

浙江大学

光电信息综合实验

交通标志识别



姓名 赵鼎 吕泽嘉 姚罡

指导教师 楼东武

年级与专业 15 级信息工程

所在学院 信息与电子工程学院

目 录

一、综述	3
1 引言	3
2 文献综述	3
二、关键技术和算法	5
1 图像分割——快速圆检测算法	5
1.1 弧线提取	5
1.1.1 边缘检测	5
1.1.2 弧线检测	6
1.1.3 弧线凸性检测	6
1.1.4 象限综合分类	7
1.2 圆形检测	7
1.2.1 弧线选择策略	8
1.2.2 中心估计	9
1.2.3 参数估计	10
1.3 单标志提取	12
1.3.1 圆形验证	12
1.3.2 标志提取	13
2 BP 网络识别	14
2.1 PCA 降维	14
2.2 网络训练	14
2.2.1 数据集建立	14
2.2.2 网络模型	15
2.2.3 参数设置	15
2.2.4 训练结果	15
三、测试模型与系统实践	17
1 搭建视频流	17
2 控制指令的分发与解析	18

3 以视觉流处理为核心的系统网络搭建与扩展	18
四、总结语	20

摘要

本文基于快速圆检测算法和定制化的 BP 神经网络模型，提出并实际测试了一种高速交通标志识别算法，算法在测试用数据集和实际系统测试中都具有极高的准确性和极快的处理速度，能够很好地满足未来视觉智能驾驶系统的实时性要求。基于这一算法，本文搭建了基于服务器-客户端模型的嵌入式智能小车视觉处理平台，系统拥有多线程扩展能力，可以作为工业级应用系统的原型参考。

关键词：圆检测，神经网络，速度，准确性，系统构建

一、综述

1 引言

随着社会经济的发展，信息产业和汽车工业发生了日新月异的变化，以无人驾驶为代表的汽车智能化是机器视觉和传统汽车工业结合的结果。无人驾驶是通过车载传感系统感知道路环境，自动规划行车路线并控制车辆到达预定目标，集自动控制、体系结构、人工智能、视觉计算等众多技术于一体，是计算机科学、模式识别和智能控制技术高度发展的产物。相比人工驾驶，无人驾驶具有安全性和可靠性高，能够实现车辆高密度运行，适应大客流运营需要的优点。无人驾驶中的一项重要内容便是交通标志的识别。在实际应用中，出于安全性的考虑，要求交通标志的识别具有高准确性和低延迟性。

本文主要介绍了如何利用椭圆检测算法和卷积神经网络实现对交通标志的抓取和识别。其中快速圆检测算法用来抓取图像中的交通标志，BP神经网络用来实现对交通标志的分类。为了验证算法的实际效果，我们将算法模型应用于智能小车中，实现交通标志识别的系统构建。

2 文献综述

机器视觉领域用于交通标志识别的方法有很多，国内外很多学者对交通标志识别方向进行了研究，比如基于模板匹配的识别方法，基于 HSI 与 SVM 的识别方法，HIS、形态学和 SVM 相结合的识别方法等等，浙江大学的何耀平提出一种 Adaboost 算法和 SVM 融合用于自然场景下的交通标志识别算法，该算法先用 Adaboost 筛选出最可能的图像子集，之后利用 SVM 做最终识别，该算法识别率较高。其中传统的模板匹配方法存在只允许平行移动、处理速度慢等局限性，若原图像中的匹配目标发生旋转或大小变化，该算法不能得到预期结果。其它算法尽管在实验测试中有较高的识别准确率，但在识别速度上却不尽如人意。在对交通标志的抓取或者识别上较高的时间复杂度和空间复杂度，导致大量时间开销和较高的硬件要求，这对于实时性要求较高的无人驾驶领域是不能接受的。本文提出了基于快速圆检测和 BP 神经网络的算法，该算法框架的创新性在弧线级别的圆检测，相比于像素级的圆形检测，该检测算法的运算量大大减少，之后 BP 网络具有快速的处理速度和高准确率，在保持识别准确率的同时大大降低了运算时间，使得其在对实时性要求较高的领域应用成为可能。

本论文的余下内容主要分为以下几个部分：交通标志识别算法框架介绍、算法应用于智能小车的整体框架、总结概述。

二、关键技术和算法

交通标志识别的实际应用场景往往有极高的实时性需求，除去网络传输带来的延迟之外，设计更具鲁棒性，识别速度和精度更高更快的算法是改善系统性能最关键的一环。本项目提出了基于快速圆检测和神经网络模型的检测识别算法，可以快速地从图像流信息中提取出交通标志并识别其分类标识。

1 图像分割——快速圆检测算法

圆形是常见的图形，也是交通标志最据代表性和普适性地特征，在许多计算机视觉处理中都含有圆形检测这一操作。一个快速而且稳定的圆形检测算法在圆形交通标志识别中显得十分重要。本文中我们放弃传统基于像素点范畴的检测，而是采用在弧线范畴检测多个弧线是不是属于同一个椭圆。之后利用平行弦的中点估计椭圆中心，其他的参数通过霍夫曼变换在奇异参数空间积累投票估计，最后只留下最接近圆形的图形。全部流程可以参考图表一该算法具有良好的检测速度和准确性，最快可以达到每张图片 10ms。

1.1 弧线提取

这部分我们需要从输入图像中提取弧线，首先检测边缘点，之后把他们归类到弧线，最后按照边缘线的方向和凸性进行弧线分类。

1.1.1 边缘检测

我们定义边缘点 $e_i = (x_i, y_i, \theta_i)$ ，其中 x_i, y_i 是像素点位置，像素点梯度相位为 θ_i 。我们采用具有自适应阈值的 Canny 边缘检测算法，该算法具有良好的速度和准确性，所需要全部参数都可以通过该算法计算得到。

1.1.2 弧线检测

定义函数 $D : e_i \rightarrow (\top, \perp)$, 其中

$$\begin{aligned} D(e_i) &= \text{sign}(\tan(\theta_i)) \\ &= \text{sign}(dx) \cdot \text{sign}(dy) \end{aligned} \quad (1)$$

分类结果 D 与边缘点 e_i 的 θ_i 参数有关, 但是我们不需要去求出准确 θ_i 的值, 简单对 Canny 算法中的 Sobel 求导微元 dx 和 dy 求符号函数即可。当垂直梯度方向 $dy = 0$ 或者水平梯度方向 $dx = 0$, 我们就舍弃该边缘点。

我们把属于相同分类结果的边缘点连接起来得到弧线 α^k :

$$\alpha^k = \{(e_i^k, \dots, e_{N^k}^k) : \forall i, j \wedge D(e_i^k) = D(e_j^k) \wedge \text{isConnected}(e_i^k, e_j^k)\} \quad (2)$$

N^k 代表弧线 α^k 边缘点的个数, 边缘点 e_i^k 和 e_j^k 的连接可以通过两个像素点的 8 领域连通性来检测。有些弧线 α^k 特征值较弱, 不足以判断椭圆, 此类需要被舍弃。如:

$$\alpha^k = \begin{cases} 0 & \text{if } N^k < Th_{length} \\ 0 & \text{if } OMA^k < Th_{oma} \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

Th_{length} 代表弧线 α_k 的长度, $N^k < Th_{length}$ 意味着该弧线较短, 应当舍弃; OMA^k 代表最小包围全部弧线点的矩形区域像素点数, $OMA^k < Th_{oma}$ 意味着存在弧线交点, 该弧线不属于椭圆边界, 也应当舍弃。

1.1.3 弧线凸性检测

定义函数 $C : \alpha^k \rightarrow (\top, \perp)$, \top 代表弧线上凸, \perp 代表弧线下凸。对于 $\forall i, (e_i^k \in OMA^k)$, 弧线上面部分区域为 O^k , 下面部分为 U^k , 弧线凸性表述如下:

$$\alpha^k = \begin{cases} \top & \text{if } Area(U^k) > Area(O^k) \\ \perp & \text{if } Area(U^k) < Area(O^k) \end{cases} \quad (4)$$

对于 $Area(U^k) = Area(O^k)$, 该情况凸性不可判断, 直接舍弃掉。

1.1.4 象限综合分类

根据已知的函数 D 和函数 C , 我们定义函数 $Q: \alpha^k \rightarrow \{I, II, III, IV\}$, 即该函数会将弧线分成四类, 这也是之后判断的最佳分类数目。函数计算如下:

$$\alpha^k = \begin{cases} I & \text{if } \langle D(\alpha^k), C(\alpha^k) \rangle = \langle \top, \top \rangle \\ II & \text{if } \langle D(\alpha^k), C(\alpha^k) \rangle = \langle \perp, \top \rangle \\ III & \text{if } \langle D(\alpha^k), C(\alpha^k) \rangle = \langle \top, \perp \rangle \\ IV & \text{if } \langle D(\alpha^k), C(\alpha^k) \rangle = \langle \perp, \perp \rangle \end{cases} \quad (5)$$

该部分整体的效果可以见于下图

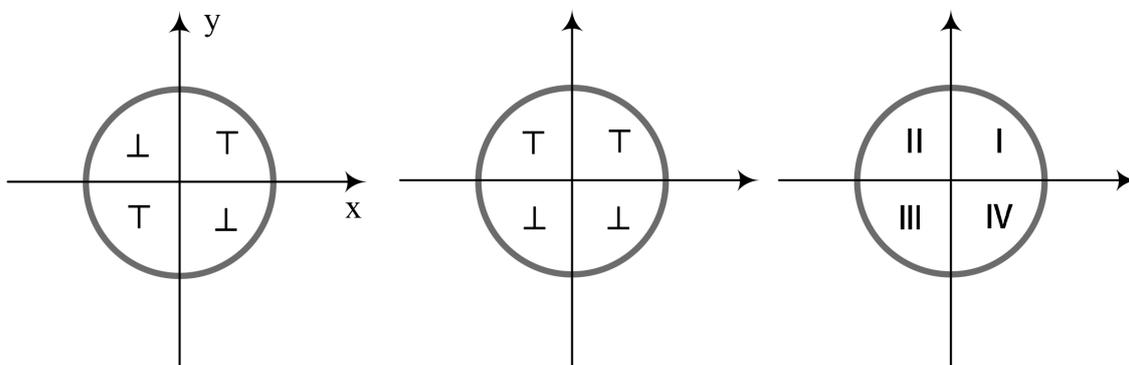


Figure 1.1: 象限综合分类

1.2 圆形检测

定义候选椭圆为三元组 ε_i , 比如椭圆可以由满足类似 $\tau = (\alpha^a, \alpha^b, \alpha^c)$ 的三段弧线来描述, 候选椭圆构成的集合为 ε 。椭圆 ε_i 的参数可以通过霍夫曼变换来计算估计。本文算法为单目标识别, 最终想要得到的是圆形 ψ 并且 $\psi \subseteq \varepsilon$ 。

1.2.1 弧线选择策略

假设 N_α 是经过前一阶段处理后剩余图片弧线的数目，定义 Γ^0 是这些弧线组成的所有可能的三元组的集合， Γ^0 中三元组数目可能会非常大并且有许多错误的结果，比如多个三元组都属于同一个椭圆。我们需要采用一定的策略去选择出真实有效的部分，即选出 Γ^0 的子集 Γ^{abc} 。

定义弧线对 $p^{ab} = (\alpha^a, \alpha^b)$ ，定义三元组为两个共享同段弧线的弧线对，即 $\tau^{abc} = \{(p^{ab}, p^{cd}) \mid \alpha^b \equiv \alpha^d\}$ 。弧线选择策略可以描述为，首先选出满足凸性和位置的弧线对，之后假设两个弧线位于相同椭圆边界上，计算椭圆中心。该策略最后会找到符合要求的三元组 Γ^{abc} 来描述椭圆。

- **凸性限制**：每一个弧线对必须是象限分类结果的子队列，默认按照逆时针方向。

$$A(p^{ab}) = \begin{cases} + & \text{if } (Q(\alpha^a), Q(\alpha^b)) \in \{(I, II), (II, III), (III, IV), (IV, I)\} \\ - & \text{otherwise} \end{cases} \quad (6)$$

其中 + 代表属于，- 代表不属于，经过该判断我们可以得到子集 $\Gamma^1 \subseteq \Gamma^0$ ，即：

$$\tau^{abc} \in \Gamma^1 \Leftrightarrow (\tau^{abc} \in \Gamma^0) \wedge (A(p^{ab}) \wedge (A(p^{cd}))) \quad (7)$$

- **位置限制**：主要检查弧线对中两个弧线的位置，并舍弃连通性不高的弧线对。

$$M(p^{ab}) = \begin{cases} + & \text{if } (Q(\alpha^a), Q(\alpha^b)) \equiv (I, II) \wedge (|L^a.x - R^b.x| < Th_{pos}) \\ + & \text{if } (Q(\alpha^a), Q(\alpha^b)) \equiv (II, III) \wedge (|L^a.y - R^b.y| < Th_{pos}) \\ + & \text{if } (Q(\alpha^a), Q(\alpha^b)) \equiv (III, IV) \wedge (|L^b.x - R^a.x| < Th_{pos}) \\ + & \text{if } (Q(\alpha^a), Q(\alpha^b)) \equiv (IV, I) \wedge (|L^b.x - R^a.y| < Th_{pos}) \\ - & \text{otherwise} \end{cases} \quad (8)$$

其中， L 和 R 分别代表弧线最左侧和最右侧的点，两个值的相似度由阈值 Th_{pos} 来控制。经过 M 函数的判断，我们可以得到子集 $\Gamma^2 \subseteq \Gamma^1$ ，即：

$$\tau^{abc} \in \Gamma^2 \Leftrightarrow (\tau^{abc} \in \Gamma^1) \wedge (M(p^{ab}) \wedge (M(p^{cd}))) \quad (9)$$

- **相同椭圆限制**：要求三条弧线 $\alpha^a, \alpha^b, \alpha^c$ 属于相同的椭圆，即弧线 α^a, α^b 的中心 C^{ab}

和弧线 α^c, α^d 的中心 C^{cd} 在阈值 $Th_{centers}$ 下近似认为相同：

$$H(\tau^{abc}) = \begin{cases} + & \text{if } |C^{ab} - C^{cd}| < Th_{centers} \\ - & \text{otherwise} \end{cases} \quad (10)$$

同样的可以得到子集 $\Gamma^3 \subseteq \Gamma^2$ ，即：

$$\tau^{abc} \in \Gamma^3 \Leftrightarrow (\tau^{abc} \in \Gamma^2) \wedge H(\tau^{abc}) \quad (11)$$

最后我们得到的 $\tau_i^{abc} \in \Gamma^3$ 即为候选椭圆 ε_i 。

1.2.2 中心估计

我们知道椭圆具有平行弦的中点共线的性质，由此估计弧线对 p^{ab} 的中心 C^{ab} 。

定义平行弦总数为 N_s ，两组平行弦分别平行于直线 $\overline{L^a M^b}$ 和直线 $\overline{R^b M^a}$ (M 是弧线中点，即 $M^k \equiv e^k_{\lfloor N^k/2 \rfloor}$)，平行弦斜率分别为 r_1^{ab}, r_2^{ab} ，中点构成集合 H_1^{ab}, H_2^{ab} ，两组平行弦的公共中垂线分别为 l_1^{ab}, l_2^{ab} 。 l_1^{ab}, l_2^{ab} 的斜率分别为 t_1^{ab}, t_2^{ab} 。

此处我们采用具有良好鲁棒性的 Theil-Sen 算法来近似估计中垂线。该算法会返回直线 l_1^{ab}, l_2^{ab} 的所有可能的斜率集合 S_1^{ab}, S_2^{ab} ，斜率会在之后的参数估计中使用。我们定义 H_1^{ab}, H_2^{ab} 的分别均值为 $\widetilde{H}_1^{ab}, \widetilde{H}_2^{ab}$ ，那么弧线中心点为：

$$C.x = \frac{\widetilde{H}_2.y - t_2 \widetilde{H}_2.x - \widetilde{H}_1.y + t_1 \widetilde{H}_1.x}{t_2 - t_1} \quad (12)$$

$$C.y = \frac{t_1 \widetilde{H}_2.y - t_2 \widetilde{H}_1.y + t_1 t_2 (\widetilde{H}_1.x - \widetilde{H}_2.x)}{t_2 - t_1} \quad (13)$$

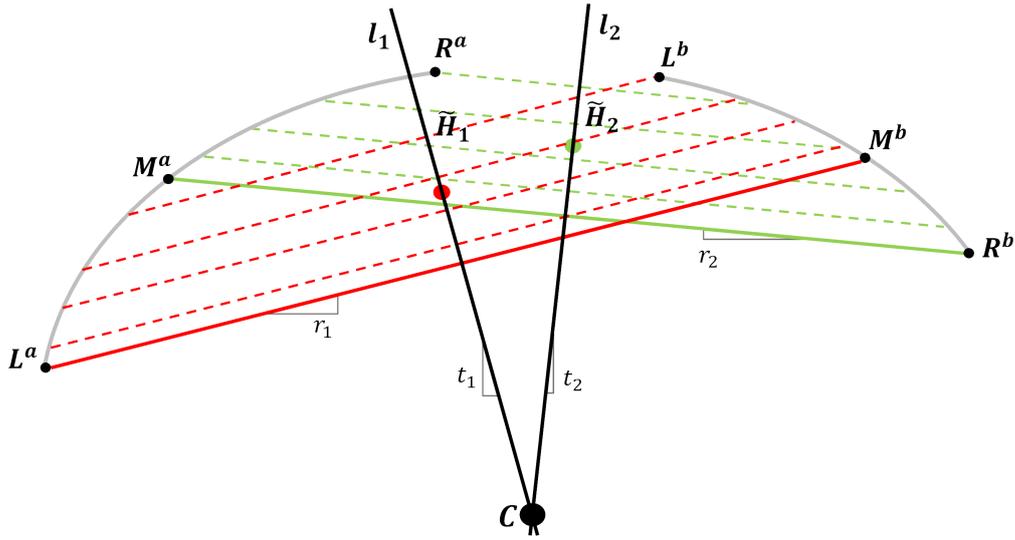


Figure 1.2: 中心估计

1.2.3 参数估计

对经过之前处理得到的三元组 $\tau^{abc} = (p^{ab}, p^{dc})$ 做参数估计，主要估计椭圆的两个半轴长。我们将参数空间分解，并选择半轴比 $N = B/A$ 和方向 ρ 作为新的参数。定义 q_1, q_3 是平行弦的斜率，定义 q_2, q_4 是通过 Theil-Sen 算法估计出的斜率集合，对应关系如下表所示：

Table 1.1: $q_1q_2q_3q_4$ 定义

q_1	q_2	q_3	q_4
r_1^{ab}	S_1^{ab}	r_1^{cd}	S_1^{cd}
r_1^{ab}	S_1^{ab}	r_2^{cd}	S_2^{cd}
r_2^{ab}	S_2^{ab}	r_2^{cd}	S_2^{cd}
r_2^{ab}	S_2^{ab}	r_1^{cd}	S_1^{cd}

令：

$$\gamma = q_1q_2 - q_3q_4 \quad (14)$$

$$\beta = (q_3 + q_4 + 1)(q_1 + q_2) - (q_1q_2 + 1)(q_3 + q_4) \quad (15)$$

可以计算得到：

$$K_+ = \frac{-\beta + \sqrt{\beta^2 + 4\gamma^2}}{2\gamma} \quad (16)$$

$$N_+ = \sqrt{\frac{(q_1 - K_+)(q_2 - K_+)}{(1 + q_1 K_+)(1 + q_2 K_+)}} \quad (17)$$

接下来就可以计算出 N 和 ρ 的值，识别物为圆形标志所以我们希望半轴比 $N = B/A$ 在阈值 Th_{circle} 尽可能接近 1，但是实际道路情况下由于相机广角等问题并不能够保证完美的圆形，较大的阈值又会导致错误率的上升，这就对阈值 Th_{circle} 的选择提出了较高的要求。

$$N = \begin{cases} N_+ & \text{if } |N_+ - 1| \leq Th_{circle} \\ \frac{1}{N_+} & \text{otherwise} \end{cases} \quad (18)$$

$$\rho = \begin{cases} \arctan(K_+) & \text{if } |N_+ - 1| \leq Th_{circle} \\ \arctan(K_+) + \frac{\pi}{2} & \text{otherwise} \end{cases} \quad (19)$$

我们选择最大的 N 和 ρ 的值作为结果。之后计算椭圆的半长轴长 A ，已知三条弧线 $\alpha^a, \alpha^b, \alpha^c$ 边缘点 e_i 的位置 (x_i, y_i) ，令：

$$x_0 = \frac{(x_i - x_c) + (y_i - y_c)K}{\sqrt{K^2 + 1}} \quad (20)$$

$$y_0 = \frac{-(x_i - x_c)K + (y_i - y_c)}{\sqrt{K^2 + 1}} \quad (21)$$

$$A_x = \sqrt{\frac{x_0^2 N^2 + y_0^2}{N^2(K^2 + 1)}} \quad (22)$$

可以得到：

$$A = \frac{A_X}{\cos(\rho)} \quad (23)$$

同样的我们选择最大的 A 作为半长轴长，那么半短轴长为 $B = A \cdot N$

1.3 单标志提取

经过上述处理可以得到候选圆形集合 Ψ^0 。本文主要实现的是对单个交通标志的提取和检测，在得到候选圆形以及计算其参数之后需要进一步根据参数验证圆形。这些条件属于必要非充分条件，可以提高对标志的识别能力。

1.3.1 圆形验证

- 有效性分析候选圆形 $\psi_i \in \Psi^0$ 的参数方程为 $f(x, y) = 1$ ，如果有集合 \mathbb{B} ：

$$\mathbb{B} = \{\forall(x_i, y_i) | (x_i, y_i) \in \psi_i \wedge |f(x_i, y_i) - 1| < 0.1\} \quad (24)$$

如果对于函数 $V : (\psi_i) \rightarrow (+, -)$ 并且满足：

$$V(\psi_i) = \begin{cases} + & \text{if } \frac{|\mathbb{B}|}{|\alpha^a| + |\alpha^b| + |\alpha^c|} - Th_{score} > 0 \\ - & \text{otherwise} \end{cases} \quad (25)$$

符号 $+$ 代笔该圆形 ψ_i 有效。该函数处理得到子集 $\Psi^1 \subseteq \Psi_0$ ，并满足：

$$(\psi \in \Psi^1) \Leftrightarrow (\psi \in \Psi^0) \wedge V(\psi_i) \quad (26)$$

- 聚类分析根据 $\forall i, \psi_i \in \Psi^1$ 的半轴、相位、中心等信息的差异对 ψ_i 做聚类处理：

$$\Upsilon^0 : \sqrt{(\psi_i.x_c - \psi_j.x_c)^2 + (\psi_i.y_c - \psi_j.y_c)^2} - \min(\psi_i.B, \psi_j.B) \times 0.1 < 0 \quad (27)$$

$$\Upsilon^1 : \frac{|\psi_i.A - \psi_j.A|}{\max(\psi_i.A, \psi_j.A)} - 0.1 < 0 \quad (28)$$

$$\Upsilon^2 : \frac{|\psi_i.B - \psi_j.B|}{\max(\psi_i.B, \psi_j.B)} - 0.1 < 0 \quad (29)$$

$$\Upsilon^3 : \left(\frac{\angle(\psi_i.\rho - \psi_j.\rho)}{\pi} - 0.1 < 0 \right) \wedge \left(\frac{\psi_i.B}{\psi_i.A} - 0.9 < 0 \right) \wedge \left(\frac{\psi_j.B}{\psi_j.A} - 0.9 < 0 \right) \quad (30)$$

如果对于函数 $C : (\psi_i, \psi_j) \rightarrow (+, -)$, 有:

$$C(\psi_i, \psi_j) = \begin{cases} + & \text{if } \Upsilon^0 \wedge \Upsilon^1 \wedge \Upsilon^2 \wedge \Upsilon^3 \\ - & \text{otherwise} \end{cases} \quad (31)$$

两个圆形满足 $\Upsilon^0 \wedge \Upsilon^1 \wedge \Upsilon^2 \wedge \Upsilon^3$, 这两个圆形是由同一个圆形衍生而来, 只保留下来一个即可; 否则是两个不同的圆形。该函数处理得到子集 $\Psi^2 \subseteq \Psi_1$, 并满足:

$$(\psi \in \Psi^2) \Leftrightarrow (\psi \in \Psi^1) \wedge C(\psi_i, \psi_j) \quad (32)$$

1.3.2 标志提取

下面做圆形检测算法最后的提取工作, 经过上述过程得到圆形集合 Ψ^2 , 选择 $\psi \in \Psi^2$, 对于函数 $\Delta(\psi_i) \rightarrow (+, -)$:

$$\Delta(\psi_i) = \begin{cases} + & \text{if } (Area(\psi_i) > Th_{distance}) \wedge (isMax(Area(\psi_i))) \\ - & \text{otherwise} \end{cases} \quad (33)$$

最后得到唯一的 ψ , 其中 $Th_{distance}$ 是 ψ_i 面积大小的阈值, 该阈值对圆形识别距离有直接影响。

2 BP 网络识别

BP 网络是一种典型的多层前向网络，又叫误差逆传播网络。一般由输入层、隐藏层和输出层组成，其是输入到输出的非线性映射，避免了传统识别算法中复杂的特征提取和数据重建过程，是在空间或时间上的采样，具有权重共享，减少了权值的数量，降低了计算量、局部区域感知等特点，具有很强的鲁棒性，在输入是多维图像时表现的最为明显。BP 网络识别速度快，准确率高等优点都是交通识别通常采用的模板匹配算法所不能比的。

识别过程为对之前圆形检测得到的图像 ψ 先做 PCA 主成分提取得到 X_k 维度降低到 388×1 ，之后将 X_k 输入到训练好的网络中得到预测结果。

2.1 PCA 降维

PCA 将数据投射到一个低维子空间实现降维，在我们的例子中，原始输入图像的大小不便于直接运算，降维的目标是在压缩数据的同时尽可能地保留原始信息。首先，将输入的图像 ψ 做归一化处理得到 λ ，大小为 40×40 ，设向量 w 为映射向量， W 是包含所有映射向量为列向量的矩阵，那么有优化函数：

$$\begin{aligned} \min tr(W^T A W) \\ \text{s.t. } W^T W = I \end{aligned} \quad (34)$$

其中 A 为协方差矩阵。计算得到最优的 W ，PCA 降维的输出为：

$$X_k = W^T X \quad (35)$$

该操作很好的保留了图像数据的内在信息，同时也避免了无关信息的影响提高了后续处理的速度。

2.2 网络训练

2.2.1 数据集建立

考虑到不同摄像头的参数有差异，传感器采集到的图像数据也有其特殊性，简单地采用开源数据集作为训练数据很难在实测摄像头采集到的图像上得到期待的结果，为了提升系统的表现，我们采用自制数据集的方式进行网络训练。我们开发了脚本，为小车摄像

头全自动地建立数据集，对每一个标签下的交通标识，在摄像头的视角下运行 2.1 提到的圆形检测算法，以稳定帧率连续采集数据。数据集格式类似与德国交通标志识别数据集，但包含了摄像头本身的特征。最后使用的数据集中共有样本 5730 个，将数据集打乱顺序后我们选择前 4500 个样本作为训练集，剩下的作为测试集。

2.2.2 网络模型

本文中综合识别速度和准确性设计了四层全连接网络，输入层为 388 节点，两个隐藏层依次为 300,100 个节点，输出层为 13 个节点，即最终识别 12 种正样本标志，最后一类为负样本。

2.2.3 参数设置

- **激活函数**本文使用 sigmoid 函数，其中设置 $\beta = 1$, $\alpha = 0.6$

$$f(x) = \beta \times \frac{1 - e^{-\alpha x}}{1 + e^{-\alpha x}} \quad (36)$$

- **学习率和动量**神经网络节点的权重值为 w ，损失函数 E 为 softmax 函数， t 为当前迭代次数，学习率 η 和动量 μ 表述如下：

$$\Delta w(t) = -\eta \frac{dE}{dw}(t) + \mu \Delta w(t-1) \quad (37)$$

学习率 η 和动量 μ 均设置为 0.1，总迭代次数为 100 次。

2.2.4 训练结果

最终模型在测试集上得到了 94.3% 的识别准确率，训练过程如下：

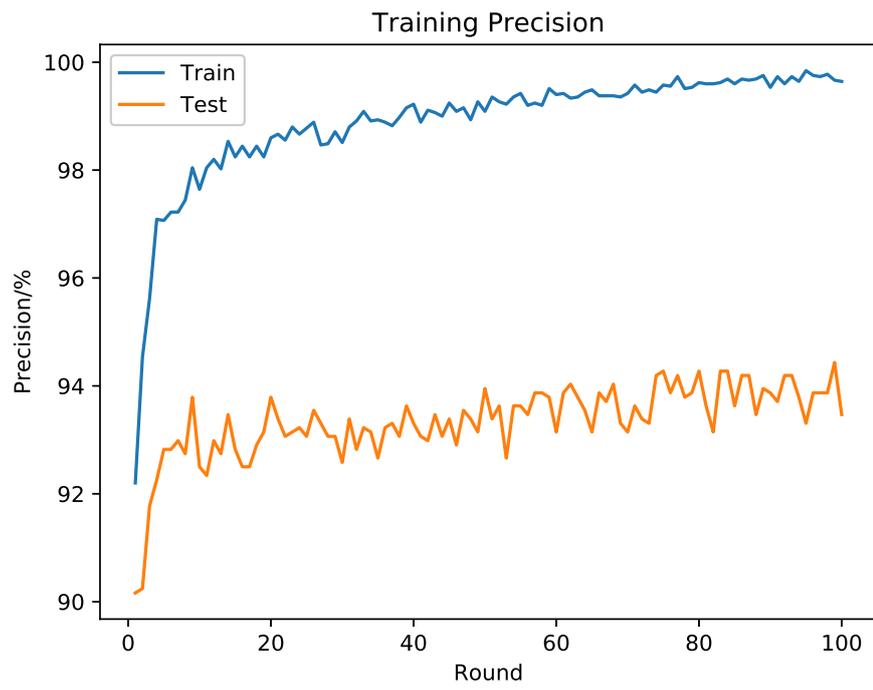


Figure 2.1: 准确率变化

三、测试模型与系统实践

实际测试系统采用基于树莓派作为处理终端，USB 摄像头作为感知设备的智能小车。传感器方面，小车搭载广角鱼眼镜头，手动对焦。处理器方面，小车搭载树莓派 3B+ 开发板和电机伺服电路，拥有 1.4GHz 64 位 4 核 ARM Cortex-A53 CPU，1G LPDDR2 Ram，虽不能胜任高强度的图像处理任务，在此作为网络接受终端以及控制信号终端的处理器性能绰绰有余。

1 搭建视频流

树莓派 3 提供了标准的相机串行接口 CSI 供官方摄像头使用，也可以使用第三方 USB 摄像头，用同样的底层固件库驱动，初始化固件完成后，在 /dev 目录下发现 video0 设备标识摄像头已经准备好被使用。软件方面，选用 linux 下的开源视频监控 mjpegstreamer，配置视频参数为 v4l2 格式，320*240 像素分辨率 16 位 RGB 彩色图像，每一帧的大小约为 150KB。

树莓派 3B 上配备有以太网卡和 wifi 模块，同上位机终端之间可以采用以太网直连或是通过路由器局域网环境连接两种方式，有线方式连接拥有延迟低的优点，但使得小车运动空间被网线限制。无线局域网的方式更符合未来对于自动驾驶的要求，但延时不便控制，受制于路由设备和整体网络带宽。

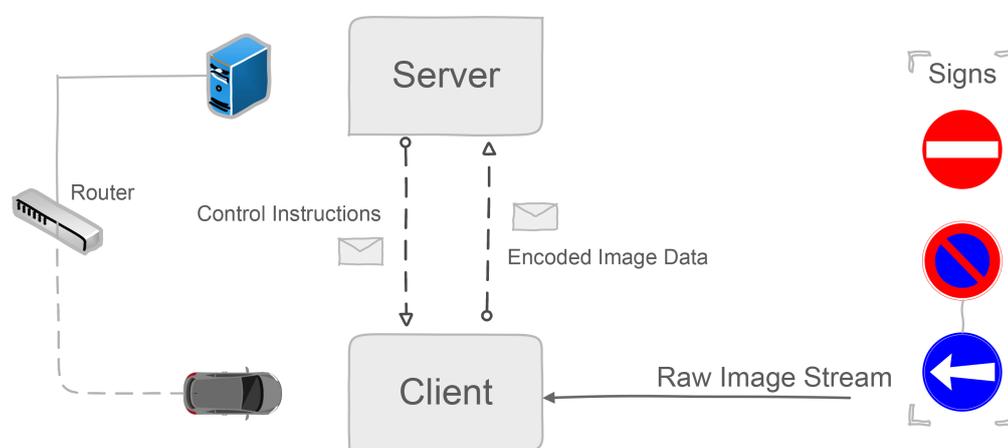


Figure 1.1: 系统整体架构

2 控制指令的分发与解析

小车摄像头捕捉到的视频数据以帧流的形式通过网络串流到上位机，上位机运行视觉处理程序后返回一个字符串形式的控制字，通过 socket 接口下发到客户端小车，小车解析控制字，并与上一次接受到的控制字比较，若两者相同，则保持原先的运动状态，若两者不同，则对新接受到的控制字进行解析后选择进入指定的运动状态。例如，“stop”标志的控制字约定为“5”，则小车若接受到 $msg="5"$ ，并且当前运行的控制字是“1”（前进），这时，小车便会跳转状态为静止。

小车的运动控制状态机如图所示：

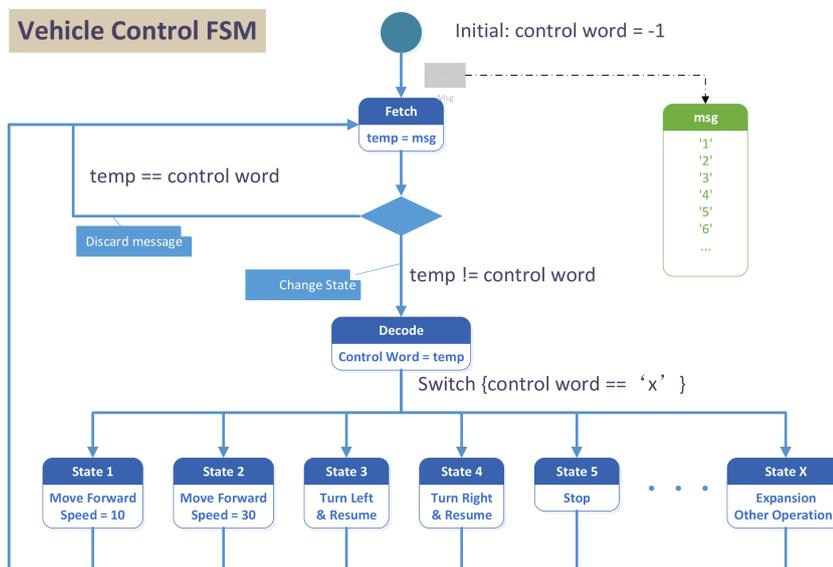


Figure 2.1: 运动控制状态机

3 以视觉流处理为核心的系统网络搭建与扩展

本文测试所用模型可以做进一步扩展，对于运算量较大的视觉图像处理可以在云端运行，在区域性的服务主机运算能力受限时，可以使用云端计算资源，以图像帧为单位发送至公网指定服务器等待响应。服务器端程序的多线程多用户服务能力使得它能够挂载多个客户端，我们目前已经成功试验了单个服务器多客户端场景下的系统运行情况，在未来5G通信时代解决了高速率高并发的难题之后，必然会极大地降低图像数据和指令传输的延时，届时，由云端计算驱动的嵌入式视觉认知系统将不再遥远。

我们设想未来每一台车辆都可以配备有独立的视频串流系统，对不同的相机产生的视

频数据标定后接入 5G 网络，云端服务器运行相同的服务程序，分别响应客户机小车的图像处理请求。实现大规模的标志识别无人驾驶车辆视觉识别服务。

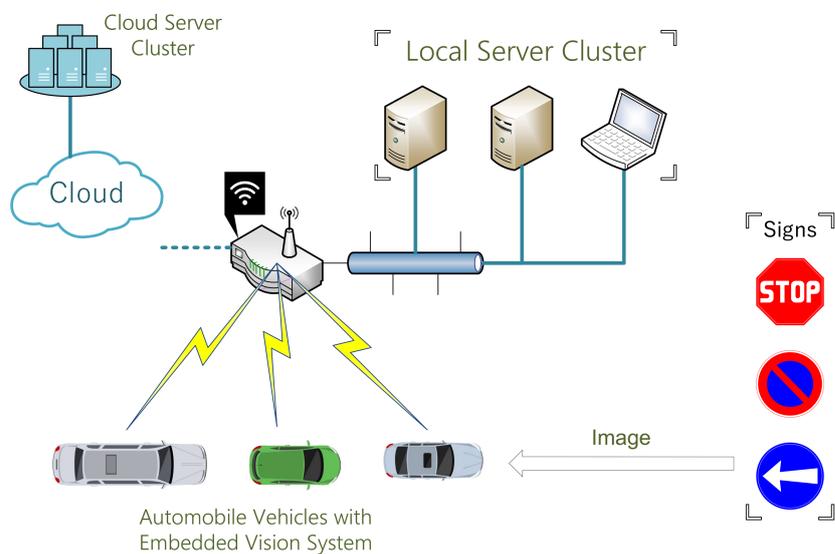


Figure 3.1: 以视觉流处理为核心的系统网络搭建与扩展

四、总结语

本文采用快速的圆形检测算法和基于 BP 神经网络的标志识别模型，搭建并测试了高效的交通标志识别系统。并且将识别算法应用到基于 CS 模型的智能网络小车控制系统中。小车采集到的原始图像数据通过网络接口发送到上位机，运行识别程序后返回控制指令到小车。整套系统运行流畅，识别速度快，且预判距离大，能够模拟实际道路环境下的压力运行，具有一定的实际应用价值，其识别算法和应用方案可以被应用到真实的无人驾驶系统中做进一步测试。