

```

1 declare module '*.vue' {
2   import type { DefineComponent } from 'vue'
3   const component: DefineComponent<{}, {}, any>
4   export default component
5 }
6 <!DOCTYPE html>
7 <html lang="">
8   <head>
9     <meta charset="UTF-8">
10    <link rel="icon" href="/favicon.ico">
11    <meta name="viewport" content="width=device-width, initial-scale=1.0">
12    <title>Vite App</title>
13  </head>
14  <body>
15    <div id="app"></div>
16    <script type="module" src="/src/main.ts"></script>
17  </body>
18 </html>
19 import { fileURLToPath, URL } from 'node:url'
20 import { defineConfig } from 'vite'
21 import vue from '@vitejs/plugin-vue'
22 import vueDevTools from 'vite-plugin-vue-devtools'
23 export default defineConfig({
24   plugins: [
25     vue(),
26     vueDevTools(),
27   ],
28   resolve: {
29     alias: {
30       '@': fileURLToPath(new URL('./src', import.meta.url))
31     },
32   },
33 })
34 import { createRouter, createWebHistory } from 'vue-router';
35 import FileUploadVue from '@components/FileUpload.vue';
36 const router = createRouter({
37   history: createWebHistory(import.meta.env.BASE_URL),
38   routes: [
39     {
40       path: '/',
41       name: 'home',
42       component: FileUploadVue,
43     },
44   ],
45 });
46 router.beforeEach((to, from, next) => {
47   next();
48 });
49 export default router;
50 import './assets/main.css'

```

```

1  import { createApp } from 'vue'
2  import App from './App.vue'
3  import router from './router'
4  const app = createApp(App)
5  app.use(router)
6  app.mount('#app')
7  <!-- This icon is from <https:
8  <template>
9    <svg
10      xmlns="http://www.w3.org/2000/svg"
11      xmlns:xlink="http://www.w3.org/1999/xlink"
12      aria-hidden="true"
13      role="img"
14      class="iconify iconify--mdi"
15      width="24"
16      height="24"
17      preserveAspectRatio="xMidYMid meet"
18      viewBox="0 0 24 24"
19    >
20      <path
21        d="M20 18v-4h-3v1h-2v-1H9v1H7v-1H4v4h16M6.33 8l-1.74 4H7v-1h2v1h6v-1h2v1h2.41l-
22 1.74-4H6.33M9 5v1h6V5H9m12.84 7.61c1.22 16.48 16.8V18c0 .53-.21 1-.6 1.41c-.44-.85 59-
23 1.4 59H4c-.55 0-1-.19-1.4-.59C2.21 19 2 18.53 2 18v-4.59c0-.32 0.06-.58 16-.8L4.5 7.22C4.84 6.41
24 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
25 1.41v1h.67c.88 0 1.49 41 1.83 1.22l2.34 5.39z"
26      fill="currentColor"
27    ></path>
28  </svg>
29 </template>
30 <template>
31   <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="currentColor">
32     <path
33       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
34 5.5 0 0 0-.114-7.665 5.5 5.5 0 0 0-7.666-.105l-.61.61z"
35     />
36   </svg>
37 </template>
38 <template>
39   <svg xmlns="http://www.w3.org/2000/svg" width="18" height="20" fill="currentColor">
40     <path
41       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-
42 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
43 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-
44 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-
45 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-
46 4.606a1 1 0 1 0 .894-1.789l-.894 1.789zm9 1l.447-.894a1 1 0 0 0-.894 0L9 1zm-2.447 1.06a1 1 0 1
47 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-
48 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
49 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
50 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 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
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
3 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-
4 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
5 1 .894-1.789zM8 9v2.5h2V9H8zm8.553-4.894l-2 1 .894 1.789 2-1-.894-1.789zm.894 0l-2-1-.894
6 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-
7 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-
8 1.789-2-1-.894 1.789zM0 5v2.5h2V5H0zm9.447 13.106l-2-1-.894 1.789 2 1 .894-1.789zm0 1.789l2-
9 1-.894-1.789-2 1 .894 1.789zM10 19v-2.5H8V19h2zm-6.553-3.894l-2-1-.894 1.789 2 1 .894-
10 1.789zM2 15v-2.5H0V15h2zm13.447 1.894l2-1-.894-1.789-2 1 .894 1.789zM18 15v-2.5h-2V15h2z"
11 />
12 </svg>
13 </template>
14 <template>
15 <svg xmlns="http://www.w3.org/2000/svg" width="20" height="17" fill="currentColor">
16 <path
17 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-
18 1.666L9.447 3.086zM1 2.253L.447 1.42A1 1 0 0 0 0 2.253h1zm0 13H0a1 1 0 0 0 1.553.833L1
19 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
20 1.42zM19 2.253h1a1 1 0 0 0-.447-.833L19 2.253zm0 13l-.553.833A1 1 0 0 0 20 15.253h-1zm-
21 9.553-.833a1 1 0 1 0 1.107 1.666L9.447 14.42zM9 2.253v13h2v-13H9zm1.553-.833C9.203.523 7.42
22 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
23 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
24 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-
25 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-
26 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
27 16.421 0 14.5 0v2zm3.5.253v13h2v-13h-2zm1.553 12.167C18.203 13.523 16.421 13 14.5 13v2c1.573
28 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
29 15.431 12.928 15 14.5 15v-2z"
30 />
31 </svg>
32 </template>
33 <template>
34 <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="currentColor">
35 <path
36 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-
37 4-4l.707-.707A1 1 0 0 0 11 14v1zm-4.707-1.293a1 1 0 0 0 0-1.414 1.414l1.414-
38 1.414zm-.707.707l-.707-.707.707.707zM9 11v-1a1 1 0 0 0-.707.293L9 11zm-4 0h1a1 1 0 0 0-1-
39 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
40 1v6h2V7h-2zm0 6a1 1 0 0 1-1 1v2a3 3 0 0 0 3-3h-2zm-1 1h-2v2h2v-2zm-3 1v4h2v-4h-2zm1.707
41 3.293l-4-4-1.414 1.414 4 4 1.414-1.414zM11 14H7v2h4v-2zm-4 0c-.276 0-.525-.111-.707-.293l-
42 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
43 1.414zM9 12h4v-2H9v2zm4 0a3 3 0 0 0 3-3h-2a1 1 0 0 1-1 1v2zm3-3V3h-2v6h2zm0-6a3 3 0 0 0-
44 3-3v2a1 1 0 0 1 1 1h2zm-3-3H3v2h10V0zM3 0a3 3 0 0 0-3 3h2a1 1 0 0 1 1-1V0zM0 3v6h2V3H0zm0
45 6a3 3 0 0 0 3 3v-2a1 1 0 0 1-1 1H0zm3 3h2v-2H3v2zm1-1v4h2v-4H4zm1.707 4.707l.586-.586-
46 1.414-1.414-.586.586 1.414 1.414z"
47 />
48 </svg>
49 </template>
50 <template>

```

```
1      <div class="file-uploader">
2          <!-- 主要内容区域：文件上传和视频播放并列 -->
3          <div class="main-content">
4              <!-- 左侧上传区域 -->
5              <div class="upload-container">
6                  <div class="upload-section">
7                      <div class="file-input-group">
8                          <h3>视频文件 (MP4)</h3>
9                          <input
10                             type="file"
11                             accept="video/mp4"
12                             @change="handleVideoSelect"
13                             :disabled="uploading"
14                         />
15                          <div v-if="videoFile" class="selected-file">
16                              已选择: {{ videoFile.name }}
17                          </div>
18                      </div>
19                      <div class="file-input-group">
20                          <h3>文本文件 (TXT)</h3>
21                          <input
22                             type="file"
23                             accept=".txt"
24                             @change="handleTextSelect"
25                             :disabled="uploading"
26                         />
27                          <div v-if="textFile" class="selected-file">
28                              已选择: {{ textFile.name }}
29                          </div>
30                      </div>
31                  </div>
32                  <!-- 上传按钮 -->
33                  <button
34                      @click="handleUploadAndProcess"
35                      :disabled="!canUpload || uploading"
36                      class="action-button"
37                  >
38                      {{ uploading ? '上传中...' : '上传文件' }}
39                  </button>
40                  <!-- 处理按钮 -->
41                  <button
42                      v-if="groupId && !processing"
43                      @click="processFiles"
44                      :disabled="processing || isProcessed"
45                      class="action-button process-button"
46                  >
47                      {{ isProcessed ? '处理完成' : '开始处理' }}
48                  </button>
49                  <!-- 上传进度条 -->
50                  <div v-if="uploading" class="progress-container">
```

```

1      <div class="progress-label">上传进度</div>
2      <div class="progress-bar">
3          <div class="progress-fill" :style="{ width: `${uploadProgress}%` }"></div>
4      </div>
5      <div class="progress-text">{{ uploadProgress }}%</div>
6  </div>
7  <!-- 处理进度状态 -->
8  <div v-if="processing" class="progress-container">
9      <div class="progress-label">处理中...</div>
10     <div class="processing-spinner"></div>
11 </div>
12 <!-- 简化的状态消息 -->
13 <div v-if="message" class="status-message" :class="messageType">
14     {{ message }}
15 </div>
16 </div>
17 <!-- 右侧视频区域 -->
18 <div v-if="processedVideoUrl && isProcessed" class="video-section">
19     <div class="video-player-container">
20         <h4>处理后的视频</h4>
21         <!-- 视频播放器 -->
22         <div class="video-wrapper">
23             <video
24                 ref="videoPlayer"
25                 class="video-player"
26                 controls
27                 @error="handleVideoError"
28                 @loadeddata="handleVideoLoaded"
29                 crossorigin="anonymous"
30             >
31                 <source :src="processedVideoUrl" type="video/mp4; codecs='avc1.42E01E,
32 mp4a.40.2'">
33                 您的浏览器不支持 HTML5 视频播放
34             </video>
35             <!-- 加载提示 -->
36             <div v-if="isVideoLoading" class="video-loading">
37                 <div class="loading-spinner"></div>
38                 <span>视频加载中...</span>
39             </div>
40         </div>
41         <!-- 视频控制按钮 -->
42         <div class="video-controls">
43             <button
44                 @click="handleDownload"
45                 class="control-button download-button"
46             >
47                 <span class="button-icon"></span>
48                 下载视频
49             </button>
50             <button
    
```

```
1          v-if="videoError"
2          @click="reloadVideo"
3          class="control-button reload-button"
4      >
5          <span class="button-icon">🔄</span>
6          重新加载
7      </button>
8  </div>
9  <!-- 错误提示 -->
10 <div v-if="videoError" class="video-error">
11     <p>{{ videoError }}</p>
12 </div>
13 </div>
14 </div>
15 </div>
16 <!-- 底部日志区域 -->
17 <div v-if="showProcessLog" class="result-section">
18     <h3>处理结果</h3>
19     <div class="log-header">
20         <span>{{ isProcessed ? '处理完成' : (hasError ? '处理出错' : '处理状态') }}</span>
21     </div>
22     <!-- 处理日志 -->
23     <div v-if="logContent" class="log-content-wrapper">
24         <pre class="log-content">{{ logContent }}</pre>
25     </div>
26 </div>
27 <!-- 错误信息显示 -->
28 <div v-if="logContent && logContent.includes('错误')" class="error-container">
29     <div class="error-message">
30         <h4>处理过程中遇到问题: </h4>
31         <pre class="error-details">{{ logContent }}</pre>
32     </div>
33 </div>
34 </div>
35 </template>
36 <script setup lang="ts">
37 import { ref, computed, onMounted } from 'vue';
38 import axios from 'axios';
39 const videoFile = ref<File | null>(null);
40 const textFile = ref<File | null>(null);
41 const uploading = ref(false);
42 const processing = ref(false);
43 const uploadProgress = ref(0);
44 const message = ref("");
45 const messageType = ref<'success' | 'error'>('success');
46 const groupId = ref<number | null>(null);
47 const isProcessed = ref(false);
48 const showProcessLog = ref(false);
49 const hasError = ref(false);
50 const logContent = ref("");
```

```
1  const detailedError = ref("");
2  const showDetailedError = ref(false);
3  const processedVideoUrl = ref<string | null>(null);
4  const videoPlayer = ref<HTMLVideoElement | null>(null);
5  const showResult = ref(false);
6  const isVideoLoading = ref(true);
7  const videoError = ref<string | null>(null);
8  const canUpload = computed(() => {
9      return videoFile.value && textFile.value && !uploading.value;
10 });
11 const handleVideoSelect = (e: Event) => {
12     const input = e.target as HTMLInputElement;
13     if (input.files && input.files[0]) {
14         const file = input.files[0];
15         if (file.type === 'video/mp4') {
16             videoFile.value = file;
17             resetVideoState();
18             showProcessLog.value = false;
19             isProcessed.value = false;
20         } else {
21             showMessage('请选择 MP4 格式的视频文件', 'error');
22         }
23     }
24 };
25 const handleTextSelect = (e: Event) => {
26     const input = e.target as HTMLInputElement;
27     if (input.files && input.files[0]) {
28         const file = input.files[0];
29         if (file.type === 'text/plain') {
30             textFile.value = file;
31         } else {
32             showMessage('请选择 TXT 格式的文本文件', 'error');
33         }
34     }
35 };
36 const showMessage = (msg: string, type: 'success' | 'error') => {
37     message.value = msg;
38     messageType.value = type;
39     setTimeout(() => {
40         message.value = "";
41     }, 3000);
42 };
43 const updateLogContent = (content: any) => {
44     if (typeof content === 'string') {
45         logContent.value = content;
46     } else if (content && typeof content === 'object') {
47         try {
48             if (content.message) {
49                 logContent.value = content.message;
50             } else if (content.status && content.result) {
```

```
1      logContent.value = `状态: ${content.status}\n 结果: ${JSON.stringify(content.result, null, 2)}`;
2    } else {
3      logContent.value = JSON.stringify(content, null, 2);
4    }
5  } catch (e) {
6    logContent.value = '无法解析日志内容';
7  }
8  } else {
9    logContent.value = '无日志内容';
10 }
11 };
12 const getVideoFileName = () => {
13   if (videoFile.value) {
14     const originalName = videoFile.value.name;
15     const baseName = originalName.replace(/\.mp4$/, '');
16     return `${baseName}_processed.mp4`;
17   }
18   return 'processed_video.mp4';
19 };
20 const handleVideoLoaded = () => {
21   isVideoLoading.value = false;
22   videoError.value = null;
23 };
24 const handleVideoError = async (e: Event) => {
25   const target = e.target as HTMLVideoElement;
26   isVideoLoading.value = false;
27   let errorMessage = '视频加载失败: ';
28   if (target.error) {
29     switch (target.error.code) {
30       case MediaError.MEDIA_ERR_ABORTED:
31         errorMessage += '加载被中断';
32         break;
33       case MediaError.MEDIA_ERR_NETWORK:
34         errorMessage += '网络错误';
35         break;
36       case MediaError.MEDIA_ERR_DECODE:
37       case MediaError.MEDIA_ERR_SRC_NOT_SUPPORTED:
38         errorMessage += '视频格式不支持';
39         await retryLoadVideo();
40         break;
41       default:
42         errorMessage += `未知错误 (${target.error.code})`;
43     }
44   }
45   videoError.value = errorMessage;
46   console.error('视频加载错误:', {
47     url: processedVideoUrl.value,
48     error: target.error,
49     errorMessage
50   });
```



```
1  };
2  const reloadVideo = () => {
3    if (videoPlayer.value) {
4      isVideoLoading.value = true;
5      videoError.value = null;
6      videoPlayer.value.load();
7    }
8  };
9  const handleDownload = async () => {
10   if (!processedVideoUrl.value) {
11     showMessage('视频文件不可用', 'error');
12     return;
13   }
14   try {
15     const timestamp = new Date().getTime();
16     const downloadUrl = `${processedVideoUrl.value}?t=${timestamp}`;
17     const checkResponse = await fetch(downloadUrl, {
18       method: 'HEAD'
19     });
20     if (!checkResponse.ok) {
21       throw new Error(`文件访问失败: ${checkResponse.status}`);
22     }
23     const response = await fetch(downloadUrl);
24     const blob = await response.blob();
25     const contentType = response.headers.get('content-type');
26     if (!contentType || !contentType.includes('video/mp4')) {
27       console.warn('警告: 响应的内容类型不是 video/mp4:', contentType);
28     }
29     const url = window.URL.createObjectURL(blob);
30     const link = document.createElement('a');
31     link.href = url;
32     link.download = getVideoFileName();
33     document.body.appendChild(link);
34     link.click();
35     setTimeout(() => {
36       document.body.removeChild(link);
37       window.URL.revokeObjectURL(url);
38     }, 100);
39     showMessage('下载开始', 'success');
40   } catch (error: any) {
41     console.error('下载失败:', error);
42     showMessage(`下载失败: ${error.message}`, 'error');
43     logContent.value += `下载错误: ${error.message}\n`;
44   }
45 };
46 const checkVideoAccessibility = async (url: string) => {
47   try {
48     const response = await fetch(url, {
49       method: 'HEAD',
50       mode: 'cors'
```

```
1    });
2    if (!response.ok) {
3      throw new Error(`HTTP error! status: ${response.status}`);
4    }
5    return true;
6  } catch (e) {
7    console.error('视频访问检查失败:', e);
8    return false;
9  }
10 };
11 const checkVideoFormat = async (url: string): Promise<boolean> => {
12   try {
13     const response = await fetch(url, {
14       method: 'HEAD'
15     });
16     if (!response.ok) {
17       throw new Error(`HTTP error! status: ${response.status}`);
18     }
19     const contentType = response.headers.get('content-type');
20     return contentType !== null && contentType.includes('video/mp4');
21   } catch (e) {
22     console.error('视频格式检查失败:', e);
23     return false;
24   }
25 };
26 const retryLoadVideo = async () => {
27   if (!processedVideoUrl.value) return;
28   try {
29     isVideoLoading.value = true;
30     videoError.value = null;
31     const timestamp = new Date().getTime();
32     const newUrl = `${processedVideoUrl.value}?t=${timestamp}`;
33     const isValidFormat = await checkVideoFormat(newUrl);
34     if (!isValidFormat) {
35       throw new Error('视频格式不正确');
36     }
37     processedVideoUrl.value = newUrl;
38     if (videoPlayer.value) {
39       videoPlayer.value.load();
40     }
41   } catch (error: any) {
42     console.error('重试加载失败:', error);
43     videoError.value = `无法加载视频: ${error.message}`;
44     isVideoLoading.value = false;
45   }
46 };
47 const handleUploadAndProcess = async () => {
48   if (!videoFile.value || !textField.value) return;
49   uploading.value = true;
50   uploadProgress.value = 0;
```

```
1    showResult.value = true;
2    showProcessLog.value = true;
3    logContent.value = '开始上传文件...\n';
4    try {
5        const formData = new FormData();
6        formData.append('video', videoFile.value);
7        formData.append('text', textFile.value);
8        const uploadResponse = await axios.post(
9            'http://localhost:8000/api/upload/',
10           formData,
11           {
12               headers: {
13                   'Content-Type': 'multipart/form-data',
14               },
15               onUploadProgress: (progressEvent) => {
16                   if (progressEvent.total) {
17                       uploadProgress.value = Math.round(
18                           (progressEvent.loaded * 100) / progressEvent.total
19                       );
20                   }
21               }
22           }
23       );
24       if (uploadResponse.data.status !== 'success') {
25           throw new Error(uploadResponse.data.error || '文件上传失败');
26       }
27       const groupId = uploadResponse.data.group_id;
28       logContent.value += '文件上传成功, 开始处理...\n';
29       uploading.value = false;
30       processing.value = true;
31       const processResponse = await axios.post(
32           `http://localhost:8000/api/process/${groupId}/`
33       );
34       if (processResponse.data.status === 'success') {
35           processing.value = false;
36           isProcessed.value = true;
37           showMessage('处理完成! ', 'success');
38           logContent.value += '处理完成! \n';
39           const videoUrl = processResponse.data.video_url;
40           const timestamp = new Date().getTime();
41           const fullVideoUrl = `${videoUrl}?t=${timestamp}`;
42           const isValidFormat = await checkVideoFormat(fullVideoUrl);
43           if (isValidFormat) {
44               processedVideoUrl.value = fullVideoUrl;
45               logContent.value += `视频可访问: ${fullVideoUrl}\n`;
46               showResult.value = true;
47               showProcessLog.value = true;
48               isVideoLoading.value = true;
49               videoError.value = null;
50           } else {
```

```
1      throw new Error('处理后的视频格式不正确或无法访问');
2    }
3  } else {
4    throw new Error(processResponse.data.message || '视频处理失败');
5  }
6  } catch (error: any) {
7    const errorMessage = error.response?.data?.error ||
8      error.response?.data?.message ||
9      error.message ||
10     '操作失败';
11    showMessage(errorMessage, 'error');
12    logContent.value += `错误: ${errorMessage}\n`;
13    hasError.value = true;
14    console.error('操作错误:', error);
15  } finally {
16    uploading.value = false;
17    processing.value = false;
18  }
19 };
20 const resetVideoState = () => {
21   isVideoLoading.value = true;
22   videoError.value = null;
23   processedVideoUrl.value = null;
24 };
25 const checkBackendStatus = async () => {
26   try {
27     await axios.get('http://localhost:8000/api/health/');
28     return true;
29   } catch (error) {
30     console.error('后端服务检查失败:', error);
31     return false;
32   }
33 };
34 onMounted(async () => {
35   const isBackendAvailable = await checkBackendStatus();
36   if (!isBackendAvailable) {
37     showMessage('无法连接到后端服务, 请确保服务已启动', 'error');
38   }
39 });
40 const handleProcessError = (error: any) => {
41   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。';
42   if (error.response?.data?.error) {
43     detailedError.value = error.response.data.error;
44   } else if (error.response?.data?.result?.script_stderr) {
45     detailedError.value = error.response.data.result.script_stderr;
46   } else if (error.message) {
47     detailedError.value = error.message;
48   }
49 };
50 </script>
```

```

1  <style scoped>
2  .file-uploader {
3      max-width: 1200px;
4      margin: 3rem auto;
5      padding: 2.5rem;
6      border: none;
7      border-radius: 16px;
8      background: #ffffff;
9      box-shadow: 0 8px 30px rgba(0, 0, 0, 0.08);
10     transition: all 0.3s ease;
11 }
12 .main-content {
13     display: grid;
14     grid-template-columns: 1fr 1fr;
15     gap: 2rem;
16     margin-bottom: 2rem;
17     min-height: 500px;
18 }
19 .upload-container {
20     display: flex;
21     flex-direction: column;
22     gap: 1.5rem;
23 }
24 .upload-section {
25     display: flex;
26     flex-direction: column;
27     gap: 1.5rem;
28 }
29 .file-input-group {
30     background: #f8fafc;
31     padding: 1.5rem;
32     border-radius: 12px;
33     border: 1px solid rgba(0, 0, 0, 0.05);
34     transition: all 0.3s ease;
35 }
36 .file-input-group:hover {
37     transform: translateY(-2px);
38     box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);
39 }
40 .file-input-group h3 {
41     margin: 0 0 1rem 0;
42     font-size: 1.1rem;
43     color: #1a1a1a;
44     font-weight: 600;
45 }
46 .selected-file {
47     margin-top: 0.8rem;
48     font-size: 0.95rem;
49     color: #4a5568;
50     padding: 0.5rem;

```

```

1     background: rgba(66, 185, 131, 0.1);
2     border-radius: 6px;
3 }
4 .action-button {
5     width: 100%;
6     padding: 1rem;
7     background: linear-gradient(135deg, #42b983 0%, #3aa876 100%);
8     color: white;
9     border: none;
10    border-radius: 8px;
11    cursor: pointer;
12    font-size: 1.1rem;
13    margin-bottom: 1.2rem;
14    transition: all 0.3s ease;
15    font-weight: 500;
16    letter-spacing: 0.5px;
17    box-shadow: 0 4px 12px rgba(66, 185, 131, 0.2);
18 }
19 .action-button:hover:not(:disabled) {
20     transform: translateY(-2px);
21     box-shadow: 0 6px 16px rgba(66, 185, 131, 0.3);
22 }
23 .action-button:disabled {
24     background: linear-gradient(135deg, #a8d5c2 0%, #9ecbb8 100%);
25     cursor: not-allowed;
26     box-shadow: none;
27 }
28 .process-button {
29     background: linear-gradient(135deg, #4a90e2 0%, #357abd 100%);
30     box-shadow: 0 4px 12px rgba(74, 144, 226, 0.2);
31 }
32 .process-button:hover:not(:disabled) {
33     box-shadow: 0 6px 16px rgba(74, 144, 226, 0.3);
34 }
35 .process-button:disabled {
36     background: linear-gradient(135deg, #a8c4e2 0%, #9ab8d9 100%);
37 }
38 .progress-container {
39     background: #f8fafc;
40     padding: 1.2rem;
41     border-radius: 10px;
42     box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.05);
43 }
44 .progress-label {
45     margin-bottom: 0.8rem;
46     color: #4a5568;
47     font-weight: 500;
48 }
49 .progress-bar {
50     height: 10px;

```

```

1    background: #e2e8f0;
2    border-radius: 6px;
3    overflow: hidden;
4    box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.05);
5  }
6  .progress-fill {
7    height: 100%;
8    background: linear-gradient(90deg, #42b983 0%, #3aa876 100%);
9    transition: width 0.4s ease;
10   box-shadow: 0 2px 4px rgba(66, 185, 131, 0.2);
11  }
12  .progress-text {
13    text-align: right;
14    font-size: 0.95rem;
15    color: #4a5568;
16    margin-top: 0.5rem;
17    font-weight: 500;
18  }
19  .processing-spinner {
20    width: 30px;
21    height: 30px;
22    border: 3px solid rgba(66, 185, 131, 0.1);
23    border-top: 3px solid #42b983;
24    border-radius: 50%;
25    animation: spin 1s linear infinite;
26    margin: 1rem auto;
27  }
28  .status-message {
29    padding: 1.2rem;
30    margin: 1.2rem 0;
31    border-radius: 10px;
32    text-align: center;
33    font-weight: 500;
34    animation: fadeIn 0.3s ease;
35  }
36  .success {
37    background: linear-gradient(135deg, #d4edda 0%, #c3e6cb 100%);
38    color: #155724;
39    box-shadow: 0 2px 8px rgba(21, 87, 36, 0.1);
40  }
41  .error {
42    background: linear-gradient(135deg, #f8d7da 0%, #f5c6cb 100%);
43    color: #721c24;
44    box-shadow: 0 2px 8px rgba(114, 28, 36, 0.1);
45  }
46  .result-section {
47    margin-top: 2rem;
48    padding: 1.5rem;
49    background: #f8fafc;
50    border-radius: 12px;

```

```
1     box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);
2     display: block;
3 }
4 .log-header {
5     margin-bottom: 1rem;
6     font-weight: 500;
7 }
8 .log-content-wrapper {
9     margin-bottom: 1.2rem;
10    height: 100%;
11 }
12 .log-content {
13     max-height: 300px;
14     overflow-y: auto;
15     background: #ffffff;
16     padding: 1.2rem;
17     border-radius: 8px;
18     font-family: 'Monaco', 'Menlo', monospace;
19     line-height: 1.5;
20     border: 1px solid #e2e8f0;
21 }
22 .video-section {
23     background: #ffffff;
24     border-radius: 12px;
25     box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);
26     height: fit-content;
27     position: sticky;
28     top: 2rem;
29 }
30 .video-player-container {
31     background: #ffffff;
32     border-radius: 12px;
33     overflow: hidden;
34 }
35 .video-wrapper {
36     aspect-ratio: 16/9;
37     background: #000000;
38     position: relative;
39 }
40 .video-player {
41     width: 100%;
42     max-width: 100%;
43     display: block;
44     border-radius: 8px;
45 }
46 .video-loading {
47     position: absolute;
48     top: 0;
49     left: 0;
50     right: 0;
```



```

1    bottom: 0;
2    display: flex;
3    flex-direction: column;
4    align-items: center;
5    justify-content: center;
6    background: rgba(0, 0, 0, 0.8);
7    color: white;
8    backdrop-filter: blur(4px);
9  }
10  .loading-spinner {
11    width: 48px;
12    height: 48px;
13    border: 4px solid rgba(255, 255, 255, 0.2);
14    border-top: 4px solid #ffffff;
15    border-radius: 50%;
16    animation: spin 1s linear infinite;
17    margin-bottom: 1.2rem;
18  }
19  .video-controls {
20    padding: 1rem;
21    border-top: 1px solid #e2e8f0;
22    background: #f8f9fa;
23  }
24  .control-button {
25    display: inline-flex;
26    align-items: center;
27    gap: 0.8rem;
28    padding: 0.8rem 1.5rem;
29    border: none;
30    border-radius: 8px;
31    cursor: pointer;
32    font-size: 1rem;
33    transition: all 0.3s ease;
34    font-weight: 500;
35    letter-spacing: 0.5px;
36  }
37  .download-button {
38    background: linear-gradient(135deg, #42b983 0%, #3aa876 100%);
39    color: white;
40    text-decoration: none;
41    box-shadow: 0 4px 12px rgba(66, 185, 131, 0.2);
42  }
43  .download-button:hover {
44    transform: translateY(-2px);
45    box-shadow: 0 6px 16px rgba(66, 185, 131, 0.3);
46  }
47  .reload-button {
48    background: linear-gradient(135deg, #4a90e2 0%, #357abd 100%);
49    color: white;
50    box-shadow: 0 4px 12px rgba(74, 144, 226, 0.2);

```

```

1  }
2  .reload-button:hover {
3      transform: translateY(-2px);
4      box-shadow: 0 6px 16px rgba(74, 144, 226, 0.3);
5  }
6  .button-icon {
7      font-size: 1.3rem;
8  }
9  .video-error {
10     margin-top: 1.2rem;
11     padding: 1rem;
12     background: linear-gradient(135deg, #f8d7da 0%, #f5c6cb 100%);
13     border: 1px solid #f5c6cb;
14     border-radius: 8px;
15     color: #721c24;
16     text-align: center;
17     font-weight: 500;
18     box-shadow: 0 2px 8px rgba(114, 28, 36, 0.1);
19 }
20 .error-container {
21     margin: 1.2rem 0;
22     padding: 1.5rem;
23     background: linear-gradient(135deg, #fff3f3 0%, #ffe8e8 100%);
24     border: 1px solid #ffcdd2;
25     border-radius: 10px;
26     box-shadow: 0 4px 12px rgba(211, 47, 47, 0.1);
27 }
28 .error-message h4 {
29     color: #d32f2f;
30     margin: 0 0 1rem 0;
31     font-weight: 600;
32 }
33 .error-details {
34     background: #ffffff;
35     padding: 1.2rem;
36     border-radius: 8px;
37     font-size: 0.95rem;
38     margin: 0;
39     white-space: pre-wrap;
40     word-break: break-word;
41     box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.05);
42     font-family: 'Monaco', 'Menlo', monospace;
43     line-height: 1.5;
44 }
45 @keyframes spin {
46     0% { transform: rotate(0deg); }
47     100% { transform: rotate(360deg); }
48 }
49 @keyframes fadeIn {
50     from { opacity: 0; transform: translateY(-10px); }

```

```

1      to { opacity: 1; transform: translateY(0); }
2  }
3  .log-content::-webkit-scrollbar,
4  .error-details::-webkit-scrollbar {
5      width: 8px;
6  }
7  .log-content::-webkit-scrollbar-track,
8  .error-details::-webkit-scrollbar-track {
9      background: #f1f1f1;
10     border-radius: 4px;
11 }
12 .log-content::-webkit-scrollbar-thumb,
13 .error-details::-webkit-scrollbar-thumb {
14     background: #c1c1c1;
15     border-radius: 4px;
16 }
17 .log-content::-webkit-scrollbar-thumb:hover,
18 .error-details::-webkit-scrollbar-thumb:hover {
19     background: #a8a8a8;
20 }
21 input[type="file"] {
22     width: 100%;
23     padding: 0.8rem;
24     border: 2px dashed #e2e8f0;
25     border-radius: 8px;
26     cursor: pointer;
27     transition: all 0.3s ease;
28 }
29 input[type="file"]:hover {
30     border-color: #42b983;
31     background: rgba(66, 185, 131, 0.05);
32 }
33 input[type="file"]:disabled {
34     border-color: #e2e8f0;
35     background: #f8f9fa;
36     cursor: not-allowed;
37 }
38 @media (max-width: 1024px) {
39     .main-content {
40         grid-template-columns: 1fr;
41     }
42     .video-section {
43         position: relative;
44         top: 0;
45     }
46 }
47 </style>
48 <template>
49     <div>
50         <FileUpload />

```

```

1     </div>
2 </template>
3 <script setup>
4 import FileUpload from '@components/FileUpload.vue'
5 </script>
6 import argparse
7 import os
8 import os.path as osp
9 import numpy as np
10 import cv2
11 import torch
12 import gc
13 import sys
14 from pathlib import Path
15 os.environ["HYDRA_FULL_ERROR"] = "1"
16 current_dir = os.path.dirname(os.path.abspath(__file__))
17 parent_dir = os.path.dirname(current_dir)
18 if parent_dir not in sys.path:
19     sys.path.insert(0, parent_dir)
20 sam2_dir = os.path.join(parent_dir, "sam2")
21 if sam2_dir not in sys.path:
22     sys.path.insert(0, sam2_dir)
23 sam2_inner_dir = os.path.join(sam2_dir, "sam2")
24 if sam2_inner_dir not in sys.path:
25     sys.path.insert(0, sam2_inner_dir)
26 os.environ["HYDRA_CONFIG_PATH"] = os.path.join(parent_dir, "sam2", "sam2", "configs")
27 from build_sam import build_sam2_video_predictor
28 color = [(255, 0, 0)]
29 def load_txt(gt_path):
30     with open(gt_path, 'r') as f:
31         gt = f.readlines()
32         prompts = {}
33         for fid, line in enumerate(gt):
34             x, y, w, h = map(float, line.split(','))
35             x, y, w, h = int(x), int(y), int(w), int(h)
36             prompts[fid] = ((x, y, x + w, y + h), 0)
37         return prompts
38 def determine_model_cfg(model_path):
39     config_base = os.path.join(parent_dir, "sam2", "sam2", "configs", "samurai")
40     if "large" in model_path or "_l" in model_path:
41         return os.path.join(config_base, "sam2.1_hiera_l.yaml")
42     elif "base_plus" in model_path or "_b+" in model_path:
43         return os.path.join(config_base, "sam2.1_hiera_b+.yaml")
44     elif "base" in model_path or "_b" in model_path:
45         return os.path.join(config_base, "sam2.1_hiera_b+.yaml")
46     elif "small" in model_path or "_s" in model_path:
47         return os.path.join(config_base, "sam2.1_hiera_s.yaml")
48     elif "tiny" in model_path or "_t" in model_path:
49         return os.path.join(config_base, "sam2.1_hiera_t.yaml")
50     else:

```

```

1         return os.path.join(config_base, "sam2.1_hiera_b+.yaml")
2     def prepare_frames_or_path(video_path):
3         if video_path.endswith(".mp4") or osp.isdir(video_path):
4             return video_path
5         else:
6             raise ValueError("Invalid video_path format. Should be .mp4 or a directory of jpg frames.")
7     def main(args):
8         output_dir = os.path.join(os.path.dirname(os.path.dirname(os.path.dirname(__file__))),
9                                   'samurai_django', 'myproject', 'media', 'processed_videos')
10        os.makedirs(output_dir, exist_ok=True)
11        input_filename = os.path.basename(args.video_path)
12        output_filename = os.path.splitext(input_filename)[0] + "_processed" +
13        os.path.splitext(input_filename)[1]
14        args.output_path = os.path.join(output_dir, output_filename)
15        args.video_output_path = args.output_path
16        if not os.path.isabs(args.model_path):
17            args.model_path = os.path.join(os.path.dirname(os.path.dirname(__file__)),
18                                           "sam2", "checkpoints",
19                                           os.path.basename(args.model_path))
20        if not os.path.exists(args.model_path):
21            raise FileNotFoundError(f"模型文件不存在: {args.model_path}")
22        model_cfg = determine_model_cfg(args.model_path)
23        predictor = build_sam2_video_predictor(model_cfg, args.model_path, device="cuda:0")
24        frames_or_path = prepare_frames_or_path(args.video_path)
25        prompts = load_txt(args.txt_path)
26        frame_rate = 30
27        if args.save_to_video:
28            if osp.isdir(args.video_path):
29                frames = sorted([osp.join(args.video_path, f) for f in os.listdir(args.video_path) if
30f.endswith(("jpg", ".jpeg", ".JPG", ".JPEG"))])
31                loaded_frames = [cv2.imread(frame_path) for frame_path in frames]
32                height, width = loaded_frames[0].shape[:2]
33            else:
34                cap = cv2.VideoCapture(args.video_path)
35                frame_rate = cap.get(cv2.CAP_PROP_FPS)
36                loaded_frames = []
37                while True:
38                    ret, frame = cap.read()
39                    if not ret:
40                        break
41                    loaded_frames.append(frame)
42                cap.release()
43                height, width = loaded_frames[0].shape[:2]
44                if len(loaded_frames) == 0:
45                    raise ValueError("No frames were loaded from the video.")
46                fourcc = cv2.VideoWriter_fourcc(*'avc1')
47                out = cv2.VideoWriter(args.video_output_path, fourcc, frame_rate, (width, height))
48                with torch.inference_mode(), torch.autocast("cuda", dtype=torch.float16):
49                    state = predictor.init_state(frames_or_path, offload_video_to_cpu=True)
50                    bbox, track_label = prompts[0]

```

```

1      _, _, masks = predictor.add_new_points_or_box(state, box=bbox, frame_idx=0, obj_id=0)
2      for frame_idx, object_ids, masks in predictor.propagate_in_video(state):
3          mask_to_vis = {}
4          bbox_to_vis = {}
5          for obj_id, mask in zip(object_ids, masks):
6              mask = mask[0].cpu().numpy()
7              mask = mask > 0.0
8              non_zero_indices = np.argwhere(mask)
9              if len(non_zero_indices) == 0:
10                 bbox = [0, 0, 0, 0]
11             else:
12                 y_min, x_min = non_zero_indices.min(axis=0).tolist()
13                 y_max, x_max = non_zero_indices.max(axis=0).tolist()
14                 bbox = [x_min, y_min, x_max - x_min, y_max - y_min]
15             bbox_to_vis[obj_id] = bbox
16             mask_to_vis[obj_id] = mask
17         if args.save_to_video:
18             img = loaded_frames[frame_idx]
19             for obj_id, mask in mask_to_vis.items():
20                 mask_img = np.zeros((height, width, 3), np.uint8)
21                 mask_img[mask] = color[(obj_id + 1) % len(color)]
22                 img = cv2.addWeighted(img, 1, mask_img, 0.2, 0)
23             for obj_id, bbox in bbox_to_vis.items():
24                 cv2.rectangle(img, (bbox[0], bbox[1]), (bbox[0] + bbox[2], bbox[1] + bbox[3]),
25 color[obj_id % len(color)], 2)
26             out.write(img)
27         if args.save_to_video:
28             out.release()
29     del predictor, state
30     gc.collect()
31     torch.clear_autocast_cache()
32     torch.cuda.empty_cache()
33     try:
34         if os.path.exists(args.output_path):
35             print(f"处理完成，视频已保存到: {args.output_path}")
36             relative_path = os.path.relpath(args.output_path,
37 os.path.join(os.path.dirname(os.path.dirname(os.path.dirname(__file__))),
38 'samurai_django', 'media'))
39             return True, relative_path
40         else:
41             print(f"处理失败：找不到输出文件 {args.output_path}")
42             return False, None
43     except Exception as e:
44         print(f"处理过程中出错: {str(e)}")
45         return False, None
46 if __name__ == "__main__":
47     parser = argparse.ArgumentParser(description='视频处理脚本')
48     parser.add_argument('--video_path', type=str, required=True, help='输入视频路径')
49     parser.add_argument('--txt_path', type=str, required=True, help='输入文本路径')

```

```

1      default_model_path = os.path.join(os.path.dirname(os.path.dirname(__file__)),
2                                          "sam2", "checkpoints", "sam2.1_hiera_base_plus.pt")
3      parser.add_argument('--model_path', type=str, default=default_model_path, help='模型路径')
4      parser.add_argument("--save_to_video", default=True, help="Save results to a video.")
5      args = parser.parse_args()
6      success, output_path = main(args)
7      exit(0 if success else 1)
8  :root {
9      --vt-c-white: #ffffff;
10     --vt-c-white-soft: #f8f8f8;
11     --vt-c-white-mute: #f2f2f2;
12     --vt-c-black: #181818;
13     --vt-c-black-soft: #222222;
14     --vt-c-black-mute: #282828;
15     --vt-c-indigo: #2c3e50;
16     --vt-c-divider-light-1: rgba(60, 60, 60, 0.29);
17     --vt-c-divider-light-2: rgba(60, 60, 60, 0.12);
18     --vt-c-divider-dark-1: rgba(84, 84, 84, 0.65);
19     --vt-c-divider-dark-2: rgba(84, 84, 84, 0.48);
20     --vt-c-text-light-1: var(--vt-c-indigo);
21     --vt-c-text-light-2: rgba(60, 60, 60, 0.66);
22     --vt-c-text-dark-1: var(--vt-c-white);
23     --vt-c-text-dark-2: rgba(235, 235, 235, 0.64);
24 }
25 :root {
26     --color-background: var(--vt-c-white);
27     --color-background-soft: var(--vt-c-white-soft);
28     --color-background-mute: var(--vt-c-white-mute);
29     --color-border: var(--vt-c-divider-light-2);
30     --color-border-hover: var(--vt-c-divider-light-1);
31     --color-heading: var(--vt-c-text-light-1);
32     --color-text: var(--vt-c-text-light-1);
33     --section-gap: 160px;
34 }
35 @media (prefers-color-scheme: dark) {
36     :root {
37         --color-background: var(--vt-c-black);
38         --color-background-soft: var(--vt-c-black-soft);
39         --color-background-mute: var(--vt-c-black-mute);
40         --color-border: var(--vt-c-divider-dark-2);
41         --color-border-hover: var(--vt-c-divider-dark-1);
42         --color-heading: var(--vt-c-text-dark-1);
43         --color-text: var(--vt-c-text-dark-2);
44     }
45 }
46 *,
47 *::before,
48 *::after {
49     box-sizing: border-box;
50     margin: 0;

```

```
1     font-weight: normal;
2 }
3 body {
4     min-height: 100vh;
5     color: var(--color-text);
6     background: var(--color-background);
7     transition:
8         color 0.5s,
9         background-color 0.5s;
10    line-height: 1.6;
11    font-family:
12        Inter,
13        -apple-system,
14        BlinkMacSystemFont,
15        'Segoe UI',
16        Roboto,
17        Oxygen,
18        Ubuntu,
19        Cantarell,
20        'Fira Sans',
21        'Droid Sans',
22        'Helvetica Neue',
23        sans-serif;
24    font-size: 15px;
25    text-rendering: optimizeLegibility;
26    -webkit-font-smoothing: antialiased;
27    -moz-osx-font-smoothing: grayscale;
28 }
29 import torch
30 from torchvision.ops.bboxes import box_area
31 import numpy as np
32 def box_cxcywh_to_xyxy(x):
33     x_c, y_c, w, h = x.unbind(-1)
34     b = [(x_c - 0.5 * w), (y_c - 0.5 * h),
35          (x_c + 0.5 * w), (y_c + 0.5 * h)]
36     return torch.stack(b, dim=-1)
37 def box_xywh_to_xyxy(x):
38     x1, y1, w, h = x.unbind(-1)
39     b = [x1, y1, x1 + w, y1 + h]
40     return torch.stack(b, dim=-1)
41 def box_xyxy_to_xywh(x):
42     x1, y1, x2, y2 = x.unbind(-1)
43     b = [x1, y1, x2 - x1, y2 - y1]
44     return torch.stack(b, dim=-1)
45 def box_xyxy_to_cxcywh(x):
46     x0, y0, x1, y1 = x.unbind(-1)
47     b = [(x0 + x1) / 2, (y0 + y1) / 2,
48          (x1 - x0), (y1 - y0)]
49     return torch.stack(b, dim=-1)
50 def box_iou(boxes1, boxes2):
```



```

1      area1 = box_area(boxes1)
2      area2 = box_area(boxes2)
3      lt = torch.max(boxes1[:, :2], boxes2[:, :2])
4      rb = torch.min(boxes1[:, 2:], boxes2[:, 2:])
5      wh = (rb - lt).clamp(min=0)
6      inter = wh[:, 0] * wh[:, 1]
7      union = area1 + area2 - inter
8      iou = inter / union
9      return iou, union
10     def generalized_box_iou(boxes1, boxes2):
11         iou, union = box_iou(boxes1, boxes2)
12         lt = torch.min(boxes1[:, :2], boxes2[:, :2])
13         rb = torch.max(boxes1[:, 2:], boxes2[:, 2:])
14         wh = (rb - lt).clamp(min=0)
15         area = wh[:, 0] * wh[:, 1]
16         return iou - (area - union) / area, iou
17     def giou_loss(boxes1, boxes2):
18         giou, iou = generalized_box_iou(boxes1, boxes2)
19         return (1 - giou).mean(), iou
20     def clip_box(box: list, H, W, margin=0):
21         x1, y1, w, h = box
22         x2, y2 = x1 + w, y1 + h
23         x1 = min(max(0, x1), W-margin)
24         x2 = min(max(margin, x2), W)
25         y1 = min(max(0, y1), H-margin)
26         y2 = min(max(margin, y2), H)
27         w = max(margin, x2-x1)
28         h = max(margin, y2-y1)
29         return [x1, y1, w, h]
30     import math
31     import torch
32     import torch.nn.functional as F
33     def generate_bbox_mask(bbox_mask, bbox):
34         b, h, w = bbox_mask.shape
35         for i in range(b):
36             bbox_i = bbox[i].cpu().tolist()
37             bbox_mask[i, int(bbox_i[1]):int(bbox_i[1] + bbox_i[3] - 1), int(bbox_i[0]):int(bbox_i[0] +
38 bbox_i[2] - 1)] = 1
39         return bbox_mask
40     def generate_mask_cond(cfg, bs, device, gt_bbox):
41         template_size = cfg.DATA.TEMPLATE.SIZE
42         stride = cfg.MODEL.BACKBONE.STRIDE
43         template_feat_size = template_size // stride
44         if cfg.MODEL.BACKBONE.CE_TEMPLATE_RANGE == 'ALL':
45             box_mask_z = None
46         elif cfg.MODEL.BACKBONE.CE_TEMPLATE_RANGE == 'CTR_POINT':
47             if template_feat_size == 8:
48                 index = slice(3, 4)
49             elif template_feat_size == 12:
50                 index = slice(5, 6)

```

```

1         elif template_feat_size == 7:
2             index = slice(3, 4)
3         elif template_feat_size == 14:
4             index = slice(6, 7)
5         else:
6             raise NotImplementedError
7         box_mask_z = torch.zeros([bs, template_feat_size, template_feat_size], device=device)
8         box_mask_z[:, index, index] = 1
9         box_mask_z = box_mask_z.flatten(1).to(torch.bool)
10    elif cfg.MODEL.BACKBONE.CE_TEMPLATE_RANGE == 'CTR_REC':
11        if template_feat_size == 8:
12            index = slice(3, 5)
13        elif template_feat_size == 12:
14            index = slice(5, 7)
15        elif template_feat_size == 7:
16            index = slice(3, 4)
17        else:
18            raise NotImplementedError
19        box_mask_z = torch.zeros([bs, template_feat_size, template_feat_size], device=device)
20        box_mask_z[:, index, index] = 1
21        box_mask_z = box_mask_z.flatten(1).to(torch.bool)
22    elif cfg.MODEL.BACKBONE.CE_TEMPLATE_RANGE == 'GT_BOX':
23        box_mask_z = torch.zeros([bs, template_size, template_size], device=device)
24        box_mask_z = generate_bbox_mask(box_mask_z, gt_bbox * template_size).unsqueeze(1).to(
25            torch.float)
26        box_mask_z = F.interpolate(box_mask_z, scale_factor=1. / cfg.MODEL.BACKBONE.STRIDE,
27    mode='bilinear',
28                                align_corners=False)
29        box_mask_z = box_mask_z.flatten(1).to(torch.bool)
30    else:
31        raise NotImplementedError
32    return box_mask_z
33
34    def adjust_keep_rate(epoch, warmup_epochs, total_epochs, ITERS_PER_EPOCH, base_keep_rate=0.5,
35    max_keep_rate=1, iters=-1):
36        if epoch < warmup_epochs:
37            return 1
38        if epoch >= total_epochs:
39            return base_keep_rate
40        if iters == -1:
41            iters = epoch * ITERS_PER_EPOCH
42        total_iters = ITERS_PER_EPOCH * (total_epochs - warmup_epochs)
43        iters = iters - ITERS_PER_EPOCH * warmup_epochs
44        keep_rate = base_keep_rate + (max_keep_rate - base_keep_rate) \
45            * (math.cos(iters / total_iters * math.pi) + 1) * 0.5
46        return keep_rate
47
48    @import './base.css';
49    #app {
50        max-width: 1280px;
51        margin: 0 auto;
52        padding: 2rem;

```

```

1     font-weight: normal;
2 }
3 a,
4 .green {
5     text-decoration: none;
6     color: hsla(160, 100%, 37%, 1);
7     transition: 0.4s;
8     padding: 3px;
9 }
10 @media (hover: hover) {
11     a:hover {
12         background-color: hsla(160, 100%, 37%, 0.2);
13     }
14 }
15 @media (min-width: 1024px) {
16     body {
17         display: flex;
18         place-items: center;
19     }
20     #app {
21         display: grid;
22         grid-template-columns: 1fr 1fr;
23         padding: 0 2rem;
24     }
25 }
26 from abc import ABC
27 import torch
28 import torch.nn as nn
29 import torch.nn.functional as F
30 class FocalLoss(nn.Module, ABC):
31     def __init__(self, alpha=2, beta=4):
32         super(FocalLoss, self).__init__()
33         self.alpha = alpha
34         self.beta = beta
35     def forward(self, prediction, target):
36         positive_index = target.eq(1).float()
37         negative_index = target.lt(1).float()
38         negative_weights = torch.pow(1 - target, self.beta)
39         prediction = torch.clamp(prediction, 1e-12)
40         positive_loss = torch.log(prediction) * torch.pow(1 - prediction, self.alpha) * positive_index
41         negative_loss = torch.log(1 - prediction) * torch.pow(prediction,
42                                                                                                     self.alpha) *
43         negative_weights * negative_index
44         num_positive = positive_index.float().sum()
45         positive_loss = positive_loss.sum()
46         negative_loss = negative_loss.sum()
47         if num_positive == 0:
48             loss = -negative_loss
49         else:
50             loss = -(positive_loss + negative_loss) / num_positive

```

```

1         return loss
2     class LBHinge(nn.Module):
3         def __init__(self, error_metric=nn.MSELoss(), threshold=None, clip=None):
4             super().__init__()
5             self.error_metric = error_metric
6             self.threshold = threshold if threshold is not None else -100
7             self.clip = clip
8         def forward(self, prediction, label, target_bb=None):
9             negative_mask = (label < self.threshold).float()
10            positive_mask = (1.0 - negative_mask)
11            prediction = negative_mask * F.relu(prediction) + positive_mask * prediction
12            loss = self.error_metric(prediction, positive_mask * label)
13            if self.clip is not None:
14                loss = torch.min(loss, torch.tensor([self.clip], device=loss.device))
15            return loss
16
17 import argparse
18 import gc
19 import os
20 import os.path as osp
21 import pdb
22 import cv2
23 import numpy as np
24 import torch
25 from loguru import logger
26 from tqdm import tqdm
27 from sam2.build_sam import build_sam2_video_predictor
28
29 def load_test_video_list(testing_list_path):
30     with open(testing_list_path, 'r') as f:
31         test_videos = [line.strip() for line in f.readlines()]
32     return test_videos
33
34 def load_gt(gt_path):
35     with open(gt_path, 'r') as f:
36         gt = f.readlines()
37     prompts = {}
38     fid = 0
39     for line in gt:
40         x, y, w, h = map(int, line.split(','))
41         prompts[fid] = ((x, y, x+w, y+h), 0)
42         fid += 1
43     return prompts
44
45 def get_ckpt_and_cfg(tracker_name, model_name):
46     assert tracker_name in ["sam2.1", "samurai"], "Invalid tracker name"
47     assert model_name in ["tiny", "small", "base_plus", "large"], "Invalid model name"
48     model_ckpt = f"sam2/checkpoints/sam2.1_hiera_{model_name}.pt"
49     if model_name == "base_plus":
50         model_cfg = f"configs/{tracker_name}/sam2.1_hiera_b+.yaml"
51     else:
52         model_cfg = f"configs/{tracker_name}/sam2.1_hiera_{model_name[0]}.yaml"
53     return model_ckpt, model_cfg
54
55 def split_list(video_list, num_chunks):

```

```

1     chunk_size = len(video_list) // num_chunks
2     return [video_list[i:i+chunk_size] for i in range(0, len(video_list), chunk_size)]
3 def inference_chunk(dataset_path, tracker_name, model_name, chunk_videos, result_folder):
4     exp_name = "test"
5     model_ckpt, model_cfg = get_ckpt_and_cfg(tracker_name, model_name)
6     for vid, video in enumerate(chunk_videos):
7         cat_name = video.split('-')[0]
8         cid_name = video.split('-')[1]
9         video_basename = video.strip()
10        frame_folder = osp.join(dataset_path, cat_name, video.strip(), "img")
11        num_frames = len(os.listdir(osp.join(dataset_path, cat_name, video.strip(), "img")))
12        height, width = cv2.imread(osp.join(frame_folder, "00000001.jpg")).shape[:2]
13        logger.info(f"Running video [{vid+1}/{len(chunk_videos)}]: {video} with {num_frames} frames
14        ({height}x{width})")
15        predictor = build_sam2_video_predictor(model_cfg, model_ckpt, device="cuda:0")
16        predictions = []
17        with torch.inference_mode(), torch.autocast("cuda", dtype=torch.float16):
18            state = predictor.init_state(frame_folder, offload_video_to_cpu=True,
19            offload_state_to_cpu=True)
20            prompts = load_gt(osp.join(dataset_path, cat_name, video.strip(), "groundtruth.txt"))
21            bbox, track_label = prompts[0]
22            frame_idx, object_ids, masks = predictor.add_new_points_or_box(state, box=bbox,
23            frame_idx=0, obj_id=0)
24            for frame_idx, object_ids, masks in predictor.propagate_in_video(state):
25                mask_to_vis = {}
26                bbox_to_vis = {}
27                assert len(masks) == 1 and len(object_ids) == 1, "Only one object is supported
28                right now"
29                for obj_id, mask in zip(object_ids, masks):
30                    mask = mask[0].cpu().numpy()
31                    mask = mask > 0.0
32                    non_zero_indices = np.argwhere(mask)
33                    if len(non_zero_indices) == 0:
34                        bbox = [0, 0, 0, 0]
35                    else:
36                        y_min, x_min = non_zero_indices.min(axis=0).tolist()
37                        y_max, x_max = non_zero_indices.max(axis=0).tolist()
38                        bbox = [x_min, y_min, x_max-x_min, y_max-y_min]
39                        bbox_to_vis[obj_id] = bbox
40                        mask_to_vis[obj_id] = mask
41                predictions.append(bbox_to_vis)
42            os.makedirs(result_folder, exist_ok=True)
43            with open(osp.join(result_folder, f'{video_basename}.txt'), 'w') as f:
44                for pred in predictions:
45                    x, y, w, h = pred[0]
46                    f.write(f"{x},{y},{w},{h}\n")
47            del predictor
48            del state
49            gc.collect()
50            torch.clear_autocast_cache()
    
```

```

1         torch.cuda.empty_cache()
2     def main():
3         parser = argparse.ArgumentParser()
4         parser.add_argument("--dataset_path", type=str, default="data/LaSOT-ext")
5         parser.add_argument("--tracker_name", type=str, default="samurai")
6         parser.add_argument("--model_name", type=str, default="large")
7         parser.add_argument("--chunk_idx", type=int, default=0)
8         parser.add_argument("--num_chunks", type=int, default=1)
9         parser.add_argument("--exp_name", type=str, default="test")
10        parser.add_argument("--root_result_folder", type=str, default="results")
11        args = parser.parse_args()
12        test_videos = load_test_video_list("data/LaSOT-ext/testing_set.txt")
13        chunk_video_list = split_list(test_videos, args.num_chunks)
14        chunk_videos = chunk_video_list[args.chunk_idx]
15        logger.info(f"Chunk ID: {args.chunk_idx}, Number of videos: {len(chunk_videos)} (from
16        {chunk_videos[0]} to {chunk_videos[-1]})")
17        exp_result_folder = osp.join(args.root_result_folder, args.tracker_name,
18        f"{args.exp_name}_{args.model_name}")
19        inference_chunk(args.dataset_path, args.tracker_name, args.model_name, chunk_videos,
20        exp_result_folder)
21    if __name__ == "__main__":
22        main()
23    import functools
24    import torch
25    import copy
26    from collections import OrderedDict
27    class TensorDict(OrderedDict):
28        def concat(self, other):
29            return TensorDict(self, **other)
30        def copy(self):
31            return TensorDict(super(TensorDict, self).copy())
32        def __deepcopy__(self, memodict={}):
33            return TensorDict(copy.deepcopy(list(self), memodict))
34        def __getattr__(self, name):
35            if not hasattr(torch.Tensor, name):
36                raise AttributeError('\TensorDict\ object has not attribute \{\}\'.format(name))
37            def apply_attr(*args, **kwargs):
38                return TensorDict({n: getattr(e, name)(*args, **kwargs) if hasattr(e, name) else e for n, e
39            in self.items()})
40            return apply_attr
41        def attribute(self, attr: str, *args):
42            return TensorDict({n: getattr(e, attr, *args) for n, e in self.items()})
43        def apply(self, fn, *args, **kwargs):
44            return TensorDict({n: fn(e, *args, **kwargs) for n, e in self.items()})
45        @staticmethod
46        def _iterable(a):
47            return isinstance(a, (TensorDict, list))
48    class TensorList(list):
49        def __init__(self, list_of_tensors = None):
50            if list_of_tensors is None:

```

```
1         list_of_tensors = list()
2         super(TensorList, self).__init__(list_of_tensors)
3     def __deepcopy__(self, memodict={}):
4         return TensorList(copy.deepcopy(list(self), memodict))
5     def __getitem__(self, item):
6         if isinstance(item, int):
7             return super(TensorList, self).__getitem__(item)
8         elif isinstance(item, (tuple, list)):
9             return TensorList([super(TensorList, self).__getitem__(i) for i in item])
10        else:
11            return TensorList(super(TensorList, self).__getitem__(item))
12    def __add__(self, other):
13        if TensorList._iterable(other):
14            return TensorList([e1 + e2 for e1, e2 in zip(self, other)])
15        return TensorList([e + other for e in self])
16    def __radd__(self, other):
17        if TensorList._iterable(other):
18            return TensorList([e2 + e1 for e1, e2 in zip(self, other)])
19        return TensorList([other + e for e in self])
20    def __iadd__(self, other):
21        if TensorList._iterable(other):
22            for i, e2 in enumerate(other):
23                self[i] += e2
24        else:
25            for i in range(len(self)):
26                self[i] += other
27        return self
28    def __sub__(self, other):
29        if TensorList._iterable(other):
30            return TensorList([e1 - e2 for e1, e2 in zip(self, other)])
31        return TensorList([e - other for e in self])
32    def __rsub__(self, other):
33        if TensorList._iterable(other):
34            return TensorList([e2 - e1 for e1, e2 in zip(self, other)])
35        return TensorList([other - e for e in self])
36    def __isub__(self, other):
37        if TensorList._iterable(other):
38            for i, e2 in enumerate(other):
39                self[i] -= e2
40        else:
41            for i in range(len(self)):
42                self[i] -= other
43        return self
44    def __mul__(self, other):
45        if TensorList._iterable(other):
46            return TensorList([e1 * e2 for e1, e2 in zip(self, other)])
47        return TensorList([e * other for e in self])
48    def __rmul__(self, other):
49        if TensorList._iterable(other):
50            return TensorList([e2 * e1 for e1, e2 in zip(self, other)])
```

```

1         return TensorList([other * e for e in self])
2     def __imul__(self, other):
3         if TensorList_iterable(other):
4             for i, e2 in enumerate(other):
5                 self[i] *= e2
6         else:
7             for i in range(len(self)):
8                 self[i] *= other
9         return self
10    def __truediv__(self, other):
11        if TensorList_iterable(other):
12            return TensorList([e1 / e2 for e1, e2 in zip(self, other)])
13        return TensorList([e / other for e in self])
14    def __rtruediv__(self, other):
15        {"name": "uav_building1", "path": "data_seq/UAV123/building1", "startFrame": 1,
16 "endFrame": 469, "nz": 6,
17 "ext": "jpg", "anno_path": "anno/UAV123/building1.txt", "object_class": "other"},
18 {"name": "uav_building2", "path": "data_seq/UAV123/building2", "startFrame": 1,
19 "endFrame": 577, "nz": 6,
20 "ext": "jpg", "anno_path": "anno/UAV123/building2.txt", "object_class": "other"},
21 {"name": "uav_building3", "path": "data_seq/UAV123/building3", "startFrame": 1,
22 "endFrame": 829, "nz": 6,
23 "ext": "jpg", "anno_path": "anno/UAV123/building3.txt", "object_class": "other"},
24 {"name": "uav_building4", "path": "data_seq/UAV123/building4", "startFrame": 1,
25 "endFrame": 787, "nz": 6,
26 "ext": "jpg", "anno_path": "anno/UAV123/building4.txt", "object_class": "other"},
27 {"name": "uav_building5", "path": "data_seq/UAV123/building5", "startFrame": 1,
28 "endFrame": 481, "nz": 6,
29 "ext": "jpg", "anno_path": "anno/UAV123/building5.txt", "object_class": "other"},
30 {"name": "uav_car1_1", "path": "data_seq/UAV123/car1", "startFrame": 1, "endFrame":
31 751, "nz": 6,
32 "ext": "jpg", "anno_path": "anno/UAV123/car1_1.txt", "object_class": "car"},
33 {"name": "uav_car1_2", "path": "data_seq/UAV123/car1", "startFrame": 751, "endFrame":
34 1627, "nz": 6,
35 "ext": "jpg", "anno_path": "anno/UAV123/car1_2.txt", "object_class": "car"},
36 {"name": "uav_car1_3", "path": "data_seq/UAV123/car1", "startFrame": 1627,
37 "endFrame": 2629, "nz": 6,
38 "ext": "jpg", "anno_path": "anno/UAV123/car1_3.txt", "object_class": "car"},
39 {"name": "uav_car10", "path": "data_seq/UAV123/car10", "startFrame": 1, "endFrame":
40 1405, "nz": 6,
41 "ext": "jpg", "anno_path": "anno/UAV123/car10.txt", "object_class": "car"},
42 {"name": "uav_car11", "path": "data_seq/UAV123/car11", "startFrame": 1, "endFrame":
43 337, "nz": 6,
44 "ext": "jpg", "anno_path": "anno/UAV123/car11.txt", "object_class": "car"},
45 {"name": "uav_car12", "path": "data_seq/UAV123/car12", "startFrame": 1, "endFrame":
46 499, "nz": 6,
47 "ext": "jpg", "anno_path": "anno/UAV123/car12.txt", "object_class": "car"},
48 {"name": "uav_car13", "path": "data_seq/UAV123/car13", "startFrame": 1, "endFrame":
49 415, "nz": 6,
50 "ext": "jpg", "anno_path": "anno/UAV123/car13.txt", "object_class": "car"},
    
```



```

1      {"name": "uav_car14", "path": "data_seq/UAV123/car14", "startFrame": 1, "endFrame":
2 1327, "nz": 6,
3      "ext": "jpg", "anno_path": "anno/UAV123/car14.txt", "object_class": "car"},
4      {"name": "uav_car15", "path": "data_seq/UAV123/car15", "startFrame": 1, "endFrame":
5 469, "nz": 6,
6      "ext": "jpg", "anno_path": "anno/UAV123/car15.txt", "object_class": "car"},
7      {"name": "uav_car16_1", "path": "data_seq/UAV123/car16", "startFrame": 1, "endFrame":
8 415, "nz": 6,
9      "ext": "jpg", "anno_path": "anno/UAV123/car16_1.txt", "object_class": "car"},
10     {"name": "uav_car16_2", "path": "data_seq/UAV123/car16", "startFrame": 415,
11 "endFrame": 1993, "nz": 6,
12     "ext": "jpg", "anno_path": "anno/UAV123/car16_2.txt", "object_class": "car"},
13     {"name": "uav_car17", "path": "data_seq/UAV123/car17", "startFrame": 1, "endFrame":
14 1057, "nz": 6,
15     "ext": "jpg", "anno_path": "anno/UAV123/car17.txt", "object_class": "car"},
16     {"name": "uav_car18", "path": "data_seq/UAV123/car18", "startFrame": 1, "endFrame":
17 1207, "nz": 6,
18     "ext": "jpg", "anno_path": "anno/UAV123/car18.txt", "object_class": "car"},
19     {"name": "uav_car1_s", "path": "data_seq/UAV123/car1_s", "startFrame": 1, "endFrame":
20 1475, "nz": 6,
21     "ext": "jpg", "anno_path": "anno/UAV123/car1_s.txt", "object_class": "car"},
22     {"name": "uav_car2", "path": "data_seq/UAV123/car2", "startFrame": 1, "endFrame":
23 1321, "nz": 6,
24     "ext": "jpg", "anno_path": "anno/UAV123/car2.txt", "object_class": "car"},
25     {"name": "uav_car2_s", "path": "data_seq/UAV123/car2_s", "startFrame": 1, "endFrame":
26 320, "nz": 6,
27     "ext": "jpg", "anno_path": "anno/UAV123/car2_s.txt", "object_class": "car"},
28     {"name": "uav_car3", "path": "data_seq/UAV123/car3", "startFrame": 1, "endFrame":
29 1717, "nz": 6,
30     "ext": "jpg", "anno_path": "anno/UAV123/car3.txt", "object_class": "car"},
31     {"name": "uav_car3_s", "path": "data_seq/UAV123/car3_s", "startFrame": 1, "endFrame":
32 1300, "nz": 6,
33     "ext": "jpg", "anno_path": "anno/UAV123/car3_s.txt", "object_class": "car"},
34     {"name": "uav_car4", "path": "data_seq/UAV123/car4", "startFrame": 1, "endFrame":
35 1345, "nz": 6,
36     "ext": "jpg", "anno_path": "anno/UAV123/car4.txt", "object_class": "car"},
37     {"name": "uav_car4_s", "path": "data_seq/UAV123/car4_s", "startFrame": 1, "endFrame":
38 830, "nz": 6,
39     "ext": "jpg", "anno_path": "anno/UAV123/car4_s.txt", "object_class": "car"},
40     {"name": "uav_car5", "path": "data_seq/UAV123/car5", "startFrame": 1, "endFrame": 745,
41 "nz": 6,
42     "ext": "jpg", "anno_path": "anno/UAV123/car5.txt", "object_class": "car"},
43     {"name": "uav_car6_1", "path": "data_seq/UAV123/car6", "startFrame": 1, "endFrame":
44 487, "nz": 6,
45     "ext": "jpg", "anno_path": "anno/UAV123/car6_1.txt", "object_class": "car"},
46     {"name": "uav_car6_2", "path": "data_seq/UAV123/car6", "startFrame": 487, "endFrame":
47 1807, "nz": 6,
48     "ext": "jpg", "anno_path": "anno/UAV123/car6_2.txt", "object_class": "car"},
49     {"name": "uav_car6_3", "path": "data_seq/UAV123/car6", "startFrame": 1807,
50 "endFrame": 2953, "nz": 6,

```

```

1         "ext": "jpg", "anno_path": "anno/UAV123/car6_3.txt", "object_class": "car"},
2         {"name": "uav_car6_4", "path": "data_seq/UAV123/car6", "startFrame": 2953,
3 "endFrame": 3925, "nz": 6,
4         "ext": "jpg", "anno_path": "anno/UAV123/car6_4.txt", "object_class": "car"},
5         {"name": "uav_car6_5", "path": "data_seq/UAV123/car6", "startFrame": 3925,
6 "endFrame": 4861, "nz": 6,
7         "ext": "jpg", "anno_path": "anno/UAV123/car6_5.txt", "object_class": "car"},
8         {"name": "uav_car7", "path": "data_seq/UAV123/car7", "startFrame": 1, "endFrame":
9 1033, "nz": 6,
10        "ext": "jpg", "anno_path": "anno/UAV123/car7.txt", "object_class": "car"},
11        {"name": "uav_car8_1", "path": "data_seq/UAV123/car8", "startFrame": 1, "endFrame":
12 1357, "nz": 6,
13        "ext": "jpg", "anno_path": "anno/UAV123/car8_1.txt", "object_class": "car"},
14        {"name": "uav_car8_2", "path": "data_seq/UAV123/car8", "startFrame": 1357,
15 "endFrame": 2575, "nz": 6,
16        "ext": "jpg", "anno_path": "anno/UAV123/car8_2.txt", "object_class": "car"},
17        {"name": "uav_car9", "path": "data_seq/UAV123/car9", "startFrame": 1, "endFrame":
18 1879, "nz": 6,
19        "ext": "jpg", "anno_path": "anno/UAV123/car9.txt", "object_class": "car"},
20        {"name": "uav_group1_1", "path": "data_seq/UAV123/group1", "startFrame": 1,
21 "endFrame": 1333, "nz": 6,
22        "ext": "jpg", "anno_path": "anno/UAV123/group1_1.txt", "object_class": "person"},
23        {"name": "uav_group1_2", "path": "data_seq/UAV123/group1", "startFrame": 1333,
24 "endFrame": 2515, "nz": 6,
25        "ext": "jpg", "anno_path": "anno/UAV123/group1_2.txt", "object_class": "person"},
26        {"name": "uav_group1_3", "path": "data_seq/UAV123/group1", "startFrame": 2515,
27 "endFrame": 3925, "nz": 6,
28        "ext": "jpg", "anno_path": "anno/UAV123/group1_3.txt", "object_class": "person"},
29        {"name": "uav_group1_4", "path": "data_seq/UAV123/group1", "startFrame": 3925,
30 "endFrame": 4873, "nz": 6,
31        "ext": "jpg", "anno_path": "anno/UAV123/group1_4.txt", "object_class": "person"},
32        {"name": "uav_group2_1", "path": "data_seq/UAV123/group2", "startFrame": 1,
33 "endFrame": 907, "nz": 6,
34        "ext": "jpg", "anno_path": "anno/UAV123/group2_1.txt", "object_class": "person"},
35        {"name": "uav_group2_2", "path": "data_seq/UAV123/group2", "startFrame": 907,
36 "endFrame": 1771, "nz": 6,
37        "ext": "jpg", "anno_path": "anno/UAV123/group2_2.txt", "object_class": "person"},
38        {"name": "uav_group2_3", "path": "data_seq/UAV123/group2", "startFrame": 1771,
39 "endFrame": 2683, "nz": 6,
40        "ext": "jpg", "anno_path": "anno/UAV123/group2_3.txt", "object_class": "person"},
41        {"name": "uav_group3_1", "path": "data_seq/UAV123/group3", "startFrame": 1,
42 "endFrame": 1567, "nz": 6,
43        "ext": "jpg", "anno_path": "anno/UAV123/group3_1.txt", "object_class": "person"},
44        {"name": "uav_group3_2", "path": "data_seq/UAV123/group3", "startFrame": 1567,
45 "endFrame": 2827, "nz": 6,
46        "ext": "jpg", "anno_path": "anno/UAV123/group3_2.txt", "object_class": "person"},
47        {"name": "uav_group3_3", "path": "data_seq/UAV123/group3", "startFrame": 2827,
48 "endFrame": 4369, "nz": 6,
49        "ext": "jpg", "anno_path": "anno/UAV123/group3_3.txt", "object_class": "person"},
50        {"name": "uav_group3_4", "path": "data_seq/UAV123/group3", "startFrame": 4369,
    
```

```

1  "endFrame": 5527, "nz": 6,
2      "ext": "jpg", "anno_path": "anno/UAV123/group3_4.txt", "object_class": "person"},
3      {"name": "uav_person1", "path": "data_seq/UAV123/person1", "startFrame": 1,
4  "endFrame": 799, "nz": 6,
5      "ext": "jpg", "anno_path": "anno/UAV123/person1.txt", "object_class": "person"},
6      {"name": "uav_person10", "path": "data_seq/UAV123/person10", "startFrame": 1,
7  "endFrame": 1021, "nz": 6,
8      "ext": "jpg", "anno_path": "anno/UAV123/person10.txt", "object_class": "person"},
9      {"name": "uav_person11", "path": "data_seq/UAV123/person11", "startFrame": 1,
10 "endFrame": 721, "nz": 6,
11     "ext": "jpg", "anno_path": "anno/UAV123/person11.txt", "object_class": "person"},
12     {"name": "uav_person12_1", "path": "data_seq/UAV123/person12", "startFrame": 1,
13 "endFrame": 601, "nz": 6,
14     "ext": "jpg", "anno_path": "anno/UAV123/person12_1.txt", "object_class": "person"},
15     {"name": "uav_person12_2", "path": "data_seq/UAV123/person12", "startFrame": 601,
16 "endFrame": 1621, "nz": 6,
17     "ext": "jpg", "anno_path": "anno/UAV123/person12_2.txt", "object_class": "person"},
18     {"name": "uav_person13", "path": "data_seq/UAV123/person13", "startFrame": 1,
19 "endFrame": 883, "nz": 6,
20     "ext": "jpg", "anno_path": "anno/UAV123/person13.txt", "object_class": "person"},
21     {"name": "uav_person14_1", "path": "data_seq/UAV123/person14", "startFrame": 1,
22 "endFrame": 847, "nz": 6,
23     "ext": "jpg", "anno_path": "anno/UAV123/person14_1.txt", "object_class": "person"},
24     {"name": "uav_person14_2", "path": "data_seq/UAV123/person14", "startFrame": 847,
25 "endFrame": 1813, "nz": 6,
26     "ext": "jpg", "anno_path": "anno/UAV123/person14_2.txt", "object_class": "person"},
27     {"name": "uav_person14_3", "path": "data_seq/UAV123/person14", "startFrame": 1813,
28 "endFrame": 2923,
29     "nz": 6, "ext": "jpg", "anno_path": "anno/UAV123/person14_3.txt", "object_class":
30 "person"},
31     {"name": "uav_person15", "path": "data_seq/UAV123/person15", "startFrame": 1,
32 "endFrame": 1339, "nz": 6,
33     "ext": "jpg", "anno_path": "anno/UAV123/person15.txt", "object_class": "person"},
34     {"name": "uav_person16", "path": "data_seq/UAV123/person16", "startFrame": 1,
35 "endFrame": 1147, "nz": 6,
36     "ext": "jpg", "anno_path": "anno/UAV123/person16.txt", "object_class": "person"},
37     {"name": "uav_person17_1", "path": "data_seq/UAV123/person17", "startFrame": 1,
38 "endFrame": 1501, "nz": 6,
39     "ext": "jpg", "anno_path": "anno/UAV123/person17_1.txt", "object_class": "person"},
40     {"name": "uav_person17_2", "path": "data_seq/UAV123/person17", "startFrame": 1501,
41 "endFrame": 2347,
42     "nz": 6, "ext": "jpg", "anno_path": "anno/UAV123/person17_2.txt", "object_class":
43 "person"},
44     {"name": "uav_person18", "path": "data_seq/UAV123/person18", "startFrame": 1,
45 "endFrame": 1393, "nz": 6,
46     "ext": "jpg", "anno_path": "anno/UAV123/person18.txt", "object_class": "person"},
47     {"name": "uav_person19_1", "path": "data_seq/UAV123/person19", "startFrame": 1,
48 "endFrame": 1243, "nz": 6,
49     "ext": "jpg", "anno_path": "anno/UAV123/person19_1.txt", "object_class": "person"},
50     {"name": "uav_person19_2", "path": "data_seq/UAV123/person19", "startFrame": 1243,

```

```

1  "endFrame": 2791,
2      "nz": 6, "ext": "jpg", "anno_path": "anno/UAV123/person19_2.txt", "object_class":
3  "person"},
4      {"name": "uav_person19_3", "path": "data_seq/UAV123/person19", "startFrame": 2791,
5  "endFrame": 4357,
6      "nz": 6, "ext": "jpg", "anno_path": "anno/UAV123/person19_3.txt", "object_class":
7  "person"},
8      {"name": "uav_person1_s", "path": "data_seq/UAV123/person1_s", "startFrame": 1,
9  "endFrame": 1600, "nz": 6,
10     "ext": "jpg", "anno_path": "anno/UAV123/person1_s.txt", "object_class": "person"},
11     {"name": "uav_person2_1", "path": "data_seq/UAV123/person2", "startFrame": 1,
12 "endFrame": 1189, "nz": 6,
13     "ext": "jpg", "anno_path": "anno/UAV123/person2_1.txt", "object_class": "person"},
14     {"name": "uav_person2_2", "path": "data_seq/UAV123/person2", "startFrame": 1189,
15 "endFrame": 2623, "nz": 6,
16     "ext": "jpg", "anno_path": "anno/UAV123/person2_2.txt", "object_class": "person"},
17     {"name": "uav_person20", "path": "data_seq/UAV123/person20", "startFrame": 1,
18 "endFrame": 1783, "nz": 6,
19     "ext": "jpg", "anno_path": "anno/UAV123/person20.txt", "object_class": "person"},
20     {"name": "uav_person21", "path": "data_seq/UAV123/person21", "startFrame": 1,
21 "endFrame": 487, "nz": 6,
22     "ext": "jpg", "anno_path": "anno/UAV123/person21.txt", "object_class": "person"},
23     {"name": "uav_person22", "path": "data_seq/UAV123/person22", "startFrame": 1,
24 "endFrame": 199, "nz": 6,
25     "ext": "jpg", "anno_path": "anno/UAV123/person22.txt", "object_class": "person"},
26     {"name": "uav_person23", "path": "data_seq/UAV123/person23", "startFrame": 1,
27 "endFrame": 397, "nz": 6,
28     "ext": "jpg", "anno_path": "anno/UAV123/person23.txt", "object_class": "person"},
29     {"name": "uav_person2_s", "path": "data_seq/UAV123/person2_s", "startFrame": 1,
30 "endFrame": 250, "nz": 6,
31     "ext": "jpg", "anno_path": "anno/UAV123/person2_s.txt", "object_class": "person"},
32     {"name": "uav_person3", "path": "data_seq/UAV123/person3", "startFrame": 1,
33 "endFrame": 643, "nz": 6,
34     "ext": "jpg", "anno_path": "anno/UAV123/person3.txt", "object_class": "person"},
35     {"name": "uav_person3_s", "path": "data_seq/UAV123/person3_s", "startFrame": 1,
36 "endFrame": 505, "nz": 6,
37     "ext": "jpg", "anno_path": "anno/UAV123/person3_s.txt", "object_class": "person"},
38     {"name": "uav_person4_1", "path": "data_seq/UAV123/person4", "startFrame": 1,
39 "endFrame": 1501, "nz": 6,
40     "ext": "jpg", "anno_path": "anno/UAV123/person4_1.txt", "object_class": "person"},
41     {"name": "uav_person4_2", "path": "data_seq/UAV123/person4", "startFrame": 1501,
42 "endFrame": 2743, "nz": 6,
43     "ext": "jpg", "anno_path": "anno/UAV123/person4_2.txt", "object_class": "person"},
44     {"name": "uav_person5_1", "path": "data_seq/UAV123/person5", "startFrame": 1,
45 "endFrame": 877, "nz": 6,
46     "ext": "jpg", "anno_path": "anno/UAV123/person5_1.txt", "object_class": "person"},
47     {"name": "uav_person5_2", "path": "data_seq/UAV123/person5", "startFrame": 877,
48 "endFrame": 2101, "nz": 6,
49     "ext": "jpg", "anno_path": "anno/UAV123/person5_2.txt", "object_class": "person"},
50     {"name": "uav_person6", "path": "data_seq/UAV123/person6", "startFrame": 1,
    
```

```

1  "endFrame": 901, "nz": 6,
2      "ext": "jpg", "anno_path": "anno/UAV123/person6.txt", "object_class": "person"},
3      {"name": "uav_person7_1", "path": "data_seq/UAV123/person7", "startFrame": 1,
4  "endFrame": 1249, "nz": 6,
5      "ext": "jpg", "anno_path": "anno/UAV123/person7_1.txt", "object_class": "person"},
6      {"name": "uav_person7_2", "path": "data_seq/UAV123/person7", "startFrame": 1249,
7  "endFrame": 2065, "nz": 6,
8      "ext": "jpg", "anno_path": "anno/UAV123/person7_2.txt", "object_class": "person"},
9      {"name": "uav_person8_1", "path": "data_seq/UAV123/person8", "startFrame": 1,
10 "endFrame": 1075, "nz": 6,
11     "ext": "jpg", "anno_path": "anno/UAV123/person8_1.txt", "object_class": "person"},
12     {"name": "uav_person8_2", "path": "data_seq/UAV123/person8", "startFrame": 1075,
13 "endFrame": 1525, "nz": 6,
14     "ext": "jpg", "anno_path": "anno/UAV123/person8_2.txt", "object_class": "person"},
15     {"name": "uav_person9", "path": "data_seq/UAV123/person9", "startFrame": 1,
16 "endFrame": 661, "nz": 6,
17     "ext": "jpg", "anno_path": "anno/UAV123/person9.txt", "object_class": "person"},
18     {"name": "uav_truck1", "path": "data_seq/UAV123/truck1", "startFrame": 1, "endFrame":
19 463, "nz": 6,
20     "ext": "jpg", "anno_path": "anno/UAV123/truck1.txt", "object_class": "truck"},
21     {"name": "uav_truck2", "path": "data_seq/UAV123/truck2", "startFrame": 1, "endFrame":
22 385, "nz": 6,
23     "ext": "jpg", "anno_path": "anno/UAV123/truck2.txt", "object_class": "truck"},
24     {"name": "uav_truck3", "path": "data_seq/UAV123/truck3", "startFrame": 1, "endFrame":
25 535, "nz": 6,
26     "ext": "jpg", "anno_path": "anno/UAV123/truck3.txt", "object_class": "truck"},
27     {"name": "uav_truck4_1", "path": "data_seq/UAV123/truck4", "startFrame": 1,
28 "endFrame": 577, "nz": 6,
29     "ext": "jpg", "anno_path": "anno/UAV123/truck4_1.txt", "object_class": "truck"},
30     {"name": "uav_truck4_2", "path": "data_seq/UAV123/truck4", "startFrame": 577,
31 "endFrame": 1261, "nz": 6,
32     "ext": "jpg", "anno_path": "anno/UAV123/truck4_2.txt", "object_class": "truck"},
33     {"name": "uav_uav1_1", "path": "data_seq/UAV123/uav1", "startFrame": 1, "endFrame":
34 1555, "nz": 6,
35     "ext": "jpg", "anno_path": "anno/UAV123/uav1_1.txt", "object_class": "aircraft"},
36     {"name": "uav_uav1_2", "path": "data_seq/UAV123/uav1", "startFrame": 1555,
37 "endFrame": 2377, "nz": 6,
38     "ext": "jpg", "anno_path": "anno/UAV123/uav1_2.txt", "object_class": "aircraft"},
39     {"name": "uav_uav1_3", "path": "data_seq/UAV123/uav1", "startFrame": 2473,
40 "endFrame": 3469, "nz": 6,
41     "ext": "jpg", "anno_path": "anno/UAV123/uav1_3.txt", "object_class": "aircraft"},
42     {"name": "uav_uav2", "path": "data_seq/UAV123/uav2", "startFrame": 1, "endFrame":
43 133, "nz": 6,
44     "ext": "jpg", "anno_path": "anno/UAV123/uav2.txt", "object_class": "aircraft"},
45     {"name": "uav_uav3", "path": "data_seq/UAV123/uav3", "startFrame": 1, "endFrame":
46 265, "nz": 6,
47     "ext": "jpg", "anno_path": "anno/UAV123/uav3.txt", "object_class": "aircraft"},
48     {"name": "uav_uav4", "path": "data_seq/UAV123/uav4", "startFrame": 1, "endFrame":
49 157, "nz": 6,
50     "ext": "jpg", "anno_path": "anno/UAV123/uav4.txt", "object_class": "aircraft"},
    
```

```

1         {"name": "uav_uav5", "path": "data_seq/UAV123/uav5", "startFrame": 1, "endFrame":
2 139, "nz": 6,
3         "ext": "jpg", "anno_path": "anno/UAV123/uav5.txt", "object_class": "aircraft"},
4         {"name": "uav_uav6", "path": "data_seq/UAV123/uav6", "startFrame": 1, "endFrame":
5 109, "nz": 6,
6         "ext": "jpg", "anno_path": "anno/UAV123/uav6.txt", "object_class": "aircraft"},
7         {"name": "uav_uav7", "path": "data_seq/UAV123/uav7", "startFrame": 1, "endFrame":
8 373, "nz": 6,
9         "ext": "jpg", "anno_path": "anno/UAV123/uav7.txt", "object_class": "aircraft"},
10        {"name": "uav_uav8", "path": "data_seq/UAV123/uav8", "startFrame": 1, "endFrame":
11 301, "nz": 6,
12        "ext": "jpg", "anno_path": "anno/UAV123/uav8.txt", "object_class": "aircraft"},
13        {"name": "uav_wakeboard1", "path": "data_seq/UAV123/wakeboard1", "startFrame": 1,
14 "endFrame": 421, "nz": 6,
15        "ext": "jpg", "anno_path": "anno/UAV123/wakeboard1.txt", "object_class": "person"},
16        {"name": "uav_wakeboard10", "path": "data_seq/UAV123/wakeboard10", "startFrame":
17 1, "endFrame": 469,
18        "nz": 6, "ext": "jpg", "anno_path": "anno/UAV123/wakeboard10.txt", "object_class":
19 "person"},
20        {"name": "uav_wakeboard2", "path": "data_seq/UAV123/wakeboard2", "startFrame": 1,
21 "endFrame": 733, "nz": 6,
22        "ext": "jpg", "anno_path": "anno/UAV123/wakeboard2.txt", "object_class": "person"},
23        {"name": "uav_wakeboard3", "path": "data_seq/UAV123/wakeboard3", "startFrame": 1,
24 "endFrame": 823, "nz": 6,
25        "ext": "jpg", "anno_path": "anno/UAV123/wakeboard3.txt", "object_class": "person"},
26        {"name": "uav_wakeboard4", "path": "data_seq/UAV123/wakeboard4", "startFrame": 1,
27 "endFrame": 697, "nz": 6,
28        "ext": "jpg", "anno_path": "anno/UAV123/wakeboard4.txt", "object_class": "person"},
29        {"name": "uav_wakeboard5", "path": "data_seq/UAV123/wakeboard5", "startFrame": 1,
30 "endFrame": 1675, "nz": 6,
31        "ext": "jpg", "anno_path": "anno/UAV123/wakeboard5.txt", "object_class": "person"},
32        {"name": "uav_wakeboard6", "path": "data_seq/UAV123/wakeboard6", "startFrame": 1,
33 "endFrame": 1165, "nz": 6,
34        "ext": "jpg", "anno_path": "anno/UAV123/wakeboard6.txt", "object_class": "person"},
35        {"name": "uav_wakeboard7", "path": "data_seq/UAV123/wakeboard7", "startFrame": 1,
36 "endFrame": 199, "nz": 6,
37        "ext": "jpg", "anno_path": "anno/UAV123/wakeboard7.txt", "object_class": "person"},
38        {"name": "uav_wakeboard8", "path": "data_seq/UAV123/wakeboard8", "startFrame": 1,
39 "endFrame": 1543, "nz": 6,
40        "ext": "jpg", "anno_path": "anno/UAV123/wakeboard8.txt", "object_class": "person"},
41        {"name": "uav_wakeboard9", "path": "data_seq/UAV123/wakeboard9", "startFrame": 1,
42 "endFrame": 355, "nz": 6,
43        "ext": "jpg", "anno_path": "anno/UAV123/wakeboard9.txt", "object_class": "person"}
44    ]
45    return sequence_info_list
46
47 import numpy as np
48 from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
49 import os
50 from lib.test.utils.load_text import load_text
51 class TrackingNetDataset(BaseDataset):

```

```

1     def __init__(self):
2         super().__init__()
3         self.base_path = self.env_settings.trackingnet_path
4         sets = 'TEST'
5         if not isinstance(sets, (list, tuple)):
6             if sets == 'TEST':
7                 sets = ['TEST']
8             elif sets == 'TRAIN':
9                 sets = ['TRAIN_{}'.format(i) for i in range(5)]
10        self.sequence_list = self._list_sequences(self.base_path, sets)
11    def get_sequence_list(self):
12        return SequenceList([self._construct_sequence(set, seq_name) for set, seq_name in
13self.sequence_list])
14    def _construct_sequence(self, set, sequence_name):
15        anno_path = '{}/{}/anno/{}.txt'.format(self.base_path, set, sequence_name)
16        ground_truth_rect = load_text(str(anno_path), delimiter=',', dtype=np.float64,
17backend='numpy')
18        frames_path = '{}/{}/frames/'.format(self.base_path, set, sequence_name)
19        frame_list = [frame for frame in os.listdir(frames_path) if frame.endswith(".jpg")]
20        frame_list.sort(key=lambda f: int(f[:-4]))
21        frames_list = [os.path.join(frames_path, frame) for frame in frame_list]
22        return Sequence(sequence_name, frames_list, 'trackingnet', ground_truth_rect.reshape(-1,
234))
24    def __len__(self):
25        return len(self.sequence_list)
26    def _list_sequences(self, root, set_ids):
27        sequence_list = []
28        for s in set_ids:
29            anno_dir = os.path.join(root, s, "anno")
30            sequences_cur_set = [(s, os.path.splitext(f)[0]) for f in os.listdir(anno_dir) if
31f.endswith('.txt')]
32            sequence_list += sequences_cur_set
33        return sequence_list
34    import importlib
35    import os
36    from collections import OrderedDict
37    from lib.test.evaluation.environment import env_settings
38    import time
39    import cv2 as cv
40    from lib.utils.lmdb_utils import decode_img
41    from pathlib import Path
42    import numpy as np
43    def trackerlist(name: str, parameter_name: str, dataset_name: str, run_ids = None, display_name: str =
44    None,
45                result_only=False):
46        if run_ids is None or isinstance(run_ids, int):
47            run_ids = [run_ids]
48        return [Tracker(name, parameter_name, dataset_name, run_id, display_name, result_only) for
49run_id in run_ids]
50    class Tracker:

```

```

1      def __init__(self, name: str, parameter_name: str, dataset_name: str, run_id: int = None,
2 display_name: str = None,
3         result_only=False):
4          assert run_id is None or isinstance(run_id, int)
5          self.name = name
6          self.parameter_name = parameter_name
7          self.dataset_name = dataset_name
8          self.run_id = run_id
9          self.display_name = display_name
10         env = env_settings()
11         if self.run_id is None:
12             self.results_dir = '{}/{}/{}'.format(env.results_path, self.name, self.parameter_name)
13         else:
14             self.results_dir = '{}/{}/{:03d}'.format(env.results_path, self.name, self.parameter_name,
15 self.run_id)
16         if result_only:
17             self.results_dir = '{}/{}/{}'.format(env.results_path, self.name)
18             tracker_module_abspath = os.path.abspath(os.path.join(os.path.dirname(__file__),
19 .., 'tracker', '%s.py' %
20 self.name))
21             if os.path.isfile(tracker_module_abspath):
22                 tracker_module = importlib.import_module('lib.test.tracker.{}'.format(self.name))
23                 self.tracker_class = tracker_module.get_tracker_class()
24             else:
25                 self.tracker_class = None
26         def create_tracker(self, params):
27             tracker = self.tracker_class(params, self.dataset_name)
28             return tracker
29         def run_sequence(self, seq, debug=None):
30             params = self.get_parameters()
31             debug_ = debug
32             if debug is None:
33                 debug_ = getattr(params, 'debug', 0)
34             params.debug = debug_
35             init_info = seq.init_info()
36             tracker = self.create_tracker(params)
37             output = self._track_sequence(tracker, seq, init_info)
38             return output
39         def _track_sequence(self, tracker, seq, init_info):
40             output = {'target_bbox': [],
41 'time': []}
42             if tracker.params.save_all_boxes:
43                 output['all_boxes'] = []
44                 output['all_scores'] = []
45             def _store_outputs(tracker_out: dict, defaults=None):
46                 defaults = {} if defaults is None else defaults
47                 for key in output.keys():
48                     val = tracker_out.get(key, defaults.get(key, None))
49                     if key in tracker_out or val is not None:
50                         output[key].append(val)

```



```

1         image = self._read_image(seq.frames[0])
2         start_time = time.time()
3         out = tracker.initialize(image, init_info)
4         if out is None:
5             out = {}
6         prev_output = OrderedDict(out)
7         init_default = {'target_bbox': init_info.get('init_bbox'),
8                         'time': time.time() - start_time}
9         if tracker.params.save_all_boxes:
10             init_default['all_boxes'] = out['all_boxes']
11             init_default['all_scores'] = out['all_scores']
12         _store_outputs(out, init_default)
13         for frame_num, frame_path in enumerate(seq.frames[1:], start=1):
14             image = self._read_image(frame_path)
15             start_time = time.time()
16             info = seq.frame_info(frame_num)
17             info['previous_output'] = prev_output
18             if len(seq.ground_truth_rect) > 1:
19                 info['gt_bbox'] = seq.ground_truth_rect[frame_num]
20             out = tracker.track(image, info)
21             prev_output = OrderedDict(out)
22             _store_outputs(out, {'time': time.time() - start_time})
23         for key in ['target_bbox', 'all_boxes', 'all_scores']:
24             if key in output and len(output[key]) <= 1:
25                 output.pop(key)
26         return output
27     def run_video(self, videofilepath, optional_box=None, debug=None, visdom_info=None,
28 save_results=False):
29         params = self.get_parameters()
30         debug_ = debug
31         if debug is None:
32             debug_ = getattr(params, 'debug', 0)
33         params.debug = debug_
34         params.tracker_name = self.name
35         params.param_name = self.parameter_name
36         multiobj_mode = getattr(params, 'multiobj_mode', getattr(self.tracker_class,
37 'multiobj_mode', 'default'))
38         if multiobj_mode == 'default':
39             tracker = self.create_tracker(params)
40         elif multiobj_mode == 'parallel':
41             tracker = MultiObjectWrapper(self.tracker_class, params, self.visdom, fast_load=True)
42         else:
43             raise ValueError('Unknown multi object mode {}'.format(multiobj_mode))
44         assert os.path.isfile(videofilepath), "Invalid param {}".format(videofilepath)
45         ", videofilepath must be a valid videofile"
46         output_boxes = []
47         cap = cv.VideoCapture(videofilepath)
48         display_name = 'Display: ' + tracker.params.tracker_name
49         cv.namedWindow(display_name, cv.WINDOW_NORMAL | cv.WINDOW_KEEPRATIO)
50         cv.resizeWindow(display_name, 960, 720)
    
```

```

1      success, frame = cap.read()
2      cv.imshow(display_name, frame)
3      def _build_init_info(box):
4          return {'init_bbox': box}
5      if success is not True:
6          print("Read frame from {} failed.".format(videofilepath))
7          exit(-1)
8      if optional_box is not None:
9          assert isinstance(optional_box, (list, tuple))
10         assert len(optional_box) == 4, "valid box's format is [x,y,w,h]"
11         tracker.initialize(frame, _build_init_info(optional_box))
12         output_boxes.append(optional_box)
13     else:
14         while True:
15             frame_disp = frame.copy()
16             cv.putText(frame_disp, 'Select target ROI and press ENTER', (20, 30),
17 cv.FONT_HERSHEY_COMPLEX_SMALL,
18                 1.5, (0, 0, 0), 1)
19             x, y, w, h = cv.selectROI(display_name, frame_disp, fromCenter=False)
20             init_state = [x, y, w, h]
21             tracker.initialize(frame, _build_init_info(init_state))
22             output_boxes.append(init_state)
23             break
24     while True:
25         ret, frame = cap.read()
26         if frame is None:
27             break
28         frame_disp = frame.copy()
29         out = tracker.track(frame)
30         state = [int(s) for s in out['target_bbox']]
31         output_boxes.append(state)
32         cv.rectangle(frame_disp, (state[0], state[1]), (state[2] + state[0], state[3] + state[1]),
33             (0, 255, 0), 5)
34         font_color = (0, 0, 0)
35         cv.putText(frame_disp, 'Tracking!', (20, 30), cv.FONT_HERSHEY_COMPLEX_SMALL, 1,
36             font_color, 1)
37         cv.putText(frame_disp, 'Press r to reset', (20, 55), cv.FONT_HERSHEY_COMPLEX_SMALL,
38 1,
39             font_color, 1)
40         cv.putText(frame_disp, 'Press q to quit', (20, 80), cv.FONT_HERSHEY_COMPLEX_SMALL,
41 1,
42             font_color, 1)
43         cv.imshow(display_name, frame_disp)
44         key = cv.waitKey(1)
45         if key == ord('q'):
46             break
47         elif key == ord('r'):
48             ret, frame = cap.read()
49             frame_disp = frame.copy()
50             cv.putText(frame_disp, 'Select target ROI and press ENTER', (20, 30),
    
```

```

1  cv.FONT_HERSHEY_COMPLEX_SMALL, 1.5,
2      (0, 0, 0), 1)
3      cv.imshow(display_name, frame_disp)
4      x, y, w, h = cv.selectROI(display_name, frame_disp, fromCenter=False)
5      init_state = [x, y, w, h]
6      tracker.initialize(frame, _build_init_info(init_state))
7      output_boxes.append(init_state)
8  cap.release()
9  cv.destroyAllWindows()
10 if save_results:
11     if not os.path.exists(self.results_dir):
12         os.makedirs(self.results_dir)
13     video_name = Path(videofilepath).stem
14     base_results_path = os.path.join(self.results_dir, 'video_{}'.format(video_name))
15     tracked_bb = np.array(output_boxes).astype(int)
16     bbox_file = '{}.txt'.format(base_results_path)
17     np.savetxt(bbox_file, tracked_bb, delimiter='\t', fmt='%d')
18 def get_parameters(self):
19     param_module = importlib.import_module('lib.test.parameter.{}'.format(self.name))
20     params = param_module.parameters(self.parameter_name)
21     return params
22 def _read_image(self, image_file: str):
23     if isinstance(image_file, str):
24         im = cv.imread(image_file)
25         return cv.cvtColor(im, cv.COLOR_BGR2RGB)
26     elif isinstance(image_file, list) and len(image_file) == 2:
27         return decode_img(image_file[0], image_file[1])
28     else:
29         raise ValueError("type of image_file should be str or list")
30 import os
31 import numpy as np
32 from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
33 from lib.test.utils.load_text import load_text, load_str
34 class TNL2kDataset(BaseDataset):
35     def __init__(self):
36         super().__init__()
37         self.base_path = self.env_settings.tnl2k_path
38         self.sequence_list = self._get_sequence_list()
39     def _get_sequence_list(self):
40         return SequenceList([self._construct_sequence(s) for s in self.sequence_list])
41     def _construct_sequence(self, sequence_name):
42         anno_path = '{}/groundtruth.txt'.format(self.base_path, sequence_name)
43         ground_truth_rect = load_text(str(anno_path), delimiter=',', dtype=np.float64)
44         text_dsp_path = '{}/language.txt'.format(self.base_path, sequence_name)
45         text_dsp = load_str(text_dsp_path)
46         frames_path = '{}/imgs'.format(self.base_path, sequence_name)
47         frames_list = [f for f in os.listdir(frames_path)]
48         frames_list = sorted(frames_list)
49         frames_list = ['{}/{}'.format(frames_path, frame_i) for frame_i in frames_list]
50         return Sequence(sequence_name, frames_list, 'tnl2k', ground_truth_rect.reshape(-1, 4),

```

```

1  text_dsp=text_dsp)
2      def __len__(self):
3          return len(self.sequence_list)
4      def _get_sequence_list(self):
5          sequence_list = []
6          for seq in os.listdir(self.base_path):
7              if os.path.isdir(os.path.join(self.base_path, seq)):
8                  sequence_list.append(seq)
9          return sequence_list
10 import numpy as np
11 from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
12 import os
13 import glob
14 import six
15 class TC128Dataset(BaseDataset):
16     def __init__(self):
17         super().__init__()
18         self.base_path = self.env_settings.tc128_path
19         self.anno_files = sorted(glob.glob(
20             os.path.join(self.base_path, '*/*_gt.txt')))
21         self.seq_dirs = [os.path.dirname(f) for f in self.anno_files]
22         self.seq_names = [os.path.basename(d) for d in self.seq_dirs]
23         self.range_files = [glob.glob(os.path.join(d, '*_frames.txt'))[0] for d in self.seq_dirs]
24     def get_sequence_list(self):
25         return SequenceList([self._construct_sequence(s) for s in self.seq_names])
26     def _construct_sequence(self, sequence_name):
27         if isinstance(sequence_name, six.string_types):
28             if not sequence_name in self.seq_names:
29                 raise Exception('Sequence {} not found.'.format(sequence_name))
30             index = self.seq_names.index(sequence_name)
31             frames = np.loadtxt(self.range_files[index], dtype=int, delimiter=',')
32             img_files = [os.path.join(self.seq_dirs[index], 'img/%04d.jpg' % f) for f in range(frames[0],
33 frames[1] + 1)]
34             anno = np.loadtxt(self.anno_files[index], delimiter=',')
35             assert len(img_files) == len(anno)
36             assert anno.shape[1] == 4
37             return Sequence(sequence_name, img_files, 'tc128', anno.reshape(-1, 4))
38     def __len__(self):
39         return len(self.seq_names)
40 import numpy as np
41 from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
42 import os
43 import glob
44 import six
45 class TC128CEDataset(BaseDataset):
46     def __init__(self):
47         super().__init__()
48         self.base_path = self.env_settings.tc128_path
49         self.anno_files = sorted(glob.glob(
50             os.path.join(self.base_path, '*/*_gt.txt')))

```

```

1         self.anno_files = [s for s in self.anno_files if "_ce" in s]
2         self.seq_dirs = [os.path.dirname(f) for f in self.anno_files]
3         self.seq_names = [os.path.basename(d) for d in self.seq_dirs]
4         self.range_files = [glob.glob(os.path.join(d, '*_frames.txt'))[0] for d in self.seq_dirs]
5     def get_sequence_list(self):
6         return SequenceList([self._construct_sequence(s) for s in self.seq_names])
7     def _construct_sequence(self, sequence_name):
8         if isinstance(sequence_name, six.string_types):
9             if not sequence_name in self.seq_names:
10                 raise Exception('Sequence {} not found.'.format(sequence_name))
11                 index = self.seq_names.index(sequence_name)
12                 frames = np.loadtxt(self.range_files[index], dtype=int, delimiter=',')
13                 img_files = [os.path.join(self.seq_dirs[index], 'img/%04d.jpg' % f) for f in range(frames[0],
14 frames[1] + 1)]
15                 anno = np.loadtxt(self.anno_files[index], delimiter=',')
16                 assert len(img_files) == len(anno)
17                 assert anno.shape[1] == 4
18                 return Sequence(sequence_name, img_files, 'tc128', anno.reshape(-1, 4))
19     def __len__(self):
20         return len(self.seq_names)
21 import numpy as np
22 import multiprocessing
23 import os
24 import sys
25 from itertools import product
26 from collections import OrderedDict
27 from lib.test.evaluation import Sequence, Tracker
28 import torch
29 def _save_tracker_output(seq: Sequence, tracker: Tracker, output: dict):
30     if not os.path.exists(tracker.results_dir):
31         print("create tracking result dir:", tracker.results_dir)
32         os.makedirs(tracker.results_dir)
33     if seq.dataset in ['trackingnet', 'got10k']:
34         if not os.path.exists(os.path.join(tracker.results_dir, seq.dataset)):
35             os.makedirs(os.path.join(tracker.results_dir, seq.dataset))
36     if seq.dataset in ['trackingnet', 'got10k']:
37         base_results_path = os.path.join(tracker.results_dir, seq.dataset, seq.name)
38     else:
39         base_results_path = os.path.join(tracker.results_dir, seq.name)
40     def save_bb(file, data):
41         tracked_bb = np.array(data).astype(int)
42         np.savetxt(file, tracked_bb, delimiter='\t', fmt='%d')
43     def save_time(file, data):
44         exec_times = np.array(data).astype(float)
45         np.savetxt(file, exec_times, delimiter='\t', fmt='%f')
46     def save_score(file, data):
47         scores = np.array(data).astype(float)
48         np.savetxt(file, scores, delimiter='\t', fmt='%0.2f')
49 def run_sequence(seq: Sequence, tracker: Tracker, debug=False, num_gpu=8):
50     try:

```

```

1         worker_name = multiprocessing.current_process().name
2         worker_id = int(worker_name[worker_name.find('-') + 1:]) - 1
3         gpu_id = worker_id % num_gpu
4         torch.cuda.set_device(gpu_id)
5     except:
6         pass
7 import numpy as np
8 from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
9 from lib.test.utils.load_text import load_text
10 class OTBDataset(BaseDataset):
11     def __init__(self):
12         super().__init__()
13         self.base_path = self.env_settings.otb_path
14         self.sequence_info_list = self._get_sequence_info_list()
15     def get_sequence_list(self):
16         return SequenceList([self._construct_sequence(s) for s in self.sequence_info_list])
17     def _construct_sequence(self, sequence_info):
18         sequence_path = sequence_info['path']
19         nz = sequence_info['nz']
20         ext = sequence_info['ext']
21         start_frame = sequence_info['startFrame']
22         end_frame = sequence_info['endFrame']
23         init_omit = 0
24         if 'initOmit' in sequence_info:
25             init_omit = sequence_info['initOmit']
26         frames =
27         ['{base_path}/{sequence_path}/{frame:0{nz}}.{ext}'.format(base_path=self.base_path,
28             sequence_path=sequence_path, frame=frame_num, nz=nz, ext=ext) for frame_num in
29         range(start_frame+init_omit, end_frame+1)]
30         anno_path = '{}{}/groundtruth.txt'.format(self.base_path, sequence_info['name'])
31         ground_truth_rect = load_text(str(anno_path), delimiter=',', None, dtype=np.float64,
32         backend='numpy')
33         return Sequence(sequence_info['name'], frames, 'otb', ground_truth_rect[init_omit:,:],
34             object_class=sequence_info['object_class'])
35     def __len__(self):
36         return len(self.sequence_info_list)
37     def _get_sequence_info_list(self):
38         sequence_info_list = [
39             {"name": "Basketball", "path": "Basketball/img", "startFrame": 1, "endFrame": 725, "nz":
40 4, "ext": ".jpg", "anno_path": "Basketball/groundtruth_rect.txt",
41             "object_class": "person"},
42             {"name": "Biker", "path": "Biker/img", "startFrame": 1, "endFrame": 142, "nz": 4, "ext":
43 ".jpg", "anno_path": "Biker/groundtruth_rect.txt",
44             "object_class": "person head"},
45             {"name": "Bird1", "path": "Bird1/img", "startFrame": 1, "endFrame": 408, "nz": 4, "ext":
46 ".jpg", "anno_path": "Bird1/groundtruth_rect.txt",
47             "object_class": "bird"},
48             {"name": "Bird2", "path": "Bird2/img", "startFrame": 1, "endFrame": 99, "nz": 4, "ext": ".jpg",
49 "anno_path": "Bird2/groundtruth_rect.txt",
50             "object_class": "bird"},

```

```

1      {"name": "BlurBody", "path": "BlurBody/img", "startFrame": 1, "endFrame": 334, "nz": 4,
2      "ext": "jpg", "anno_path": "BlurBody/groundtruth_rect.txt",
3      "object_class": "person"},
4      {"name": "BlurCar1", "path": "BlurCar1/img", "startFrame": 247, "endFrame": 988, "nz": 4,
5      "ext": "jpg", "anno_path": "BlurCar1/groundtruth_rect.txt",
6      "object_class": "car"},
7      {"name": "BlurCar2", "path": "BlurCar2/img", "startFrame": 1, "endFrame": 585, "nz": 4,
8      "ext": "jpg", "anno_path": "BlurCar2/groundtruth_rect.txt",
9      "object_class": "car"},
10     {"name": "BlurCar3", "path": "BlurCar3/img", "startFrame": 3, "endFrame": 359, "nz": 4,
11     "ext": "jpg", "anno_path": "BlurCar3/groundtruth_rect.txt",
12     "object_class": "car"},
13     {"name": "BlurCar4", "path": "BlurCar4/img", "startFrame": 18, "endFrame": 397, "nz": 4,
14     "ext": "jpg", "anno_path": "BlurCar4/groundtruth_rect.txt",
15     "object_class": "car"},
16     {"name": "BlurFace", "path": "BlurFace/img", "startFrame": 1, "endFrame": 493, "nz": 4,
17     "ext": "jpg", "anno_path": "BlurFace/groundtruth_rect.txt",
18     "object_class": "face"},
19     {"name": "BlurOwl", "path": "BlurOwl/img", "startFrame": 1, "endFrame": 631, "nz": 4,
20     "ext": "jpg", "anno_path": "BlurOwl/groundtruth_rect.txt",
21     "object_class": "other"},
22     {"name": "Board", "path": "Board/img", "startFrame": 1, "endFrame": 698, "nz": 5, "ext":
23     "jpg", "anno_path": "Board/groundtruth_rect.txt",
24     "object_class": "other"},
25     {"name": "Bolt", "path": "Bolt/img", "startFrame": 1, "endFrame": 350, "nz": 4, "ext": "jpg",
26     "anno_path": "Bolt/groundtruth_rect.txt",
27     "object_class": "person"},
28     {"name": "Bolt2", "path": "Bolt2/img", "startFrame": 1, "endFrame": 293, "nz": 4, "ext":
29     "jpg", "anno_path": "Bolt2/groundtruth_rect.txt",
30     "object_class": "person"},
31     {"name": "Box", "path": "Box/img", "startFrame": 1, "endFrame": 1161, "nz": 4, "ext": "jpg",
32     "anno_path": "Box/groundtruth_rect.txt",
33     "object_class": "other"},
34     {"name": "Boy", "path": "Boy/img", "startFrame": 1, "endFrame": 602, "nz": 4, "ext": "jpg",
35     "anno_path": "Boy/groundtruth_rect.txt",
36     "object_class": "face"},
37     {"name": "Car1", "path": "Car1/img", "startFrame": 1, "endFrame": 1020, "nz": 4, "ext":
38     "jpg", "anno_path": "Car1/groundtruth_rect.txt",
39     "object_class": "car"},
40     {"name": "Car2", "path": "Car2/img", "startFrame": 1, "endFrame": 913, "nz": 4, "ext":
41     "jpg", "anno_path": "Car2/groundtruth_rect.txt",
42     "object_class": "car"},
43     {"name": "Car24", "path": "Car24/img", "startFrame": 1, "endFrame": 3059, "nz": 4, "ext":
44     "jpg", "anno_path": "Car24/groundtruth_rect.txt",
45     "object_class": "car"},
46     {"name": "Car4", "path": "Car4/img", "startFrame": 1, "endFrame": 659, "nz": 4, "ext":
47     "jpg", "anno_path": "Car4/groundtruth_rect.txt",
48     "object_class": "car"},
49     {"name": "CarDark", "path": "CarDark/img", "startFrame": 1, "endFrame": 393, "nz": 4,
50     "ext": "jpg", "anno_path": "CarDark/groundtruth_rect.txt",

```

```

1         "object_class": "car"},
2         {"name": "CarScale", "path": "CarScale/img", "startFrame": 1, "endFrame": 252, "nz": 4,
3 "ext": "jpg", "anno_path": "CarScale/groundtruth_rect.txt",
4         "object_class": "car"},
5         {"name": "ClifBar", "path": "ClifBar/img", "startFrame": 1, "endFrame": 472, "nz": 4, "ext":
6 "jpg", "anno_path": "ClifBar/groundtruth_rect.txt",
7         "object_class": "other"},
8         {"name": "Coke", "path": "Coke/img", "startFrame": 1, "endFrame": 291, "nz": 4, "ext":
9 "jpg", "anno_path": "Coke/groundtruth_rect.txt",
10        "object_class": "other"},
11        {"name": "Couple", "path": "Couple/img", "startFrame": 1, "endFrame": 140, "nz": 4, "ext":
12 "jpg", "anno_path": "Couple/groundtruth_rect.txt",
13        "object_class": "person"},
14        {"name": "Coupon", "path": "Coupon/img", "startFrame": 1, "endFrame": 327, "nz": 4,
15 "ext": "jpg", "anno_path": "Coupon/groundtruth_rect.txt",
16        "object_class": "other"},
17        {"name": "Crossing", "path": "Crossing/img", "startFrame": 1, "endFrame": 120, "nz": 4,
18 "ext": "jpg", "anno_path": "Crossing/groundtruth_rect.txt",
19        "object_class": "person"},
20        {"name": "Crowds", "path": "Crowds/img", "startFrame": 1, "endFrame": 347, "nz": 4, "ext":
21 "jpg", "anno_path": "Crowds/groundtruth_rect.txt",
22        "object_class": "person"},
23        {"name": "Dancer", "path": "Dancer/img", "startFrame": 1, "endFrame": 225, "nz": 4, "ext":
24 "jpg", "anno_path": "Dancer/groundtruth_rect.txt",
25        "object_class": "person"},
26        {"name": "Dancer2", "path": "Dancer2/img", "startFrame": 1, "endFrame": 150, "nz": 4,
27 "ext": "jpg", "anno_path": "Dancer2/groundtruth_rect.txt",
28        "object_class": "person"},
29        {"name": "David", "path": "David/img", "startFrame": 300, "endFrame": 770, "nz": 4, "ext":
30 "jpg", "anno_path": "David/groundtruth_rect.txt",
31        "object_class": "face"},
32        {"name": "David2", "path": "David2/img", "startFrame": 1, "endFrame": 537, "nz": 4, "ext":
33 "jpg", "anno_path": "David2/groundtruth_rect.txt",
34        "object_class": "face"},
35        {"name": "David3", "path": "David3/img", "startFrame": 1, "endFrame": 252, "nz": 4, "ext":
36 "jpg", "anno_path": "David3/groundtruth_rect.txt",
37        "object_class": "person"},
38        {"name": "Deer", "path": "Deer/img", "startFrame": 1, "endFrame": 71, "nz": 4, "ext": "jpg",
39 "anno_path": "Deer/groundtruth_rect.txt",
40        "object_class": "mammal"},
41        {"name": "Diving", "path": "Diving/img", "startFrame": 1, "endFrame": 215, "nz": 4, "ext":
42 "jpg", "anno_path": "Diving/groundtruth_rect.txt",
43        "object_class": "person"},
44        {"name": "Dog", "path": "Dog/img", "startFrame": 1, "endFrame": 127, "nz": 4, "ext": "jpg",
45 "anno_path": "Dog/groundtruth_rect.txt",
46        "object_class": "dog"},
47        {"name": "Dog1", "path": "Dog1/img", "startFrame": 1, "endFrame": 1350, "nz": 4, "ext":
48 "jpg", "anno_path": "Dog1/groundtruth_rect.txt",
49        "object_class": "dog"},
50        {"name": "Doll", "path": "Doll/img", "startFrame": 1, "endFrame": 3872, "nz": 4, "ext": "jpg",

```



```

1  "anno_path": "Doll/groundtruth_rect.txt",
2      "object_class": "other"},
3      {"name": "DragonBaby", "path": "DragonBaby/img", "startFrame": 1, "endFrame": 113,
4  "nz": 4, "ext": "jpg", "anno_path": "DragonBaby/groundtruth_rect.txt",
5      "object_class": "face"},
6      {"name": "Dudek", "path": "Dudek/img", "startFrame": 1, "endFrame": 1145, "nz": 4, "ext":
7  "jpg", "anno_path": "Dudek/groundtruth_rect.txt",
8      "object_class": "face"},
9      {"name": "FaceOcc1", "path": "FaceOcc1/img", "startFrame": 1, "endFrame": 892, "nz": 4,
10 "ext": "jpg", "anno_path": "FaceOcc1/groundtruth_rect.txt",
11      "object_class": "face"},
12      {"name": "FaceOcc2", "path": "FaceOcc2/img", "startFrame": 1, "endFrame": 812, "nz": 4,
13 "ext": "jpg", "anno_path": "FaceOcc2/groundtruth_rect.txt",
14      "object_class": "face"},
15      {"name": "Fish", "path": "Fish/img", "startFrame": 1, "endFrame": 476, "nz": 4, "ext": "jpg",
16 "anno_path": "Fish/groundtruth_rect.txt",
17      "object_class": "other"},
18      {"name": "FleetFace", "path": "FleetFace/img", "startFrame": 1, "endFrame": 707, "nz": 4,
19 "ext": "jpg", "anno_path": "FleetFace/groundtruth_rect.txt",
20      "object_class": "face"},
21      {"name": "Football", "path": "Football/img", "startFrame": 1, "endFrame": 362, "nz": 4,
22 "ext": "jpg", "anno_path": "Football/groundtruth_rect.txt",
23      "object_class": "person head"},
24      {"name": "Football1", "path": "Football1/img", "startFrame": 1, "endFrame": 74, "nz": 4,
25 "ext": "jpg", "anno_path": "Football1/groundtruth_rect.txt",
26      "object_class": "face"},
27      {"name": "Freeman1", "path": "Freeman1/img", "startFrame": 1, "endFrame": 326, "nz":
28 4, "ext": "jpg", "anno_path": "Freeman1/groundtruth_rect.txt",
29      "object_class": "face"},
30      {"name": "Freeman3", "path": "Freeman3/img", "startFrame": 1, "endFrame": 460, "nz":
31 4, "ext": "jpg", "anno_path": "Freeman3/groundtruth_rect.txt",
32      "object_class": "face"},
33      {"name": "Freeman4", "path": "Freeman4/img", "startFrame": 1, "endFrame": 283, "nz":
34 4, "ext": "jpg", "anno_path": "Freeman4/groundtruth_rect.txt",
35      "object_class": "face"},
36      {"name": "Girl", "path": "Girl/img", "startFrame": 1, "endFrame": 500, "nz": 4, "ext": "jpg",
37 "anno_path": "Girl/groundtruth_rect.txt",
38      "object_class": "face"},
39      {"name": "Girl2", "path": "Girl2/img", "startFrame": 1, "endFrame": 1500, "nz": 4, "ext":
40 "jpg", "anno_path": "Girl2/groundtruth_rect.txt",
41      "object_class": "person"},
42      {"name": "Gym", "path": "Gym/img", "startFrame": 1, "endFrame": 767, "nz": 4, "ext":
43 "jpg", "anno_path": "Gym/groundtruth_rect.txt",
44      "object_class": "person"},
45      {"name": "Human2", "path": "Human2/img", "startFrame": 1, "endFrame": 1128, "nz": 4,
46 "ext": "jpg", "anno_path": "Human2/groundtruth_rect.txt",
47      "object_class": "person"},
48      {"name": "Human3", "path": "Human3/img", "startFrame": 1, "endFrame": 1698, "nz": 4,
49 "ext": "jpg", "anno_path": "Human3/groundtruth_rect.txt",
50      "object_class": "person"},

```

```

1      {"name": "Human4_2", "path": "Human4/img", "startFrame": 1, "endFrame": 667, "nz": 4,
2  "ext": "jpg", "anno_path": "Human4/groundtruth_rect.2.txt",
3      "object_class": "person"},
4      {"name": "Human4", "path": "Human4/img", "startFrame": 1, "endFrame": 667, "nz": 4,
5  "ext": "jpg", "anno_path": "Human4/groundtruth_rect.2.txt",
6      "object_class": "person"},
7      {"name": "Human5", "path": "Human5/img", "startFrame": 1, "endFrame": 713, "nz": 4,
8  "ext": "jpg", "anno_path": "Human5/groundtruth_rect.txt",
9      "object_class": "person"},
10     {"name": "Human6", "path": "Human6/img", "startFrame": 1, "endFrame": 792, "nz": 4,
11  "ext": "jpg", "anno_path": "Human6/groundtruth_rect.txt",
12     "object_class": "person"},
13     {"name": "Human7", "path": "Human7/img", "startFrame": 1, "endFrame": 250, "nz": 4,
14  "ext": "jpg", "anno_path": "Human7/groundtruth_rect.txt",
15     "object_class": "person"},
16     {"name": "Human8", "path": "Human8/img", "startFrame": 1, "endFrame": 128, "nz": 4,
17  "ext": "jpg", "anno_path": "Human8/groundtruth_rect.txt",
18     "object_class": "person"},
19     {"name": "Human9", "path": "Human9/img", "startFrame": 1, "endFrame": 305, "nz": 4,
20  "ext": "jpg", "anno_path": "Human9/groundtruth_rect.txt",
21     "object_class": "person"},
22     {"name": "Ironman", "path": "Ironman/img", "startFrame": 1, "endFrame": 166, "nz": 4,
23  "ext": "jpg", "anno_path": "Ironman/groundtruth_rect.txt",
24     "object_class": "person head"},
25     {"name": "Jogging", "path": "Jogging/img", "startFrame": 1, "endFrame": 307, "nz": 4,
26  "ext": "jpg", "anno_path": "Jogging/groundtruth_rect.1.txt",
27     "object_class": "person"},
28     {"name": "Jump", "path": "Jump/img", "startFrame": 1, "endFrame": 122, "nz": 4, "ext":
29  "jpg", "anno_path": "Jump/groundtruth_rect.txt",
30     "object_class": "person"},
31     {"name": "Jumping", "path": "Jumping/img", "startFrame": 1, "endFrame": 313, "nz": 4,
32  "ext": "jpg", "anno_path": "Jumping/groundtruth_rect.txt",
33     "object_class": "face"},
34     {"name": "KiteSurf", "path": "KiteSurf/img", "startFrame": 1, "endFrame": 84, "nz": 4, "ext":
35  "jpg", "anno_path": "KiteSurf/groundtruth_rect.txt",
36     "object_class": "face"},
37     {"name": "Lemming", "path": "Lemming/img", "startFrame": 1, "endFrame": 1336, "nz":
38  4, "ext": "jpg", "anno_path": "Lemming/groundtruth_rect.txt",
39     "object_class": "other"},
40     {"name": "Liquor", "path": "Liquor/img", "startFrame": 1, "endFrame": 1741, "nz": 4, "ext":
41  "jpg", "anno_path": "Liquor/groundtruth_rect.txt",
42     "object_class": "other"},
43     {"name": "Man", "path": "Man/img", "startFrame": 1, "endFrame": 134, "nz": 4, "ext": "jpg",
44  "anno_path": "Man/groundtruth_rect.txt",
45     "object_class": "face"},
46     {"name": "Matrix", "path": "Matrix/img", "startFrame": 1, "endFrame": 100, "nz": 4, "ext":
47  "jpg", "anno_path": "Matrix/groundtruth_rect.txt",
48     "object_class": "person head"},
49     {"name": "Mhyang", "path": "Mhyang/img", "startFrame": 1, "endFrame": 1490, "nz": 4,
50  "ext": "jpg", "anno_path": "Mhyang/groundtruth_rect.txt",

```

```

1         "object_class": "face"},
2         {"name": "MotorRolling", "path": "MotorRolling/img", "startFrame": 1, "endFrame": 164,
3 "nz": 4, "ext": "jpg", "anno_path": "MotorRolling/groundtruth_rect.txt",
4         "object_class": "vehicle"},
5         {"name": "MountainBike", "path": "MountainBike/img", "startFrame": 1, "endFrame": 228,
6 "nz": 4, "ext": "jpg", "anno_path": "MountainBike/groundtruth_rect.txt",
7         "object_class": "bicycle"},
8         {"name": "Panda", "path": "Panda/img", "startFrame": 1, "endFrame": 1000, "nz": 4, "ext":
9 "jpg", "anno_path": "Panda/groundtruth_rect.txt",
10        "object_class": "mammal"},
11        {"name": "RedTeam", "path": "RedTeam/img", "startFrame": 1, "endFrame": 1918, "nz":
12 4, "ext": "jpg", "anno_path": "RedTeam/groundtruth_rect.txt",
13        "object_class": "vehicle"},
14        {"name": "Rubik", "path": "Rubik/img", "startFrame": 1, "endFrame": 1997, "nz": 4, "ext":
15 "jpg", "anno_path": "Rubik/groundtruth_rect.txt",
16        "object_class": "other"},
17        {"name": "Shaking", "path": "Shaking/img", "startFrame": 1, "endFrame": 365, "nz": 4,
18 "ext": "jpg", "anno_path": "Shaking/groundtruth_rect.txt",
19        "object_class": "face"},
20        {"name": "Singer1", "path": "Singer1/img", "startFrame": 1, "endFrame": 351, "nz": 4,
21 "ext": "jpg", "anno_path": "Singer1/groundtruth_rect.txt",
22        "object_class": "person"},
23        {"name": "Singer2", "path": "Singer2/img", "startFrame": 1, "endFrame": 366, "nz": 4,
24 "ext": "jpg", "anno_path": "Singer2/groundtruth_rect.txt",
25        "object_class": "person"},
26        {"name": "Skater", "path": "Skater/img", "startFrame": 1, "endFrame": 160, "nz": 4, "ext":
27 "jpg", "anno_path": "Skater/groundtruth_rect.txt",
28        "object_class": "person"},
29        {"name": "Skater2", "path": "Skater2/img", "startFrame": 1, "endFrame": 435, "nz": 4, "ext":
30 "jpg", "anno_path": "Skater2/groundtruth_rect.txt",
31        "object_class": "person"},
32        {"name": "Skating1", "path": "Skating1/img", "startFrame": 1, "endFrame": 400, "nz": 4,
33 "ext": "jpg", "anno_path": "Skating1/groundtruth_rect.txt",
34        "object_class": "person"},
35        {"name": "Skating2", "path": "Skating2/img", "startFrame": 1, "endFrame": 473, "nz": 4,
36 "ext": "jpg", "anno_path": "Skating2/groundtruth_rect.1.txt",
37        "object_class": "person"},
38        {"name": "Skating2_1", "path": "Skating2/img", "startFrame": 1, "endFrame": 473, "nz": 4,
39 "ext": "jpg", "anno_path": "Skating2/groundtruth_rect.1.txt",
40        "object_class": "person"},
41        {"name": "Skating2_2", "path": "Skating2/img", "startFrame": 1, "endFrame": 473, "nz": 4,
42 "ext": "jpg", "anno_path": "Skating2/groundtruth_rect.2.txt",
43        "object_class": "person"},
44        {"name": "Skiing", "path": "Skiing/img", "startFrame": 1, "endFrame": 81, "nz": 4, "ext":
45 "jpg", "anno_path": "Skiing/groundtruth_rect.txt",
46        "object_class": "person"},
47        {"name": "Soccer", "path": "Soccer/img", "startFrame": 1, "endFrame": 392, "nz": 4, "ext":
48 "jpg", "anno_path": "Soccer/groundtruth_rect.txt",
49        "object_class": "face"},
50        {"name": "Subway", "path": "Subway/img", "startFrame": 1, "endFrame": 175, "nz": 4,

```

```

1  "ext": "jpg", "anno_path": "Subway/groundtruth_rect.txt",
2      "object_class": "person"},
3      {"name": "Surfer", "path": "Surfer/img", "startFrame": 1, "endFrame": 376, "nz": 4, "ext":
4  "jpg", "anno_path": "Surfer/groundtruth_rect.txt",
5      "object_class": "person head"},
6      {"name": "Suv", "path": "Suv/img", "startFrame": 1, "endFrame": 945, "nz": 4, "ext": "jpg",
7  "anno_path": "Suv/groundtruth_rect.txt",
8      "object_class": "car"},
9      {"name": "Sylvester", "path": "Sylvester/img", "startFrame": 1, "endFrame": 1345, "nz": 4,
10 "ext": "jpg", "anno_path": "Sylvester/groundtruth_rect.txt",
11      "object_class": "other"},
12      {"name": "Tiger1", "path": "Tiger1/img", "startFrame": 1, "endFrame": 354, "nz": 4, "ext":
13 "jpg", "anno_path": "Tiger1/groundtruth_rect.txt", "initOmit": 5,
14      "object_class": "other"},
15      {"name": "Tiger2", "path": "Tiger2/img", "startFrame": 1, "endFrame": 365, "nz": 4, "ext":
16 "jpg", "anno_path": "Tiger2/groundtruth_rect.txt",
17      "object_class": "other"},
18      {"name": "Toy", "path": "Toy/img", "startFrame": 1, "endFrame": 271, "nz": 4, "ext": "jpg",
19 "anno_path": "Toy/groundtruth_rect.txt",
20      "object_class": "other"},
21      {"name": "Trans", "path": "Trans/img", "startFrame": 1, "endFrame": 124, "nz": 4, "ext":
22 "jpg", "anno_path": "Trans/groundtruth_rect.txt",
23      "object_class": "other"},
24      {"name": "Trellis", "path": "Trellis/img", "startFrame": 1, "endFrame": 569, "nz": 4, "ext":
25 "jpg", "anno_path": "Trellis/groundtruth_rect.txt",
26      "object_class": "face"},
27      {"name": "Twinnings", "path": "Twinnings/img", "startFrame": 1, "endFrame": 472, "nz":
28 4, "ext": "jpg", "anno_path": "Twinnings/groundtruth_rect.txt",
29      "object_class": "other"},
30      {"name": "Vase", "path": "Vase/img", "startFrame": 1, "endFrame": 271, "nz": 4, "ext":
31 "jpg", "anno_path": "Vase/groundtruth_rect.txt",
32      "object_class": "other"},
33      {"name": "Walking", "path": "Walking/img", "startFrame": 1, "endFrame": 412, "nz": 4,
34 "ext": "jpg", "anno_path": "Walking/groundtruth_rect.txt",
35      "object_class": "person"},
36      {"name": "Walking2", "path": "Walking2/img", "startFrame": 1, "endFrame": 500, "nz": 4,
37 "ext": "jpg", "anno_path": "Walking2/groundtruth_rect.txt",
38      "object_class": "person"},
39      {"name": "Woman", "path": "Woman/img", "startFrame": 1, "endFrame": 597, "nz": 4,
40 "ext": "jpg", "anno_path": "Woman/groundtruth_rect.txt",
41      "object_class": "person"}
42  ]
43  return sequence_info_list
44  import numpy as np
45  from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
46  from lib.test.utils.load_text import load_text
47  class NFSDataset(BaseDataset):
48      def __init__(self):
49          super().__init__()
50          self.base_path = self.env_settings.nfs_path
    
```

```

1         self.sequence_info_list = self._get_sequence_info_list()
2     def get_sequence_list(self):
3         return SequenceList([self._construct_sequence(s) for s in self.sequence_info_list])
4     def _construct_sequence(self, sequence_info):
5         sequence_path = sequence_info['path']
6         nz = sequence_info['nz']
7         ext = sequence_info['ext']
8         start_frame = sequence_info['startFrame']
9         end_frame = sequence_info['endFrame']
10        init_omit = 0
11        if 'initOmit' in sequence_info:
12            init_omit = sequence_info['initOmit']
13        frames =
14        ['{base_path}/{sequence_path}/{frame:0{nz}}.{ext}'.format(base_path=self.base_path,
15            sequence_path=sequence_path, frame=frame_num, nz=nz, ext=ext) for frame_num in
16        range(start_frame+init_omit, end_frame+1)]
17        anno_path = f"{self.base_path}/{sequence_info['name'][:4]}/30/groundtruth.txt"
18        ground_truth_rect = load_text(str(anno_path), delimiter=',', dtype=np.float64)
19        return Sequence(sequence_info['name'][:4:], frames, 'nfs', ground_truth_rect[init_omit:],
20            object_class=sequence_info['object_class'])
21    def __len__(self):
22        return len(self.sequence_info_list)
23    def _get_sequence_info_list(self):
24        sequence_info_list = [
25            {"name": "nfs_Gymnastics", "path": "sequences/Gymnastics", "startFrame": 1,
26            "endFrame": 368, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_Gymnastics.txt", "object_class": "person",
27            'occlusion': False},
28            {"name": "nfs_MachLoop_jet", "path": "sequences/MachLoop_jet", "startFrame": 1,
29            "endFrame": 99, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_MachLoop_jet.txt", "object_class":
30            "aircraft", 'occlusion': False},
31            {"name": "nfs_Skiing_red", "path": "sequences/Skiing_red", "startFrame": 1, "endFrame":
32            69, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_Skiing_red.txt", "object_class": "person", 'occlusion':
33            False},
34            {"name": "nfs_Skydiving", "path": "sequences/Skydiving", "startFrame": 1, "endFrame":
35            196, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_Skydiving.txt", "object_class": "person", 'occlusion':
36            True},
37            {"name": "nfs_airboard_1", "path": "sequences/airboard_1", "startFrame": 1, "endFrame":
38            425, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_airboard_1.txt", "object_class": "ball", 'occlusion':
39            False},
40            {"name": "nfs_airplane_landing", "path": "sequences/airplane_landing", "startFrame": 1,
41            "endFrame": 81, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_airplane_landing.txt", "object_class":
42            "aircraft", 'occlusion': False},
43            {"name": "nfs_airtable_3", "path": "sequences/airtable_3", "startFrame": 1, "endFrame":
44            482, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_airtable_3.txt", "object_class": "ball", 'occlusion': False},
45            {"name": "nfs_basketball_1", "path": "sequences/basketball_1", "startFrame": 1,
46            "endFrame": 282, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_1.txt", "object_class": "ball",
47            'occlusion': False},
48            {"name": "nfs_basketball_2", "path": "sequences/basketball_2", "startFrame": 1,
49            "endFrame": 102, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_2.txt", "object_class": "ball",
50            'occlusion': False},

```

```

1         {"name": "nfs_basketball_3", "path": "sequences/basketball_3", "startFrame": 1,
2 "endFrame": 421, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_3.txt", "object_class": "ball",
3 'occlusion': False},
4         {"name": "nfs_basketball_6", "path": "sequences/basketball_6", "startFrame": 1,
5 "endFrame": 224, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_6.txt", "object_class": "ball",
6 'occlusion': False},
7         {"name": "nfs_basketball_7", "path": "sequences/basketball_7", "startFrame": 1,
8 "endFrame": 240, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_7.txt", "object_class":
9 "person", 'occlusion': True},
10        {"name": "nfs_basketball_player", "path": "sequences/basketball_player", "startFrame": 1,
11 "endFrame": 369, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_player.txt", "object_class":
12 "person", 'occlusion': True},
13        {"name": "nfs_basketball_player_2", "path": "sequences/basketball_player_2",
14 "startFrame": 1, "endFrame": 437, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_basketball_player_2.txt",
15 "object_class": "person", 'occlusion': False},
16        {"name": "nfs_beach_flipback_person", "path": "sequences/beach_flipback_person",
17 "startFrame": 1, "endFrame": 61, "nz": 5, "ext": "jpg", "anno_path":
18 "anno/nfs_beach_flipback_person.txt", "object_class": "person head", 'occlusion': False},
19        {"name": "nfs_bee", "path": "sequences/bee", "startFrame": 1, "endFrame": 45, "nz": 5,
20 "ext": "jpg", "anno_path": "anno/nfs_bee.txt", "object_class": "insect", 'occlusion': False},
21        {"name": "nfs_biker_acrobat", "path": "sequences/biker_acrobat", "startFrame": 1,
22 "endFrame": 128, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_biker_acrobat.txt", "object_class":
23 "bicycle", 'occlusion': False},
24        {"name": "nfs_biker_all_1", "path": "sequences/biker_all_1", "startFrame": 1, "endFrame":
25 113, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_biker_all_1.txt", "object_class": "person", 'occlusion':
26 False},
27        {"name": "nfs_biker_head_2", "path": "sequences/biker_head_2", "startFrame": 1,
28 "endFrame": 132, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_biker_head_2.txt", "object_class":
29 "person head", 'occlusion': False},
30        {"name": "nfs_biker_head_3", "path": "sequences/biker_head_3", "startFrame": 1,
31 "endFrame": 254, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_biker_head_3.txt", "object_class":
32 "person head", 'occlusion': False},
33        {"name": "nfs_biker_upper_body", "path": "sequences/biker_upper_body", "startFrame":
34 1, "endFrame": 194, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_biker_upper_body.txt", "object_class":
35 "person", 'occlusion': False},
36        {"name": "nfs_biker_whole_body", "path": "sequences/biker_whole_body", "startFrame":
37 1, "endFrame": 572, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_biker_whole_body.txt", "object_class":
38 "person", 'occlusion': True},
39        {"name": "nfs_billiard_2", "path": "sequences/billiard_2", "startFrame": 1, "endFrame":
40 604, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_billiard_2.txt", "object_class": "ball", 'occlusion': False},
41        {"name": "nfs_billiard_3", "path": "sequences/billiard_3", "startFrame": 1, "endFrame":
42 698, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_billiard_3.txt", "object_class": "ball", 'occlusion': False},
43        {"name": "nfs_billiard_6", "path": "sequences/billiard_6", "startFrame": 1, "endFrame":
44 771, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_billiard_6.txt", "object_class": "ball", 'occlusion': False},
45        {"name": "nfs_billiard_7", "path": "sequences/billiard_7", "startFrame": 1, "endFrame":
46 724, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_billiard_7.txt", "object_class": "ball", 'occlusion': False},
47        {"name": "nfs_billiard_8", "path": "sequences/billiard_8", "startFrame": 1, "endFrame":
48 778, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_billiard_8.txt", "object_class": "ball", 'occlusion': False},
49        {"name": "nfs_bird_2", "path": "sequences/bird_2", "startFrame": 1, "endFrame": 476, "nz":
50 5, "ext": "jpg", "anno_path": "anno/nfs_bird_2.txt", "object_class": "bird", 'occlusion': False},
    
```

```

1         {"name": "nfs_book", "path": "sequences/book", "startFrame": 1, "endFrame": 288, "nz":
2 5, "ext": "jpg", "anno_path": "anno/nfs_book.txt", "object_class": "other", 'occlusion': False},
3         {"name": "nfs_bottle", "path": "sequences/bottle", "startFrame": 1, "endFrame": 2103,
4 "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bottle.txt", "object_class": "other", 'occlusion': False},
5         {"name": "nfs_bowling_1", "path": "sequences/bowling_1", "startFrame": 1, "endFrame":
6 303, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bowling_1.txt", "object_class": "ball", 'occlusion': True},
7         {"name": "nfs_bowling_2", "path": "sequences/bowling_2", "startFrame": 1, "endFrame":
8 710, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bowling_2.txt", "object_class": "ball", 'occlusion': True},
9         {"name": "nfs_bowling_3", "path": "sequences/bowling_3", "startFrame": 1, "endFrame":
10 271, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bowling_3.txt", "object_class": "ball", 'occlusion': True},
11         {"name": "nfs_bowling_6", "path": "sequences/bowling_6", "startFrame": 1, "endFrame":
12 260, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bowling_6.txt", "object_class": "ball", 'occlusion': False},
13         {"name": "nfs_bowling_ball", "path": "sequences/bowling_ball", "startFrame": 1,
14 "endFrame": 275, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bowling_ball.txt", "object_class": "ball",
15 'occlusion': True},
16         {"name": "nfs_bunny", "path": "sequences/bunny", "startFrame": 1, "endFrame": 705,
17 "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_bunny.txt", "object_class": "mammal", 'occlusion': False},
18         {"name": "nfs_car", "path": "sequences/car", "startFrame": 1, "endFrame": 2020, "nz": 5,
19 "ext": "jpg", "anno_path": "anno/nfs_car.txt", "object_class": "car", 'occlusion': True},
20         {"name": "nfs_car_camaro", "path": "sequences/car_camaro", "startFrame": 1,
21 "endFrame": 36, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_camaro.txt", "object_class": "car",
22 'occlusion': False},
23         {"name": "nfs_car_drifting", "path": "sequences/car_drifting", "startFrame": 1,
24 "endFrame": 173, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_drifting.txt", "object_class": "car",
25 'occlusion': False},
26         {"name": "nfs_car_jumping", "path": "sequences/car_jumping", "startFrame": 1,
27 "endFrame": 22, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_jumping.txt", "object_class": "car",
28 'occlusion': False},
29         {"name": "nfs_car_rc_rolling", "path": "sequences/car_rc_rolling", "startFrame": 1,
30 "endFrame": 62, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_rc_rolling.txt", "object_class": "car",
31 'occlusion': False},
32         {"name": "nfs_car_rc_rotating", "path": "sequences/car_rc_rotating", "startFrame": 1,
33 "endFrame": 80, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_rc_rotating.txt", "object_class": "car",
34 'occlusion': False},
35         {"name": "nfs_car_side", "path": "sequences/car_side", "startFrame": 1, "endFrame": 108,
36 "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_side.txt", "object_class": "car", 'occlusion': False},
37         {"name": "nfs_car_white", "path": "sequences/car_white", "startFrame": 1, "endFrame":
38 2063, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_car_white.txt", "object_class": "car", 'occlusion': False},
39         {"name": "nfs_cheetah", "path": "sequences/cheetah", "startFrame": 1, "endFrame": 167,
40 "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_cheetah.txt", "object_class": "mammal", 'occlusion': True},
41         {"name": "nfs_cup", "path": "sequences/cup", "startFrame": 1, "endFrame": 1281, "nz": 5,
42 "ext": "jpg", "anno_path": "anno/nfs_cup.txt", "object_class": "other", 'occlusion': False},
43         {"name": "nfs_cup_2", "path": "sequences/cup_2", "startFrame": 1, "endFrame": 182, "nz":
44 5, "ext": "jpg", "anno_path": "anno/nfs_cup_2.txt", "object_class": "other", 'occlusion': False},
45         {"name": "nfs_dog", "path": "sequences/dog", "startFrame": 1, "endFrame": 1030, "nz": 5,
46 "ext": "jpg", "anno_path": "anno/nfs_dog.txt", "object_class": "dog", 'occlusion': True},
47         {"name": "nfs_dog_1", "path": "sequences/dog_1", "startFrame": 1, "endFrame": 168, "nz":
48 5, "ext": "jpg", "anno_path": "anno/nfs_dog_1.txt", "object_class": "dog", 'occlusion': False},
49         {"name": "nfs_dog_3", "path": "sequences/dog_3", "startFrame": 1, "endFrame": 200, "nz":
50 5, "ext": "jpg", "anno_path": "anno/nfs_dog_3.txt", "object_class": "dog", 'occlusion': False},
    
```

```

1         {"name": "nfs_dogs", "path": "sequences/dogs", "startFrame": 1, "endFrame": 198, "nz":
2 5, "ext": "jpg", "anno_path": "anno/nfs_dogs.txt", "object_class": "dog", 'occlusion': True},
3         {"name": "nfs_dollar", "path": "sequences/dollar", "startFrame": 1, "endFrame": 1426,
4 "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_dollar.txt", "object_class": "other", 'occlusion': False},
5         {"name": "nfs_drone", "path": "sequences/drone", "startFrame": 1, "endFrame": 70, "nz":
6 5, "ext": "jpg", "anno_path": "anno/nfs_drone.txt", "object_class": "aircraft", 'occlusion': False},
7         {"name": "nfs_ducks_lake", "path": "sequences/ducks_lake", "startFrame": 1, "endFrame":
8 107, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_ducks_lake.txt", "object_class": "bird", 'occlusion':
9 False},
10        {"name": "nfs_exit", "path": "sequences/exit", "startFrame": 1, "endFrame": 359, "nz": 5,
11 "ext": "jpg", "anno_path": "anno/nfs_exit.txt", "object_class": "other", 'occlusion': False},
12        {"name": "nfs_first", "path": "sequences/first", "startFrame": 1, "endFrame": 435, "nz": 5,
13 "ext": "jpg", "anno_path": "anno/nfs_first.txt", "object_class": "other", 'occlusion': False},
14        {"name": "nfs_flower", "path": "sequences/flower", "startFrame": 1, "endFrame": 448, "nz":
15 5, "ext": "jpg", "anno_path": "anno/nfs_flower.txt", "object_class": "other", 'occlusion': False},
16        {"name": "nfs_footbal_skill", "path": "sequences/footbal_skill", "startFrame": 1,
17 "endFrame": 131, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_footbal_skill.txt", "object_class": "ball",
18 'occlusion': True},
19        {"name": "nfs_helicopter", "path": "sequences/helicopter", "startFrame": 1, "endFrame":
20 310, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_helicopter.txt", "object_class": "aircraft", 'occlusion':
21 False},
22        {"name": "nfs_horse_jumping", "path": "sequences/horse_jumping", "startFrame": 1,
23 "endFrame": 117, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_horse_jumping.txt", "object_class":
24 "horse", 'occlusion': True},
25        {"name": "nfs_horse_running", "path": "sequences/horse_running", "startFrame": 1,
26 "endFrame": 139, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_horse_running.txt", "object_class":
27 "horse", 'occlusion': False},
28        {"name": "nfs_iceskating_6", "path": "sequences/iceskating_6", "startFrame": 1,
29 "endFrame": 603, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_iceskating_6.txt", "object_class": "person",
30 'occlusion': False},
31        {"name": "nfs_jellyfish_5", "path": "sequences/jellyfish_5", "startFrame": 1, "endFrame":
32 746, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_jellyfish_5.txt", "object_class": "invertebrate",
33 'occlusion': False},
34        {"name": "nfs_kid_swing", "path": "sequences/kid_swing", "startFrame": 1, "endFrame":
35 169, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_kid_swing.txt", "object_class": "person", 'occlusion':
36 False},
37        {"name": "nfs_motorcross", "path": "sequences/motorcross", "startFrame": 1,
38 "endFrame": 39, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_motorcross.txt", "object_class": "vehicle",
39 'occlusion': True},
40        {"name": "nfs_motorcross_kawasaki", "path": "sequences/motorcross_kawasaki",
41 "startFrame": 1, "endFrame": 65, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_motorcross_kawasaki.txt",
42 "object_class": "vehicle", 'occlusion': False},
43        {"name": "nfs_parkour", "path": "sequences/parkour", "startFrame": 1, "endFrame": 58,
44 "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_parkour.txt", "object_class": "person head", 'occlusion':
45 False},
46        {"name": "nfs_person_scooter", "path": "sequences/person_scooter", "startFrame": 1,
47 "endFrame": 413, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_person_scooter.txt", "object_class":
48 "person", 'occlusion': True},
49        {"name": "nfs_pingpong_2", "path": "sequences/pingpong_2", "startFrame": 1,
50 "endFrame": 1277, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_pingpong_2.txt", "object_class": "ball",
    
```



```

1  'occlusion': False},
2      {"name": "nfs_pingpong_7", "path": "sequences/pingpong_7", "startFrame": 1,
3  "endFrame": 1290, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_pingpong_7.txt", "object_class": "ball",
4  'occlusion': False},
5      {"name": "nfs_pingpong_8", "path": "sequences/pingpong_8", "startFrame": 1,
6  "endFrame": 296, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_pingpong_8.txt", "object_class": "ball",
7  'occlusion': False},
8      {"name": "nfs_purse", "path": "sequences/purse", "startFrame": 1, "endFrame": 968, "nz":
9  5, "ext": "jpg", "anno_path": "anno/nfs_purse.txt", "object_class": "other", 'occlusion': False},
10     {"name": "nfs_rubber", "path": "sequences/rubber", "startFrame": 1, "endFrame": 1328,
11 "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_rubber.txt", "object_class": "other", 'occlusion': False},
12     {"name": "nfs_running", "path": "sequences/running", "startFrame": 1, "endFrame": 677,
13 "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_running.txt", "object_class": "person", 'occlusion': False},
14     {"name": "nfs_running_100_m", "path": "sequences/running_100_m", "startFrame": 1,
15 "endFrame": 313, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_running_100_m.txt", "object_class":
16 "person", 'occlusion': True},
17     {"name": "nfs_running_100_m_2", "path": "sequences/running_100_m_2", "startFrame":
18 1, "endFrame": 337, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_running_100_m_2.txt", "object_class":
19 "person", 'occlusion': True},
20     {"name": "nfs_running_2", "path": "sequences/running_2", "startFrame": 1, "endFrame":
21 363, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_running_2.txt", "object_class": "person", 'occlusion':
22 False},
23     {"name": "nfs_shuffleboard_1", "path": "sequences/shuffleboard_1", "startFrame": 1,
24 "endFrame": 42, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffleboard_1.txt", "object_class": "other",
25 'occlusion': False},
26     {"name": "nfs_shuffleboard_2", "path": "sequences/shuffleboard_2", "startFrame": 1,
27 "endFrame": 41, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffleboard_2.txt", "object_class": "other",
28 'occlusion': False},
29     {"name": "nfs_shuffleboard_4", "path": "sequences/shuffleboard_4", "startFrame": 1,
30 "endFrame": 62, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffleboard_4.txt", "object_class": "other",
31 'occlusion': False},
32     {"name": "nfs_shuffleboard_5", "path": "sequences/shuffleboard_5", "startFrame": 1,
33 "endFrame": 32, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffleboard_5.txt", "object_class": "other",
34 'occlusion': False},
35     {"name": "nfs_shuffleboard_6", "path": "sequences/shuffleboard_6", "startFrame": 1,
36 "endFrame": 52, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffleboard_6.txt", "object_class": "other",
37 'occlusion': False},
38     {"name": "nfs_shuffletable_2", "path": "sequences/shuffletable_2", "startFrame": 1,
39 "endFrame": 372, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffletable_2.txt", "object_class":
40 "other", 'occlusion': False},
41     {"name": "nfs_shuffletable_3", "path": "sequences/shuffletable_3", "startFrame": 1,
42 "endFrame": 368, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffletable_3.txt", "object_class":
43 "other", 'occlusion': False},
44     {"name": "nfs_shuffletable_4", "path": "sequences/shuffletable_4", "startFrame": 1,
45 "endFrame": 101, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_shuffletable_4.txt", "object_class":
46 "other", 'occlusion': False},
47     {"name": "nfs_ski_long", "path": "sequences/ski_long", "startFrame": 1, "endFrame": 274,
48 "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_ski_long.txt", "object_class": "person", 'occlusion': False},
49     {"name": "nfs_soccer_ball", "path": "sequences/soccer_ball", "startFrame": 1, "endFrame":
50 163, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_soccer_ball.txt", "object_class": "ball", 'occlusion':

```

```

1 False},
2         {"name": "nfs_soccer_ball_2", "path": "sequences/soccer_ball_2", "startFrame": 1,
3 "endFrame": 1934, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_soccer_ball_2.txt", "object_class": "ball",
4 'occlusion': False},
5         {"name": "nfs_soccer_ball_3", "path": "sequences/soccer_ball_3", "startFrame": 1,
6 "endFrame": 1381, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_soccer_ball_3.txt", "object_class": "ball",
7 'occlusion': False},
8         {"name": "nfs_soccer_player_2", "path": "sequences/soccer_player_2", "startFrame": 1,
9 "endFrame": 475, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_soccer_player_2.txt", "object_class":
10 "person", 'occlusion': False},
11         {"name": "nfs_soccer_player_3", "path": "sequences/soccer_player_3", "startFrame": 1,
12 "endFrame": 319, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_soccer_player_3.txt", "object_class":
13 "person", 'occlusion': True},
14         {"name": "nfs_stop_sign", "path": "sequences/stop_sign", "startFrame": 1, "endFrame":
15 302, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_stop_sign.txt", "object_class": "other", 'occlusion':
16 False},
17         {"name": "nfs_suv", "path": "sequences/suv", "startFrame": 1, "endFrame": 2584, "nz": 5,
18 "ext": "jpg", "anno_path": "anno/nfs_suv.txt", "object_class": "car", 'occlusion': False},
19         {"name": "nfs_tiger", "path": "sequences/tiger", "startFrame": 1, "endFrame": 1556, "nz":
20 5, "ext": "jpg", "anno_path": "anno/nfs_tiger.txt", "object_class": "mammal", 'occlusion': False},
21         {"name": "nfs_walking", "path": "sequences/walking", "startFrame": 1, "endFrame": 555,
22 "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_walking.txt", "object_class": "person", 'occlusion': False},
23         {"name": "nfs_walking_3", "path": "sequences/walking_3", "startFrame": 1, "endFrame":
24 1427, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_walking_3.txt", "object_class": "person", 'occlusion':
25 False},
26         {"name": "nfs_water_ski_2", "path": "sequences/water_ski_2", "startFrame": 1,
27 "endFrame": 47, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_water_ski_2.txt", "object_class": "person",
28 'occlusion': False},
29         {"name": "nfs_yoyo", "path": "sequences/yoyo", "startFrame": 1, "endFrame": 67, "nz": 5,
30 "ext": "jpg", "anno_path": "anno/nfs_yoyo.txt", "object_class": "other", 'occlusion': False},
31         {"name": "nfs_zebra_fish", "path": "sequences/zebra_fish", "startFrame": 1, "endFrame":
32 671, "nz": 5, "ext": "jpg", "anno_path": "anno/nfs_zebra_fish.txt", "object_class": "fish", 'occlusion': False},
33     ]
34     return sequence_info_list
35 from lib.test.evaluation.environment import EnvSettings
36 from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
37 from lib.utils.lmdb_utils import *
38 class LaSOTImdbDataset(BaseDataset):
39     def __init__(self):
40         super().__init__()
41         self.base_path = self.env_settings.lasot_lmdb_path
42         self.sequence_list = self._get_sequence_list()
43         self.clean_list = self.clean_seq_list()
44     def clean_seq_list(self):
45         clean_lst = []
46         for i in range(len(self.sequence_list)):
47             cls, _ = self.sequence_list[i].split('-')
48             clean_lst.append(cls)
49         return clean_lst
50     def get_sequence_list(self):

```

```

1         return SequenceList([self._construct_sequence(s) for s in self.sequence_list])
2     def __len__(self):
3         return len(self.sequence_list)
4     def _get_sequence_list(self):
5 import numpy as np
6 from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
7 from lib.test.utils.load_text import load_text
8 class LaSOTExtensionSubsetDataset(BaseDataset):
9     def __init__(self):
10         super().__init__()
11         self.base_path = self.env_settings.lasot_extension_subset_path
12         self.sequence_list = self._get_sequence_list()
13         self.clean_list = self.clean_seq_list()
14     def clean_seq_list(self):
15         clean_lst = []
16         for i in range(len(self.sequence_list)):
17             cls, _ = self.sequence_list[i].split('-')
18             clean_lst.append(cls)
19         return clean_lst
20     def get_sequence_list(self):
21         return SequenceList([self._construct_sequence(s) for s in self.sequence_list])
22     def _construct_sequence(self, sequence_name):
23         class_name = sequence_name.split('-')[0]
24         anno_path = '{}/{}/{}/groundtruth.txt'.format(self.base_path, class_name, sequence_name)
25         ground_truth_rect = load_text(str(anno_path), delimiter=',', dtype=np.float64)
26         occlusion_label_path = '{}/{}/{}/full_occlusion.txt'.format(self.base_path, class_name,
27 sequence_name)
28         full_occlusion = load_text(str(occlusion_label_path), delimiter=',', dtype=np.float64,
29 backend='numpy')
30         out_of_view_label_path = '{}/{}/{}/out_of_view.txt'.format(self.base_path, class_name,
31 sequence_name)
32         out_of_view = load_text(str(out_of_view_label_path), delimiter=',', dtype=np.float64,
33 backend='numpy')
34         target_visible = np.logical_and(full_occlusion == 0, out_of_view == 0)
35         frames_path = '{}/{}/{}/img'.format(self.base_path, class_name, sequence_name)
36         frames_list = ['{}/{:08d}.jpg'.format(frames_path, frame_number) for frame_number in
37 range(1, ground_truth_rect.shape[0] + 1)]
38         target_class = class_name
39         return Sequence(sequence_name, frames_list, 'lasot_extension_subset',
40 ground_truth_rect.reshape(-1, 4),
41 object_class=target_class, target_visible=target_visible)
42     def __len__(self):
43         return len(self.sequence_list)
44     def _get_sequence_list(self):
45 return sequence_list
46 import numpy as np
47 from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
48 from lib.test.utils.load_text import load_text
49 class LaSOTDataset(BaseDataset):
50     def __init__(self):

```

```

1         super().__init__()
2         self.base_path = self.env_settings.lasot_path
3         self.sequence_list = self._get_sequence_list()
4         self.clean_list = self.clean_seq_list()
5     def clean_seq_list(self):
6         clean_lst = []
7         for i in range(len(self.sequence_list)):
8             cls, _ = self.sequence_list[i].split('-')
9             clean_lst.append(cls)
10        return clean_lst
11    def get_sequence_list(self):
12        return SequenceList([self._construct_sequence(s) for s in self.sequence_list])
13    def __len__(self):
14        return len(self.sequence_list)
15    import numpy as np
16    from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList
17    from lib.test.utils.load_text import load_text
18    import os
19    class ITBDataset(BaseDataset):
20        def __len__(self):
21            return len(self.sequence_info_list)
22        def get_fileNames(self, rootdir):
23            fs = []
24            fs_all = []
25            for root, dirs, files in os.walk(rootdir, topdown=True):
26                files.sort()
27                files.sort(key=len)
28                if files is not None:
29                    for name in files:
30                        _, ending = os.path.splitext(name)
31                        if ending == ".jpg":
32                            _, root_ = os.path.split(root)
33                            fs.append(os.path.join(root_, name))
34                            fs_all.append(os.path.join(root, name))
35            return fs_all, fs
36        def _get_sequence_info_list(self, base_path):
37            sequence_info_list = []
38            for scene in os.listdir(base_path):
39                if '.' in scene:
40                    continue
41                videos = os.listdir(os.path.join(base_path, scene))
42                for video in videos:
43                    _, fs = self.get_fileNames(os.path.join(base_path, scene, video))
44                    video_tmp = {"name": video, "path": scene + '/' + video, "startFrame": 1,
45                                "endFrame": len(fs),
46                                "nz": len(fs[0].split('/')[0].split('.')[0]), "ext": ".jpg",
47                                "anno_path": scene + '/' + video + "/groundtruth.txt",
48                                "object_class": "unknown"}
49                    sequence_info_list.append(video_tmp)
50            return sequence_info_list

```