按照上周的计划,我这周主要在看《Fast Training of Pose Detectors in the Fourier Domain》这篇论文。这篇论文提供了一种快速训练 Pose Detector 的方法， 这种方法同时适应对平面内变换、平面外变换和非刚性变换的数据集的训练，但前提是用于训练的数据的变换是循环的。

训练的流程大体如下：

首先构造一个表示一张图片的所有变换的模型：

$$C_Q(\mathbf{x}) = \begin{bmatrix} (Q^0\mathbf{x})^T \\ (Q^1\mathbf{x})^T \\ \vdots \\ (Q^{s-1}\mathbf{x})^T \end{bmatrix}$$

其中: x 为该图片的特征向量，长度为 m * 1

Q 是一个长度为 m*m 的正交矩阵

Qx 则表示图片的增量转换

前面已经提到，用于训练的数据的变换是循环的，因此有 $Q^s = Q^0 = I$ 的性质，最后 $C_Q(\mathbf{x})$ 就储存的一个样本数据的所有变换。然后用包含所有样本数据的矩阵进行训练，这样就达到同时训练不同变换的效果。

论文还给出了用于水平变换的矩阵：

$$P = \begin{bmatrix} 0_{s-1}^T & 1 \\ I_{s-1} & 0_{s-1} \end{bmatrix}$$

其中 0_{s-1} 是一个长度为(s-1)*1 的零矩阵。该矩阵能将图片水平向右变换，即

$$(x_1, x_2, x_3, \ldots, x_s) \rightarrow (x_s, x_1, x_2, \ldots, x_{s-1})$$

论文提出这个 Q 矩阵不需要显式地去计算，我的理解是，如果我们用于训练的数据包含所有的变换，则我们输入的已经是完整的 $C_Q(\mathbf{x})$，因此 Q 就不需要计算了。

论文还给出了该方法的 demo MATLAB 代码，我选取了一组包含 16 个样本的训练集，其中每个样本包含 6 个变换，而每张图片的大小为 128*200，最后训练时间为 0.05s 左右。