```
declare module '*.vue' {
  import type { DefineComponent } from 'vue'
  const component: DefineComponent<{}, {}, any>
  export default component
}
<!DOCTYPE html>
<html lang="">
  <head>
    <meta charset="UTF-8">
    <link rel="icon" href="/favicon.ico">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Vite App</title>
  </head>
  <body>
    <div id="app"></div>
    <script type="module" src="/src/main.ts"></script>
  </body>
</html>
import { fileURLToPath, URL } from 'node:url'
import { defineConfig } from 'vite'
import vue from '@vitejs/plugin-vue'
import vueDevTools from 'vite-plugin-vue-devtools'
export default defineConfig({
  plugins: [
    vue(),
    vueDevTools(),
  ],
  resolve: {
    alias: {
      '@': fileURLToPath(new URL('./src', import.meta.url))
    },
  },
})
import { createRouter, createWebHistory } from 'vue-router';
import FileUploadVue from '@/components/FileUpload.vue';
const router = createRouter({
    history: createWebHistory(import.meta.env.BASE_URL),
    routes: [
        {
            path: '/',
            name: 'home',
            component: FileUploadVue,
        },
    ],
});
router.beforeEach((to, from, next) => {
    next();
});
export default router;
import './assets/main.css'
```

```
import { createApp } from 'vue'
import App from './App.vue'
import router from './router'
const app = createApp(App)
app.use(router)
app.mount('#app')
<!-- This icon is from <https:
<template>
  <svg
    xmlns="http://www.w3.org/2000/svg"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    aria-hidden="true"
    role="img"
    class="iconify iconify--mdi"
    width="24"
    height="24"
    preserveAspectRatio="xMidYMid meet"
    viewBox="0 0 24 24"
  >
    <path
      d="M20 18v-4h-3v1h-2v-1H9v1H7v-1H4v4h16M6.33 8l-1.74 4H7v-1h2v1h6v-1h2v1h2.41l-1.74-4H6.33M9 5v1h6V5H9m12.84 7.61c.1.22.16.48.16.8V18c0 .53-.21 1-.6 1.41c-.4.4-.85.59-1.4.59H4c-.55 0-1-.19-1.4-.59C2.21 19 2 18.53 2 18v-4.59c0-.32.06-.58.16-.8L4.5 7.22C4.84 6.41 5.45 6 6.33 6H7V5c0-.55.18-1 .57-1.41C7.96 3.2 8.44 3 9 3h6c.56 0 1.04.2 1.43.59c.39.41.57.86.57 1.41v1h.67c.88 0 1.49.41 1.83 1.22l2.34 5.39z"
      fill="currentColor"
    ></path>
  </svg>
</template>
<template>
  <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="currentColor">
    <path
      d="M10 3.22l-.61-.6a5.5 5.5 0 0 0-7.666.105 5.5 5.5 0 0 0-.114 7.665L10 18.78l8.39-8.4a5.5 5.5 0 0 0-.114-7.665 5.5 5.5 0 0 0-7.666-.105l-.61.61z"
    />
  </svg>
</template>
<template>
  <svg xmlns="http://www.w3.org/2000/svg" width="18" height="20" fill="currentColor">
    <path
      d="M11.447 8.894a1 1 0 1 0-.894-1.789l.894 1.789zm-2.894-.789a1 1 0 1 0 .894 1.789l-.894-1.789zm0 1.789a1 1 0 1 0 .894-1.789l-.894 1.789zM7.447 7.106a1 1 0 1 0-.894 1.789l.894-1.789zM10 9a1 1 0 1 0-2 0h2zm-2 2.5a1 1 0 1 0 2 0H8zm9.447-5.606a1 1 0 1 0-.894-1.789l.894 1.789zm-2.894-.789a1 1 0 1 0 .894 1.789l-.894-1.789zm2 .789a1 1 0 1 0 .894-1.789l-.894 1.789zm-1.106-2.789a1 1 0 1 0-.894 1.789l.894-1.789zM18 5a1 1 0 1 0-2 0h2zm-2 2.5a1 1 0 1 0 2 0h-2zm-5.447-4.606a1 1 0 1 0 .894-1.789l-.894 1.789zM9 1l.447-.894a1 1 0 0 0-.894 0L9 1zm-2.447.106a1 1 0 1 0 .894 1.789l-.894-1.789zm-6 3a1 1 0 1 0 .894 1.789L.553 4.106zm2.894.789a1 1 0 1 0-.894-1.789l.894 1.789zm-2-.789a1 1 0 1 0-.894 1.789l.894-1.789zm1.106 2.789a1 1 0 1 0 .894-1.789l-.894 1.789zM2 5a1 1 0 1 0-2 0h2zM0 7.5a1 1 0 1 0 2 0H0zm8.553 12.394a1 1 0 1 0 .894-1.789l-.894 1.789zm-1.106-2.789a1 1 0 1 0-.894 1.789l.894-1.789zm1.106 1a1 1 0 1 0 .894-1.789l-.894 1.789zm-
```

```
1.789zm2.894.789a1 1 0 1 0-.894-1.789l.894 1.789zM8 19a1 1 0 1 0 2 0H8zm2-2.5a1 1 0 1 0-2 0h2zm-7.447.394a1 1 0 1 0 .894-1.789l-.894 1.789zM1 15H0a1 1 0 0 0 .553.894L1 15zm1-2.5a1 1 0 1 0-2 0h2zm12.553 2.606a1 1 0 1 0 .894 1.789l-.894-1.789zM17 15l.447.894A1 1 0 0 0 18 15h-1zm1-2.5a1 1 0 1 0-2 0h2zm-7.447-5.394l-2 1 .894 1.789 2-1-.894-1.789zm-1.106 1l-2-1-.894 1.789 2 1 .894-1.789zM8 9v2.5h2V9H8zm8.553-4.894l-2 1 .894 1.789 2-1-.894-1.789zm.894 0l-2-1-.894 1.789 2 1 .894-1.789zM16 5v2.5h2V5h-2zm-4.553-3.894l-2-1-.894 1.789 2 1 .894-1.789zm-2.894-1l-2 1 .894 1.789 2-1L8.553.106zM1.447 5.894l2-1-.894-1.789-2 1 .894 1.789zm-.894 0l2 1 .894-1.789-2-1-.894 1.789zM0 5v2.5h2V5H0zm9.447 13.106l-2-1-.894 1.789 2 1 .894-1.789zm0 1.789l2-1-.894-1.789-2 1 .894 1.789zM10 19v-2.5H8V19h2zm-6.553-3.894l-2-1-.894 1.789 2 1 .894-1.789zM2 15v-2.5H0V15h2zm13.447 1.894l2-1-.894-1.789-2 1 .894 1.789zM18 15v-2.5h-2V15h2z"
      />
  </svg>
</template>
<template>
  <svg xmlns="http://www.w3.org/2000/svg" width="20" height="17" fill="currentColor">
    <path
      d="M11 2.253a1 1 0 1 0-2 0h2zm-2 13a1 1 0 1 0 2 0H9zm.447-12.167a1 1 0 1 0 1.107-1.666L9.447 3.086zM1 2.253L.447 1.42A1 1 0 0 0 0 2.253h1zm0 13H0a1 1 0 0 0 .553.833L1 15.253zm8.447.833a1 1 0 1 0 1.107-1.666l-1.107 1.666zm0-14.666a1 1 0 1 0 1.107 1.666L9.447 1.42zM19 2.253h1a1 1 0 0 0-.447-.833L19 2.253zm0 13l-.553.833A1 1 0 0 0 20 15.253h-1zm-9.553-.833a1 1 0 1 0 1.107 1.666L9.447 14.42zM9 2.253v13h2v-13H9zm1.553-.833C9.203.523 7.42 0 5.5 0v2c1.572 0 2.961.431 3.947 1.086l1.107-1.666zM5.5 0C3.58 0 1.797.523.447 1.42l1.107 1.666C2.539 2.431 3.928 2 5.5 2V0zM0 2.253v13h2v-13H0zm1.553 13.833C2.539 15.431 3.928 15 5.5 15v-2c-1.92 0-3.703.523-5.053 1.42l1.107 1.666zM5.5 15c1.572 0 2.961.431 3.947 1.086l1.107-1.666C9.203 13.523 7.42 13 5.5 13v2zm5.053-11.914C11.539 2.431 12.928 2 14.5 2V0c-1.92 0-3.703.523-5.053 1.42l1.107 1.666zM14.5 2c1.573 0 2.961.431 3.947 1.086l1.107-1.666C18.203.523 16.421 0 14.5 0v2zm3.5 1.253v13h2v-13h-2zm1.553 13.833C18.203 13.523 16.421 13 14.5 13v2c1.573 0 2.961.431 3.947 1.086l1.107-1.666zM14.5 13c-1.92 0-3.703.523-5.053 1.42l1.107 1.666C11.539 15.431 12.928 15 14.5 15v-2z"
      />
  </svg>
</template>
<template>
  <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="currentColor">
    <path
      d="M15 4a1 1 0 1 0 0 2V4zm0 11v-1a1 1 0 0 0-1 1h1zm0 4l-.707.707A1 1 0 0 0 16 19h-1zm-4-4l.707-.707A1 1 0 0 0 11 14v1zm-4.707-1.293a1 1 0 0 0-1.414 1.414l1.414-1.414zm-.707.707l-.707.707.707.707zM9 11v-1a1 1 0 0 0-.707.293L9 11zm-4 0h1a1 1 0 0 0-1-1v1zm0 4H4a1 1 0 0 0 1.707.707L5 15zm10-9h2V4h-2v2zm2 0a1 1 0 0 1 1 1h2a3 3 0 0 0-3-3v2zm1 1v6h2V7h-2zm0 6a1 1 0 0 1-1 1v2a3 3 0 0 0 3-3h-2zm-1 1h-2v2h2v-2zm-3 1v4h2v-4h-2zm3.293 3.707l-4-4-1.414 1.414 4 4 1.414-1.414zM11 14H7v2h4v-2zm-4 0c-.276 0-.525-.111-.707-.293l-1.414 1.414C5.42 15.663 6.172 16 7 16v-2zm-.707 1.121l3.414-3.414-1.414-1.414-3.414 3.414 1.414 1.414zM9 12H4v2h5v-2zm-5 0c-.829 0-1.579.337-2.121.879l1.414 1.414C2.475 14.111 2.724 14 3 14v-2zm-2.121.879C.337 13.421 0 14.171 0 15h2c0-.276.111-.525.293-.707L.879 12.879zM0 15c0 .829.337 1.579.879 2.121l1.414-1.414C2.111 15.525 2 15.276 2 15H0zm.879 2.121C1.421 17.663 2.171 18 3 18v-2c-.276 0-.525-.111-.707-.293L.879 17.121zM3 18h4v-2H3v2zm4 0c.829 0 1.579-.337 2.121-.879l-1.414-1.414C7.525 15.889 7.276 16 7 16v2zm2.121-.879C9.663 16.579 10 15.829 10 15H8c0 .276-.111.525-.293.707l1.414 1.414zM10 15v-4H8v4h2zm6.586-.586l-.586-1.414-1.414.586.586 1.414 1.414-.586z"
      />
  </svg>
</template>
<template>
```

```
<div class="file-uploader">
  <!-- 主要内容区域：文件上传和视频播放并列 -->
  <div class="main-content">
    <!-- 左侧上传区域 -->
    <div class="upload-container">
      <div class="upload-section">
        <div class="file-input-group">
          <h3>视频文件 (MP4)</h3>
          <input
            type="file"
            accept="video/mp4"
            @change="handleVideoSelect"
            :disabled="uploading"
          />
          <div v-if="videoFile" class="selected-file">
            已选择: {{ videoFile.name }}
          </div>
        </div>
        <div class="file-input-group">
          <h3>文本文件 (TXT)</h3>
          <input
            type="file"
            accept=".txt"
            @change="handleTextSelect"
            :disabled="uploading"
          />
          <div v-if="textFile" class="selected-file">
            已选择: {{ textFile.name }}
          </div>
        </div>
      </div>
      <!-- 上传按钮 -->
      <button
        @click="handleUploadAndProcess"
        :disabled="!canUpload || uploading"
        class="action-button"
      >
        {{ uploading ? '上传中...' : '上传文件' }}
      </button>
      <!-- 处理按钮 -->
      <button
        v-if="groupId && !processing"
        @click="processFiles"
        :disabled="processing || isProcessed"
        class="action-button process-button"
      >
        {{ isProcessed ? '处理完成' : '开始处理' }}
      </button>
      <!-- 上传进度条 -->
      <div v-if="uploading" class="progress-container">
```

```
            <div class="progress-label">上传进度</div>
            <div class="progress-bar">
              <div class="progress-fill" :style="{ width: `${uploadProgress}%` }"></div>
            </div>
            <div class="progress-text">{{ uploadProgress }}%</div>
          </div>
          <!-- 处理进度状态 -->
          <div v-if="processing" class="progress-container">
            <div class="progress-label">处理中...</div>
            <div class="processing-spinner"></div>
          </div>
          <!-- 简化的状态消息 -->
          <div v-if="message" class="status-message" :class="messageType">
            {{ message }}
          </div>
        </div>
        <!-- 右侧视频区域 -->
        <div v-if="processedVideoUrl && isProcessed" class="video-section">
          <div class="video-player-container">
            <h4>处理后的视频</h4>
            <!-- 视频播放器 -->
            <div class="video-wrapper">
              <video
                ref="videoPlayer"
                class="video-player"
                controls
                @error="handleVideoError"
                @loadeddata="handleVideoLoaded"
                crossorigin="anonymous"
              >
                <source :src="processedVideoUrl" type="video/mp4; codecs='avc1.42E01E,
mp4a.40.2'">
                您的浏览器不支持 HTML5 视频播放
              </video>
              <!-- 加载提示 -->
              <div v-if="isVideoLoading" class="video-loading">
                <div class="loading-spinner"></div>
                <span>视频加载中...</span>
              </div>
            </div>
            <!-- 视频控制按钮 -->
            <div class="video-controls">
              <button
                @click="handleDownload"
                class="control-button download-button"
              >
                <span class="button-icon"></span>
                下载视频
              </button>
              <button
```

```
                    v-if="videoError"
                    @click="reloadVideo"
                    class="control-button reload-button"
                  >
                    <span class="button-icon">↻</span>
                    重新加载
                  </button>
                </div>
                <!-- 错误提示 -->
                <div v-if="videoError" class="video-error">
                  <p>{{ videoError }}</p>
                </div>
              </div>
            </div>
          </div>
          <!-- 底部日志区域 -->
          <div v-if="showProcessLog" class="result-section">
            <h3>处理结果</h3>
            <div class="log-header">
              <span>{{ isProcessed ? '处理完成' : (hasError ? '处理出错' : '处理状态') }}</span>
            </div>
            <!-- 处理日志 -->
            <div v-if="logContent" class="log-content-wrapper">
              <pre class="log-content">{{ logContent }}</pre>
            </div>
          </div>
          <!-- 错误信息显示 -->
          <div v-if="logContent && logContent.includes('错误')" class="error-container">
            <div class="error-message">
              <h4>处理过程中遇到问题：</h4>
              <pre class="error-details">{{ logContent }}</pre>
            </div>
          </div>
        </div>
      </template>
      <script setup lang="ts">
      import { ref, computed, onMounted } from 'vue';
      import axios from 'axios';
      const videoFile = ref<File | null>(null);
      const textFile = ref<File | null>(null);
      const uploading = ref(false);
      const processing = ref(false);
      const uploadProgress = ref(0);
      const message = ref('');
      const messageType = ref<'success' | 'error'>('success');
      const groupId = ref<number | null>(null);
      const isProcessed = ref(false);
      const showProcessLog = ref(false);
      const hasError = ref(false);
      const logContent = ref('');
```

```
const detailedError = ref('');
const showDetailedError = ref(false);
const processedVideoUrl = ref<string | null>(null);
const videoPlayer = ref<HTMLVideoElement | null>(null);
const showResult = ref(false);
const isVideoLoading = ref(true);
const videoError = ref<string | null>(null);
const canUpload = computed(() => {
  return videoFile.value && textFile.value && !uploading.value;
});
const handleVideoSelect = (e: Event) => {
  const input = e.target as HTMLInputElement;
  if (input.files && input.files[0]) {
    const file = input.files[0];
    if (file.type === 'video/mp4') {
      videoFile.value = file;
      resetVideoState();
      showProcessLog.value = false;
      isProcessed.value = false;
    } else {
      showMessage('请选择 MP4 格式的视频文件', 'error');
    }
  }
};
const handleTextSelect = (e: Event) => {
  const input = e.target as HTMLInputElement;
  if (input.files && input.files[0]) {
    const file = input.files[0];
    if (file.type === 'text/plain') {
      textFile.value = file;
    } else {
      showMessage('请选择 TXT 格式的文本文件', 'error');
    }
  }
};
const showMessage = (msg: string, type: 'success' | 'error') => {
  message.value = msg;
  messageType.value = type;
  setTimeout(() => {
    message.value = '';
  }, 3000);
};
const updateLogContent = (content: any) => {
  if (typeof content === 'string') {
    logContent.value = content;
  } else if (content && typeof content === 'object') {
    try {
      if (content.message) {
        logContent.value = content.message;
      } else if (content.status && content.result) {
```

```
            logContent.value = `状态: ${content.status}\n 结果: ${JSON.stringify(content.result, null, 2)}`;
          } else {
            logContent.value = JSON.stringify(content, null, 2);
          }
        } catch (e) {
          logContent.value = '无法解析日志内容';
        }
      } else {
        logContent.value = '无日志内容';
      }
    };
    const getVideoFileName = () => {
      if (videoFile.value) {
        const originalName = videoFile.value.name;
        const baseName = originalName.replace(/\.mp4$/, '');
        return `${baseName}_processed.mp4`;
      }
      return 'processed_video.mp4';
    };
    const handleVideoLoaded = () => {
      isVideoLoading.value = false;
      videoError.value = null;
    };
    const handleVideoError = async (e: Event) => {
      const target = e.target as HTMLVideoElement;
      isVideoLoading.value = false;
      let errorMessage = '视频加载失败: ';
      if (target.error) {
        switch (target.error.code) {
          case MediaError.MEDIA_ERR_ABORTED:
            errorMessage += '加载被中断';
            break;
          case MediaError.MEDIA_ERR_NETWORK:
            errorMessage += '网络错误';
            break;
          case MediaError.MEDIA_ERR_DECODE:
          case MediaError.MEDIA_ERR_SRC_NOT_SUPPORTED:
            errorMessage += '视频格式不支持';
            await retryLoadVideo();
            break;
          default:
            errorMessage += `未知错误 (${target.error.code})`;
        }
      }
      videoError.value = errorMessage;
      console.error('视频加载错误:', {
        url: processedVideoUrl.value,
        error: target.error,
        errorMessage
      });
```

```
};
const reloadVideo = () => {
    if (videoPlayer.value) {
        isVideoLoading.value = true;
        videoError.value = null;
        videoPlayer.value.load();
    }
};
const handleDownload = async () => {
    if (!processedVideoUrl.value) {
        showMessage('视频文件不可用', 'error');
        return;
    }
    try {
        const timestamp = new Date().getTime();
        const downloadUrl = `${processedVideoUrl.value}?t=${timestamp}`;
        const checkResponse = await fetch(downloadUrl, {
            method: 'HEAD'
        });
        if (!checkResponse.ok) {
            throw new Error(`文件访问失败: ${checkResponse.status}`);
        }
        const response = await fetch(downloadUrl);
        const blob = await response.blob();
        const contentType = response.headers.get('content-type');
        if (!contentType || !contentType.includes('video/mp4')) {
            console.warn('警告: 响应的内容类型不是 video/mp4:', contentType);
        }
        const url = window.URL.createObjectURL(blob);
        const link = document.createElement('a');
        link.href = url;
        link.download = getVideoFileName();
        document.body.appendChild(link);
        link.click();
        setTimeout(() => {
            document.body.removeChild(link);
            window.URL.revokeObjectURL(url);
        }, 100);
        showMessage('下载开始', 'success');
    } catch (error: any) {
        console.error('下载失败:', error);
        showMessage(`下载失败: ${error.message}`, 'error');
        logContent.value += `下载错误: ${error.message}\n`;
    }
};
const checkVideoAccessibility = async (url: string) => {
    try {
        const response = await fetch(url, {
            method: 'HEAD',
            mode: 'cors'
```

```
        });
        if (!response.ok) {
            throw new Error(`HTTP error! status: ${response.status}`);
        }
        return true;
    } catch (e) {
        console.error('视频访问检查失败:', e);
        return false;
    }
};
const checkVideoFormat = async (url: string): Promise<boolean> => {
    try {
        const response = await fetch(url, {
            method: 'HEAD'
        });
        if (!response.ok) {
            throw new Error(`HTTP error! status: ${response.status}`);
        }
        const contentType = response.headers.get('content-type');
        return contentType !== null && contentType.includes('video/mp4');
    } catch (e) {
        console.error('视频格式检查失败:', e);
        return false;
    }
};
const retryLoadVideo = async () => {
    if (!processedVideoUrl.value) return;
    try {
        isVideoLoading.value = true;
        videoError.value = null;
        const timestamp = new Date().getTime();
        const newUrl = `${processedVideoUrl.value}?t=${timestamp}`;
        const isValidFormat = await checkVideoFormat(newUrl);
        if (!isValidFormat) {
            throw new Error('视频格式不正确');
        }
        processedVideoUrl.value = newUrl;
        if (videoPlayer.value) {
            videoPlayer.value.load();
        }
    } catch (error: any) {
        console.error('重试加载失败:', error);
        videoError.value = `无法加载视频: ${error.message}`;
        isVideoLoading.value = false;
    }
};
const handleUploadAndProcess = async () => {
    if (!videoFile.value || !textFile.value) return;
    uploading.value = true;
    uploadProgress.value = 0;
```

```
    showResult.value = true;
    showProcessLog.value = true;
    logContent.value = '开始上传文件...\n';
    try {
      const formData = new FormData();
      formData.append('video', videoFile.value);
      formData.append('text', textFile.value);
      const uploadResponse = await axios.post(
        'http://localhost:8000/api/upload/',
        formData,
        {
          headers: {
            'Content-Type': 'multipart/form-data',
          },
          onUploadProgress: (progressEvent) => {
            if (progressEvent.total) {
              uploadProgress.value = Math.round(
                (progressEvent.loaded * 100) / progressEvent.total
              );
            }
          }
        }
      );
      if (uploadResponse.data.status !== 'success') {
        throw new Error(uploadResponse.data.error || '文件上传失败');
      }
      const groupId = uploadResponse.data.group_id;
      logContent.value += '文件上传成功，开始处理...\n';
      uploading.value = false;
      processing.value = true;
      const processResponse = await axios.post(
        `http://localhost:8000/api/process/${groupId}/`
      );
      if (processResponse.data.status === 'success') {
        processing.value = false;
        isProcessed.value = true;
        showMessage('处理完成！', 'success');
        logContent.value += '处理完成！\n';
        const videoUrl = processResponse.data.video_url;
        const timestamp = new Date().getTime();
        const fullVideoUrl = `${videoUrl}?t=${timestamp}`;
        const isValidFormat = await checkVideoFormat(fullVideoUrl);
        if (isValidFormat) {
          processedVideoUrl.value = fullVideoUrl;
          logContent.value += `视频可访问: ${fullVideoUrl}\n`;
          showResult.value = true;
          showProcessLog.value = true;
          isVideoLoading.value = true;
          videoError.value = null;
        } else {
```

```
            throw new Error('处理后的视频格式不正确或无法访问');
          }
        } else {
          throw new Error(processResponse.data.message || '视频处理失败');
        }
      } catch (error: any) {
        const errorMessage = error.response?.data?.error ||
                             error.response?.data?.message ||
                             error.message ||
                             '操作失败';
        showMessage(errorMessage, 'error');
        logContent.value += `错误: ${errorMessage}\n`;
        hasError.value = true;
        console.error('操作错误:', error);
      } finally {
        uploading.value = false;
        processing.value = false;
      }
    };
    const resetVideoState = () => {
      isVideoLoading.value = true;
      videoError.value = null;
      processedVideoUrl.value = null;
    };
    const checkBackendStatus = async () => {
      try {
        await axios.get('http://localhost:8000/api/health/');
        return true;
      } catch (error) {
        console.error('后端服务检查失败:', error);
        return false;
      }
    };
    onMounted(async () => {
      const isBackendAvailable = await checkBackendStatus();
      if (!isBackendAvailable) {
        showMessage('无法连接到后端服务，请确保服务已启动', 'error');
      }
    });
    const handleProcessError = (error: any) => {
      logContent.value += '\n 处理过程中出错。需要安装缺失的依赖，请查看详细错误。';
      if (error.response?.data?.error) {
        detailedError.value = error.response.data.error;
      } else if (error.response?.data?.result?.script_stderr) {
        detailedError.value = error.response.data.result.script_stderr;
      } else if (error.message) {
        detailedError.value = error.message;
      }
    };
    </script>
```

```
<style scoped>
.file-uploader {
    max-width: 1200px;
    margin: 3rem auto;
    padding: 2.5rem;
    border: none;
    border-radius: 16px;
    background: #ffffff;
    box-shadow: 0 8px 30px rgba(0, 0, 0, 0.08);
    transition: all 0.3s ease;
}
.main-content {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 2rem;
    margin-bottom: 2rem;
    min-height: 500px;
}
.upload-container {
    display: flex;
    flex-direction: column;
    gap: 1.5rem;
}
.upload-section {
    display: flex;
    flex-direction: column;
    gap: 1.5rem;
}
.file-input-group {
    background: #f8fafc;
    padding: 1.5rem;
    border-radius: 12px;
    border: 1px solid rgba(0, 0, 0, 0.05);
    transition: all 0.3s ease;
}
.file-input-group:hover {
    transform: translateY(-2px);
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);
}
.file-input-group h3 {
    margin: 0 0 1rem 0;
    font-size: 1.1rem;
    color: #1a1a1a;
    font-weight: 600;
}
.selected-file {
    margin-top: 0.8rem;
    font-size: 0.95rem;
    color: #4a5568;
    padding: 0.5rem;
```

```
    background: rgba(66, 185, 131, 0.1);
    border-radius: 6px;
}
.action-button {
    width: 100%;
    padding: 1rem;
    background: linear-gradient(135deg, #42b983 0%, #3aa876 100%);
    color: white;
    border: none;
    border-radius: 8px;
    cursor: pointer;
    font-size: 1.1rem;
    margin-bottom: 1.2rem;
    transition: all 0.3s ease;
    font-weight: 500;
    letter-spacing: 0.5px;
    box-shadow: 0 4px 12px rgba(66, 185, 131, 0.2);
}
.action-button:hover:not(:disabled) {
    transform: translateY(-2px);
    box-shadow: 0 6px 16px rgba(66, 185, 131, 0.3);
}
.action-button:disabled {
    background: linear-gradient(135deg, #a8d5c2 0%, #9ecbb8 100%);
    cursor: not-allowed;
    box-shadow: none;
}
.process-button {
    background: linear-gradient(135deg, #4a90e2 0%, #357abd 100%);
    box-shadow: 0 4px 12px rgba(74, 144, 226, 0.2);
}
.process-button:hover:not(:disabled) {
    box-shadow: 0 6px 16px rgba(74, 144, 226, 0.3);
}
.process-button:disabled {
    background: linear-gradient(135deg, #a8c4e2 0%, #9ab8d9 100%);
}
.progress-container {
    background: #f8fafc;
    padding: 1.2rem;
    border-radius: 10px;
    box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.05);
}
.progress-label {
    margin-bottom: 0.8rem;
    color: #4a5568;
    font-weight: 500;
}
.progress-bar {
    height: 10px;
```

```
    background: #e2e8f0;
    border-radius: 6px;
    overflow: hidden;
    box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.05);
}
.progress-fill {
    height: 100%;
    background: linear-gradient(90deg, #42b983 0%, #3aa876 100%);
    transition: width 0.4s ease;
    box-shadow: 0 2px 4px rgba(66, 185, 131, 0.2);
}
.progress-text {
    text-align: right;
    font-size: 0.95rem;
    color: #4a5568;
    margin-top: 0.5rem;
    font-weight: 500;
}
.processing-spinner {
    width: 30px;
    height: 30px;
    border: 3px solid rgba(66, 185, 131, 0.1);
    border-top: 3px solid #42b983;
    border-radius: 50%;
    animation: spin 1s linear infinite;
    margin: 1rem auto;
}
.status-message {
    padding: 1.2rem;
    margin: 1.2rem 0;
    border-radius: 10px;
    text-align: center;
    font-weight: 500;
    animation: fadeIn 0.3s ease;
}
.success {
    background: linear-gradient(135deg, #d4edda 0%, #c3e6cb 100%);
    color: #155724;
    box-shadow: 0 2px 8px rgba(21, 87, 36, 0.1);
}
.error {
    background: linear-gradient(135deg, #f8d7da 0%, #f5c6cb 100%);
    color: #721c24;
    box-shadow: 0 2px 8px rgba(114, 28, 36, 0.1);
}
.result-section {
    margin-top: 2rem;
    padding: 1.5rem;
    background: #f8fafc;
    border-radius: 12px;
```

```
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);
    display: block;
}
.log-header {
    margin-bottom: 1rem;
    font-weight: 500;
}
.log-content-wrapper {
    margin-bottom: 1.2rem;
    height: 100%;
}
.log-content {
    max-height: 300px;
    overflow-y: auto;
    background: #ffffff;
    padding: 1.2rem;
    border-radius: 8px;
    font-family: 'Monaco', 'Menlo', monospace;
    line-height: 1.5;
    border: 1px solid #e2e8f0;
}
.video-section {
    background: #ffffff;
    border-radius: 12px;
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);
    height: fit-content;
    position: sticky;
    top: 2rem;
}
.video-player-container {
    background: #ffffff;
    border-radius: 12px;
    overflow: hidden;
}
.video-wrapper {
    aspect-ratio: 16/9;
    background: #000000;
    position: relative;
}
.video-player {
    width: 100%;
    max-width: 100%;
    display: block;
    border-radius: 8px;
}
.video-loading {
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
```

```
    bottom: 0;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    background: rgba(0, 0, 0, 0.8);
    color: white;
    backdrop-filter: blur(4px);
}
.loading-spinner {
    width: 48px;
    height: 48px;
    border: 4px solid rgba(255, 255, 255, 0.2);
    border-top: 4px solid #ffffff;
    border-radius: 50%;
    animation: spin 1s linear infinite;
    margin-bottom: 1.2rem;
}
.video-controls {
    padding: 1rem;
    border-top: 1px solid #e2e8f0;
    background: #f8fafc;
}
.control-button {
    display: inline-flex;
    align-items: center;
    gap: 0.8rem;
    padding: 0.8rem 1.5rem;
    border: none;
    border-radius: 8px;
    cursor: pointer;
    font-size: 1rem;
    transition: all 0.3s ease;
    font-weight: 500;
    letter-spacing: 0.5px;
}
.download-button {
    background: linear-gradient(135deg, #42b983 0%, #3aa876 100%);
    color: white;
    text-decoration: none;
    box-shadow: 0 4px 12px rgba(66, 185, 131, 0.2);
}
.download-button:hover {
    transform: translateY(-2px);
    box-shadow: 0 6px 16px rgba(66, 185, 131, 0.3);
}
.reload-button {
    background: linear-gradient(135deg, #4a90e2 0%, #357abd 100%);
    color: white;
    box-shadow: 0 4px 12px rgba(74, 144, 226, 0.2);
```

```
}
.reload-button:hover {
    transform: translateY(-2px);
    box-shadow: 0 6px 16px rgba(74, 144, 226, 0.3);
}
.button-icon {
    font-size: 1.3rem;
}
.video-error {
    margin-top: 1.2rem;
    padding: 1rem;
    background: linear-gradient(135deg, #f8d7da 0%, #f5c6cb 100%);
    border: 1px solid #f5c6cb;
    border-radius: 8px;
    color: #721c24;
    text-align: center;
    font-weight: 500;
    box-shadow: 0 2px 8px rgba(114, 28, 36, 0.1);
}
.error-container {
    margin: 1.2rem 0;
    padding: 1.5rem;
    background: linear-gradient(135deg, #fff3f3 0%, #ffe8e8 100%);
    border: 1px solid #ffcdd2;
    border-radius: 10px;
    box-shadow: 0 4px 12px rgba(211, 47, 47, 0.1);
}
.error-message h4 {
    color: #d32f2f;
    margin: 0 0 1rem 0;
    font-weight: 600;
}
.error-details {
    background: #ffffff;
    padding: 1.2rem;
    border-radius: 8px;
    font-size: 0.95rem;
    margin: 0;
    white-space: pre-wrap;
    word-break: break-word;
    box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.05);
    font-family: 'Monaco', 'Menlo', monospace;
    line-height: 1.5;
}
@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
@keyframes fadeIn {
    from { opacity: 0; transform: translateY(-10px); }
```

```
    to { opacity: 1; transform: translateY(0); }
}
.log-content::-webkit-scrollbar,
.error-details::-webkit-scrollbar {
    width: 8px;
}
.log-content::-webkit-scrollbar-track,
.error-details::-webkit-scrollbar-track {
    background: #f1f1f1;
    border-radius: 4px;
}
.log-content::-webkit-scrollbar-thumb,
.error-details::-webkit-scrollbar-thumb {
    background: #c1c1c1;
    border-radius: 4px;
}
.log-content::-webkit-scrollbar-thumb:hover,
.error-details::-webkit-scrollbar-thumb:hover {
    background: #a8a8a8;
}
input[type="file"] {
    width: 100%;
    padding: 0.8rem;
    border: 2px dashed #e2e8f0;
    border-radius: 8px;
    cursor: pointer;
    transition: all 0.3s ease;
}
input[type="file"]:hover {
    border-color: #42b983;
    background: rgba(66, 185, 131, 0.05);
}
input[type="file"]:disabled {
    border-color: #e2e8f0;
    background: #f8fafc;
    cursor: not-allowed;
}
@media (max-width: 1024px) {
    .main-content {
        grid-template-columns: 1fr;
    }
    .video-section {
        position: relative;
        top: 0;
    }
}
</style>
<template>
    <div>
        <FileUpload />
```

```
    </div>
</template>
<script setup>
import FileUpload from '@/components/FileUpload.vue'
</script>
import argparse
import os
import os.path as osp
import numpy as np
import cv2
import torch
import gc
import sys
from pathlib import Path
os.environ["HYDRA_FULL_ERROR"] = "1"
current_dir = os.path.dirname(os.path.abspath(__file__))
parent_dir = os.path.dirname(current_dir)
if parent_dir not in sys.path:
    sys.path.insert(0, parent_dir)
sam2_dir = os.path.join(parent_dir, "sam2")
if sam2_dir not in sys.path:
    sys.path.insert(0, sam2_dir)
sam2_inner_dir = os.path.join(sam2_dir, "sam2")
if sam2_inner_dir not in sys.path:
    sys.path.insert(0, sam2_inner_dir)
os.environ["HYDRA_CONFIG_PATH"] = os.path.join(parent_dir, "sam2", "sam2", "configs")
from build_sam import build_sam2_video_predictor
color = [(255, 0, 0)]
def load_txt(gt_path):
    with open(gt_path, 'r') as f:
        gt = f.readlines()
    prompts = {}
    for fid, line in enumerate(gt):
        x, y, w, h = map(float, line.split(','))
        x, y, w, h = int(x), int(y), int(w), int(h)
        prompts[fid] = ((x, y, x + w, y + h), 0)
    return prompts
def determine_model_cfg(model_path):
    config_base = os.path.join(parent_dir, "sam2", "sam2", "configs", "samurai")
    if "large" in model_path or "_l" in model_path:
        return os.path.join(config_base, "sam2.1_hiera_l.yaml")
    elif "base_plus" in model_path or "_b+" in model_path:
        return os.path.join(config_base, "sam2.1_hiera_b+.yaml")
    elif "base" in model_path or "_b" in model_path:
        return os.path.join(config_base, "sam2.1_hiera_b+.yaml")
    elif "small" in model_path or "_s" in model_path:
        return os.path.join(config_base, "sam2.1_hiera_s.yaml")
    elif "tiny" in model_path or "_t" in model_path:
        return os.path.join(config_base, "sam2.1_hiera_t.yaml")
    else:
```

```
        return os.path.join(config_base, "sam2.1_hiera_b+.yaml")
def prepare_frames_or_path(video_path):
    if video_path.endswith(".mp4") or osp.isdir(video_path):
        return video_path
    else:
        raise ValueError("Invalid video_path format. Should be .mp4 or a directory of jpg frames.")
def main(args):
    output_dir = os.path.join(os.path.dirname(os.path.dirname(os.path.dirname(__file__))),
                              'samurai_django', 'myproject', 'media', 'processed_videos')
    os.makedirs(output_dir, exist_ok=True)
    input_filename = os.path.basename(args.video_path)
    output_filename = os.path.splitext(input_filename)[0] + "_processed" +
os.path.splitext(input_filename)[1]
    args.output_path = os.path.join(output_dir, output_filename)
    args.video_output_path = args.output_path
    if not os.path.isabs(args.model_path):
        args.model_path = os.path.join(os.path.dirname(os.path.dirname(__file__)),
                                       "sam2", "checkpoints",
os.path.basename(args.model_path))
    if not os.path.exists(args.model_path):
        raise FileNotFoundError(f"模型文件不存在: {args.model_path}")
    model_cfg = determine_model_cfg(args.model_path)
    predictor = build_sam2_video_predictor(model_cfg, args.model_path, device="cuda:0")
    frames_or_path = prepare_frames_or_path(args.video_path)
    prompts = load_txt(args.txt_path)
    frame_rate = 30
    if args.save_to_video:
        if osp.isdir(args.video_path):
            frames = sorted([osp.join(args.video_path, f) for f in os.listdir(args.video_path) if
f.endswith((".jpg", ".jpeg", ".JPG", ".JPEG"))])
            loaded_frames = [cv2.imread(frame_path) for frame_path in frames]
            height, width = loaded_frames[0].shape[:2]
        else:
            cap = cv2.VideoCapture(args.video_path)
            frame_rate = cap.get(cv2.CAP_PROP_FPS)
            loaded_frames = []
            while True:
                ret, frame = cap.read()
                if not ret:
                    break
                loaded_frames.append(frame)
            cap.release()
            height, width = loaded_frames[0].shape[:2]
            if len(loaded_frames) == 0:
                raise ValueError("No frames were loaded from the video.")
    fourcc = cv2.VideoWriter_fourcc(*'avc1')
    out = cv2.VideoWriter(args.video_output_path, fourcc, frame_rate, (width, height))
    with torch.inference_mode(), torch.autocast("cuda", dtype=torch.float16):
        state = predictor.init_state(frames_or_path, offload_video_to_cpu=True)
        bbox, track_label = prompts[0]
```

```
            _, _, masks = predictor.add_new_points_or_box(state, box=bbox, frame_idx=0, obj_id=0)
            for frame_idx, object_ids, masks in predictor.propagate_in_video(state):
                mask_to_vis = {}
                bbox_to_vis = {}
                for obj_id, mask in zip(object_ids, masks):
                    mask = mask[0].cpu().numpy()
                    mask = mask > 0.0
                    non_zero_indices = np.argwhere(mask)
                    if len(non_zero_indices) == 0:
                        bbox = [0, 0, 0, 0]
                    else:
                        y_min, x_min = non_zero_indices.min(axis=0).tolist()
                        y_max, x_max = non_zero_indices.max(axis=0).tolist()
                        bbox = [x_min, y_min, x_max - x_min, y_max - y_min]
                    bbox_to_vis[obj_id] = bbox
                    mask_to_vis[obj_id] = mask
                if args.save_to_video:
                    img = loaded_frames[frame_idx]
                    for obj_id, mask in mask_to_vis.items():
                        mask_img = np.zeros((height, width, 3), np.uint8)
                        mask_img[mask] = color[(obj_id + 1) % len(color)]
                        img = cv2.addWeighted(img, 1, mask_img, 0.2, 0)
                    for obj_id, bbox in bbox_to_vis.items():
                        cv2.rectangle(img, (bbox[0], bbox[1]), (bbox[0] + bbox[2], bbox[1] + bbox[3]),
color[obj_id % len(color)], 2)
                    out.write(img)
            if args.save_to_video:
                out.release()
        del predictor, state
        gc.collect()
        torch.clear_autocast_cache()
        torch.cuda.empty_cache()
        try:
            if os.path.exists(args.output_path):
                print(f"处理完成，视频已保存到: {args.output_path}")
                relative_path = os.path.relpath(args.output_path,

os.path.join(os.path.dirname(os.path.dirname(os.path.dirname(__file__))),
                                                                    'samurai_django', 'media'))
                return True, relative_path
            else:
                print(f"处理失败：找不到输出文件 {args.output_path}")
                return False, None
        except Exception as e:
            print(f"处理过程中出错: {str(e)}")
            return False, None
if __name__ == "__main__":
        parser = argparse.ArgumentParser(description='视频处理脚本')
        parser.add_argument('--video_path', type=str, required=True, help='输入视频路径')
        parser.add_argument('--txt_path', type=str, required=True, help='输入文本路径')
```

```
        default_model_path = os.path.join(os.path.dirname(os.path.dirname(__file__)),
                                          "sam2", "checkpoints", "sam2.1_hiera_base_plus.pt")
        parser.add_argument('--model_path', type=str, default=default_model_path, help='模型路径')
        parser.add_argument("--save_to_video", default=True, help="Save results to a video.")
        args = parser.parse_args()
        success, output_path = main(args)
        exit(0 if success else 1)
:root {
    --vt-c-white: #ffffff;
    --vt-c-white-soft: #f8f8f8;
    --vt-c-white-mute: #f2f2f2;
    --vt-c-black: #181818;
    --vt-c-black-soft: #222222;
    --vt-c-black-mute: #282828;
    --vt-c-indigo: #2c3e50;
    --vt-c-divider-light-1: rgba(60, 60, 60, 0.29);
    --vt-c-divider-light-2: rgba(60, 60, 60, 0.12);
    --vt-c-divider-dark-1: rgba(84, 84, 84, 0.65);
    --vt-c-divider-dark-2: rgba(84, 84, 84, 0.48);
    --vt-c-text-light-1: var(--vt-c-indigo);
    --vt-c-text-light-2: rgba(60, 60, 60, 0.66);
    --vt-c-text-dark-1: var(--vt-c-white);
    --vt-c-text-dark-2: rgba(235, 235, 235, 0.64);
}
:root {
    --color-background: var(--vt-c-white);
    --color-background-soft: var(--vt-c-white-soft);
    --color-background-mute: var(--vt-c-white-mute);
    --color-border: var(--vt-c-divider-light-2);
    --color-border-hover: var(--vt-c-divider-light-1);
    --color-heading: var(--vt-c-text-light-1);
    --color-text: var(--vt-c-text-light-1);
    --section-gap: 160px;
}
@media (prefers-color-scheme: dark) {
    :root {
        --color-background: var(--vt-c-black);
        --color-background-soft: var(--vt-c-black-soft);
        --color-background-mute: var(--vt-c-black-mute);
        --color-border: var(--vt-c-divider-dark-2);
        --color-border-hover: var(--vt-c-divider-dark-1);
        --color-heading: var(--vt-c-text-dark-1);
        --color-text: var(--vt-c-text-dark-2);
    }
}
*,
*::before,
*::after {
    box-sizing: border-box;
    margin: 0;
```

```
    font-weight: normal;
}
body {
    min-height: 100vh;
    color: var(--color-text);
    background: var(--color-background);
    transition:
        color 0.5s,
        background-color 0.5s;
    line-height: 1.6;
    font-family:
        Inter,
        -apple-system,
        BlinkMacSystemFont,
        'Segoe UI',
        Roboto,
        Oxygen,
        Ubuntu,
        Cantarell,
        'Fira Sans',
        'Droid Sans',
        'Helvetica Neue',
        sans-serif;
    font-size: 15px;
    text-rendering: optimizeLegibility;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
}
import torch
from torchvision.ops.boxes import box_area
import numpy as np
def box_cxcywh_to_xyxy(x):
        x_c, y_c, w, h = x.unbind(-1)
        b = [(x_c - 0.5 * w), (y_c - 0.5 * h),
                (x_c + 0.5 * w), (y_c + 0.5 * h)]
        return torch.stack(b, dim=-1)
def box_xywh_to_xyxy(x):
        x1, y1, w, h = x.unbind(-1)
        b = [x1, y1, x1 + w, y1 + h]
        return torch.stack(b, dim=-1)
def box_xyxy_to_xywh(x):
        x1, y1, x2, y2 = x.unbind(-1)
        b = [x1, y1, x2 - x1, y2 - y1]
        return torch.stack(b, dim=-1)
def box_xyxy_to_cxcywh(x):
        x0, y0, x1, y1 = x.unbind(-1)
        b = [(x0 + x1) / 2, (y0 + y1) / 2,
                (x1 - x0), (y1 - y0)]
        return torch.stack(b, dim=-1)
def box_iou(boxes1, boxes2):
```

```
        area1 = box_area(boxes1)
        area2 = box_area(boxes2)
        lt = torch.max(boxes1[:, :2], boxes2[:, :2])
        rb = torch.min(boxes1[:, 2:], boxes2[:, 2:])
        wh = (rb - lt).clamp(min=0)
        inter = wh[:, 0] * wh[:, 1]
        union = area1 + area2 - inter
        iou = inter / union
        return iou, union
    def generalized_box_iou(boxes1, boxes2):
        iou, union = box_iou(boxes1, boxes2)
        lt = torch.min(boxes1[:, :2], boxes2[:, :2])
        rb = torch.max(boxes1[:, 2:], boxes2[:, 2:])
        wh = (rb - lt).clamp(min=0)
        area = wh[:, 0] * wh[:, 1]
        return iou - (area - union) / area, iou
    def giou_loss(boxes1, boxes2):
        giou, iou = generalized_box_iou(boxes1, boxes2)
        return (1 - giou).mean(), iou
    def clip_box(box: list, H, W, margin=0):
        x1, y1, w, h = box
        x2, y2 = x1 + w, y1 + h
        x1 = min(max(0, x1), W-margin)
        x2 = min(max(margin, x2), W)
        y1 = min(max(0, y1), H-margin)
        y2 = min(max(margin, y2), H)
        w = max(margin, x2-x1)
        h = max(margin, y2-y1)
        return [x1, y1, w, h]
    import math
    import torch
    import torch.nn.functional as F
    def generate_bbox_mask(bbox_mask, bbox):
        b, h, w = bbox_mask.shape
        for i in range(b):
            bbox_i = bbox[i].cpu().tolist()
            bbox_mask[i, int(bbox_i[1]):int(bbox_i[1] + bbox_i[3] - 1), int(bbox_i[0]):int(bbox_i[0] +
    bbox_i[2] - 1)] = 1
        return bbox_mask
    def generate_mask_cond(cfg, bs, device, gt_bbox):
        template_size = cfg.DATA.TEMPLATE.SIZE
        stride = cfg.MODEL.BACKBONE.STRIDE
        template_feat_size = template_size // stride
        if cfg.MODEL.BACKBONE.CE_TEMPLATE_RANGE == 'ALL':
            box_mask_z = None
        elif cfg.MODEL.BACKBONE.CE_TEMPLATE_RANGE == 'CTR_POINT':
            if template_feat_size == 8:
                index = slice(3, 4)
            elif template_feat_size == 12:
                index = slice(5, 6)
```

```
            elif template_feat_size == 7:
                index = slice(3, 4)
            elif template_feat_size == 14:
                index = slice(6, 7)
            else:
                raise NotImplementedError
            box_mask_z = torch.zeros([bs, template_feat_size, template_feat_size], device=device)
            box_mask_z[:, index, index] = 1
            box_mask_z = box_mask_z.flatten(1).to(torch.bool)
        elif cfg.MODEL.BACKBONE.CE_TEMPLATE_RANGE == 'CTR_REC':
            if template_feat_size == 8:
                index = slice(3, 5)
            elif template_feat_size == 12:
                index = slice(5, 7)
            elif template_feat_size == 7:
                index = slice(3, 4)
            else:
                raise NotImplementedError
            box_mask_z = torch.zeros([bs, template_feat_size, template_feat_size], device=device)
            box_mask_z[:, index, index] = 1
            box_mask_z = box_mask_z.flatten(1).to(torch.bool)
        elif cfg.MODEL.BACKBONE.CE_TEMPLATE_RANGE == 'GT_BOX':
            box_mask_z = torch.zeros([bs, template_size, template_size], device=device)
            box_mask_z = generate_bbox_mask(box_mask_z, gt_bbox * template_size).unsqueeze(1).to(
                torch.float)
            box_mask_z = F.interpolate(box_mask_z, scale_factor=1. / cfg.MODEL.BACKBONE.STRIDE,
mode='bilinear',
                                       align_corners=False)
            box_mask_z = box_mask_z.flatten(1).to(torch.bool)
        else:
            raise NotImplementedError
        return box_mask_z
def adjust_keep_rate(epoch, warmup_epochs, total_epochs, ITERS_PER_EPOCH, base_keep_rate=0.5,
max_keep_rate=1, iters=-1):
        if epoch < warmup_epochs:
            return 1
        if epoch >= total_epochs:
            return base_keep_rate
        if iters == -1:
            iters = epoch * ITERS_PER_EPOCH
        total_iters = ITERS_PER_EPOCH * (total_epochs - warmup_epochs)
        iters = iters - ITERS_PER_EPOCH * warmup_epochs
        keep_rate = base_keep_rate + (max_keep_rate - base_keep_rate) \
            * (math.cos(iters / total_iters * math.pi) + 1) * 0.5
        return keep_rate
@import './base.css';
#app {
  max-width: 1280px;
  margin: 0 auto;
  padding: 2rem;
```

```
    font-weight: normal;
}
a,
.green {
    text-decoration: none;
    color: hsla(160, 100%, 37%, 1);
    transition: 0.4s;
    padding: 3px;
}
@media (hover: hover) {
    a:hover {
        background-color: hsla(160, 100%, 37%, 0.2);
    }
}
@media (min-width: 1024px) {
    body {
        display: flex;
        place-items: center;
    }
    #app {
        display: grid;
        grid-template-columns: 1fr 1fr;
        padding: 0 2rem;
    }
}
from abc import ABC
import torch
import torch.nn as nn
import torch.nn.functional as F
class FocalLoss(nn.Module, ABC):
        def __init__(self, alpha=2, beta=4):
                super(FocalLoss, self).__init__()
                self.alpha = alpha
                self.beta = beta
        def forward(self, prediction, target):
                positive_index = target.eq(1).float()
                negative_index = target.lt(1).float()
                negative_weights = torch.pow(1 - target, self.beta)
                prediction = torch.clamp(prediction, 1e-12)
                positive_loss = torch.log(prediction) * torch.pow(1 - prediction, self.alpha) * positive_index
                negative_loss = torch.log(1 - prediction) * torch.pow(prediction,
                                                                                self.alpha)                 *
negative_weights * negative_index
                num_positive = positive_index.float().sum()
                positive_loss = positive_loss.sum()
                negative_loss = negative_loss.sum()
                if num_positive == 0:
                        loss = -negative_loss
                else:
                        loss = -(positive_loss + negative_loss) / num_positive
```

```
            return loss
class LBHinge(nn.Module):
    def __init__(self, error_metric=nn.MSELoss(), threshold=None, clip=None):
            super().__init__()
            self.error_metric = error_metric
            self.threshold = threshold if threshold is not None else -100
            self.clip = clip
    def forward(self, prediction, label, target_bb=None):
            negative_mask = (label < self.threshold).float()
            positive_mask = (1.0 - negative_mask)
            prediction = negative_mask * F.relu(prediction) + positive_mask * prediction
            loss = self.error_metric(prediction, positive_mask * label)
            if self.clip is not None:
                loss = torch.min(loss, torch.tensor([self.clip], device=loss.device))
            return loss
import argparse
import gc
import os
import os.path as osp
import pdb
import cv2
import numpy as np
import torch
from loguru import logger
from tqdm import tqdm
from sam2.build_sam import build_sam2_video_predictor
def load_test_video_list(testing_list_path):
    with open(testing_list_path, 'r') as f:
            test_videos = [line.strip() for line in f.readlines()]
    return test_videos
def load_gt(gt_path):
    with open(gt_path, 'r') as f:
            gt = f.readlines()
    prompts = {}
    fid = 0
    for line in gt:
            x, y, w, h = map(int, line.split(','))
            prompts[fid] = ((x, y, x+w, y+h), 0)
            fid += 1
    return prompts
def get_ckpt_and_cfg(tracker_name, model_name):
    assert tracker_name in ["sam2.1", "samurai"], "Invalid tracker name"
    assert model_name in ["tiny", "small", "base_plus", "large"], "Invalid model name"
    model_ckpt = f"sam2/checkpoints/sam2.1_hiera_{model_name}.pt"
    if model_name == "base_plus":
            model_cfg = f"configs/{tracker_name}/sam2.1_hiera_b+.yaml"
    else:
            model_cfg = f"configs/{tracker_name}/sam2.1_hiera_{model_name[0]}.yaml"
    return model_ckpt, model_cfg
def split_list(video_list, num_chunks):
```

```
    chunk_size = len(video_list) // num_chunks
    return [video_list[i:i+chunk_size] for i in range(0, len(video_list), chunk_size)]
def inference_chunk(dataset_path, tracker_name, model_name, chunk_videos, result_folder):
    exp_name = "test"
    model_ckpt, model_cfg = get_ckpt_and_cfg(tracker_name, model_name)
    for vid, video in enumerate(chunk_videos):
        cat_name = video.split('-')[0]
        cid_name = video.split('-')[1]
        video_basename = video.strip()
        frame_folder = osp.join(dataset_path, cat_name, video.strip(), "img")
        num_frames = len(os.listdir(osp.join(dataset_path, cat_name, video.strip(), "img")))
        height, width = cv2.imread(osp.join(frame_folder, "00000001.jpg")).shape[:2]
        logger.info(f"Running video [{vid+1}/{len(chunk_videos)}]: {video} with {num_frames} frames
({height}x{width})")
        predictor = build_sam2_video_predictor(model_cfg, model_ckpt, device="cuda:0")
        predictions = []
        with torch.inference_mode(), torch.autocast("cuda", dtype=torch.float16):
            state = predictor.init_state(frame_folder, offload_video_to_cpu=True,
offload_state_to_cpu=True)
            prompts = load_gt(osp.join(dataset_path, cat_name, video.strip(), "groundtruth.txt"))
            bbox, track_label = prompts[0]
            frame_idx, object_ids, masks = predictor.add_new_points_or_box(state, box=bbox,
frame_idx=0, obj_id=0)
            for frame_idx, object_ids, masks in predictor.propagate_in_video(state):
                mask_to_vis = {}
                bbox_to_vis = {}
                assert len(masks) == 1 and len(object_ids) == 1, "Only one object is supported
right now"
                for obj_id, mask in zip(object_ids, masks):
                    mask = mask[0].cpu().numpy()
                    mask = mask > 0.0
                    non_zero_indices = np.argwhere(mask)
                    if len(non_zero_indices) == 0:
                        bbox = [0, 0, 0, 0]
                    else:
                        y_min, x_min = non_zero_indices.min(axis=0).tolist()
                        y_max, x_max = non_zero_indices.max(axis=0).tolist()
                        bbox = [x_min, y_min, x_max-x_min, y_max-y_min]
                    bbox_to_vis[obj_id] = bbox
                    mask_to_vis[obj_id] = mask
                predictions.append(bbox_to_vis)
        os.makedirs(result_folder, exist_ok=True)
        with open(osp.join(result_folder, f'{video_basename}.txt'), 'w') as f:
            for pred in predictions:
                x, y, w, h = pred[0]
                f.write(f"{x},{y},{w},{h}\n")
        del predictor
        del state
        gc.collect()
        torch.clear_autocast_cache()
```

```
            torch.cuda.empty_cache()
def main():
    parser = argparse.ArgumentParser()
    parser.add_argument("--dataset_path", type=str, default="data/LaSOT-ext")
    parser.add_argument("--tracker_name", type=str, default="samurai")
    parser.add_argument("--model_name", type=str, default="large")
    parser.add_argument("--chunk_idx", type=int, default=0)
    parser.add_argument("--num_chunks", type=int, default=1)
    parser.add_argument("--exp_name", type=str, default="test")
    parser.add_argument("--root_result_folder", type=str, default="results")
    args = parser.parse_args()
    test_videos = load_test_video_list("data/LaSOT-ext/testing_set.txt")
    chunk_video_list = split_list(test_videos, args.num_chunks)
    chunk_videos = chunk_video_list[args.chunk_idx]
    logger.info(f"Chunk ID: {args.chunk_idx}, Number of videos: {len(chunk_videos)} (from
{chunk_videos[0]} to {chunk_videos[-1]})")
    exp_result_folder = osp.join(args.root_result_folder, args.tracker_name,
f"{args.exp_name}_{args.model_name}")
    inference_chunk(args.dataset_path, args.tracker_name, args.model_name, chunk_videos,
exp_result_folder)
if __name__ == "__main__":
    main()
import functools
import torch
import copy
from collections import OrderedDict
class TensorDict(OrderedDict):
    def concat(self, other):
        return TensorDict(self, **other)
    def copy(self):
        return TensorDict(super(TensorDict, self).copy())
    def __deepcopy__(self, memodict={}):
        return TensorDict(copy.deepcopy(list(self), memodict))
    def __getattr__(self, name):
        if not hasattr(torch.Tensor, name):
            raise AttributeError('\'TensorDict\' object has not attribute \'{}\''.format(name))
        def apply_attr(*args, **kwargs):
            return TensorDict({n: getattr(e, name)(*args, **kwargs) if hasattr(e, name) else e for n, e
in self.items()})
        return apply_attr
    def attribute(self, attr: str, *args):
        return TensorDict({n: getattr(e, attr, *args) for n, e in self.items()})
    def apply(self, fn, *args, **kwargs):
        return TensorDict({n: fn(e, *args, **kwargs) for n, e in self.items()})
    @staticmethod
    def _iterable(a):
        return isinstance(a, (TensorDict, list))
class TensorList(list):
    def __init__(self, list_of_tensors = None):
        if list_of_tensors is None:
```

```
            list_of_tensors = list()
        super(TensorList, self).__init__(list_of_tensors)
    def __deepcopy__(self, memodict={}):
        return TensorList(copy.deepcopy(list(self), memodict))
    def __getitem__(self, item):
        if isinstance(item, int):
            return super(TensorList, self).__getitem__(item)
        elif isinstance(item, (tuple, list)):
            return TensorList([super(TensorList, self).__getitem__(i) for i in item])
        else:
            return TensorList(super(TensorList, self).__getitem__(item))
    def __add__(self, other):
        if TensorList._iterable(other):
            return TensorList([e1 + e2 for e1, e2 in zip(self, other)])
        return TensorList([e + other for e in self])
    def __radd__(self, other):
        if TensorList._iterable(other):
            return TensorList([e2 + e1 for e1, e2 in zip(self, other)])
        return TensorList([other + e for e in self])
    def __iadd__(self, other):
        if TensorList._iterable(other):
            for i, e2 in enumerate(other):
                self[i] += e2
        else:
            for i in range(len(self)):
                self[i] += other
        return self
    def __sub__(self, other):
        if TensorList._iterable(other):
            return TensorList([e1 - e2 for e1, e2 in zip(self, other)])
        return TensorList([e - other for e in self])
    def __rsub__(self, other):
        if TensorList._iterable(other):
            return TensorList([e2 - e1 for e1, e2 in zip(self, other)])
        return TensorList([other - e for e in self])
    def __isub__(self, other):
        if TensorList._iterable(other):
            for i, e2 in enumerate(other):
                self[i] -= e2
        else:
            for i in range(len(self)):
                self[i] -= other
        return self
    def __mul__(self, other):
        if TensorList._iterable(other):
            return TensorList([e1 * e2 for e1, e2 in zip(self, other)])
        return TensorList([e * other for e in self])
    def __rmul__(self, other):
        if TensorList._iterable(other):
            return TensorList([e2 * e1 for e1, e2 in zip(self, other)])
```

```
            return TensorList([other * e for e in self])
        def __imul__(self, other):
            if TensorList._iterable(other):
                for i, e2 in enumerate(other):
                    self[i] *= e2
            else:
                for i in range(len(self)):
                    self[i] *= other
            return self
        def __truediv__(self, other):
            if TensorList._iterable(other):
                return TensorList([e1 / e2 for e1, e2 in zip(self, other)])
            return TensorList([e / other for e in self])
        def __rtruediv__(self, other):
                {"name": "uav_building1", "path": "data_seq/UAV123/building1", "startFrame": 1,
"endFrame": 469, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/building1.txt", "object_class": "other"},
                {"name": "uav_building2", "path": "data_seq/UAV123/building2", "startFrame": 1,
"endFrame": 577, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/building2.txt", "object_class": "other"},
                {"name": "uav_building3", "path": "data_seq/UAV123/building3", "startFrame": 1,
"endFrame": 829, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/building3.txt", "object_class": "other"},
                {"name": "uav_building4", "path": "data_seq/UAV123/building4", "startFrame": 1,
"endFrame": 787, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/building4.txt", "object_class": "other"},
                {"name": "uav_building5", "path": "data_seq/UAV123/building5", "startFrame": 1,
"endFrame": 481, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/building5.txt", "object_class": "other"},
                {"name": "uav_car1_1", "path": "data_seq/UAV123/car1", "startFrame": 1, "endFrame":
751, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/car1_1.txt", "object_class": "car"},
                {"name": "uav_car1_2", "path": "data_seq/UAV123/car1", "startFrame": 751, "endFrame":
1627, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/car1_2.txt", "object_class": "car"},
                {"name": "uav_car1_3", "path": "data_seq/UAV123/car1", "startFrame": 1627,
"endFrame": 2629, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/car1_3.txt", "object_class": "car"},
                {"name": "uav_car10", "path": "data_seq/UAV123/car10", "startFrame": 1, "endFrame":
1405, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/car10.txt", "object_class": "car"},
                {"name": "uav_car11", "path": "data_seq/UAV123/car11", "startFrame": 1, "endFrame":
337, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/car11.txt", "object_class": "car"},
                {"name": "uav_car12", "path": "data_seq/UAV123/car12", "startFrame": 1, "endFrame":
499, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/car12.txt", "object_class": "car"},
                {"name": "uav_car13", "path": "data_seq/UAV123/car13", "startFrame": 1, "endFrame":
415, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/car13.txt", "object_class": "car"},
```

```
{"name": "uav_car14", "path": "data_seq/UAV123/car14", "startFrame": 1, "endFrame": 1327, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car14.txt", "object_class": "car"},
{"name": "uav_car15", "path": "data_seq/UAV123/car15", "startFrame": 1, "endFrame": 469, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car15.txt", "object_class": "car"},
{"name": "uav_car16_1", "path": "data_seq/UAV123/car16", "startFrame": 1, "endFrame": 415, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car16_1.txt", "object_class": "car"},
{"name": "uav_car16_2", "path": "data_seq/UAV123/car16", "startFrame": 415, "endFrame": 1993, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car16_2.txt", "object_class": "car"},
{"name": "uav_car17", "path": "data_seq/UAV123/car17", "startFrame": 1, "endFrame": 1057, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car17.txt", "object_class": "car"},
{"name": "uav_car18", "path": "data_seq/UAV123/car18", "startFrame": 1, "endFrame": 1207, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car18.txt", "object_class": "car"},
{"name": "uav_car1_s", "path": "data_seq/UAV123/car1_s", "startFrame": 1, "endFrame": 1475, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car1_s.txt", "object_class": "car"},
{"name": "uav_car2", "path": "data_seq/UAV123/car2", "startFrame": 1, "endFrame": 1321, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car2.txt", "object_class": "car"},
{"name": "uav_car2_s", "path": "data_seq/UAV123/car2_s", "startFrame": 1, "endFrame": 320, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car2_s.txt", "object_class": "car"},
{"name": "uav_car3", "path": "data_seq/UAV123/car3", "startFrame": 1, "endFrame": 1717, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car3.txt", "object_class": "car"},
{"name": "uav_car3_s", "path": "data_seq/UAV123/car3_s", "startFrame": 1, "endFrame": 1300, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car3_s.txt", "object_class": "car"},
{"name": "uav_car4", "path": "data_seq/UAV123/car4", "startFrame": 1, "endFrame": 1345, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car4.txt", "object_class": "car"},
{"name": "uav_car4_s", "path": "data_seq/UAV123/car4_s", "startFrame": 1, "endFrame": 830, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car4_s.txt", "object_class": "car"},
{"name": "uav_car5", "path": "data_seq/UAV123/car5", "startFrame": 1, "endFrame": 745, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car5.txt", "object_class": "car"},
{"name": "uav_car6_1", "path": "data_seq/UAV123/car6", "startFrame": 1, "endFrame": 487, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car6_1.txt", "object_class": "car"},
{"name": "uav_car6_2", "path": "data_seq/UAV123/car6", "startFrame": 487, "endFrame": 1807, "nz": 6,
 "ext": "jpg", "anno_path": "anno/UAV123/car6_2.txt", "object_class": "car"},
{"name": "uav_car6_3", "path": "data_seq/UAV123/car6", "startFrame": 1807, "endFrame": 2953, "nz": 6,
```

```
                "ext": "jpg", "anno_path": "anno/UAV123/car6_3.txt", "object_class": "car"},
               {"name": "uav_car6_4", "path": "data_seq/UAV123/car6", "startFrame": 2953,
"endFrame": 3925, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/car6_4.txt", "object_class": "car"},
               {"name": "uav_car6_5", "path": "data_seq/UAV123/car6", "startFrame": 3925,
"endFrame": 4861, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/car6_5.txt", "object_class": "car"},
               {"name": "uav_car7", "path": "data_seq/UAV123/car7", "startFrame": 1, "endFrame":
1033, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/car7.txt", "object_class": "car"},
               {"name": "uav_car8_1", "path": "data_seq/UAV123/car8", "startFrame": 1, "endFrame":
1357, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/car8_1.txt", "object_class": "car"},
               {"name": "uav_car8_2", "path": "data_seq/UAV123/car8", "startFrame": 1357,
"endFrame": 2575, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/car8_2.txt", "object_class": "car"},
               {"name": "uav_car9", "path": "data_seq/UAV123/car9", "startFrame": 1, "endFrame":
1879, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/car9.txt", "object_class": "car"},
               {"name": "uav_group1_1", "path": "data_seq/UAV123/group1", "startFrame": 1,
"endFrame": 1333, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/group1_1.txt", "object_class": "person"},
               {"name": "uav_group1_2", "path": "data_seq/UAV123/group1", "startFrame": 1333,
"endFrame": 2515, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/group1_2.txt", "object_class": "person"},
               {"name": "uav_group1_3", "path": "data_seq/UAV123/group1", "startFrame": 2515,
"endFrame": 3925, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/group1_3.txt", "object_class": "person"},
               {"name": "uav_group1_4", "path": "data_seq/UAV123/group1", "startFrame": 3925,
"endFrame": 4873, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/group1_4.txt", "object_class": "person"},
               {"name": "uav_group2_1", "path": "data_seq/UAV123/group2", "startFrame": 1,
"endFrame": 907, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/group2_1.txt", "object_class": "person"},
               {"name": "uav_group2_2", "path": "data_seq/UAV123/group2", "startFrame": 907,
"endFrame": 1771, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/group2_2.txt", "object_class": "person"},
               {"name": "uav_group2_3", "path": "data_seq/UAV123/group2", "startFrame": 1771,
"endFrame": 2683, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/group2_3.txt", "object_class": "person"},
               {"name": "uav_group3_1", "path": "data_seq/UAV123/group3", "startFrame": 1,
"endFrame": 1567, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/group3_1.txt", "object_class": "person"},
               {"name": "uav_group3_2", "path": "data_seq/UAV123/group3", "startFrame": 1567,
"endFrame": 2827, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/group3_2.txt", "object_class": "person"},
               {"name": "uav_group3_3", "path": "data_seq/UAV123/group3", "startFrame": 2827,
"endFrame": 4369, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/group3_3.txt", "object_class": "person"},
               {"name": "uav_group3_4", "path": "data_seq/UAV123/group3", "startFrame": 4369,
```

```
"endFrame": 5527, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/group3_4.txt", "object_class": "person"},
                {"name": "uav_person1", "path": "data_seq/UAV123/person1", "startFrame": 1,
"endFrame": 799, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person1.txt", "object_class": "person"},
                {"name": "uav_person10", "path": "data_seq/UAV123/person10", "startFrame": 1,
"endFrame": 1021, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person10.txt", "object_class": "person"},
                {"name": "uav_person11", "path": "data_seq/UAV123/person11", "startFrame": 1,
"endFrame": 721, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person11.txt", "object_class": "person"},
                {"name": "uav_person12_1", "path": "data_seq/UAV123/person12", "startFrame": 1,
"endFrame": 601, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person12_1.txt", "object_class": "person"},
                {"name": "uav_person12_2", "path": "data_seq/UAV123/person12", "startFrame": 601,
"endFrame": 1621, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person12_2.txt", "object_class": "person"},
                {"name": "uav_person13", "path": "data_seq/UAV123/person13", "startFrame": 1,
"endFrame": 883, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person13.txt", "object_class": "person"},
                {"name": "uav_person14_1", "path": "data_seq/UAV123/person14", "startFrame": 1,
"endFrame": 847, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person14_1.txt", "object_class": "person"},
                {"name": "uav_person14_2", "path": "data_seq/UAV123/person14", "startFrame": 847,
"endFrame": 1813, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person14_2.txt", "object_class": "person"},
                {"name": "uav_person14_3", "path": "data_seq/UAV123/person14", "startFrame": 1813,
"endFrame": 2923,
                    "nz": 6, "ext": "jpg", "anno_path": "anno/UAV123/person14_3.txt", "object_class":
"person"},
                {"name": "uav_person15", "path": "data_seq/UAV123/person15", "startFrame": 1,
"endFrame": 1339, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person15.txt", "object_class": "person"},
                {"name": "uav_person16", "path": "data_seq/UAV123/person16", "startFrame": 1,
"endFrame": 1147, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person16.txt", "object_class": "person"},
                {"name": "uav_person17_1", "path": "data_seq/UAV123/person17", "startFrame": 1,
"endFrame": 1501, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person17_1.txt", "object_class": "person"},
                {"name": "uav_person17_2", "path": "data_seq/UAV123/person17", "startFrame": 1501,
"endFrame": 2347,
                    "nz": 6, "ext": "jpg", "anno_path": "anno/UAV123/person17_2.txt", "object_class":
"person"},
                {"name": "uav_person18", "path": "data_seq/UAV123/person18", "startFrame": 1,
"endFrame": 1393, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person18.txt", "object_class": "person"},
                {"name": "uav_person19_1", "path": "data_seq/UAV123/person19", "startFrame": 1,
"endFrame": 1243, "nz": 6,
                    "ext": "jpg", "anno_path": "anno/UAV123/person19_1.txt", "object_class": "person"},
                {"name": "uav_person19_2", "path": "data_seq/UAV123/person19", "startFrame": 1243,
```

```
"endFrame": 2791,
                "nz": 6, "ext": "jpg", "anno_path": "anno/UAV123/person19_2.txt", "object_class":
"person"},
               {"name": "uav_person19_3", "path": "data_seq/UAV123/person19", "startFrame": 2791,
"endFrame": 4357,
                "nz": 6, "ext": "jpg", "anno_path": "anno/UAV123/person19_3.txt", "object_class":
"person"},
               {"name": "uav_person1_s", "path": "data_seq/UAV123/person1_s", "startFrame": 1,
"endFrame": 1600, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person1_s.txt", "object_class": "person"},
               {"name": "uav_person2_1", "path": "data_seq/UAV123/person2", "startFrame": 1,
"endFrame": 1189, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person2_1.txt", "object_class": "person"},
               {"name": "uav_person2_2", "path": "data_seq/UAV123/person2", "startFrame": 1189,
"endFrame": 2623, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person2_2.txt", "object_class": "person"},
               {"name": "uav_person20", "path": "data_seq/UAV123/person20", "startFrame": 1,
"endFrame": 1783, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person20.txt", "object_class": "person"},
               {"name": "uav_person21", "path": "data_seq/UAV123/person21", "startFrame": 1,
"endFrame": 487, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person21.txt", "object_class": "person"},
               {"name": "uav_person22", "path": "data_seq/UAV123/person22", "startFrame": 1,
"endFrame": 199, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person22.txt", "object_class": "person"},
               {"name": "uav_person23", "path": "data_seq/UAV123/person23", "startFrame": 1,
"endFrame": 397, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person23.txt", "object_class": "person"},
               {"name": "uav_person2_s", "path": "data_seq/UAV123/person2_s", "startFrame": 1,
"endFrame": 250, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person2_s.txt", "object_class": "person"},
               {"name": "uav_person3", "path": "data_seq/UAV123/person3", "startFrame": 1,
"endFrame": 643, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person3.txt", "object_class": "person"},
               {"name": "uav_person3_s", "path": "data_seq/UAV123/person3_s", "startFrame": 1,
"endFrame": 505, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person3_s.txt", "object_class": "person"},
               {"name": "uav_person4_1", "path": "data_seq/UAV123/person4", "startFrame": 1,
"endFrame": 1501, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person4_1.txt", "object_class": "person"},
               {"name": "uav_person4_2", "path": "data_seq/UAV123/person4", "startFrame": 1501,
"endFrame": 2743, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person4_2.txt", "object_class": "person"},
               {"name": "uav_person5_1", "path": "data_seq/UAV123/person5", "startFrame": 1,
"endFrame": 877, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person5_1.txt", "object_class": "person"},
               {"name": "uav_person5_2", "path": "data_seq/UAV123/person5", "startFrame": 877,
"endFrame": 2101, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person5_2.txt", "object_class": "person"},
               {"name": "uav_person6", "path": "data_seq/UAV123/person6", "startFrame": 1,
```

```
"endFrame": 901, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person6.txt", "object_class": "person"},
               {"name": "uav_person7_1", "path": "data_seq/UAV123/person7", "startFrame": 1,
"endFrame": 1249, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person7_1.txt", "object_class": "person"},
               {"name": "uav_person7_2", "path": "data_seq/UAV123/person7", "startFrame": 1249,
"endFrame": 2065, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person7_2.txt", "object_class": "person"},
               {"name": "uav_person8_1", "path": "data_seq/UAV123/person8", "startFrame": 1,
"endFrame": 1075, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person8_1.txt", "object_class": "person"},
               {"name": "uav_person8_2", "path": "data_seq/UAV123/person8", "startFrame": 1075,
"endFrame": 1525, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person8_2.txt", "object_class": "person"},
               {"name": "uav_person9", "path": "data_seq/UAV123/person9", "startFrame": 1,
"endFrame": 661, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/person9.txt", "object_class": "person"},
               {"name": "uav_truck1", "path": "data_seq/UAV123/truck1", "startFrame": 1, "endFrame":
463, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/truck1.txt", "object_class": "truck"},
               {"name": "uav_truck2", "path": "data_seq/UAV123/truck2", "startFrame": 1, "endFrame":
385, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/truck2.txt", "object_class": "truck"},
               {"name": "uav_truck3", "path": "data_seq/UAV123/truck3", "startFrame": 1, "endFrame":
535, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/truck3.txt", "object_class": "truck"},
               {"name": "uav_truck4_1", "path": "data_seq/UAV123/truck4", "startFrame": 1,
"endFrame": 577, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/truck4_1.txt", "object_class": "truck"},
               {"name": "uav_truck4_2", "path": "data_seq/UAV123/truck4", "startFrame": 577,
"endFrame": 1261, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/truck4_2.txt", "object_class": "truck"},
               {"name": "uav_uav1_1", "path": "data_seq/UAV123/uav1", "startFrame": 1, "endFrame":
1555, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/uav1_1.txt", "object_class": "aircraft"},
               {"name": "uav_uav1_2", "path": "data_seq/UAV123/uav1", "startFrame": 1555,
"endFrame": 2377, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/uav1_2.txt", "object_class": "aircraft"},
               {"name": "uav_uav1_3", "path": "data_seq/UAV123/uav1", "startFrame": 2473,
"endFrame": 3469, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/uav1_3.txt", "object_class": "aircraft"},
               {"name": "uav_uav2", "path": "data_seq/UAV123/uav2", "startFrame": 1, "endFrame":
133, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/uav2.txt", "object_class": "aircraft"},
               {"name": "uav_uav3", "path": "data_seq/UAV123/uav3", "startFrame": 1, "endFrame":
265, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/uav3.txt", "object_class": "aircraft"},
               {"name": "uav_uav4", "path": "data_seq/UAV123/uav4", "startFrame": 1, "endFrame":
157, "nz": 6,
                "ext": "jpg", "anno_path": "anno/UAV123/uav4.txt", "object_class": "aircraft"},
```

```
                {"name": "uav_uav5", "path": "data_seq/UAV123/uav5", "startFrame": 1, "endFrame":
139, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/uav5.txt", "object_class": "aircraft"},
                {"name": "uav_uav6", "path": "data_seq/UAV123/uav6", "startFrame": 1, "endFrame":
109, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/uav6.txt", "object_class": "aircraft"},
                {"name": "uav_uav7", "path": "data_seq/UAV123/uav7", "startFrame": 1, "endFrame":
373, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/uav7.txt", "object_class": "aircraft"},
                {"name": "uav_uav8", "path": "data_seq/UAV123/uav8", "startFrame": 1, "endFrame":
301, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/uav8.txt", "object_class": "aircraft"},
                {"name": "uav_wakeboard1", "path": "data_seq/UAV123/wakeboard1", "startFrame": 1,
"endFrame": 421, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/wakeboard1.txt", "object_class": "person"},
                {"name": "uav_wakeboard10", "path": "data_seq/UAV123/wakeboard10", "startFrame":
1, "endFrame": 469,
                 "nz": 6, "ext": "jpg", "anno_path": "anno/UAV123/wakeboard10.txt", "object_class":
"person"},
                {"name": "uav_wakeboard2", "path": "data_seq/UAV123/wakeboard2", "startFrame": 1,
"endFrame": 733, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/wakeboard2.txt", "object_class": "person"},
                {"name": "uav_wakeboard3", "path": "data_seq/UAV123/wakeboard3", "startFrame": 1,
"endFrame": 823, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/wakeboard3.txt", "object_class": "person"},
                {"name": "uav_wakeboard4", "path": "data_seq/UAV123/wakeboard4", "startFrame": 1,
"endFrame": 697, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/wakeboard4.txt", "object_class": "person"},
                {"name": "uav_wakeboard5", "path": "data_seq/UAV123/wakeboard5", "startFrame": 1,
"endFrame": 1675, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/wakeboard5.txt", "object_class": "person"},
                {"name": "uav_wakeboard6", "path": "data_seq/UAV123/wakeboard6", "startFrame": 1,
"endFrame": 1165, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/wakeboard6.txt", "object_class": "person"},
                {"name": "uav_wakeboard7", "path": "data_seq/UAV123/wakeboard7", "startFrame": 1,
"endFrame": 199, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/wakeboard7.txt", "object_class": "person"},
                {"name": "uav_wakeboard8", "path": "data_seq/UAV123/wakeboard8", "startFrame": 1,
"endFrame": 1543, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/wakeboard8.txt", "object_class": "person"},
                {"name": "uav_wakeboard9", "path": "data_seq/UAV123/wakeboard9", "startFrame": 1,
"endFrame": 355, "nz": 6,
                 "ext": "jpg", "anno_path": "anno/UAV123/wakeboard9.txt", "object_class": "person"}
            ]
            return sequence_info_list
import numpy as np
from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
import os
from lib.test.utils.load_text import load_text
class TrackingNetDataset(BaseDataset):
```

```
    def __init__(self):
        super().__init__()
        self.base_path = self.env_settings.trackingnet_path
        sets = 'TEST'
        if not isinstance(sets, (list, tuple)):
            if sets == 'TEST':
                sets = ['TEST']
            elif sets == 'TRAIN':
                sets = ['TRAIN_{}'.format(i) for i in range(5)]
        self.sequence_list = self._list_sequences(self.base_path, sets)
    def get_sequence_list(self):
        return SequenceList([self._construct_sequence(set, seq_name) for set, seq_name in
self.sequence_list])
    def _construct_sequence(self, set, sequence_name):
        anno_path = '{}/{}/anno/{}.txt'.format(self.base_path, set, sequence_name)
        ground_truth_rect = load_text(str(anno_path), delimiter=',', dtype=np.float64,
backend='numpy')
        frames_path = '{}/{}/frames/{}'.format(self.base_path, set, sequence_name)
        frame_list = [frame for frame in os.listdir(frames_path) if frame.endswith(".jpg")]
        frame_list.sort(key=lambda f: int(f[:-4]))
        frames_list = [os.path.join(frames_path, frame) for frame in frame_list]
        return Sequence(sequence_name, frames_list, 'trackingnet', ground_truth_rect.reshape(-1,
4))
    def __len__(self):
        return len(self.sequence_list)
    def _list_sequences(self, root, set_ids):
        sequence_list = []
        for s in set_ids:
            anno_dir = os.path.join(root, s, "anno")
            sequences_cur_set = [(s, os.path.splitext(f)[0]) for f in os.listdir(anno_dir) if
f.endswith('.txt')]
            sequence_list += sequences_cur_set
        return sequence_list
import importlib
import os
from collections import OrderedDict
from lib.test.evaluation.environment import env_settings
import time
import cv2 as cv
from lib.utils.lmdb_utils import decode_img
from pathlib import Path
import numpy as np
def trackerlist(name: str, parameter_name: str, dataset_name: str, run_ids = None, display_name: str =
None,
                result_only=False):
    if run_ids is None or isinstance(run_ids, int):
        run_ids = [run_ids]
    return [Tracker(name, parameter_name, dataset_name, run_id, display_name, result_only) for
run_id in run_ids]
class Tracker:
```

```
    def __init__(self, name: str, parameter_name: str, dataset_name: str, run_id: int = None,
display_name: str = None,
                 result_only=False):
        assert run_id is None or isinstance(run_id, int)
        self.name = name
        self.parameter_name = parameter_name
        self.dataset_name = dataset_name
        self.run_id = run_id
        self.display_name = display_name
        env = env_settings()
        if self.run_id is None:
            self.results_dir = '{}/{}/{}'.format(env.results_path, self.name, self.parameter_name)
        else:
            self.results_dir = '{}/{}/{}_{:03d}'.format(env.results_path, self.name, self.parameter_name,
self.run_id)
        if result_only:
            self.results_dir = '{}/{}'.format(env.results_path, self.name)
        tracker_module_abspath = os.path.abspath(os.path.join(os.path.dirname(__file__),
                                                              '..', 'tracker', '%s.py' %
self.name))
        if os.path.isfile(tracker_module_abspath):
            tracker_module = importlib.import_module('lib.test.tracker.{}'.format(self.name))
            self.tracker_class = tracker_module.get_tracker_class()
        else:
            self.tracker_class = None
    def create_tracker(self, params):
        tracker = self.tracker_class(params, self.dataset_name)
        return tracker
    def run_sequence(self, seq, debug=None):
        params = self.get_parameters()
        debug_ = debug
        if debug is None:
            debug_ = getattr(params, 'debug', 0)
        params.debug = debug_
        init_info = seq.init_info()
        tracker = self.create_tracker(params)
        output = self._track_sequence(tracker, seq, init_info)
        return output
    def _track_sequence(self, tracker, seq, init_info):
        output = {'target_bbox': [],
                  'time': []}
        if tracker.params.save_all_boxes:
            output['all_boxes'] = []
            output['all_scores'] = []
        def _store_outputs(tracker_out: dict, defaults=None):
            defaults = {} if defaults is None else defaults
            for key in output.keys():
                val = tracker_out.get(key, defaults.get(key, None))
                if key in tracker_out or val is not None:
                    output[key].append(val)
```

```
            image = self._read_image(seq.frames[0])
            start_time = time.time()
            out = tracker.initialize(image, init_info)
            if out is None:
                out = {}
            prev_output = OrderedDict(out)
            init_default = {'target_bbox': init_info.get('init_bbox'),
                            'time': time.time() - start_time}
            if tracker.params.save_all_boxes:
                init_default['all_boxes'] = out['all_boxes']
                init_default['all_scores'] = out['all_scores']
            _store_outputs(out, init_default)
            for frame_num, frame_path in enumerate(seq.frames[1:], start=1):
                image = self._read_image(frame_path)
                start_time = time.time()
                info = seq.frame_info(frame_num)
                info['previous_output'] = prev_output
                if len(seq.ground_truth_rect) > 1:
                    info['gt_bbox'] = seq.ground_truth_rect[frame_num]
                out = tracker.track(image, info)
                prev_output = OrderedDict(out)
                _store_outputs(out, {'time': time.time() - start_time})
            for key in ['target_bbox', 'all_boxes', 'all_scores']:
                if key in output and len(output[key]) <= 1:
                    output.pop(key)
            return output
        def run_video(self, videofilepath, optional_box=None, debug=None, visdom_info=None,
save_results=False):
            params = self.get_parameters()
            debug_ = debug
            if debug is None:
                debug_ = getattr(params, 'debug', 0)
            params.debug = debug_
            params.tracker_name = self.name
            params.param_name = self.parameter_name
            multiobj_mode = getattr(params, 'multiobj_mode', getattr(self.tracker_class,
'multiobj_mode', 'default'))
            if multiobj_mode == 'default':
                tracker = self.create_tracker(params)
            elif multiobj_mode == 'parallel':
                tracker = MultiObjectWrapper(self.tracker_class, params, self.visdom, fast_load=True)
            else:
                raise ValueError('Unknown multi object mode {}'.format(multiobj_mode))
            assert os.path.isfile(videofilepath), "Invalid param {}".format(videofilepath)
            ", videofilepath must be a valid videofile"
            output_boxes = []
            cap = cv.VideoCapture(videofilepath)
            display_name = 'Display: ' + tracker.params.tracker_name
            cv.namedWindow(display_name, cv.WINDOW_NORMAL | cv.WINDOW_KEEPRATIO)
            cv.resizeWindow(display_name, 960, 720)
```

```
            success, frame = cap.read()
            cv.imshow(display_name, frame)
            def _build_init_info(box):
                return {'init_bbox': box}
            if success is not True:
                print("Read frame from {} failed.".format(videofilepath))
                exit(-1)
            if optional_box is not None:
                assert isinstance(optional_box, (list, tuple))
                assert len(optional_box) == 4, "valid box's foramt is [x,y,w,h]"
                tracker.initialize(frame, _build_init_info(optional_box))
                output_boxes.append(optional_box)
            else:
                while True:
                    frame_disp = frame.copy()
                    cv.putText(frame_disp, 'Select target ROI and press ENTER', (20, 30),
cv.FONT_HERSHEY_COMPLEX_SMALL,
                               1.5, (0, 0, 0), 1)
                    x, y, w, h = cv.selectROI(display_name, frame_disp, fromCenter=False)
                    init_state = [x, y, w, h]
                    tracker.initialize(frame, _build_init_info(init_state))
                    output_boxes.append(init_state)
                    break
            while True:
                ret, frame = cap.read()
                if frame is None:
                    break
                frame_disp = frame.copy()
                out = tracker.track(frame)
                state = [int(s) for s in out['target_bbox']]
                output_boxes.append(state)
                cv.rectangle(frame_disp, (state[0], state[1]), (state[2] + state[0], state[3] + state[1]),
                             (0, 255, 0), 5)
                font_color = (0, 0, 0)
                cv.putText(frame_disp, 'Tracking!', (20, 30), cv.FONT_HERSHEY_COMPLEX_SMALL, 1,
                           font_color, 1)
                cv.putText(frame_disp, 'Press r to reset', (20, 55), cv.FONT_HERSHEY_COMPLEX_SMALL,
1,
                           font_color, 1)
                cv.putText(frame_disp, 'Press q to quit', (20, 80), cv.FONT_HERSHEY_COMPLEX_SMALL,
1,
                           font_color, 1)
                cv.imshow(display_name, frame_disp)
                key = cv.waitKey(1)
                if key == ord('q'):
                    break
                elif key == ord('r'):
                    ret, frame = cap.read()
                    frame_disp = frame.copy()
                    cv.putText(frame_disp, 'Select target ROI and press ENTER', (20, 30),
```

```
cv.FONT_HERSHEY_COMPLEX_SMALL, 1.5,
                                    (0, 0, 0), 1)
                    cv.imshow(display_name, frame_disp)
                    x, y, w, h = cv.selectROI(display_name, frame_disp, fromCenter=False)
                    init_state = [x, y, w, h]
                    tracker.initialize(frame, _build_init_info(init_state))
                    output_boxes.append(init_state)
            cap.release()
            cv.destroyAllWindows()
            if save_results:
                if not os.path.exists(self.results_dir):
                    os.makedirs(self.results_dir)
                video_name = Path(videofilepath).stem
                base_results_path = os.path.join(self.results_dir, 'video_{}'.format(video_name))
                tracked_bb = np.array(output_boxes).astype(int)
                bbox_file = '{}.txt'.format(base_results_path)
                np.savetxt(bbox_file, tracked_bb, delimiter='\t', fmt='%d')
        def get_parameters(self):
            param_module = importlib.import_module('lib.test.parameter.{}'.format(self.name))
            params = param_module.parameters(self.parameter_name)
            return params
        def _read_image(self, image_file: str):
            if isinstance(image_file, str):
                im = cv.imread(image_file)
                return cv.cvtColor(im, cv.COLOR_BGR2RGB)
            elif isinstance(image_file, list) and len(image_file) == 2:
                return decode_img(image_file[0], image_file[1])
            else:
                raise ValueError("type of image_file should be str or list")
import os
import numpy as np
from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
from lib.test.utils.load_text import load_text, load_str
class TNL2kDataset(BaseDataset):
        def __init__(self):
            super().__init__()
            self.base_path = self.env_settings.tnl2k_path
            self.sequence_list = self._get_sequence_list()
        def get_sequence_list(self):
            return SequenceList([self._construct_sequence(s) for s in self.sequence_list])
        def _construct_sequence(self, sequence_name):
            anno_path = '{}/{}/groundtruth.txt'.format(self.base_path, sequence_name)
            ground_truth_rect = load_text(str(anno_path), delimiter=',', dtype=np.float64)
            text_dsp_path = '{}/{}/language.txt'.format(self.base_path, sequence_name)
            text_dsp = load_str(text_dsp_path)
            frames_path = '{}/{}/imgs'.format(self.base_path, sequence_name)
            frames_list = [f for f in os.listdir(frames_path)]
            frames_list = sorted(frames_list)
            frames_list = ['{}/{}'.format(frames_path, frame_i) for frame_i in frames_list]
            return Sequence(sequence_name, frames_list, 'tnl2k', ground_truth_rect.reshape(-1, 4),
```

```
text_dsp=text_dsp)
    def __len__(self):
        return len(self.sequence_list)
    def _get_sequence_list(self):
        sequence_list = []
        for seq in os.listdir(self.base_path):
            if os.path.isdir(os.path.join(self.base_path, seq)):
                sequence_list.append(seq)
        return sequence_list
import numpy as np
from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
import os
import glob
import six
class TC128Dataset(BaseDataset):
    def __init__(self):
        super().__init__()
        self.base_path = self.env_settings.tc128_path
        self.anno_files = sorted(glob.glob(
            os.path.join(self.base_path, '*/*_gt.txt')))
        self.seq_dirs = [os.path.dirname(f) for f in self.anno_files]
        self.seq_names = [os.path.basename(d) for d in self.seq_dirs]
        self.range_files = [glob.glob(os.path.join(d, '*_frames.txt'))[0] for d in self.seq_dirs]
    def get_sequence_list(self):
        return SequenceList([self._construct_sequence(s) for s in self.seq_names])
    def _construct_sequence(self, sequence_name):
        if isinstance(sequence_name, six.string_types):
            if not sequence_name in self.seq_names:
                raise Exception('Sequence {} not found.'.format(sequence_name))
            index = self.seq_names.index(sequence_name)
        frames = np.loadtxt(self.range_files[index], dtype=int, delimiter=',')
        img_files = [os.path.join(self.seq_dirs[index], 'img/%04d.jpg' % f) for f in range(frames[0],
frames[1] + 1)]
        anno = np.loadtxt(self.anno_files[index], delimiter=',')
        assert len(img_files) == len(anno)
        assert anno.shape[1] == 4
        return Sequence(sequence_name, img_files, 'tc128', anno.reshape(-1, 4))
    def __len__(self):
        return len(self.seq_names)
import numpy as np
from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
import os
import glob
import six
class TC128CEDataset(BaseDataset):
    def __init__(self):
        super().__init__()
        self.base_path = self.env_settings.tc128_path
        self.anno_files = sorted(glob.glob(
            os.path.join(self.base_path, '*/*_gt.txt')))
```

```
            self.anno_files = [s for s in self.anno_files if ''_ce'' in s]
            self.seq_dirs = [os.path.dirname(f) for f in self.anno_files]
            self.seq_names = [os.path.basename(d) for d in self.seq_dirs]
            self.range_files = [glob.glob(os.path.join(d, '*_frames.txt'))[0] for d in self.seq_dirs]
        def get_sequence_list(self):
            return SequenceList([self._construct_sequence(s) for s in self.seq_names])
        def _construct_sequence(self, sequence_name):
            if isinstance(sequence_name, six.string_types):
                if not sequence_name in self.seq_names:
                    raise Exception('Sequence {} not found.'.format(sequence_name))
                index = self.seq_names.index(sequence_name)
            frames = np.loadtxt(self.range_files[index], dtype=int, delimiter=',')
            img_files = [os.path.join(self.seq_dirs[index], 'img/%04d.jpg' % f) for f in range(frames[0],
frames[1] + 1)]
            anno = np.loadtxt(self.anno_files[index], delimiter=',')
            assert len(img_files) == len(anno)
            assert anno.shape[1] == 4
            return Sequence(sequence_name, img_files, 'tc128', anno.reshape(-1, 4))
        def __len__(self):
            return len(self.seq_names)
    import numpy as np
    import multiprocessing
    import os
    import sys
    from itertools import product
    from collections import OrderedDict
    from lib.test.evaluation import Sequence, Tracker
    import torch
    def _save_tracker_output(seq: Sequence, tracker: Tracker, output: dict):
        if not os.path.exists(tracker.results_dir):
            print("create tracking result dir:", tracker.results_dir)
            os.makedirs(tracker.results_dir)
        if seq.dataset in ['trackingnet', 'got10k']:
            if not os.path.exists(os.path.join(tracker.results_dir, seq.dataset)):
                os.makedirs(os.path.join(tracker.results_dir, seq.dataset))
        if seq.dataset in ['trackingnet', 'got10k']:
            base_results_path = os.path.join(tracker.results_dir, seq.dataset, seq.name)
        else:
            base_results_path = os.path.join(tracker.results_dir, seq.name)
        def save_bb(file, data):
            tracked_bb = np.array(data).astype(int)
            np.savetxt(file, tracked_bb, delimiter='\t', fmt='%d')
        def save_time(file, data):
            exec_times = np.array(data).astype(float)
            np.savetxt(file, exec_times, delimiter='\t', fmt='%f')
        def save_score(file, data):
            scores = np.array(data).astype(float)
            np.savetxt(file, scores, delimiter='\t', fmt='%.2f')
    def run_sequence(seq: Sequence, tracker: Tracker, debug=False, num_gpu=8):
        try:
```

```
            worker_name = multiprocessing.current_process().name
            worker_id = int(worker_name[worker_name.find('-') + 1:]) - 1
            gpu_id = worker_id % num_gpu
            torch.cuda.set_device(gpu_id)
        except:
            pass
import numpy as np
from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
from lib.test.utils.load_text import load_text
class OTBDataset(BaseDataset):
    def __init__(self):
        super().__init__()
        self.base_path = self.env_settings.otb_path
        self.sequence_info_list = self._get_sequence_info_list()
    def get_sequence_list(self):
        return SequenceList([self._construct_sequence(s) for s in self.sequence_info_list])
    def _construct_sequence(self, sequence_info):
        sequence_path = sequence_info['path']
        nz = sequence_info['nz']
        ext = sequence_info['ext']
        start_frame = sequence_info['startFrame']
        end_frame = sequence_info['endFrame']
        init_omit = 0
        if 'initOmit' in sequence_info:
            init_omit = sequence_info['initOmit']
        frames = ['{base_path}/{sequence_path}/{frame:0{nz}}.{ext}'.format(base_path=self.base_path,
        sequence_path=sequence_path, frame=frame_num, nz=nz, ext=ext) for frame_num in range(start_frame+init_omit, end_frame+1)]
        anno_path = '{}/{}/groundtruth.txt'.format(self.base_path, sequence_info['name'])
        ground_truth_rect = load_text(str(anno_path), delimiter=(',', None), dtype=np.float64, backend='numpy')
        return Sequence(sequence_info['name'], frames, 'otb', ground_truth_rect[init_omit:,:],
                        object_class=sequence_info['object_class'])
    def __len__(self):
        return len(self.sequence_info_list)
    def _get_sequence_info_list(self):
        sequence_info_list = [
            {"name": "Basketball", "path": "Basketball/img", "startFrame": 1, "endFrame": 725, "nz": 4, "ext": "jpg", "anno_path": "Basketball/groundtruth_rect.txt",
             "object_class": "person"},
            {"name": "Biker", "path": "Biker/img", "startFrame": 1, "endFrame": 142, "nz": 4, "ext": "jpg", "anno_path": "Biker/groundtruth_rect.txt",
             "object_class": "person head"},
            {"name": "Bird1", "path": "Bird1/img", "startFrame": 1, "endFrame": 408, "nz": 4, "ext": "jpg", "anno_path": "Bird1/groundtruth_rect.txt",
             "object_class": "bird"},
            {"name": "Bird2", "path": "Bird2/img", "startFrame": 1, "endFrame": 99, "nz": 4, "ext": "jpg", "anno_path": "Bird2/groundtruth_rect.txt",
             "object_class": "bird"},
```

```
{"name": "BlurBody", "path": "BlurBody/img", "startFrame": 1, "endFrame": 334, "nz": 4,
"ext": "jpg", "anno_path": "BlurBody/groundtruth_rect.txt",
 "object_class": "person"},
{"name": "BlurCar1", "path": "BlurCar1/img", "startFrame": 247, "endFrame": 988, "nz": 4,
"ext": "jpg", "anno_path": "BlurCar1/groundtruth_rect.txt",
 "object_class": "car"},
{"name": "BlurCar2", "path": "BlurCar2/img", "startFrame": 1, "endFrame": 585, "nz": 4,
"ext": "jpg", "anno_path": "BlurCar2/groundtruth_rect.txt",
 "object_class": "car"},
{"name": "BlurCar3", "path": "BlurCar3/img", "startFrame": 3, "endFrame": 359, "nz": 4,
"ext": "jpg", "anno_path": "BlurCar3/groundtruth_rect.txt",
 "object_class": "car"},
{"name": "BlurCar4", "path": "BlurCar4/img", "startFrame": 18, "endFrame": 397, "nz": 4,
"ext": "jpg", "anno_path": "BlurCar4/groundtruth_rect.txt",
 "object_class": "car"},
{"name": "BlurFace", "path": "BlurFace/img", "startFrame": 1, "endFrame": 493, "nz": 4,
"ext": "jpg", "anno_path": "BlurFace/groundtruth_rect.txt",
 "object_class": "face"},
{"name": "BlurOwl", "path": "BlurOwl/img", "startFrame": 1, "endFrame": 631, "nz": 4,
"ext": "jpg", "anno_path": "BlurOwl/groundtruth_rect.txt",
 "object_class": "other"},
{"name": "Board", "path": "Board/img", "startFrame": 1, "endFrame": 698, "nz": 5, "ext":
"jpg", "anno_path": "Board/groundtruth_rect.txt",
 "object_class": "other"},
{"name": "Bolt", "path": "Bolt/img", "startFrame": 1, "endFrame": 350, "nz": 4, "ext": "jpg",
"anno_path": "Bolt/groundtruth_rect.txt",
 "object_class": "person"},
{"name": "Bolt2", "path": "Bolt2/img", "startFrame": 1, "endFrame": 293, "nz": 4, "ext":
"jpg", "anno_path": "Bolt2/groundtruth_rect.txt",
 "object_class": "person"},
{"name": "Box", "path": "Box/img", "startFrame": 1, "endFrame": 1161, "nz": 4, "ext": "jpg",
"anno_path": "Box/groundtruth_rect.txt",
 "object_class": "other"},
{"name": "Boy", "path": "Boy/img", "startFrame": 1, "endFrame": 602, "nz": 4, "ext": "jpg",
"anno_path": "Boy/groundtruth_rect.txt",
 "object_class": "face"},
{"name": "Car1", "path": "Car1/img", "startFrame": 1, "endFrame": 1020, "nz": 4, "ext":
"jpg", "anno_path": "Car1/groundtruth_rect.txt",
 "object_class": "car"},
{"name": "Car2", "path": "Car2/img", "startFrame": 1, "endFrame": 913, "nz": 4, "ext":
"jpg", "anno_path": "Car2/groundtruth_rect.txt",
 "object_class": "car"},
{"name": "Car24", "path": "Car24/img", "startFrame": 1, "endFrame": 3059, "nz": 4, "ext":
"jpg", "anno_path": "Car24/groundtruth_rect.txt",
 "object_class": "car"},
{"name": "Car4", "path": "Car4/img", "startFrame": 1, "endFrame": 659, "nz": 4, "ext":
"jpg", "anno_path": "Car4/groundtruth_rect.txt",
 "object_class": "car"},
{"name": "CarDark", "path": "CarDark/img", "startFrame": 1, "endFrame": 393, "nz": 4,
"ext": "jpg", "anno_path": "CarDark/groundtruth_rect.txt",
```

```
                    "object_class": "car"},
                {"name": "CarScale", "path": "CarScale/img", "startFrame": 1, "endFrame": 252, "nz": 4,
"ext": "jpg", "anno_path": "CarScale/groundtruth_rect.txt",
                    "object_class": "car"},
                {"name": "ClifBar", "path": "ClifBar/img", "startFrame": 1, "endFrame": 472, "nz": 4, "ext":
"jpg", "anno_path": "ClifBar/groundtruth_rect.txt",
                    "object_class": "other"},
                {"name": "Coke", "path": "Coke/img", "startFrame": 1, "endFrame": 291, "nz": 4, "ext":
"jpg", "anno_path": "Coke/groundtruth_rect.txt",
                    "object_class": "other"},
                {"name": "Couple", "path": "Couple/img", "startFrame": 1, "endFrame": 140, "nz": 4, "ext":
"jpg", "anno_path": "Couple/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "Coupon", "path": "Coupon/img", "startFrame": 1, "endFrame": 327, "nz": 4,
"ext": "jpg", "anno_path": "Coupon/groundtruth_rect.txt",
                    "object_class": "other"},
                {"name": "Crossing", "path": "Crossing/img", "startFrame": 1, "endFrame": 120, "nz": 4,
"ext": "jpg", "anno_path": "Crossing/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "Crowds", "path": "Crowds/img", "startFrame": 1, "endFrame": 347, "nz": 4, "ext":
"jpg", "anno_path": "Crowds/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "Dancer", "path": "Dancer/img", "startFrame": 1, "endFrame": 225, "nz": 4, "ext":
"jpg", "anno_path": "Dancer/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "Dancer2", "path": "Dancer2/img", "startFrame": 1, "endFrame": 150, "nz": 4,
"ext": "jpg", "anno_path": "Dancer2/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "David", "path": "David/img", "startFrame": 300, "endFrame": 770, "nz": 4, "ext":
"jpg", "anno_path": "David/groundtruth_rect.txt",
                    "object_class": "face"},
                {"name": "David2", "path": "David2/img", "startFrame": 1, "endFrame": 537, "nz": 4, "ext":
"jpg", "anno_path": "David2/groundtruth_rect.txt",
                    "object_class": "face"},
                {"name": "David3", "path": "David3/img", "startFrame": 1, "endFrame": 252, "nz": 4, "ext":
"jpg", "anno_path": "David3/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "Deer", "path": "Deer/img", "startFrame": 1, "endFrame": 71, "nz": 4, "ext": "jpg",
"anno_path": "Deer/groundtruth_rect.txt",
                    "object_class": "mammal"},
                {"name": "Diving", "path": "Diving/img", "startFrame": 1, "endFrame": 215, "nz": 4, "ext":
"jpg", "anno_path": "Diving/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "Dog", "path": "Dog/img", "startFrame": 1, "endFrame": 127, "nz": 4, "ext": "jpg",
"anno_path": "Dog/groundtruth_rect.txt",
                    "object_class": "dog"},
                {"name": "Dog1", "path": "Dog1/img", "startFrame": 1, "endFrame": 1350, "nz": 4, "ext":
"jpg", "anno_path": "Dog1/groundtruth_rect.txt",
                    "object_class": "dog"},
                {"name": "Doll", "path": "Doll/img", "startFrame": 1, "endFrame": 3872, "nz": 4, "ext": "jpg",
```

```
"anno_path": "Doll/groundtruth_rect.txt",
                "object_class": "other"},
               {"name": "DragonBaby", "path": "DragonBaby/img", "startFrame": 1, "endFrame": 113,
"nz": 4, "ext": "jpg", "anno_path": "DragonBaby/groundtruth_rect.txt",
                "object_class": "face"},
               {"name": "Dudek", "path": "Dudek/img", "startFrame": 1, "endFrame": 1145, "nz": 4, "ext":
"jpg", "anno_path": "Dudek/groundtruth_rect.txt",
                "object_class": "face"},
               {"name": "FaceOcc1", "path": "FaceOcc1/img", "startFrame": 1, "endFrame": 892, "nz": 4,
"ext": "jpg", "anno_path": "FaceOcc1/groundtruth_rect.txt",
                "object_class": "face"},
               {"name": "FaceOcc2", "path": "FaceOcc2/img", "startFrame": 1, "endFrame": 812, "nz": 4,
"ext": "jpg", "anno_path": "FaceOcc2/groundtruth_rect.txt",
                "object_class": "face"},
               {"name": "Fish", "path": "Fish/img", "startFrame": 1, "endFrame": 476, "nz": 4, "ext": "jpg",
"anno_path": "Fish/groundtruth_rect.txt",
                "object_class": "other"},
               {"name": "FleetFace", "path": "FleetFace/img", "startFrame": 1, "endFrame": 707, "nz": 4,
"ext": "jpg", "anno_path": "FleetFace/groundtruth_rect.txt",
                "object_class": "face"},
               {"name": "Football", "path": "Football/img", "startFrame": 1, "endFrame": 362, "nz": 4,
"ext": "jpg", "anno_path": "Football/groundtruth_rect.txt",
                "object_class": "person head"},
               {"name": "Football1", "path": "Football1/img", "startFrame": 1, "endFrame": 74, "nz": 4,
"ext": "jpg", "anno_path": "Football1/groundtruth_rect.txt",
                "object_class": "face"},
               {"name": "Freeman1", "path": "Freeman1/img", "startFrame": 1, "endFrame": 326, "nz":
4, "ext": "jpg", "anno_path": "Freeman1/groundtruth_rect.txt",
                "object_class": "face"},
               {"name": "Freeman3", "path": "Freeman3/img", "startFrame": 1, "endFrame": 460, "nz":
4, "ext": "jpg", "anno_path": "Freeman3/groundtruth_rect.txt",
                "object_class": "face"},
               {"name": "Freeman4", "path": "Freeman4/img", "startFrame": 1, "endFrame": 283, "nz":
4, "ext": "jpg", "anno_path": "Freeman4/groundtruth_rect.txt",
                "object_class": "face"},
               {"name": "Girl", "path": "Girl/img", "startFrame": 1, "endFrame": 500, "nz": 4, "ext": "jpg",
"anno_path": "Girl/groundtruth_rect.txt",
                "object_class": "face"},
               {"name": "Girl2", "path": "Girl2/img", "startFrame": 1, "endFrame": 1500, "nz": 4, "ext":
"jpg", "anno_path": "Girl2/groundtruth_rect.txt",
                "object_class": "person"},
               {"name": "Gym", "path": "Gym/img", "startFrame": 1, "endFrame": 767, "nz": 4, "ext":
"jpg", "anno_path": "Gym/groundtruth_rect.txt",
                "object_class": "person"},
               {"name": "Human2", "path": "Human2/img", "startFrame": 1, "endFrame": 1128, "nz": 4,
"ext": "jpg", "anno_path": "Human2/groundtruth_rect.txt",
                "object_class": "person"},
               {"name": "Human3", "path": "Human3/img", "startFrame": 1, "endFrame": 1698, "nz": 4,
"ext": "jpg", "anno_path": "Human3/groundtruth_rect.txt",
                "object_class": "person"},
```

```
{"name": "Human4_2", "path": "Human4/img", "startFrame": 1, "endFrame": 667, "nz": 4,
"ext": "jpg", "anno_path": "Human4/groundtruth_rect.2.txt",
"object_class": "person"},
{"name": "Human4", "path": "Human4/img", "startFrame": 1, "endFrame": 667, "nz": 4,
"ext": "jpg", "anno_path": "Human4/groundtruth_rect.2.txt",
"object_class": "person"},
{"name": "Human5", "path": "Human5/img", "startFrame": 1, "endFrame": 713, "nz": 4,
"ext": "jpg", "anno_path": "Human5/groundtruth_rect.txt",
"object_class": "person"},
{"name": "Human6", "path": "Human6/img", "startFrame": 1, "endFrame": 792, "nz": 4,
"ext": "jpg", "anno_path": "Human6/groundtruth_rect.txt",
"object_class": "person"},
{"name": "Human7", "path": "Human7/img", "startFrame": 1, "endFrame": 250, "nz": 4,
"ext": "jpg", "anno_path": "Human7/groundtruth_rect.txt",
"object_class": "person"},
{"name": "Human8", "path": "Human8/img", "startFrame": 1, "endFrame": 128, "nz": 4,
"ext": "jpg", "anno_path": "Human8/groundtruth_rect.txt",
"object_class": "person"},
{"name": "Human9", "path": "Human9/img", "startFrame": 1, "endFrame": 305, "nz": 4,
"ext": "jpg", "anno_path": "Human9/groundtruth_rect.txt",
"object_class": "person"},
{"name": "Ironman", "path": "Ironman/img", "startFrame": 1, "endFrame": 166, "nz": 4,
"ext": "jpg", "anno_path": "Ironman/groundtruth_rect.txt",
"object_class": "person head"},
{"name": "Jogging", "path": "Jogging/img", "startFrame": 1, "endFrame": 307, "nz": 4,
"ext": "jpg", "anno_path": "Jogging/groundtruth_rect.1.txt",
"object_class": "person"},
{"name": "Jump", "path": "Jump/img", "startFrame": 1, "endFrame": 122, "nz": 4, "ext":
"jpg", "anno_path": "Jump/groundtruth_rect.txt",
"object_class": "person"},
{"name": "Jumping", "path": "Jumping/img", "startFrame": 1, "endFrame": 313, "nz": 4,
"ext": "jpg", "anno_path": "Jumping/groundtruth_rect.txt",
"object_class": "face"},
{"name": "KiteSurf", "path": "KiteSurf/img", "startFrame": 1, "endFrame": 84, "nz": 4, "ext":
"jpg", "anno_path": "KiteSurf/groundtruth_rect.txt",
"object_class": "face"},
{"name": "Lemming", "path": "Lemming/img", "startFrame": 1, "endFrame": 1336, "nz":
4, "ext": "jpg", "anno_path": "Lemming/groundtruth_rect.txt",
"object_class": "other"},
{"name": "Liquor", "path": "Liquor/img", "startFrame": 1, "endFrame": 1741, "nz": 4, "ext":
"jpg", "anno_path": "Liquor/groundtruth_rect.txt",
"object_class": "other"},
{"name": "Man", "path": "Man/img", "startFrame": 1, "endFrame": 134, "nz": 4, "ext": "jpg",
"anno_path": "Man/groundtruth_rect.txt",
"object_class": "face"},
{"name": "Matrix", "path": "Matrix/img", "startFrame": 1, "endFrame": 100, "nz": 4, "ext":
"jpg", "anno_path": "Matrix/groundtruth_rect.txt",
"object_class": "person head"},
{"name": "Mhyang", "path": "Mhyang/img", "startFrame": 1, "endFrame": 1490, "nz": 4,
"ext": "jpg", "anno_path": "Mhyang/groundtruth_rect.txt",
```

```
                    "object_class": "face"},
                {"name": "MotorRolling", "path": "MotorRolling/img", "startFrame": 1, "endFrame": 164,
"nz": 4, "ext": "jpg", "anno_path": "MotorRolling/groundtruth_rect.txt",
                    "object_class": "vehicle"},
                {"name": "MountainBike", "path": "MountainBike/img", "startFrame": 1, "endFrame": 228,
"nz": 4, "ext": "jpg", "anno_path": "MountainBike/groundtruth_rect.txt",
                    "object_class": "bicycle"},
                {"name": "Panda", "path": "Panda/img", "startFrame": 1, "endFrame": 1000, "nz": 4, "ext":
"jpg", "anno_path": "Panda/groundtruth_rect.txt",
                    "object_class": "mammal"},
                {"name": "RedTeam", "path": "RedTeam/img", "startFrame": 1, "endFrame": 1918, "nz":
4, "ext": "jpg", "anno_path": "RedTeam/groundtruth_rect.txt",
                    "object_class": "vehicle"},
                {"name": "Rubik", "path": "Rubik/img", "startFrame": 1, "endFrame": 1997, "nz": 4, "ext":
"jpg", "anno_path": "Rubik/groundtruth_rect.txt",
                    "object_class": "other"},
                {"name": "Shaking", "path": "Shaking/img", "startFrame": 1, "endFrame": 365, "nz": 4,
"ext": "jpg", "anno_path": "Shaking/groundtruth_rect.txt",
                    "object_class": "face"},
                {"name": "Singer1", "path": "Singer1/img", "startFrame": 1, "endFrame": 351, "nz": 4,
"ext": "jpg", "anno_path": "Singer1/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "Singer2", "path": "Singer2/img", "startFrame": 1, "endFrame": 366, "nz": 4,
"ext": "jpg", "anno_path": "Singer2/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "Skater", "path": "Skater/img", "startFrame": 1, "endFrame": 160, "nz": 4, "ext":
"jpg", "anno_path": "Skater/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "Skater2", "path": "Skater2/img", "startFrame": 1, "endFrame": 435, "nz": 4, "ext":
"jpg", "anno_path": "Skater2/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "Skating1", "path": "Skating1/img", "startFrame": 1, "endFrame": 400, "nz": 4,
"ext": "jpg", "anno_path": "Skating1/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "Skating2", "path": "Skating2/img", "startFrame": 1, "endFrame": 473, "nz": 4,
"ext": "jpg", "anno_path": "Skating2/groundtruth_rect.1.txt",
                    "object_class": "person"},
                {"name": "Skating2_1", "path": "Skating2/img", "startFrame": 1, "endFrame": 473, "nz": 4,
"ext": "jpg", "anno_path": "Skating2/groundtruth_rect.1.txt",
                    "object_class": "person"},
                {"name": "Skating2_2", "path": "Skating2/img", "startFrame": 1, "endFrame": 473, "nz": 4,
"ext": "jpg", "anno_path": "Skating2/groundtruth_rect.2.txt",
                    "object_class": "person"},
                {"name": "Skiing", "path": "Skiing/img", "startFrame": 1, "endFrame": 81, "nz": 4, "ext":
"jpg", "anno_path": "Skiing/groundtruth_rect.txt",
                    "object_class": "person"},
                {"name": "Soccer", "path": "Soccer/img", "startFrame": 1, "endFrame": 392, "nz": 4, "ext":
"jpg", "anno_path": "Soccer/groundtruth_rect.txt",
                    "object_class": "face"},
                {"name": "Subway", "path": "Subway/img", "startFrame": 1, "endFrame": 175, "nz": 4,
```

```
"ext": "jpg", "anno_path": "Subway/groundtruth_rect.txt",
                    "object_class": "person"},
                   {"name": "Surfer", "path": "Surfer/img", "startFrame": 1, "endFrame": 376, "nz": 4, "ext":
"jpg", "anno_path": "Surfer/groundtruth_rect.txt",
                    "object_class": "person head"},
                   {"name": "Suv", "path": "Suv/img", "startFrame": 1, "endFrame": 945, "nz": 4, "ext": "jpg",
"anno_path": "Suv/groundtruth_rect.txt",
                    "object_class": "car"},
                   {"name": "Sylvester", "path": "Sylvester/img", "startFrame": 1, "endFrame": 1345, "nz": 4,
"ext": "jpg", "anno_path": "Sylvester/groundtruth_rect.txt",
                    "object_class": "other"},
                   {"name": "Tiger1", "path": "Tiger1/img", "startFrame": 1, "endFrame": 354, "nz": 4, "ext":
"jpg", "anno_path": "Tiger1/groundtruth_rect.txt", "initOmit": 5,
                    "object_class": "other"},
                   {"name": "Tiger2", "path": "Tiger2/img", "startFrame": 1, "endFrame": 365, "nz": 4, "ext":
"jpg", "anno_path": "Tiger2/groundtruth_rect.txt",
                    "object_class": "other"},
                   {"name": "Toy", "path": "Toy/img", "startFrame": 1, "endFrame": 271, "nz": 4, "ext": "jpg",
"anno_path": "Toy/groundtruth_rect.txt",
                    "object_class": "other"},
                   {"name": "Trans", "path": "Trans/img", "startFrame": 1, "endFrame": 124, "nz": 4, "ext":
"jpg", "anno_path": "Trans/groundtruth_rect.txt",
                    "object_class": "other"},
                   {"name": "Trellis", "path": "Trellis/img", "startFrame": 1, "endFrame": 569, "nz": 4, "ext":
"jpg", "anno_path": "Trellis/groundtruth_rect.txt",
                    "object_class": "face"},
                   {"name": "Twinnings", "path": "Twinnings/img", "startFrame": 1, "endFrame": 472, "nz":
4, "ext": "jpg", "anno_path": "Twinnings/groundtruth_rect.txt",
                    "object_class": "other"},
                   {"name": "Vase", "path": "Vase/img", "startFrame": 1, "endFrame": 271, "nz": 4, "ext":
"jpg", "anno_path": "Vase/groundtruth_rect.txt",
                    "object_class": "other"},
                   {"name": "Walking", "path": "Walking/img", "startFrame": 1, "endFrame": 412, "nz": 4,
"ext": "jpg", "anno_path": "Walking/groundtruth_rect.txt",
                    "object_class": "person"},
                   {"name": "Walking2", "path": "Walking2/img", "startFrame": 1, "endFrame": 500, "nz": 4,
"ext": "jpg", "anno_path": "Walking2/groundtruth_rect.txt",
                    "object_class": "person"},
                   {"name": "Woman", "path": "Woman/img", "startFrame": 1, "endFrame": 597, "nz": 4,
"ext": "jpg", "anno_path": "Woman/groundtruth_rect.txt",
                    "object_class": "person"}
               ]
               return sequence_info_list
import numpy as np
from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
from lib.test.utils.load_text import load_text
class NFSDataset(BaseDataset):
    def __init__(self):
        super().__init__()
        self.base_path = self.env_settings.nfs_path
```

```
            self.sequence_info_list = self._get_sequence_info_list()
        def get_sequence_list(self):
            return SequenceList([self._construct_sequence(s) for s in self.sequence_info_list])
        def _construct_sequence(self, sequence_info):
            sequence_path = sequence_info['path']
            nz = sequence_info['nz']
            ext = sequence_info['ext']
            start_frame = sequence_info['startFrame']
            end_frame = sequence_info['endFrame']
            init_omit = 0
            if 'initOmit' in sequence_info:
                init_omit = sequence_info['initOmit']
            frames =
['{base_path}/{sequence_path}/{frame:0{nz}}.{ext}'.format(base_path=self.base_path,
            sequence_path=sequence_path, frame=frame_num, nz=nz, ext=ext) for frame_num in
range(start_frame+init_omit, end_frame+1)]
            anno_path = f"{self.base_path}/{sequence_info['name'][4:]}/30/groundtruth.txt"
            ground_truth_rect = load_text(str(anno_path), delimiter=',', dtype=np.float64)
            return Sequence(sequence_info['name'][4:], frames, 'nfs', ground_truth_rect[init_omit:,:],
                            object_class=sequence_info['object_class'])
        def __len__(self):
            return len(self.sequence_info_list)
        def _get_sequence_info_list(self):
            sequence_info_list = [
                {"name": "nfs_Gymnastics", "path": "sequences/Gymnastics", "startFrame": 1,
"endFrame": 368, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_Gymnastics.txt", "object_class": "person",
'occlusion': False},
                {"name": "nfs_MachLoop_jet", "path": "sequences/MachLoop_jet", "startFrame": 1,
"endFrame": 99, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_MachLoop_jet.txt", "object_class":
"aircraft", 'occlusion': False},
                {"name": "nfs_Skiing_red", "path": "sequences/Skiing_red", "startFrame": 1, "endFrame":
69, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_Skiing_red.txt", "object_class": "person", 'occlusion':
False},
                {"name": "nfs_Skydiving", "path": "sequences/Skydiving", "startFrame": 1, "endFrame":
196, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_Skydiving.txt", "object_class": "person", 'occlusion':
True},
                {"name": "nfs_airboard_1", "path": "sequences/airboard_1", "startFrame": 1, "endFrame":
425, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_airboard_1.txt", "object_class": "ball", 'occlusion':
False},
                {"name": "nfs_airplane_landing", "path": "sequences/airplane_landing", "startFrame": 1,
"endFrame": 81, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_airplane_landing.txt", "object_class":
"aircraft", 'occlusion': False},
                {"name": "nfs_airtable_3", "path": "sequences/airtable_3", "startFrame": 1, "endFrame":
482, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_airtable_3.txt", "object_class": "ball", 'occlusion': False},
                {"name": "nfs_basketball_1", "path": "sequences/basketball_1", "startFrame": 1,
"endFrame": 282, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_1.txt", "object_class": "ball",
'occlusion': False},
                {"name": "nfs_basketball_2", "path": "sequences/basketball_2", "startFrame": 1,
"endFrame": 102, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_2.txt", "object_class": "ball",
'occlusion': False},
```

{"name": "nfs_basketball_3", "path": "sequences/basketball_3", "startFrame": 1, "endFrame": 421, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_3.txt", "object_class": "ball", 'occlusion': False},
{"name": "nfs_basketball_6", "path": "sequences/basketball_6", "startFrame": 1, "endFrame": 224, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_6.txt", "object_class": "ball", 'occlusion': False},
{"name": "nfs_basketball_7", "path": "sequences/basketball_7", "startFrame": 1, "endFrame": 240, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_7.txt", "object_class": "person", 'occlusion': True},
{"name": "nfs_basketball_player", "path": "sequences/basketball_player", "startFrame": 1, "endFrame": 369, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_player.txt", "object_class": "person", 'occlusion': True},
{"name": "nfs_basketball_player_2", "path": "sequences/basketball_player_2", "startFrame": 1, "endFrame": 437, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_player_2.txt", "object_class": "person", 'occlusion': False},
{"name": "nfs_beach_flipback_person", "path": "sequences/beach_flipback_person", "startFrame": 1, "endFrame": 61, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_beach_flipback_person.txt", "object_class": "person head", 'occlusion': False},
{"name": "nfs_bee", "path": "sequences/bee", "startFrame": 1, "endFrame": 45, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bee.txt", "object_class": "insect", 'occlusion': False},
{"name": "nfs_biker_acrobat", "path": "sequences/biker_acrobat", "startFrame": 1, "endFrame": 128, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_biker_acrobat.txt", "object_class": "bicycle", 'occlusion': False},
{"name": "nfs_biker_all_1", "path": "sequences/biker_all_1", "startFrame": 1, "endFrame": 113, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_biker_all_1.txt", "object_class": "person", 'occlusion': False},
{"name": "nfs_biker_head_2", "path": "sequences/biker_head_2", "startFrame": 1, "endFrame": 132, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_biker_head_2.txt", "object_class": "person head", 'occlusion': False},
{"name": "nfs_biker_head_3", "path": "sequences/biker_head_3", "startFrame": 1, "endFrame": 254, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_biker_head_3.txt", "object_class": "person head", 'occlusion': False},
{"name": "nfs_biker_upper_body", "path": "sequences/biker_upper_body", "startFrame": 1, "endFrame": 194, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_biker_upper_body.txt", "object_class": "person", 'occlusion': False},
{"name": "nfs_biker_whole_body", "path": "sequences/biker_whole_body", "startFrame": 1, "endFrame": 572, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_biker_whole_body.txt", "object_class": "person", 'occlusion': True},
{"name": "nfs_billiard_2", "path": "sequences/billiard_2", "startFrame": 1, "endFrame": 604, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_billiard_2.txt", "object_class": "ball", 'occlusion': False},
{"name": "nfs_billiard_3", "path": "sequences/billiard_3", "startFrame": 1, "endFrame": 698, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_billiard_3.txt", "object_class": "ball", 'occlusion': False},
{"name": "nfs_billiard_6", "path": "sequences/billiard_6", "startFrame": 1, "endFrame": 771, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_billiard_6.txt", "object_class": "ball", 'occlusion': False},
{"name": "nfs_billiard_7", "path": "sequences/billiard_7", "startFrame": 1, "endFrame": 724, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_billiard_7.txt", "object_class": "ball", 'occlusion': False},
{"name": "nfs_billiard_8", "path": "sequences/billiard_8", "startFrame": 1, "endFrame": 778, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_billiard_8.txt", "object_class": "ball", 'occlusion': False},
{"name": "nfs_bird_2", "path": "sequences/bird_2", "startFrame": 1, "endFrame": 476, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bird_2.txt", "object_class": "bird", 'occlusion': False},

```
{"name": "nfs_book", "path": "sequences/book", "startFrame": 1, "endFrame": 288, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_book.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_bottle", "path": "sequences/bottle", "startFrame": 1, "endFrame": 2103, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bottle.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_bowling_1", "path": "sequences/bowling_1", "startFrame": 1, "endFrame": 303, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bowling_1.txt", "object_class": "ball", 'occlusion': True},
{"name": "nfs_bowling_2", "path": "sequences/bowling_2", "startFrame": 1, "endFrame": 710, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bowling_2.txt", "object_class": "ball", 'occlusion': True},
{"name": "nfs_bowling_3", "path": "sequences/bowling_3", "startFrame": 1, "endFrame": 271, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bowling_3.txt", "object_class": "ball", 'occlusion': True},
{"name": "nfs_bowling_6", "path": "sequences/bowling_6", "startFrame": 1, "endFrame": 260, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bowling_6.txt", "object_class": "ball", 'occlusion': False},
{"name": "nfs_bowling_ball", "path": "sequences/bowling_ball", "startFrame": 1, "endFrame": 275, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bowling_ball.txt", "object_class": "ball", 'occlusion': True},
{"name": "nfs_bunny", "path": "sequences/bunny", "startFrame": 1, "endFrame": 705, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bunny.txt", "object_class": "mammal", 'occlusion': False},
{"name": "nfs_car", "path": "sequences/car", "startFrame": 1, "endFrame": 2020, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car.txt", "object_class": "car", 'occlusion': True},
{"name": "nfs_car_camaro", "path": "sequences/car_camaro", "startFrame": 1, "endFrame": 36, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_camaro.txt", "object_class": "car", 'occlusion': False},
{"name": "nfs_car_drifting", "path": "sequences/car_drifting", "startFrame": 1, "endFrame": 173, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_drifting.txt", "object_class": "car", 'occlusion': False},
{"name": "nfs_car_jumping", "path": "sequences/car_jumping", "startFrame": 1, "endFrame": 22, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_jumping.txt", "object_class": "car", 'occlusion': False},
{"name": "nfs_car_rc_rolling", "path": "sequences/car_rc_rolling", "startFrame": 1, "endFrame": 62, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_rc_rolling.txt", "object_class": "car", 'occlusion': False},
{"name": "nfs_car_rc_rotating", "path": "sequences/car_rc_rotating", "startFrame": 1, "endFrame": 80, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_rc_rotating.txt", "object_class": "car", 'occlusion': False},
{"name": "nfs_car_side", "path": "sequences/car_side", "startFrame": 1, "endFrame": 108, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_side.txt", "object_class": "car", 'occlusion': False},
{"name": "nfs_car_white", "path": "sequences/car_white", "startFrame": 1, "endFrame": 2063, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_white.txt", "object_class": "car", 'occlusion': False},
{"name": "nfs_cheetah", "path": "sequences/cheetah", "startFrame": 1, "endFrame": 167, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_cheetah.txt", "object_class": "mammal", 'occlusion': True},
{"name": "nfs_cup", "path": "sequences/cup", "startFrame": 1, "endFrame": 1281, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_cup.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_cup_2", "path": "sequences/cup_2", "startFrame": 1, "endFrame": 182, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_cup_2.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_dog", "path": "sequences/dog", "startFrame": 1, "endFrame": 1030, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_dog.txt", "object_class": "dog", 'occlusion': True},
{"name": "nfs_dog_1", "path": "sequences/dog_1", "startFrame": 1, "endFrame": 168, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_dog_1.txt", "object_class": "dog", 'occlusion': False},
{"name": "nfs_dog_3", "path": "sequences/dog_3", "startFrame": 1, "endFrame": 200, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_dog_3.txt", "object_class": "dog", 'occlusion': False},
```

{"name": "nfs_dogs", "path": "sequences/dogs", "startFrame": 1, "endFrame": 198, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_dogs.txt", "object_class": "dog", 'occlusion': True},
{"name": "nfs_dollar", "path": "sequences/dollar", "startFrame": 1, "endFrame": 1426, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_dollar.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_drone", "path": "sequences/drone", "startFrame": 1, "endFrame": 70, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_drone.txt", "object_class": "aircraft", 'occlusion': False},
{"name": "nfs_ducks_lake", "path": "sequences/ducks_lake", "startFrame": 1, "endFrame": 107, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_ducks_lake.txt", "object_class": "bird", 'occlusion': False},
{"name": "nfs_exit", "path": "sequences/exit", "startFrame": 1, "endFrame": 359, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_exit.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_first", "path": "sequences/first", "startFrame": 1, "endFrame": 435, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_first.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_flower", "path": "sequences/flower", "startFrame": 1, "endFrame": 448, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_flower.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_footbal_skill", "path": "sequences/footbal_skill", "startFrame": 1, "endFrame": 131, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_footbal_skill.txt", "object_class": "ball", 'occlusion': True},
{"name": "nfs_helicopter", "path": "sequences/helicopter", "startFrame": 1, "endFrame": 310, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_helicopter.txt", "object_class": "aircraft", 'occlusion': False},
{"name": "nfs_horse_jumping", "path": "sequences/horse_jumping", "startFrame": 1, "endFrame": 117, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_horse_jumping.txt", "object_class": "horse", 'occlusion': True},
{"name": "nfs_horse_running", "path": "sequences/horse_running", "startFrame": 1, "endFrame": 139, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_horse_running.txt", "object_class": "horse", 'occlusion': False},
{"name": "nfs_iceskating_6", "path": "sequences/iceskating_6", "startFrame": 1, "endFrame": 603, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_iceskating_6.txt", "object_class": "person", 'occlusion': False},
{"name": "nfs_jellyfish_5", "path": "sequences/jellyfish_5", "startFrame": 1, "endFrame": 746, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_jellyfish_5.txt", "object_class": "invertebrate", 'occlusion': False},
{"name": "nfs_kid_swing", "path": "sequences/kid_swing", "startFrame": 1, "endFrame": 169, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_kid_swing.txt", "object_class": "person", 'occlusion': False},
{"name": "nfs_motorcross", "path": "sequences/motorcross", "startFrame": 1, "endFrame": 39, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_motorcross.txt", "object_class": "vehicle", 'occlusion': True},
{"name": "nfs_motorcross_kawasaki", "path": "sequences/motorcross_kawasaki", "startFrame": 1, "endFrame": 65, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_motorcross_kawasaki.txt", "object_class": "vehicle", 'occlusion': False},
{"name": "nfs_parkour", "path": "sequences/parkour", "startFrame": 1, "endFrame": 58, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_parkour.txt", "object_class": "person head", 'occlusion': False},
{"name": "nfs_person_scooter", "path": "sequences/person_scooter", "startFrame": 1, "endFrame": 413, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_person_scooter.txt", "object_class": "person", 'occlusion': True},
{"name": "nfs_pingpong_2", "path": "sequences/pingpong_2", "startFrame": 1, "endFrame": 1277, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_pingpong_2.txt", "object_class": "ball",

'occlusion': False},
{"name": "nfs_pingpong_7", "path": "sequences/pingpong_7", "startFrame": 1, "endFrame": 1290, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_pingpong_7.txt", "object_class": "ball", 'occlusion': False},
{"name": "nfs_pingpong_8", "path": "sequences/pingpong_8", "startFrame": 1, "endFrame": 296, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_pingpong_8.txt", "object_class": "ball", 'occlusion': False},
{"name": "nfs_purse", "path": "sequences/purse", "startFrame": 1, "endFrame": 968, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_purse.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_rubber", "path": "sequences/rubber", "startFrame": 1, "endFrame": 1328, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_rubber.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_running", "path": "sequences/running", "startFrame": 1, "endFrame": 677, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_running.txt", "object_class": "person", 'occlusion': False},
{"name": "nfs_running_100_m", "path": "sequences/running_100_m", "startFrame": 1, "endFrame": 313, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_running_100_m.txt", "object_class": "person", 'occlusion': True},
{"name": "nfs_running_100_m_2", "path": "sequences/running_100_m_2", "startFrame": 1, "endFrame": 337, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_running_100_m_2.txt", "object_class": "person", 'occlusion': True},
{"name": "nfs_running_2", "path": "sequences/running_2", "startFrame": 1, "endFrame": 363, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_running_2.txt", "object_class": "person", 'occlusion': False},
{"name": "nfs_shuffleboard_1", "path": "sequences/shuffleboard_1", "startFrame": 1, "endFrame": 42, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffleboard_1.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_shuffleboard_2", "path": "sequences/shuffleboard_2", "startFrame": 1, "endFrame": 41, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffleboard_2.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_shuffleboard_4", "path": "sequences/shuffleboard_4", "startFrame": 1, "endFrame": 62, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffleboard_4.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_shuffleboard_5", "path": "sequences/shuffleboard_5", "startFrame": 1, "endFrame": 32, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffleboard_5.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_shuffleboard_6", "path": "sequences/shuffleboard_6", "startFrame": 1, "endFrame": 52, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffleboard_6.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_shuffletable_2", "path": "sequences/shuffletable_2", "startFrame": 1, "endFrame": 372, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffletable_2.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_shuffletable_3", "path": "sequences/shuffletable_3", "startFrame": 1, "endFrame": 368, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffletable_3.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_shuffletable_4", "path": "sequences/shuffletable_4", "startFrame": 1, "endFrame": 101, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffletable_4.txt", "object_class": "other", 'occlusion': False},
{"name": "nfs_ski_long", "path": "sequences/ski_long", "startFrame": 1, "endFrame": 274, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_ski_long.txt", "object_class": "person", 'occlusion': False},
{"name": "nfs_soccer_ball", "path": "sequences/soccer_ball", "startFrame": 1, "endFrame": 163, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_soccer_ball.txt", "object_class": "ball", 'occlusion':

```
False},
                {"name": "nfs_soccer_ball_2", "path": "sequences/soccer_ball_2", "startFrame": 1,
"endFrame": 1934, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_soccer_ball_2.txt", "object_class": "ball",
'occlusion': False},
                {"name": "nfs_soccer_ball_3", "path": "sequences/soccer_ball_3", "startFrame": 1,
"endFrame": 1381, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_soccer_ball_3.txt", "object_class": "ball",
'occlusion': False},
                {"name": "nfs_soccer_player_2", "path": "sequences/soccer_player_2", "startFrame": 1,
"endFrame": 475, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_soccer_player_2.txt", "object_class":
"person", 'occlusion': False},
                {"name": "nfs_soccer_player_3", "path": "sequences/soccer_player_3", "startFrame": 1,
"endFrame": 319, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_soccer_player_3.txt", "object_class":
"person", 'occlusion': True},
                {"name": "nfs_stop_sign", "path": "sequences/stop_sign", "startFrame": 1, "endFrame":
302, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_stop_sign.txt", "object_class": "other", 'occlusion':
False},
                {"name": "nfs_suv", "path": "sequences/suv", "startFrame": 1, "endFrame": 2584, "nz": 5,
"ext": "jpg", "anno_path": "anno/nfs_suv.txt", "object_class": "car", 'occlusion': False},
                {"name": "nfs_tiger", "path": "sequences/tiger", "startFrame": 1, "endFrame": 1556, "nz":
5, "ext": "jpg", "anno_path": "anno/nfs_tiger.txt", "object_class": "mammal", 'occlusion': False},
                {"name": "nfs_walking", "path": "sequences/walking", "startFrame": 1, "endFrame": 555,
"nz": 5, "ext": "jpg", "anno_path": "anno/nfs_walking.txt", "object_class": "person", 'occlusion': False},
                {"name": "nfs_walking_3", "path": "sequences/walking_3", "startFrame": 1, "endFrame":
1427, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_walking_3.txt", "object_class": "person", 'occlusion':
False},
                {"name": "nfs_water_ski_2", "path": "sequences/water_ski_2", "startFrame": 1,
"endFrame": 47, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_water_ski_2.txt", "object_class": "person",
'occlusion': False},
                {"name": "nfs_yoyo", "path": "sequences/yoyo", "startFrame": 1, "endFrame": 67, "nz": 5,
"ext": "jpg", "anno_path": "anno/nfs_yoyo.txt", "object_class": "other", 'occlusion': False},
                {"name": "nfs_zebra_fish", "path": "sequences/zebra_fish", "startFrame": 1, "endFrame":
671, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_zebra_fish.txt", "object_class": "fish", 'occlusion': False},
            ]
            return sequence_info_list
from lib.test.evaluation.environment import EnvSettings
from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
from lib.utils.lmdb_utils import *
class LaSOTLmdbDataset(BaseDataset):
    def __init__(self):
        super().__init__()
        self.base_path = self.env_settings.lasot_lmdb_path
        self.sequence_list = self._get_sequence_list()
        self.clean_list = self.clean_seq_list()
    def clean_seq_list(self):
        clean_lst = []
        for i in range(len(self.sequence_list)):
            cls, _ = self.sequence_list[i].split('-')
            clean_lst.append(cls)
        return clean_lst
    def get_sequence_list(self):
```

```
            return SequenceList([self._construct_sequence(s) for s in self.sequence_list])
        def __len__(self):
            return len(self.sequence_list)
        def _get_sequence_list(self):
import numpy as np
from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
from lib.test.utils.load_text import load_text
class LaSOTExtensionSubsetDataset(BaseDataset):
        def __init__(self):
            super().__init__()
            self.base_path = self.env_settings.lasot_extension_subset_path
            self.sequence_list = self._get_sequence_list()
            self.clean_list = self.clean_seq_list()
        def clean_seq_list(self):
            clean_lst = []
            for i in range(len(self.sequence_list)):
                cls, _ = self.sequence_list[i].split('-')
                clean_lst.append(cls)
            return  clean_lst
        def get_sequence_list(self):
            return SequenceList([self._construct_sequence(s) for s in self.sequence_list])
        def _construct_sequence(self, sequence_name):
            class_name = sequence_name.split('-')[0]
            anno_path = '{}/{}/{}/groundtruth.txt'.format(self.base_path, class_name, sequence_name)
            ground_truth_rect = load_text(str(anno_path), delimiter=',', dtype=np.float64)
            occlusion_label_path = '{}/{}/{}/full_occlusion.txt'.format(self.base_path, class_name,
sequence_name)
            full_occlusion = load_text(str(occlusion_label_path), delimiter=',', dtype=np.float64,
backend='numpy')
            out_of_view_label_path = '{}/{}/{}/out_of_view.txt'.format(self.base_path, class_name,
sequence_name)
            out_of_view = load_text(str(out_of_view_label_path), delimiter=',', dtype=np.float64,
backend='numpy')
            target_visible = np.logical_and(full_occlusion == 0, out_of_view == 0)
            frames_path = '{}/{}/{}/img'.format(self.base_path, class_name, sequence_name)
            frames_list = ['{}/{:08d}.jpg'.format(frames_path, frame_number) for frame_number in
range(1, ground_truth_rect.shape[0] + 1)]
            target_class = class_name
            return Sequence(sequence_name, frames_list, 'lasot_extension_subset',
ground_truth_rect.reshape(-1, 4),
                            object_class=target_class, target_visible=target_visible)
        def __len__(self):
            return len(self.sequence_list)
        def _get_sequence_list(self):
return sequence_list
import numpy as np
from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
from lib.test.utils.load_text import load_text
class LaSOTDataset(BaseDataset):
        def __init__(self):
```

```
            super().__init__()
            self.base_path = self.env_settings.lasot_path
            self.sequence_list = self._get_sequence_list()
            self.clean_list = self.clean_seq_list()
        def clean_seq_list(self):
            clean_lst = []
            for i in range(len(self.sequence_list)):
                cls, _ = self.sequence_list[i].split('-')
                clean_lst.append(cls)
            return    clean_lst
        def get_sequence_list(self):
            return SequenceList([self._construct_sequence(s) for s in self.sequence_list])
        def __len__(self):
            return len(self.sequence_list)
    import numpy as np
    from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
    from lib.test.utils.load_text import load_text
    import os
    class ITBDataset(BaseDataset):
        def __len__(self):
            return len(self.sequence_info_list)
        def get_fileNames(self, rootdir):
            fs = []
            fs_all = []
            for root, dirs, files in os.walk(rootdir, topdown=True):
                files.sort()
                files.sort(key=len)
                if files is not None:
                    for name in files:
                        _, ending = os.path.splitext(name)
                        if ending == ".jpg":
                            _, root_ = os.path.split(root)
                            fs.append(os.path.join(root_, name))
                            fs_all.append(os.path.join(root, name))
            return fs_all, fs
        def _get_sequence_info_list(self, base_path):
            sequence_info_list = []
            for scene in os.listdir(base_path):
                if '.' in scene:
                    continue
                videos = os.listdir(os.path.join(base_path, scene))
                for video in videos:
                    _, fs = self.get_fileNames(os.path.join(base_path, scene, video))
                    video_tmp = {"name": video, "path": scene + '/' + video, "startFrame": 1,
    "endFrame": len(fs),
                                 "nz": len(fs[0].split('/')[-1].split('.')[0]), "ext": "jpg",
                                 "anno_path": scene + '/' + video + "/groundtruth.txt",
                                 "object_class": "unknown"}
                    sequence_info_list.append(video_tmp)
            return sequence_info_list
```