boxes $\{b_i^t\}_{i=1}^a, \{b_i^{t+1}\}_{i=1}^a$, where $a$ is the number of anchors per image, predict $\{dx_i, dy_i\}_{i=1}^n$. A simple linear model suffices.
- **Objective 2**: Given a pair of successive images $I_t, I_{i+1}$ and their associated label bounding boxes from a state-of-the-art model, e.g. Faster-RCNN, $\{\hat{b_i^t}\}_{i=1}^a, \{\hat{b_i^{t+1}}\}_{i=1}^a$, predict $\{dx_i, dy_i\}_{i=1}^n$. A simple linear model with regularization suffices.
- Once you are done with these two tasks, you will then need server access to complete **Objective 3**: Amend a state-of-the-art model to output bounding boxes, *along with* $dx, dy$ for each bounding box.

## 2 Learning Embedding

- **Dataset**: CIFAR10
- **Objective 1**: Given an image $I \in R^d$, learn a lower-dimensional embedding in $R^{d'}$ such that $d' << d$, using $g : R^d \rightarrow R^{d'}$. i.e., Train $\hat{I} = g^{-1}(g(I))$. A linear embedding suffices.
- **Objective 2**: Given an the lower-dimensional embedding of an image $g(I) \in R^{d'}$, run classification. Again, a linear classifier suffices. $\hat{y} = f(g(I))$.
- Once you are done with these two tasks, you will then need server access to complete **Objective 3**: Train a state-of-the-art model to accept a lower-dimensional embedding as input.

## 3 Robustness to Noise

- **Dataset**: MNIST
- **Objective 1**: Train a simple neural network–3 layers, with 1 convolutional layer and 2 fully-connected layers on MNIST. Denote this predictor $f$. Compute $var(Y) = var(f(X))$ (variance over the samples). Then, apply Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ to the samples $X + \epsilon$, and compute variance of the output $var(\tilde{Y}) = var(f(X + \epsilon))$. Repeat for increasing variance in the Gaussian. Plot $var(\tilde{Y})$ w.r.t. variance in the noise $\sigma$.
- **Objective 2**: Replace the convolutional layer with a shift module. Sample usage: shiftresnet-cifar repository. Plot the output variance w.r.t. variance in the noise, on the same plot.
- Once you are done with these two tasks, you will then need server access to complete **Objective 3**: Repeat both experiments, one with convolutions, and one with shift, on CIFAR-10, then CIFAR-100. Other paths of experiments: apply high-variance noise to single pixels.