**北京邮电大学人工智能学院**

**计算机视觉课程**

**实时目标跟踪系统设计文档**

****

**课程名称： 计算机视觉**

**实验名称： 实时目标跟踪系统**

**实验完成人：**

**姓名： 黄成梓 学号： 2021213xxx**

**姓名： 巩羽飞 学号： 2021522077**

**姓名： 李佳颉 学号： 2021522084**

**指导教师： 鲁鹏**

**日 期： 2024 年 06 月 01 日**

1. **引言**

本项目实现了一套纯算法实现的实时目标跟踪系统，使用Python实现，并结合一些恰当的外部库，实现了视频实时显示和目标跟踪功能。

报告将从系统架构、技术选型、详细设计、代码实现、性能评估和实验结果分析等方面展开介绍，接下来是报告正文。

**1.1 项目背景**

本项目尝试采用Kernelized Correlation Filters算法，结合一些视觉特征提取方法，如HOG、颜色直方图、边缘特征等，来实现一个纯算法驱动的实时目标跟踪系统。该系统不需要依赖于大规模的训练数据，理论上可以在各种复杂场景中保持较好的泛化性和跟踪效果。同时，该系统还具有实时性，可以在前端网页中进行视频流的实时显示和目标跟踪结果的展示。

这种基于纯算法的实时目标跟踪系统在一些对实时性和泛化性有较高要求的应用场景中可能会有一定的应用前景，因此本项目的研究和开发也具有一定的实践意义。

**1.2 项目功能描述**

简单来说，本项目以Kernelized Correlation Filters算法为基础，通过提取HOG特征（如图1）、颜色直方图特征（如图2）或边缘特征（如图3），更新视频中的每一帧对应的跟踪器状态和ROI，在前端网页中实时显示视频，并同时展示对用户选定的目标跟踪结果。

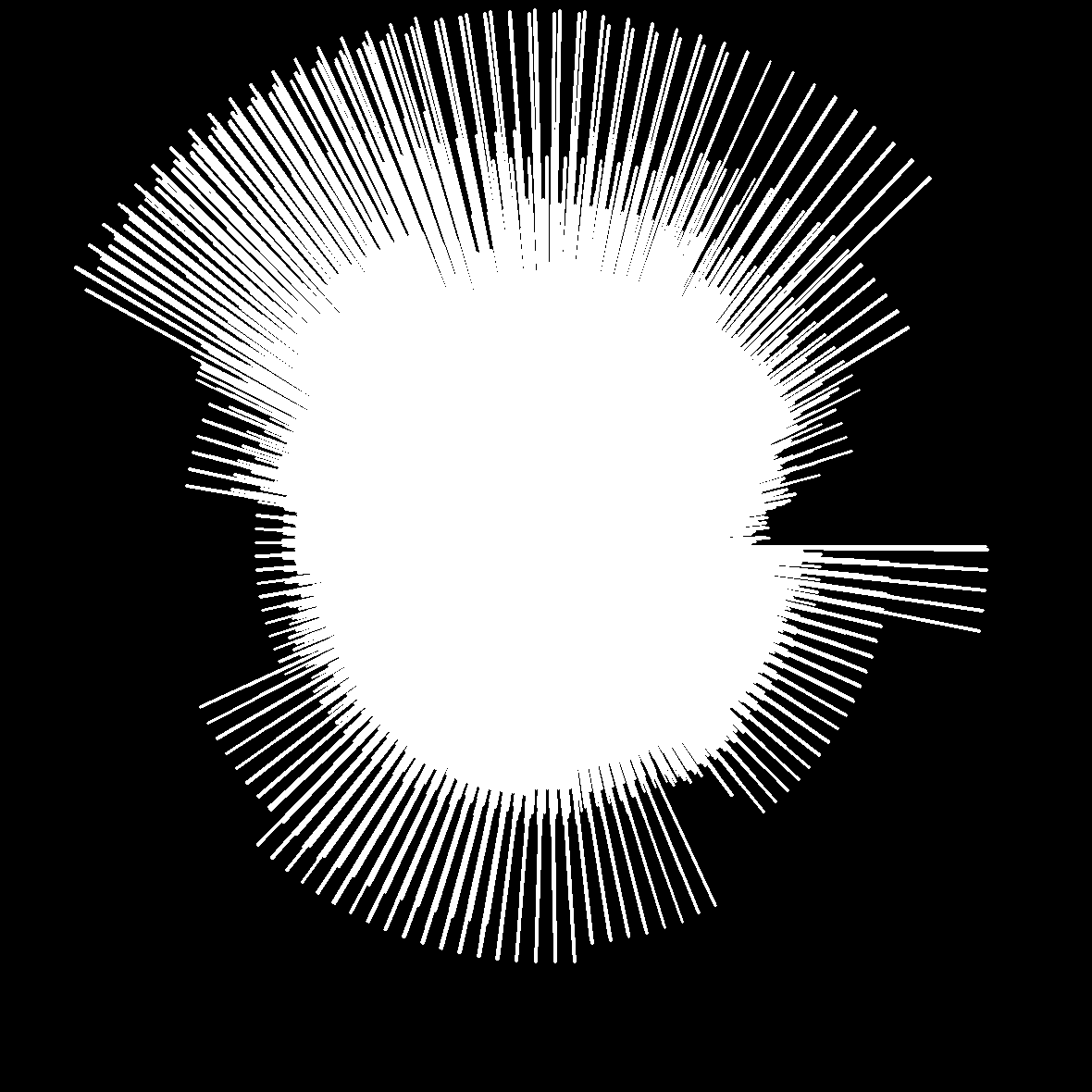


图1：HOG特征示例

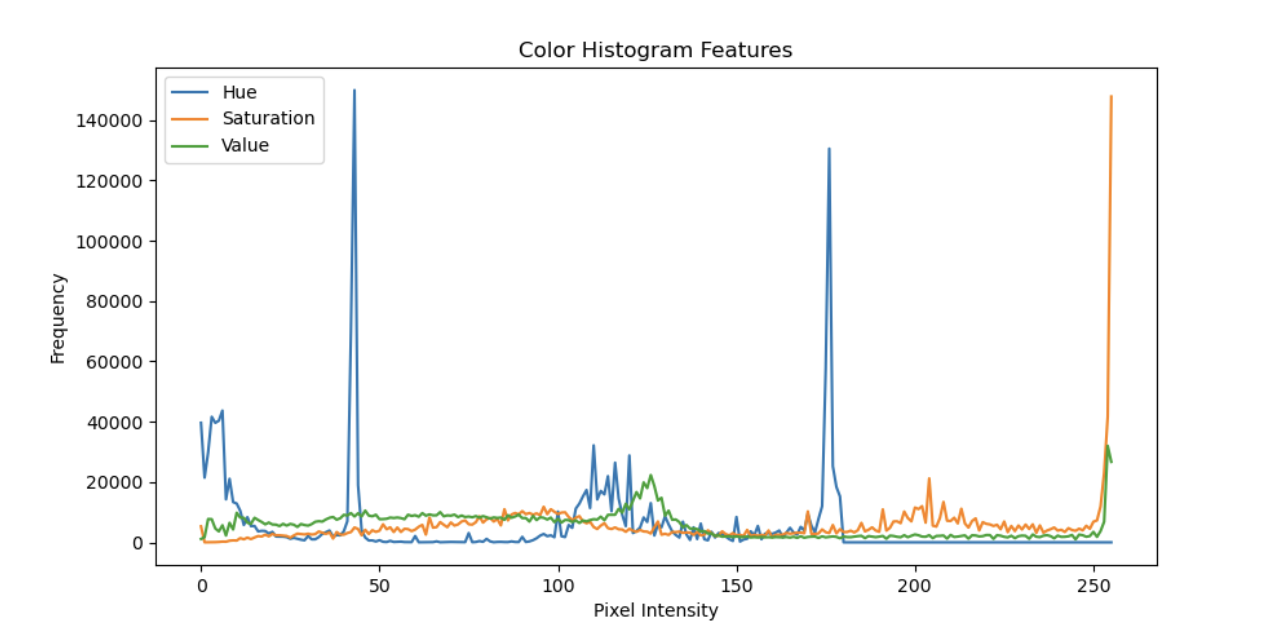


图2：颜色直方图特征示例

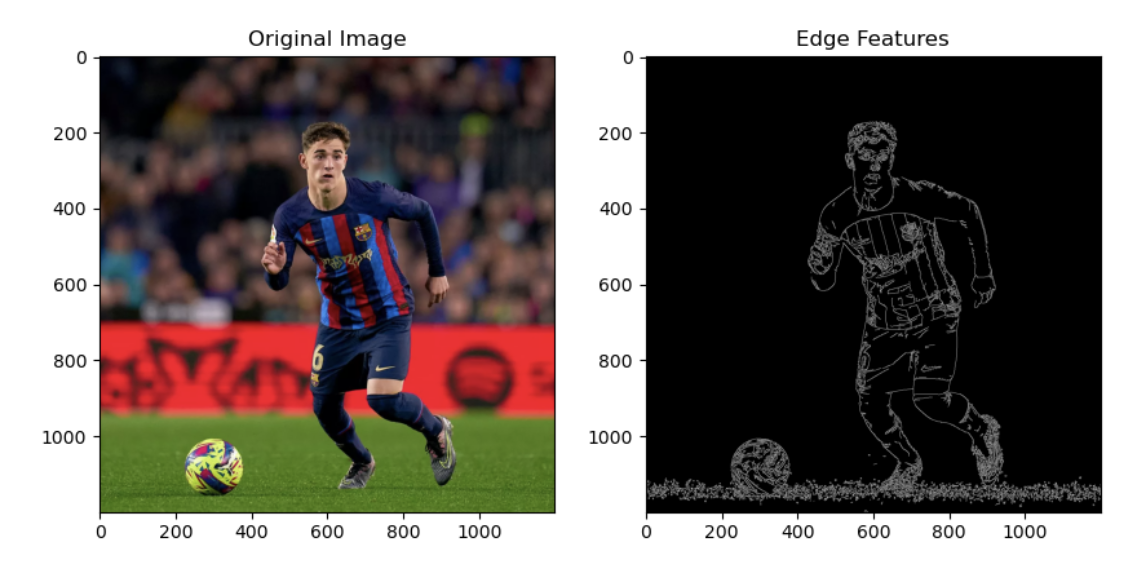


图3：边缘特征示例

1. **系统架构**

### 2.1 总体架构

为保证开发和调试的效率，系统采用模块化的架构，总体的系统框架图如图4所示，主要分为用户界面交互模块、视频帧处理模块、Flask Web Server模块和其他必要组件构成。具体每个模块的功能和任务，将放在第4节中进行详细的介绍。

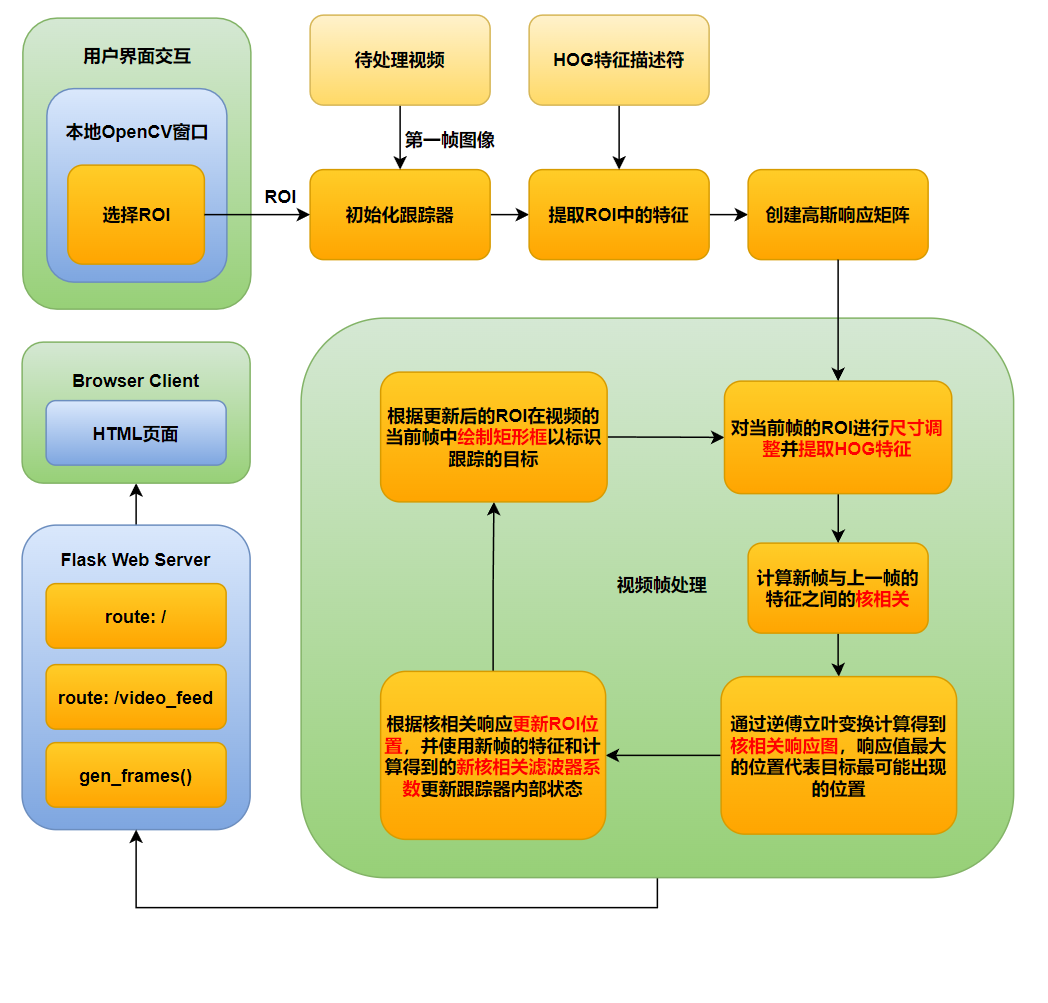


图4：实时目标跟踪系统的系统框架图

**2.2 数据流程说明**

用户首先会指定本地的一个视频作为输入，运行程序后会打开一个本地窗口和一个本地网页，分别用于框选ROI和实时显示跟踪的结果。

随后，待处理的视频会进入初始化跟踪器，利用HOG特征描述符，从第一帧图像开始提取特征并创建高斯响应矩阵。

紧接着，视频帧处理模块会对当前帧的ROI进行尺寸调整，以满足算法的需求，并计算相邻两帧特征之间的核相关，然后通过逆傅里叶变换得到相应的核相关相应图，其中，响应值最大的位置代表目标最可能出现的位置，不过这个位置还不是最终我们要的位置，还要不断更新这个ROI位置，并且使用当前的新帧特征计算一个新的核相关滤波器系数，从而更新我们前面提到的跟踪器的内部状态。

然后，这个更新后的状态才是我们要绘制矩形框的基础，不断的循环这一过程就可以实现实时目标跟踪了。

最后，通过一些由Flask实现的简单路由，可以将我们的结果展示在网页中，这样就完成了数据从输入到输出的整个流程。

**3. 技术选型**

### 3.1 外部库选择

cv2: 即OpenCV，用于处理图像和视频，在这里被用于进行图像处理，如裁剪、缩放和计算HOG特征。OpenCV 提供了许多优化良好的算法和功能，可以帮助更高效地完成计算机视觉任务。

numpy: numpy提供了多维数组对象和大量的函数来操作这些数组，矩阵运算，如傅里叶变换、卷积等操作离不开它，其提供的高度优化的数值计算功能大大提高了代码的性能。

scipy: scipy是numpy的扩展，提供了许多用户友好的和高效的数值例程，如傅里叶变换、线性代数、积分等。在本项目中被用于实现快速傅里叶变换和逆快速傅里叶变换。

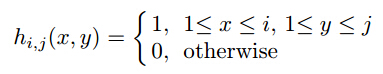
flask: 一个轻量级的Python Web框架可以更好的用于快速开发Web应用程序。flask提供了一个简单而有效的API，使得我们可以快速构建Web应用程序，并且很容易地进行扩展功能的实现。

**3.2 相关工作**

**A. Real-time compressive tracking**

有效降维：

对每一个样本z，将其与一系列矩形滤波器卷积表示。i和j分别是矩形滤波器的宽和高。然后将每一个滤波后的图像表示成列向量，再将他们联结成一个非常高维的多尺度图像特征向量。维度通常在十的六次方到十的十次方。



分析低维压缩特征：

低维特征v中的每个分量是不同尺度下空间分布矩形特征的线性组合。由于测量矩阵的系数可能是正的或负的，压缩特征计算相对像素值差的方式非常类似于Haar-like特征，如图5所示。

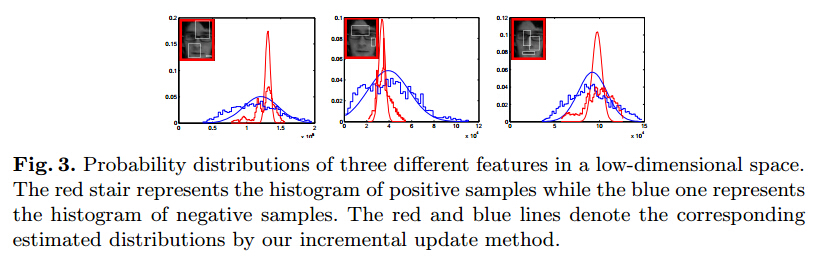
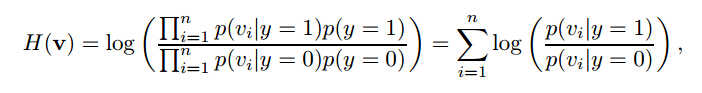


图5：低维空间中三种不同特征的概率分布，红、蓝色阶梯分别表示正、负样本的直方图

分类器的构造与更新：

对于每个样本z，它的低维表示是v。我们假定v中所有元素是独立分布的，并且用朴素贝叶斯分类器建模，其中y是表示样本标签的二元变量，假定p(y=1)=p(y=0)。



**B. Robust\_Object\_Tracking\_with\_Online\_Multiple\_Instance\_Learning**

MIL跟踪算法：

由前一帧目标位置得到当前帧目标可能位置的集合，并计算相应的特征向量（使用Haar-like特征）。使用MIL分类器计算集合中所有图像块的概率（图像块x中包含目标的概率）更新跟踪器目标位置，选出概率最大的图像块作为当前帧的目标位置。得到目标位置以后，扩展样本。

**与KCF的比较：**

与上面算法相比，KCF用了核技巧，核技巧用一句话来概括的话，就是“更高维空间中内积的快速运算”。KCF用核技巧通过核函数隐式地将X空间映射到Z空间中，不必显示地计算，就可以实现快速运算。

此外，KCF的检测更为快速。为了进一步提高处理速度，KCF对检测的过程也进行了简化加速。检测过程，也就是利用回归器与输入数据相关的过程，通过傅立叶变换求出两个单通道向量的核相关内积。

**4. 系统详细设计**

**4.1 视频输入模块**

……

**4.2 特征提取模块**

……

**4.3 目标跟踪算法模块**

……

**4.4 结果展示模块**

显然，结果展示模块是整个视觉跟踪系统非常重要的组成部分，其主要功能是将跟踪算法的输出结果以直观、友好的方式呈现给用户。对于这个项目而言，结果展示模块可以包含以下几个主要部分：

1. **实时视频显示：**

* 在界面上实时显示输入的视频流；
* 将跟踪目标用边框或其他标记在视频画面上高亮显示；
* 可以支持暂停、快进等基本操作，方便用户观察跟踪过程。

1. **跟踪状态信息展示：**

* 实时显示目标的位置坐标、尺度大小等跟踪状态参数；
* 可以以数字、图表等形式呈现跟踪质量指标，如跟踪精度、帧率等；
* 展示跟踪算法的运行时间、资源占用等性能指标。

1. **结果导出功能：**

* 提供将跟踪结果导出为视频文件、图片序列等格式的功能；
* 支持导出跟踪轨迹数据，如目标位置、尺度变化等，方便后续分析。

1. **交互式操作：**

* 允许用户手动选择跟踪目标，并观察算法的跟踪效果；
* 提供参数调整界面，使用户可以根据实际需求对算法进行微调；
* 支持多目标跟踪时，提供目标选择、切换等交互功能。

1. **辅助可视化：**

* 将跟踪过程中的中间结果，如响应图、特征图等可视化展示；
* 帮助用户直观了解算法内部运作机制，有助于调试和优化。

**5. 系统实现**

**5.1 代码结构**

……

**5.2 测试数据与结果**

视频帧率保留程度还算可以，能达到80%，以下是一个原本帧率为30的视频的跟踪效果：



图6：跟踪效果与帧率显示

**5.3 实验结果分析**

……

**6. 总结与展望**

**6.1 项目总结**

我们采用KCF核相关滤波器作为跟踪模型，能够高效地进行目标跟踪。KCF 通过核技巧将线性滤波器扩展到非线性特征空间，从而提高了跟踪性能。利用循环移位技巧，大大降低了计算复杂度，使得该算法能够实现实时跟踪。循环移位技巧允许在频域高效计算滤波器的响应，避免了空域滤波的耗时计算。

该算法具有较好的鲁棒性，能够处理目标尺度变化、遮挡等复杂场景。通过自适应的尺度估计和通道可靠性机制，有效提高了跟踪稳定性。并且，算法计算高效，在普通硬件上即可实现超过60帧率的实时跟踪，非常适合嵌入式系统和移动设备应用。

**6.2 不足与改进方向**

目前 KCF 算法主要基于手工设计的视觉特征，如HOG和Color Names，这些特征可能无法充分描述复杂场景下的目标外观。可以考虑引入深度学习特征，以提高跟踪精度。同时，尺度自适应机制还有进一步优化空间，可以探索更鲁棒的尺度估计方法，提高跟踪性能。

该算法在处理快速运动目标、严重遮挡等极端情况下仍存在一定局限性，需要进一步提高算法的鲁棒性。当前算法主要针对单目标跟踪，如何扩展到多目标跟踪也是一个值得探索的方向。可以考虑将该算法与深度学习跟踪器等其他先进跟踪方法进行融合，充分发挥各自的优势，进一步提升跟踪性能。

**7. 参考文献**

[1] Henriques， J. F.， Caseiro， R.， Martins， P.， & Batista， J. (2015). High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence， 37(3)， 583-596.

[2] Danelljan， M.， Häger， G.， Khan， F. S.， & Felsberg， M. (2014). Accurate scale estimation for robust visual tracking. In British Machine Vision Conference.

[3] Danelljan， M.， Häger， G.， Khan， F. S.， & Felsberg， M. (2015). Convolutional features for correlation filter based visual tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshops.

[4]Lukezic， A.， Vojir， T.， Čehovin Zajc， L.， Matas， J.， & Kristan， M. (2017). Discriminative correlation filter with channel and spatial reliability. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[5]Bhat， G.， Danelljan， M.， Shahbaz Khan， F.， & Felsberg， M. (2019). Learning discriminative model prediction for tracking. In Proceedings of the IEEE International Conference on Computer Vision.

Li， B.， Wu， W.， Wang， Q.， Zhang， F.， Xing， J.， & Yan， J. (2019). Siamrpn++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[6]Zhu， Z.， Wang， Q.， Li， B.， Wu， W.， Yan， J.， & Hu， W. (2018). Distractor-aware siamese networks for visual object tracking. In Proceedings of the European Conference on Computer Vision.

[7]Guo， Q.， Feng， W.， Zhou， C.， Huang， R.， Wan， L.， & Wang， S. (2017). Learning dynamic siamese network for visual object tracking. In Proceedings of the IEEE International Conference on Computer Vision.

**8. 附录**

**8.1 公式推导**

（待补充）

**8.2 讲解PPT、代码源文件和展示视频**

请见附件