

目录

交通标志识别	1
1 PCA/Fisher 交通标志识别	1
1.1 方法描述.....	1
1.2 结果与分析.....	2
2 Hog 特征+SVM 分类方法	3
2.1 方法描述.....	3
2.2 结果与分析.....	3
3 开集测试.....	4
3.1 PCA+kNN.....	4
3.1.1 不使用负样本训练集.....	4
3.1.2 使用负样本训练集.....	6
3.1.3 有无训练数据的对比.....	7
3.2 Hog+SVM 开集测试.....	7
4 作业中遇到的问题与思考.....	8

交通标志识别

1 PCA/Fisher 交通标志识别

1.1 方法描述

利用 PCA+kNN 进行交通标志识别，核心操作是将原图像投影到特征空间，然后在特征空间进行 kNN 分类。算法流程如下。

- 1) 对训练、测试图像进行预处理（转灰度、统一大小、直方图均衡化等）
- 2) 对所有训练数据样本进行 PCA 操作，获取特征空间（用特征向量矩阵表示）
- 3) 确定要保留的特征空间维数 K 和 kNN 分类近邻数 k 的大小
- 4) 在特征空间，根据 kNN 准则，对测试图像进行分类。

上述流程用流程图表示如图 1 所示。

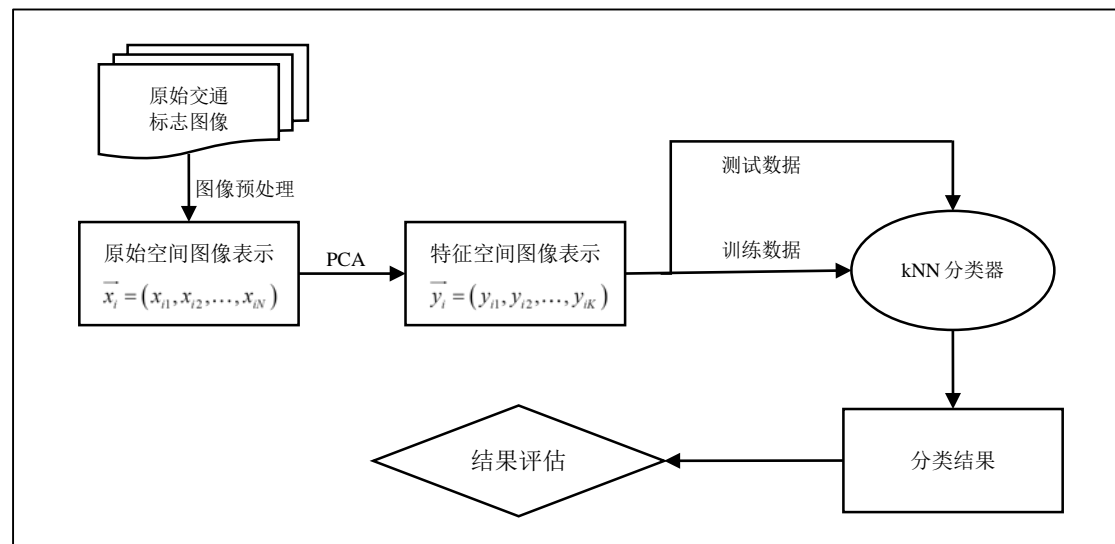


图 1 PCA+kNN 分类流程图示意图

其中，图像预处理包含转灰度图、统一图像大小等操作。此外，注意到给定数据集中，有不少图像的灰度分布并不均匀，有些看起来明显偏暗，这将会对后续的分类造成不利的影响。因此，预处理中包括了直方图均衡化操作，以尽量均衡图像的灰度分布。

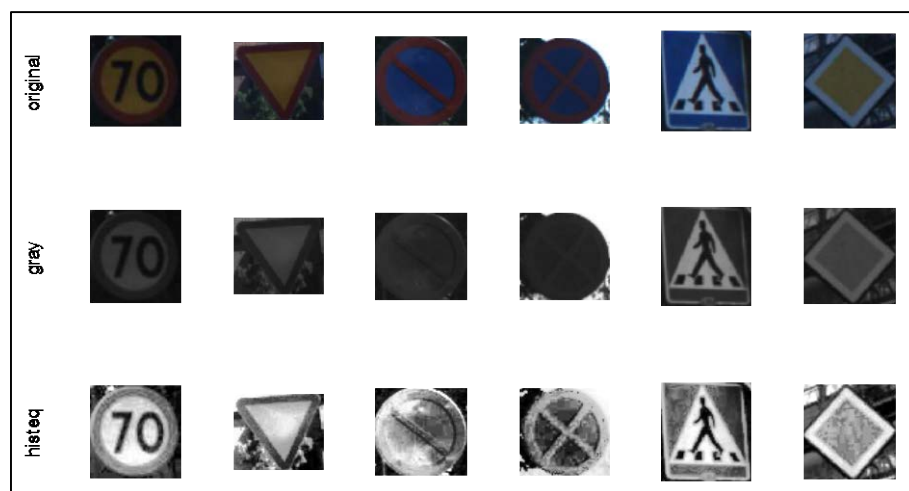


图 2 直方图均衡化后的结果

图 2 选取了数据集中亮度比较低的几幅图像，可以看出，直接转为灰度图后，图像灰度分布很不均匀，NO_PARKING 和 NO_STOPPING_NO_STANDING 标志甚至无法区分。但均衡化之后，图像的灰度分布特征得到明显改善。

1.2 结果与分析

仿真中，主成分的个数选为 1-250，kNN 中近邻数选为 1、3、5，得到的仿真结果如图 3 所示。

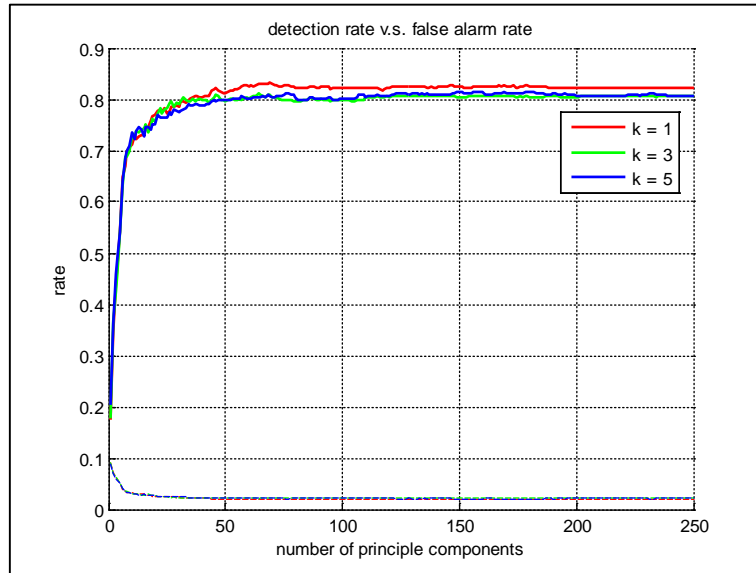


图 3 PCA+kNN 仿真结果

图 3 分别显示了 $k = 1, 3, 5$ 时，分类器检测率 P_d 和虚警率 P_{fa} 随着主成分个数 K 的变化关系。可以看出，当主成分个数达到一定数量之后，近邻数目对分类效果的影响不是太大。整体来说， $k = 1$ 时分类效果最好。选取图 3 中分类效果达到最好时的参数，得到的分类结果 Confusion Matrix 对应的图形表示如图 4 所示。

可以看出，识别率最高达到了 83.25%，效果相对不错。进一步分析发现，类 6 (NO_PARKING) 和类 7 (NO_STOPPING_NO_STANDING) 识别效果不是很好，这主要是因为两类样本非常相似，在特征空间中，两类样本形成的分布簇比较接近。

作为对比，我们注意到，当不对图像进行直方图均衡化处理时，识别率最高只能达到 65.5% (分类结果如图 5 所示)。因此，均衡化操作显著提升了 PCA+kNN 的分类效果。

对该方法的分析总结如下：

- 1) 随着选取的主成分的个数逐步增加，分类的识别率整体上逐步提升。但当主成分个数达到 75 以上之后，识别率基本稳定不变，这说明前几十个维度已经集中了几乎所有原图像的特征。同时考虑到计算量的问题，选取主成分个数在 60 左右比较合适。
- 2) 近邻数的选取对分类效果的影响相对不大。
- 3) 受 kNN 算法本身的影响，该算法识别率上限是 80% 多，即使通过提升主成分个数也无法进一步提升识别率。如果希望进一步提高识别效果，可以选取更有效的分类特征和更适合的分类算法。

该部分仿真代码文件：[pr_proj_task1.m](#)。

开集测试见第 3 部分。

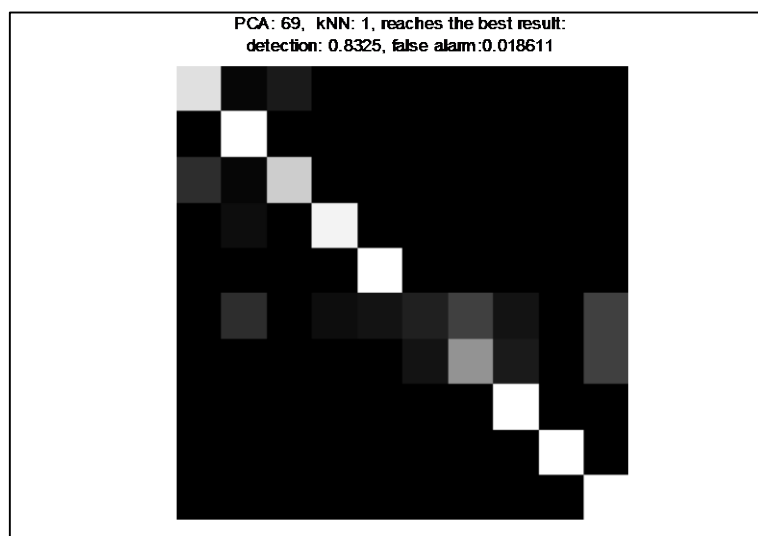


图 4 PCA+kNN 分类结果的 Confusion Matrix

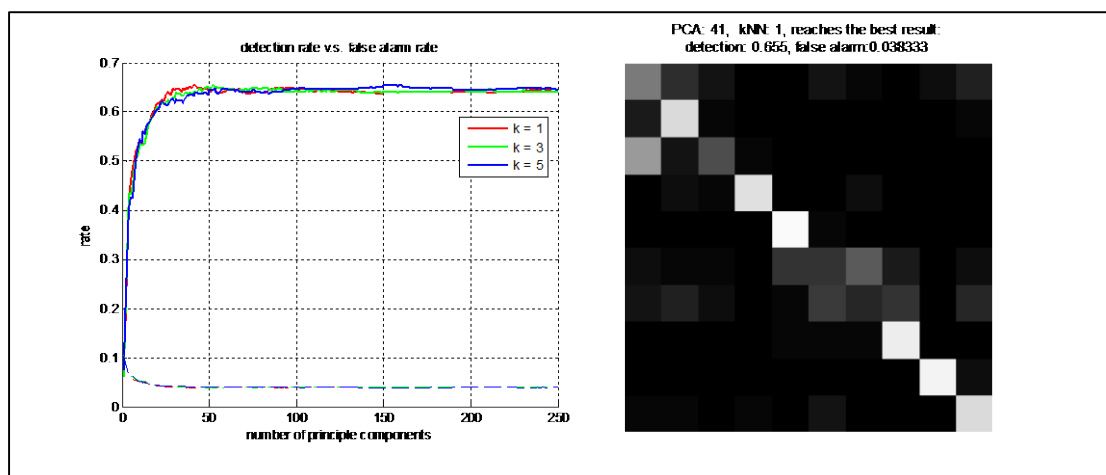


图 5 没有直方图均衡化时的分类结果

2 Hog 特征+SVM 分类方法

2.1 方法描述

如前文所述, PCA+kNN 识别率上限是 80% 多, 如果希望进一步提高识别效果, 可以选取更有效的分类特征和更适合的分类算法。其中, Hog 特征+SVM 分类算法就是一个很好的方法。

该方法操作流程与 PCA+kNN 基本相似, 不同的是, 该方法不需要进行特征空间转化。利用 Hog 特征提取方法, 得到图像的 Hog 特征之后, 可以直接使用 SVM 算法对测试图像进行分类。

2.2 结果与分析

仿真中, Hog 特征提取中, cellSize 设为 8, 图像被预处理为 32*32 大小, 因此, 每幅图像对应的 Hog 特征维度是 496。SVM 采用 multi-class SVM 分类, kernel 使用线性核, 其他参数均为 libsvm 中的默认参数。

该方法的分类结果如图 6 所示。

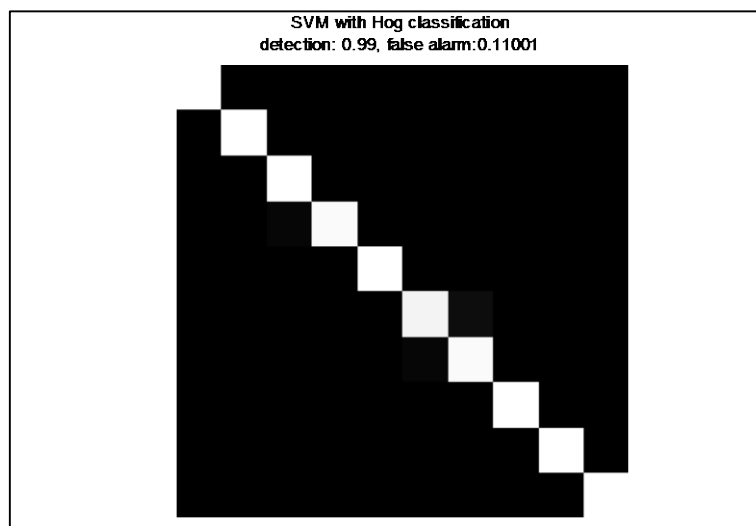


图 6 Hog+SVM 分类结果的 Confusion Matrix

对该方法的分析如下：

- 1) 从图 6 看出，除了极少量的类 6、类 7 有误分之外，其实测试样本基本全部分类正确，分类的识别率达到了 99%。
- 2) Hog 通过计算和统计图像局部区域的梯度方向直方图来构成特征，与其他特征相比，Hog 特征保持了几何和光学转化不变性，因此 Hog 特征尤其适合目标（如行人）检测。此外，SVM 能够最小化结构风险，寻求最优超平面进行分类，基于凸优化理论，该分类面是全局最优的，因此，Hog+SVM 在该实验（闭集测试）中能达到很好的识别结果。
- 3) 作为对照，我们注意到，即使没有直方图均衡化的操作，Hog+SVM 识别率依然能达到 98%，这也说明了 Hog+SVM 对于目标检测的有效性。
- 4) 相比 PCA+kNN 而言，Hog+SVM 需要设置的参数更多。尽管 Hog+SVM 效果很好，但其对参数的设置比较敏感。如，当采用非线性核时，分类效果便明显降低。实际中，需要结合经验及实验，选取最合适的参数设置。

该部分仿真代码文件：[pr_proj_task2.m](#)。

开集测试见第 3 部分。

3 开集测试

上述实验均是在闭集上进行测试的，因而识别率往往可以达到很高的效果。但如果进行开集测试，还需要注意给出合理的拒识方式，用来判断某张图像是否属于训练的 10 个类别。

下面本实验分别根据 kNN 和 SVM 的特点，并结合是否有负样本训练数据，来讨论开集测试的效果。

3.1 PCA+kNN

kNN 根据待分类数据最邻近的样本点的类别来确定未知数据的类别。一个很自然的想法就是，对最邻近的距离设计一个合理的阈值，如果距离超过了某个阈值，即使未知样本被分到某个类别，我们也认为分类失败（即不是 10 类样本之一）。

3.1.1 不使用负样本训练集

基于这个思想，我们可以在没有负样本标注数据的基础上，仅根据距离阈值设计拒识方式。但可以预见到，这种方式难以同时达到很好的识别率和拒识率，在调增阈值参数时，一种指标的提升是以另一种指标的降低为代价的。

对于 kNN，详细的拒识方式算法如下：

- 1) 对所有测试数据用 kNN 算法进行分类
- 2) 在分类过程中，分别记录每个测试数据与其 k 近邻的平均距离，即

$$d(x_i) = \frac{1}{k} \sum_{x_j \in kNN(x_i)} d(x_j - x_i)$$

- 3) 获得分类信息和距离信息后，对于被分类到同类的测试数据，用该类别内 k 近邻最大的距离对类内所有距离归一化，即

$$d_{norm}(x_i) = \frac{1}{\max_{x_j \in class(x_i)} \{d(x_j)\}} d(x_i)$$

- 4) 通过上述操作，所有测试数据的 k 近邻距离均在 (0,1) 之间，因此可以设计合适的距离阈值 α ，若 $d_{norm}(x_i) > \alpha$ ，则拒识样本 x_i 。

显然，若 α 很大，正样本识别率会很好，但负样本拒识率很低；若 α 很小，负样本拒识率升高，但正样本识别率随之下降。

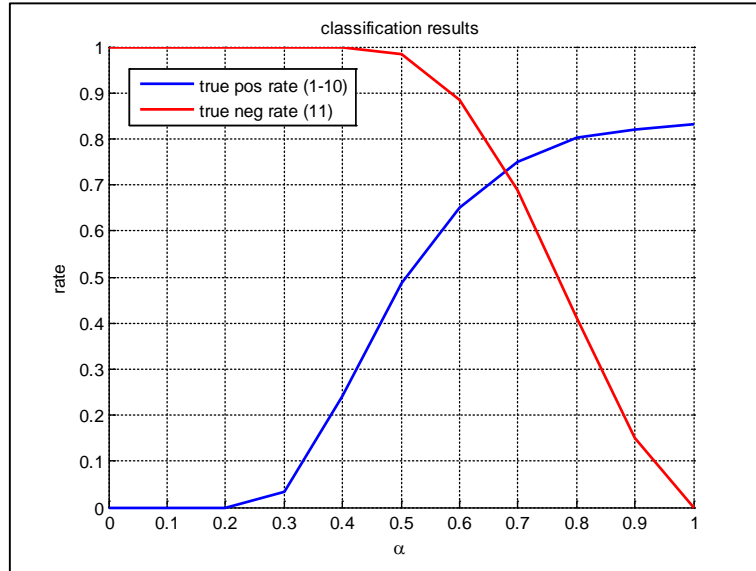


图 7 识别率与拒识率随参数 α 的变化关系

图 7 中，蓝色曲线代表正类（1-10）的识别率，红色曲线代表负类（假设为 11）的识别率，也即拒识率。因为在这种框架下，成功拒识意味着样本的归一距离超过了阈值，即被修正为第 11 类（负样本）。

可以看出，识别率与拒识率随着 α 的变化关系与之前分析的结果相一致。随着 α 的提升，识别率逐渐增加，但拒识率却逐渐减低。特别的，当 $\alpha = 1$ 时，模型等价于不对 kNN 进行任何改进，即没有阈值判断，因此拒识率降低为零。

进一步的，由于已经获取了一系列 True Positive Rate 和 True Negative Rate 的值，我们可以画出 ROC 曲线来进一步查看分类性能。需要注意的是，绘制 ROC 曲线时，我们实际上把前 10 类统一看成正类，因此，为了得到一条真正的 ROC 曲线，我们需要修正 True Positive Rate 的定义为，前 10 类的样本被成功分类为前 10 类数据，而不管类内是否分类正确。此时，可以得到如图 8 所示的 ROC 曲线。

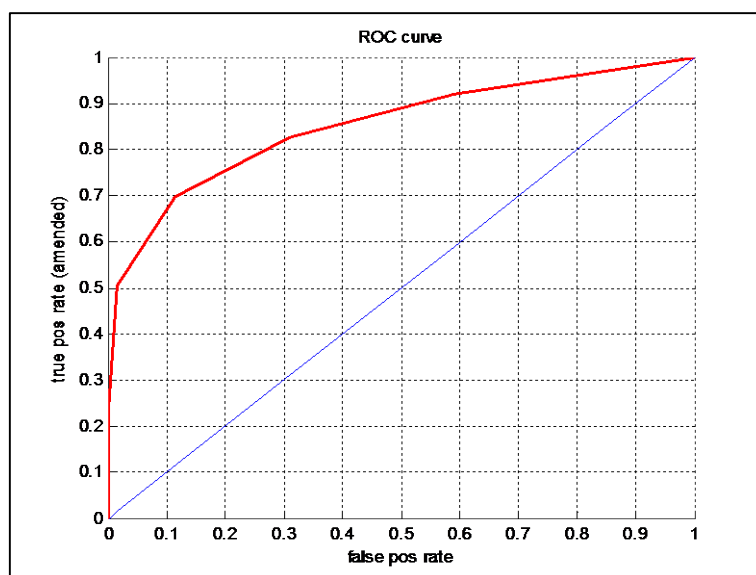


图 8 kNN 无负样本训练集情况下的分类 ROC 曲线

该部分的仿真代码文件：[pr_proj_task3_1.m](#)。

3.1.2 使用负样本训练集

为了进一步在保证识别率的基础上进一步提升拒识率，我们可以借助负样本训练数据辅助拒识判别。为了完成这一任务，我从网上另找了 112 张负样本图像，其中既包含测试图像中含有的交通标志，也包含一些新的交通标志。

基于此，问题转化为一个 11 分类的问题。我们可以同时根据距离阈值和分类结果进行判定，即如果一个测试样本被分类为 11，或者其 k 近邻距离超过阈值，均认为成功拒识。这种情况下的分类结果如图 9 所示。

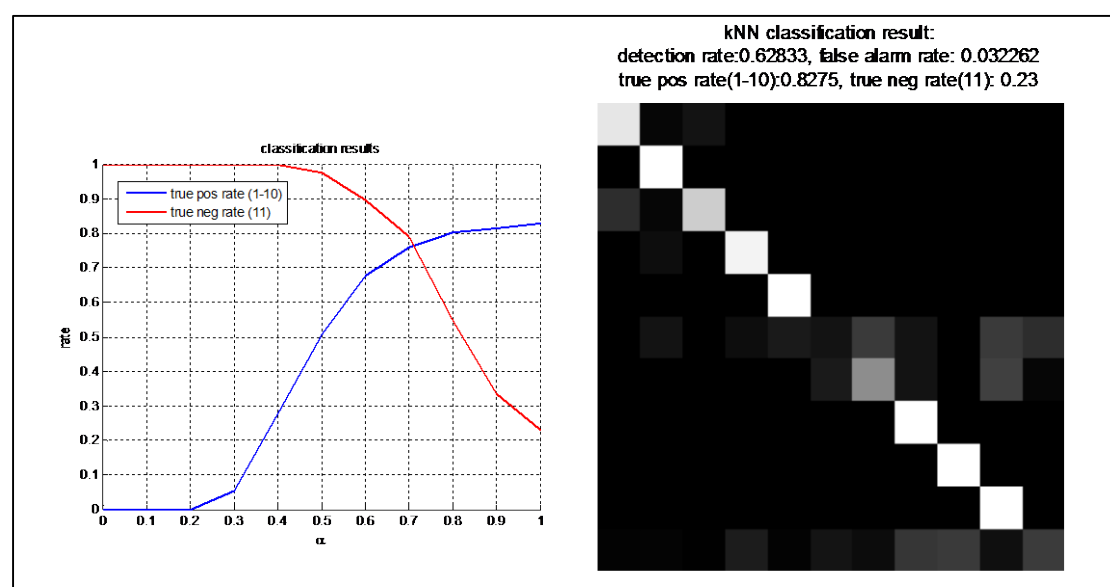


图 9 kNN 使用负样本训练集时的分类结果

(左图：TPR 与 TFR 随 α 变化关系；右图：分类结果的 Confusion Matrix)

可以看出，随着 α 逐渐增大，拒识率降低的趋势变缓了，即使在 $\alpha = 1$ 时，拒识率也在 0.23 左右。实际上，拒识率的最小值是由第 11 类的分类正确率确定的。由于第 11 类数据含有的交通标志类标较多，且训练集和测试集有很多不一样的标志，因此 kNN 对类 11 的分类准确率很低，也因此，若把该问题看作纯粹的 11 分类问题时，平均识别率从 0.83 降低到了

0.628。可以预料到，使用 Hog+SVM 分类，第 11 类的分类准确率会有所提升，但也无法达到前 10 类的准确率。该部分对应的仿真代码文件：[pr_proj_task3_2.m](#)。

3.1.3 有无训练数据的对比

进一步的，为了更直观看出添加负样本训练集对分类效果的影响，我们把有无训练集的结果绘制在同一张图中，可以得到如图 10 所示的结果。

从图 10 可以看出，加入负样本训练集之后，正样本的识别率得到微弱的改善（正样本识别率上限由 kNN 算法确定），但负样本的拒识率得到了比较好的改善。其中，负样本拒识率的改善情况是由负样本的分类正确率所确定的，因此，如果选取更加合适的负样本训练数据，拒识率可以得到更大的改善。

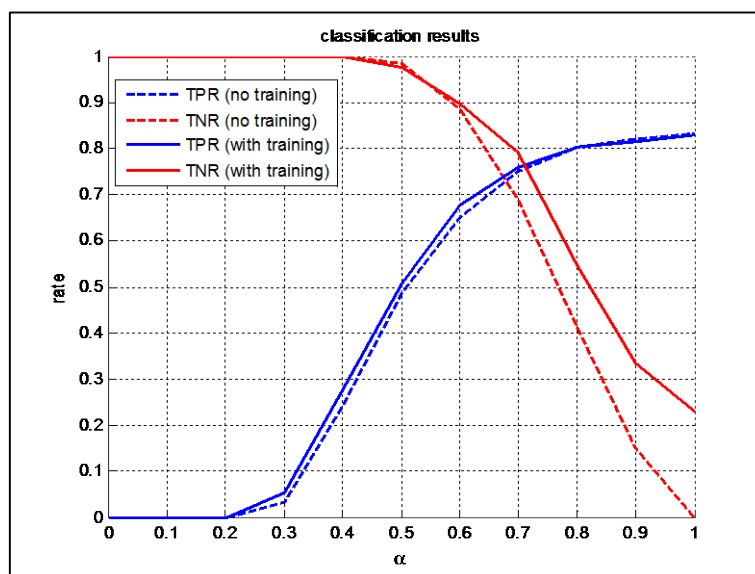


图 10 有无负样本训练集时的分类效果对比

该部分（对比）的仿真代码文件：[pr_proj_task3_contrast.m](#)。

3.2 Hog+SVM 开集测试

对于 SVM 分类，可以直接将问题看作一个 11 分类问题，没有其他参数的调整。因此，拒识判则简化为：只要样本被分类为类 11，则拒识该样本。同样采用第 3.1.2 小节中使用的人工采集的负样本训练数据，利用 Hog+SVM 分类，得到的分类结果如图 11 所示。

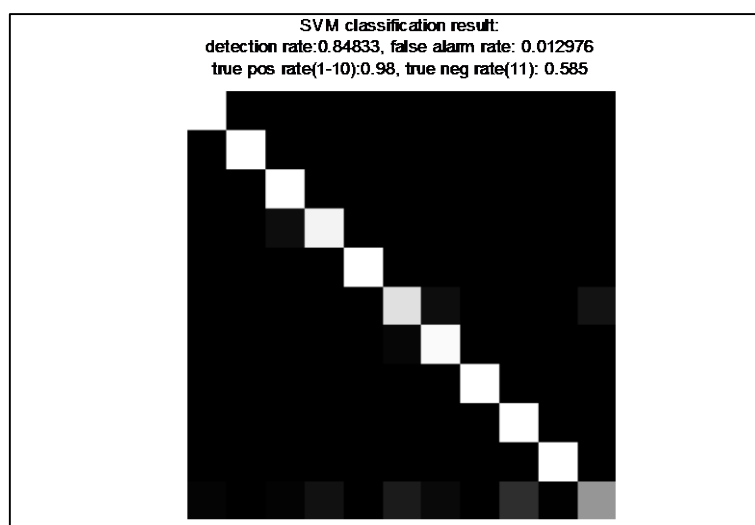


图 11 SVM 使用负样本训练集时的分类结果

可以看出, 相比于 kNN, SVM 将第 11 类的识别准确率从 0.23 提升到了 0.585。前 10 类的识别正确率依然维持在 98% 左右, 这也验证了 Hog+SVM 在目标检测问题中的有效性。由于类 11 的影响, 所有测试图像的平均识别率从 99% 降低到了 84.8%。

该部分的仿真代码文件: [pr_proj_task3_3.m](#)。

4 作业中遇到的问题与思考

在 PCA+kNN 分类中, 我遇到了这个问题, 即应该选取如下哪种方案:

- 1) 用所有 10 类训练数据进行 PCA 操作, 然后再整个特征空间里利用 kNN 分类
- 2) 仅用某一类 (以类 1 为例) 训练数据进行 PCA 操作, 这样每个类别对应了一个特征空间, 可以分别在每个特征空间内对测试数据分类, 然后采用投票机制。

结合 PCA 原理, 对此问题的初步思考是: 如果采用方案(1), 理想的情况应该是, 各类的样本在总的特征空间里呈现为相对集中的簇, 不同簇之间有一定间隔。对于方案(2), 实际上类 1 的特征空间并不包含类 2-9 的信息, 因此猜想, 类 2-9 的数据投影到类 1 的特征空间, 可能无法将后 9 类有效区分开来。因此, 即使采用投票机制, 方案(2)的效果可能也不如方案(1)。

结合以上想法, 我仅采用 10 类的测试数据 (10*40=400 个), 分别采用上述两种方案, 查看测试数据在特征空间中的分布, 根据分布的特征便可以直观地看出哪种方法更合理。

为了使数据可视化, 我们仅采用前 3 个主成分, 即在 3 维的特征空间内画出所有的测试点。根据第 1 部分的结果, 当 $K=3$ 时, 识别正确率仅在 0.4 到 0.5 之间, 但这已足以够我们直观上观察数据点的分布特征。

这两种方案对应的特征空间中数据分布分别如图 12 和图 13 所示。

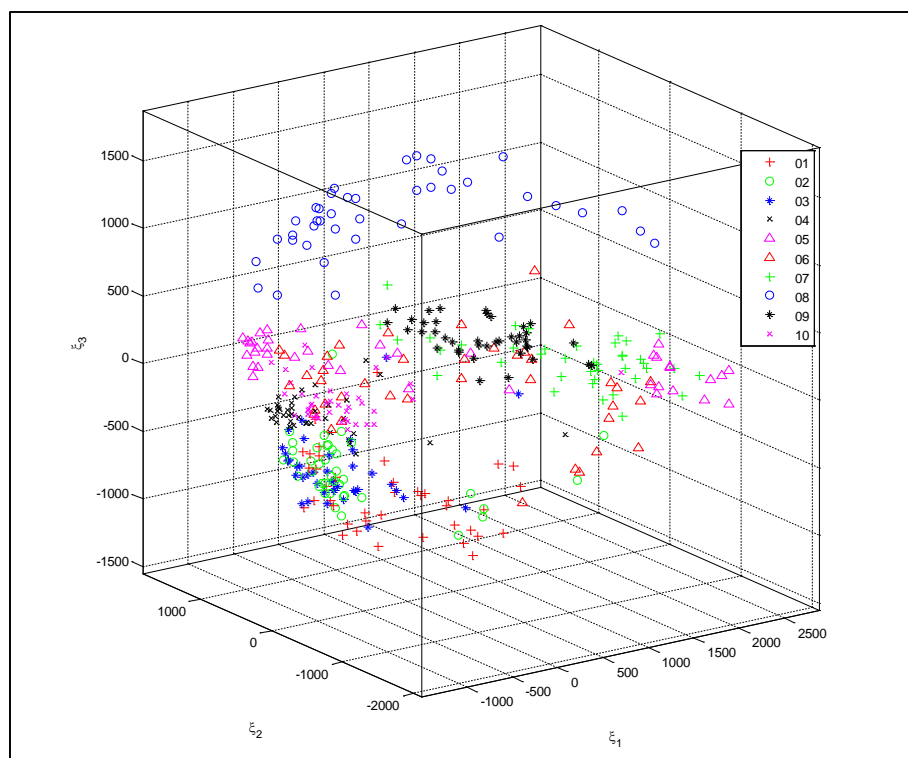


图 12 方案 1 对应的特征空间数据分布

从图 12 可以看出, 尽管有些类别存在部分重叠, 但不同的类确实分布在特征空间中的不同区域, 这是后续能正确分类的基础。同时可以看出, 类别 6 (红色三角) 几乎散布在整个特征空间内, 类内聚合性很差, 因此可以预料到, 类 6 的分类效果将会很差, 第 1 部分的

结果也印证了这个观点。在图 13 中，不同类别数据的分布则更多的混在了一起，这将会对 kNN 分类产生很不利的影响。因此，如果采用这种方法，即使训练出 10 个特征空间分别分类再投票表决，效果也难以保证。

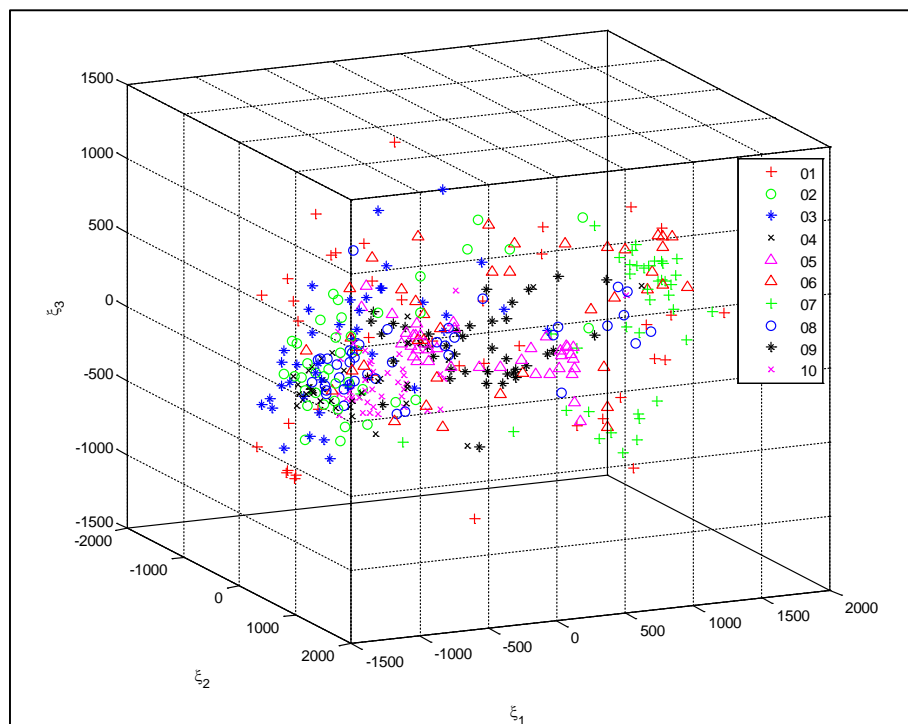


图 13 方案 2 对应的特征空间数据分布

基于此，我在第 1 部分采用了方案(1)，即用所有类别的训练数据进行 PCA 操作，然后在这个共有的特征空间内对测试样本进行分类。

图 14 给出了上述两幅图的二维显示结果，通过该图也可以直观上看出这两种方案的差别。相比方案(2)，方案(1)更符合“类内紧致，类间分离”的原则。

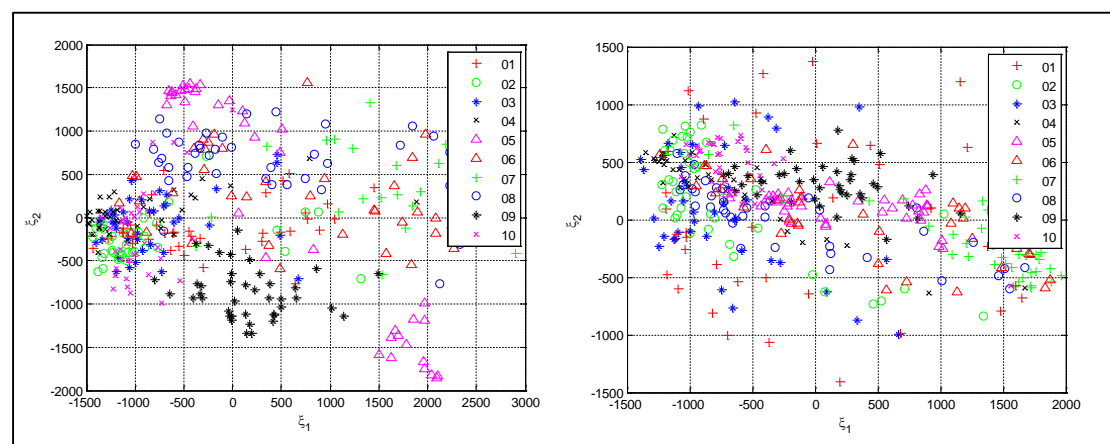


图 14 两种方案对应的特征空间中数据分布（2 维）