declare module '\*.vue' {

import type { DefineComponent } from 'vue'

const component: DefineComponent<{}, {}, any>

export default component

}

<!DOCTYPE html>

<html lang="">

<head>

<meta charset="UTF-8">

<link rel="icon" href="/favicon.ico">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Vite App</title>

</head>

<body>

<div id="app"></div>

<script type="module" src="/src/main.ts"></script>

</body>

</html>

import { fileURLToPath, URL } from 'node:url'

import { defineConfig } from 'vite'

import vue from '@vitejs/plugin-vue'

import vueDevTools from 'vite-plugin-vue-devtools'

export default defineConfig({

plugins: [

vue(),

vueDevTools(),

],

resolve: {

alias: {

'@': fileURLToPath(new URL('./src', import.meta.url))

},

},

})

import { createRouter, createWebHistory } from 'vue-router';

import FileUploadVue from '@/components/FileUpload.vue';

const router = createRouter({

history: createWebHistory(import.meta.env.BASE\_URL),

routes: [

{

path: '/',

name: 'home',

component: FileUploadVue,

},

],

});

router.beforeEach((to, from, next) => {

next();

});

export default router;

import './assets/main.css'

import { createApp } from 'vue'

import App from './App.vue'

import router from './router'

const app = createApp(App)

app.use(router)

app.mount('#app')

<!-- This icon is from <https:

<template>

<svg

xmlns="http://www.w3.org/2000/svg"

xmlns:xlink="http://www.w3.org/1999/xlink"

aria-hidden="true"

role="img"

class="iconify iconify--mdi"

width="24"

height="24"

preserveAspectRatio="xMidYMid meet"

viewBox="0 0 24 24"

>

<path

d="M20 18v-4h-3v1h-2v-1H9v1H7v-1H4v4h16M6.33 8l-1.74 4H7v-1h2v1h6v-1h2v1h2.41l-1.74-4H6.33M9 5v1h6V5H9m12.84 7.61c.1.22.16.48.16.8V18c0 .53-.21 1-.6 1.41c-.4.4-.85.59-1.4.59H4c-.55 0-1-.19-1.4-.59C2.21 19 2 18.53 2 18v-4.59c0-.32.06-.58.16-.8L4.5 7.22C4.84 6.41 5.45 6 6.33 6H7V5c0-.55.18-1 .57-1.41C7.96 3.2 8.44 3 9 3h6c.56 0 1.04.2 1.43.59c.39.41.57.86.57 1.41v1h.67c.88 0 1.49.41 1.83 1.22l2.34 5.39z"

fill="currentColor"

></path>

</svg>

</template>

<template>

<svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="currentColor">

<path

d="M10 3.22l-.61-.6a5.5 5.5 0 0 0-7.666.105 5.5 5.5 0 0 0-.114 7.665L10 18.78l8.39-8.4a5.5 5.5 0 0 0-.114-7.665 5.5 5.5 0 0 0-7.666-.105l-.61.61z"

/>

</svg>

</template>

<template>

<svg xmlns="http://www.w3.org/2000/svg" width="18" height="20" fill="currentColor">

<path

d="M11.447 8.894a1 1 0 1 0-.894-1.789l.894 1.789zm-2.894-.789a1 1 0 1 0 .894 1.789l-.894-1.789zm0 1.789a1 1 0 1 0 .894-1.789l-.894 1.789zM7.447 7.106a1 1 0 1 0-.894 1.789l.894-1.789zM10 9a1 1 0 1 0-2 0h2zm-2 2.5a1 1 0 1 0 2 0H8zm9.447-5.606a1 1 0 1 0-.894-1.789l.894 1.789zm-2.894-.789a1 1 0 1 0 .894 1.789l-.894-1.789zm2 .789a1 1 0 1 0 .894-1.789l-.894 1.789zm-1.106-2.789a1 1 0 1 0-.894 1.789l.894-1.789zM18 5a1 1 0 1 0-2 0h2zm-2 2.5a1 1 0 1 0 2 0h-2zm-5.447-4.606a1 1 0 1 0 .894-1.789l-.894 1.789zM9 1l.447-.894a1 1 0 0 0-.894 0L9 1zm-2.447.106a1 1 0 1 0 .894 1.789l-.894-1.789zm-6 3a1 1 0 1 0 .894 1.789L.553 4.106zm2.894.789a1 1 0 1 0-.894-1.789l.894 1.789zm-2-.789a1 1 0 1 0-.894 1.789l.894-1.789zm1.106 2.789a1 1 0 1 0 .894-1.789l-.894 1.789zM2 5a1 1 0 1 0-2 0h2zM0 7.5a1 1 0 1 0 2 0H0zm8.553 12.394a1 1 0 1 0 .894-1.789l-.894 1.789zm-1.106-2.789a1 1 0 1 0-.894 1.789l.894-1.789zm1.106 1a1 1 0 1 0 .894 1.789l-.894-1.789zm2.894.789a1 1 0 1 0-.894-1.789l.894 1.789zM8 19a1 1 0 1 0 2 0H8zm2-2.5a1 1 0 1 0-2 0h2zm-7.447.394a1 1 0 1 0 .894-1.789l-.894 1.789zM1 15H0a1 1 0 0 0 .553.894L1 15zm1-2.5a1 1 0 1 0-2 0h2zm12.553 2.606a1 1 0 1 0 .894 1.789l-.894-1.789zM17 15l.447.894A1 1 0 0 0 18 15h-1zm1-2.5a1 1 0 1 0-2 0h2zm-7.447-5.394l-2 1 .894 1.789 2-1-.894-1.789zm-1.106 1l-2-1-.894 1.789 2 1 .894-1.789zM8 9v2.5h2V9H8zm8.553-4.894l-2 1 .894 1.789 2-1-.894-1.789zm.894 0l-2-1-.894 1.789 2 1 .894-1.789zM16 5v2.5h2V5h-2zm-4.553-3.894l-2-1-.894 1.789 2 1 .894-1.789zm-2.894-1l-2 1 .894 1.789 2-1L8.553.106zM1.447 5.894l2-1-.894-1.789-2 1 .894 1.789zm-.894 0l2 1 .894-1.789-2-1-.894 1.789zM0 5v2.5h2V5H0zm9.447 13.106l-2-1-.894 1.789 2 1 .894-1.789zm0 1.789l2-1-.894-1.789-2 1 .894 1.789zM10 19v-2.5H8V19h2zm-6.553-3.894l-2-1-.894 1.789 2 1 .894-1.789zM2 15v-2.5H0V15h2zm13.447 1.894l2-1-.894-1.789-2 1 .894 1.789zM18 15v-2.5h-2V15h2z"

/>

</svg>

</template>

<template>

<svg xmlns="http://www.w3.org/2000/svg" width="20" height="17" fill="currentColor">

<path

d="M11 2.253a1 1 0 1 0-2 0h2zm-2 13a1 1 0 1 0 2 0H9zm.447-12.167a1 1 0 1 0 1.107-1.666L9.447 3.086zM1 2.253L.447 1.42A1 1 0 0 0 0 2.253h1zm0 13H0a1 1 0 0 0 1.553.833L1 15.253zm8.447.833a1 1 0 1 0 1.107-1.666l-1.107 1.666zm0-14.666a1 1 0 1 0 1.107 1.666L9.447 1.42zM19 2.253h1a1 1 0 0 0-.447-.833L19 2.253zm0 13l-.553.833A1 1 0 0 0 20 15.253h-1zm-9.553-.833a1 1 0 1 0 1.107 1.666L9.447 14.42zM9 2.253v13h2v-13H9zm1.553-.833C9.203.523 7.42 0 5.5 0v2c1.572 0 2.961.431 3.947 1.086l1.107-1.666zM5.5 0C3.58 0 1.797.523.447 1.42l1.107 1.666C2.539 2.431 3.928 2 5.5 2V0zM0 2.253v13h2v-13H0zm1.553 13.833C2.539 15.431 3.928 15 5.5 15v-2c-1.92 0-3.703.523-5.053 1.42l1.107 1.666zM5.5 15c1.572 0 2.961.431 3.947 1.086l1.107-1.666C9.203 13.523 7.42 13 5.5 13v2zm5.053-11.914C11.539 2.431 12.928 2 14.5 2V0c-1.92 0-3.703.523-5.053 1.42l1.107 1.666zM14.5 2c1.573 0 2.961.431 3.947 1.086l1.107-1.666C18.203.523 16.421 0 14.5 0v2zm3.5.253v13h2v-13h-2zm1.553 12.167C18.203 13.523 16.421 13 14.5 13v2c1.573 0 2.961.431 3.947 1.086l1.107-1.666zM14.5 13c-1.92 0-3.703.523-5.053 1.42l1.107 1.666C11.539 15.431 12.928 15 14.5 15v-2z"

/>

</svg>

</template>

<template>

<svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="currentColor">

<path

d="M15 4a1 1 0 1 0 0 2V4zm0 11v-1a1 1 0 0 0-1 1h1zm0 4l-.707.707A1 1 0 0 0 16 19h-1zm-4-4l.707-.707A1 1 0 0 0 11 14v1zm-4.707-1.293a1 1 0 0 0-1.414 1.414l1.414-1.414zm-.707.707l-.707-.707.707.707zM9 11v-1a1 1 0 0 0-.707.293L9 11zm-4 0h1a1 1 0 0 0-1-1v1zm0 4H4a1 1 0 0 0 1.707.707L5 15zm10-9h2V4h-2v2zm2 0a1 1 0 0 1 1 1h2a3 3 0 0 0-3-3v2zm1 1v6h2V7h-2zm0 6a1 1 0 0 1-1 1v2a3 3 0 0 0 3-3h-2zm-1 1h-2v2h2v-2zm-3 1v4h2v-4h-2zm1.707 3.293l-4-4-1.414 1.414 4 4 1.414-1.414zM11 14H7v2h4v-2zm-4 0c-.276 0-.525-.111-.707-.293l-1.414 1.414C5.42 15.663 6.172 16 7 16v-2zm-.707 1.121l3.414-3.414-1.414-1.414-3.414 3.414 1.414 1.414zM9 12h4v-2H9v2zm4 0a3 3 0 0 0 3-3h-2a1 1 0 0 1-1 1v2zm3-3V3h-2v6h2zm0-6a3 3 0 0 0-3-3v2a1 1 0 0 1 1 1h2zm-3-3H3v2h10V0zM3 0a3 3 0 0 0-3 3h2a1 1 0 0 1 1-1V0zM0 3v6h2V3H0zm0 6a3 3 0 0 0 3 3v-2a1 1 0 0 1-1-1H0zm3 3h2v-2H3v2zm1-1v4h2v-4H4zm1.707 4.707l.586-.586-1.414-1.414-.586.586 1.414 1.414z"

/>

</svg>

</template>

<template>

<div class="file-uploader">

<!-- 主要内容区域：文件上传和视频播放并列 -->

<div class="main-content">

<!-- 左侧上传区域 -->

<div class="upload-container">

<div class="upload-section">

<div class="file-input-group">

<h3>视频文件 (MP4)</h3>

<input

type="file"

accept="video/mp4"

@change="handleVideoSelect"

:disabled="uploading"

/>

<div v-if="videoFile" class="selected-file">

已选择: {{ videoFile.name }}

</div>

</div>

<div class="file-input-group">

<h3>文本文件 (TXT)</h3>

<input

type="file"

accept=".txt"

@change="handleTextSelect"

:disabled="uploading"

/>

<div v-if="textFile" class="selected-file">

已选择: {{ textFile.name }}

</div>

</div>

</div>

<!-- 上传按钮 -->

<button

@click="handleUploadAndProcess"

:disabled="!canUpload || uploading"

class="action-button"

>

{{ uploading ? '上传中...' : '上传文件' }}

</button>

<!-- 处理按钮 -->

<button

v-if="groupId && !processing"

@click="processFiles"

:disabled="processing || isProcessed"

class="action-button process-button"

>

{{ isProcessed ? '处理完成' : '开始处理' }}

</button>

<!-- 上传进度条 -->

<div v-if="uploading" class="progress-container">

<div class="progress-label">上传进度</div>

<div class="progress-bar">

<div class="progress-fill" :style="{ width: `${uploadProgress}%` }"></div>

</div>

<div class="progress-text">{{ uploadProgress }}%</div>

</div>

<!-- 处理进度状态 -->

<div v-if="processing" class="progress-container">

<div class="progress-label">处理中...</div>

<div class="processing-spinner"></div>

</div>

<!-- 简化的状态消息 -->

<div v-if="message" class="status-message" :class="messageType">

{{ message }}

</div>

</div>

<!-- 右侧视频区域 -->

<div v-if="processedVideoUrl && isProcessed" class="video-section">

<div class="video-player-container">

<h4>处理后的视频</h4>

<!-- 视频播放器 -->

<div class="video-wrapper">

<video

ref="videoPlayer"

class="video-player"

controls

@error="handleVideoError"

@loadeddata="handleVideoLoaded"

crossorigin="anonymous"

>

<source :src="processedVideoUrl" type="video/mp4; codecs='avc1.42E01E, mp4a.40.2'">

您的浏览器不支持 HTML5 视频播放

</video>

<!-- 加载提示 -->

<div v-if="isVideoLoading" class="video-loading">

<div class="loading-spinner"></div>

<span>视频加载中...</span>

</div>

</div>

<!-- 视频控制按钮 -->

<div class="video-controls">

<button

@click="handleDownload"

class="control-button download-button"

>

<span class="button-icon"></span>

下载视频

</button>

<button

v-if="videoError"

@click="reloadVideo"

class="control-button reload-button"

>

<span class="button-icon">↻</span>

重新加载

</button>

</div>

<!-- 错误提示 -->

<div v-if="videoError" class="video-error">

<p>{{ videoError }}</p>

</div>

</div>

</div>

</div>

<!-- 底部日志区域 -->

<div v-if="showProcessLog" class="result-section">

<h3>处理结果</h3>

<div class="log-header">

<span>{{ isProcessed ? '处理完成' : (hasError ? '处理出错' : '处理状态') }}</span>

</div>

<!-- 处理日志 -->

<div v-if="logContent" class="log-content-wrapper">

<pre class="log-content">{{ logContent }}</pre>

</div>

</div>

<!-- 错误信息显示 -->

<div v-if="logContent && logContent.includes('错误')" class="error-container">

<div class="error-message">

<h4>处理过程中遇到问题：</h4>

<pre class="error-details">{{ logContent }}</pre>

</div>

</div>

</div>

</template>

<script setup lang="ts">

import { ref, computed, onMounted } from 'vue';

import axios from 'axios';

const videoFile = ref<File | null>(null);

const textFile = ref<File | null>(null);

const uploading = ref(false);

const processing = ref(false);

const uploadProgress = ref(0);

const message = ref('');

const messageType = ref<'success' | 'error'>('success');

const groupId = ref<number | null>(null);

const isProcessed = ref(false);

const showProcessLog = ref(false);

const hasError = ref(false);

const logContent = ref('');

const detailedError = ref('');

const showDetailedError = ref(false);

const processedVideoUrl = ref<string | null>(null);

const videoPlayer = ref<HTMLVideoElement | null>(null);

const showResult = ref(false);

const isVideoLoading = ref(true);

const videoError = ref<string | null>(null);

const canUpload = computed(() => {

return videoFile.value && textFile.value && !uploading.value;

});

const handleVideoSelect = (e: Event) => {

const input = e.target as HTMLInputElement;

if (input.files && input.files[0]) {

const file = input.files[0];

if (file.type === 'video/mp4') {

videoFile.value = file;

resetVideoState();

showProcessLog.value = false;

isProcessed.value = false;

} else {

showMessage('请选择MP4格式的视频文件', 'error');

}

}

};

const handleTextSelect = (e: Event) => {

const input = e.target as HTMLInputElement;

if (input.files && input.files[0]) {

const file = input.files[0];

if (file.type === 'text/plain') {

textFile.value = file;

} else {

showMessage('请选择TXT格式的文本文件', 'error');

}

}

};

const showMessage = (msg: string, type: 'success' | 'error') => {

message.value = msg;

messageType.value = type;

setTimeout(() => {

message.value = '';

}, 3000);

};

const updateLogContent = (content: any) => {

if (typeof content === 'string') {

logContent.value = content;

} else if (content && typeof content === 'object') {

try {

if (content.message) {

logContent.value = content.message;

} else if (content.status && content.result) {

logContent.value = `状态: ${content.status}\n结果: ${JSON.stringify(content.result, null, 2)}`;

} else {

logContent.value = JSON.stringify(content, null, 2);

}

} catch (e) {

logContent.value = '无法解析日志内容';

}

} else {

logContent.value = '无日志内容';

}

};

const getVideoFileName = () => {

if (videoFile.value) {

const originalName = videoFile.value.name;

const baseName = originalName.replace(/\.mp4$/, '');

return `${baseName}\_processed.mp4`;

}

return 'processed\_video.mp4';

};

const handleVideoLoaded = () => {

isVideoLoading.value = false;

videoError.value = null;

};

const handleVideoError = async (e: Event) => {

const target = e.target as HTMLVideoElement;

isVideoLoading.value = false;

let errorMessage = '视频加载失败: ';

if (target.error) {

switch (target.error.code) {

case MediaError.MEDIA\_ERR\_ABORTED:

errorMessage += '加载被中断';

break;

case MediaError.MEDIA\_ERR\_NETWORK:

errorMessage += '网络错误';

break;

case MediaError.MEDIA\_ERR\_DECODE:

case MediaError.MEDIA\_ERR\_SRC\_NOT\_SUPPORTED:

errorMessage += '视频格式不支持';

await retryLoadVideo();

break;

default:

errorMessage += `未知错误 (${target.error.code})`;

}

}

videoError.value = errorMessage;

console.error('视频加载错误:', {

url: processedVideoUrl.value,

error: target.error,

errorMessage

});

};

const reloadVideo = () => {

if (videoPlayer.value) {

isVideoLoading.value = true;

videoError.value = null;

videoPlayer.value.load();

}

};

const handleDownload = async () => {

if (!processedVideoUrl.value) {

showMessage('视频文件不可用', 'error');

return;

}

try {

const timestamp = new Date().getTime();

const downloadUrl = `${processedVideoUrl.value}?t=${timestamp}`;

const checkResponse = await fetch(downloadUrl, {

method: 'HEAD'

});

if (!checkResponse.ok) {

throw new Error(`文件访问失败: ${checkResponse.status}`);

}

const response = await fetch(downloadUrl);

const blob = await response.blob();

const contentType = response.headers.get('content-type');

if (!contentType || !contentType.includes('video/mp4')) {

console.warn('警告: 响应的内容类型不是 video/mp4:', contentType);

}

const url = window.URL.createObjectURL(blob);

const link = document.createElement('a');

link.href = url;

link.download = getVideoFileName();

document.body.appendChild(link);

link.click();

setTimeout(() => {

document.body.removeChild(link);

window.URL.revokeObjectURL(url);

}, 100);

showMessage('下载开始', 'success');

} catch (error: any) {

console.error('下载失败:', error);

showMessage(`下载失败: ${error.message}`, 'error');

logContent.value += `下载错误: ${error.message}\n`;

}

};

const checkVideoAccessibility = async (url: string) => {

try {

const response = await fetch(url, {

method: 'HEAD',

mode: 'cors'

});

if (!response.ok) {

throw new Error(`HTTP error! status: ${response.status}`);

}

return true;

} catch (e) {

console.error('视频访问检查失败:', e);

return false;

}

};

const checkVideoFormat = async (url: string): Promise<boolean> => {

try {

const response = await fetch(url, {

method: 'HEAD'

});

if (!response.ok) {

throw new Error(`HTTP error! status: ${response.status}`);

}

const contentType = response.headers.get('content-type');

return contentType !== null && contentType.includes('video/mp4');

} catch (e) {

console.error('视频格式检查失败:', e);

return false;

}

};

const retryLoadVideo = async () => {

if (!processedVideoUrl.value) return;

try {

isVideoLoading.value = true;

videoError.value = null;

const timestamp = new Date().getTime();

const newUrl = `${processedVideoUrl.value}?t=${timestamp}`;

const isValidFormat = await checkVideoFormat(newUrl);

if (!isValidFormat) {

throw new Error('视频格式不正确');

}

processedVideoUrl.value = newUrl;

if (videoPlayer.value) {

videoPlayer.value.load();

}

} catch (error: any) {

console.error('重试加载失败:', error);

videoError.value = `无法加载视频: ${error.message}`;

isVideoLoading.value = false;

}

};

const handleUploadAndProcess = async () => {

if (!videoFile.value || !textFile.value) return;

uploading.value = true;

uploadProgress.value = 0;

showResult.value = true;

showProcessLog.value = true;

logContent.value = '开始上传文件...\n';

try {

const formData = new FormData();

formData.append('video', videoFile.value);

formData.append('text', textFile.value);

const uploadResponse = await axios.post(

'http://localhost:8000/api/upload/',

formData,

{

headers: {

'Content-Type': 'multipart/form-data',

},

onUploadProgress: (progressEvent) => {

if (progressEvent.total) {

uploadProgress.value = Math.round(

(progressEvent.loaded \* 100) / progressEvent.total

);

}

}

}

);

if (uploadResponse.data.status !== 'success') {

throw new Error(uploadResponse.data.error || '文件上传失败');

}

const groupId = uploadResponse.data.group\_id;

logContent.value += '文件上传成功，开始处理...\n';

uploading.value = false;

processing.value = true;

const processResponse = await axios.post(

`http://localhost:8000/api/process/${groupId}/`

);

if (processResponse.data.status === 'success') {

processing.value = false;

isProcessed.value = true;

showMessage('处理完成！', 'success');

logContent.value += '处理完成！\n';

const videoUrl = processResponse.data.video\_url;

const timestamp = new Date().getTime();

const fullVideoUrl = `${videoUrl}?t=${timestamp}`;

const isValidFormat = await checkVideoFormat(fullVideoUrl);

if (isValidFormat) {

processedVideoUrl.value = fullVideoUrl;

logContent.value += `视频可访问: ${fullVideoUrl}\n`;

showResult.value = true;

showProcessLog.value = true;

isVideoLoading.value = true;

videoError.value = null;

} else {

throw new Error('处理后的视频格式不正确或无法访问');

}

} else {

throw new Error(processResponse.data.message || '视频处理失败');

}

} catch (error: any) {

const errorMessage = error.response?.data?.error ||

error.response?.data?.message ||

error.message ||

'操作失败';

showMessage(errorMessage, 'error');

logContent.value += `错误: ${errorMessage}\n`;

hasError.value = true;

console.error('操作错误:', error);

} finally {

uploading.value = false;

processing.value = false;

}

};

const resetVideoState = () => {

isVideoLoading.value = true;

videoError.value = null;

processedVideoUrl.value = null;

};

const checkBackendStatus = async () => {

try {

await axios.get('http://localhost:8000/api/health/');

return true;

} catch (error) {

console.error('后端服务检查失败:', error);

return false;

}

};

onMounted(async () => {

const isBackendAvailable = await checkBackendStatus();

if (!isBackendAvailable) {

showMessage('无法连接到后端服务，请确保服务已启动', 'error');

}

});

const handleProcessError = (error: any) => {

logContent.value += '\n处理过程中出错。需要安装缺失的依赖，请查看详细错误。';

if (error.response?.data?.error) {

detailedError.value = error.response.data.error;

} else if (error.response?.data?.result?.script\_stderr) {

detailedError.value = error.response.data.result.script\_stderr;

} else if (error.message) {

detailedError.value = error.message;

}

};

</script>

<style scoped>

.file-uploader {

max-width: 1200px;

margin: 3rem auto;

padding: 2.5rem;

border: none;

border-radius: 16px;

background: #ffffff;

box-shadow: 0 8px 30px rgba(0, 0, 0, 0.08);

transition: all 0.3s ease;

}

.main-content {

display: grid;

grid-template-columns: 1fr 1fr;

gap: 2rem;

margin-bottom: 2rem;

min-height: 500px;

}

.upload-container {

display: flex;

flex-direction: column;

gap: 1.5rem;

}

.upload-section {

display: flex;

flex-direction: column;

gap: 1.5rem;

}

.file-input-group {

background: #f8fafc;

padding: 1.5rem;

border-radius: 12px;

border: 1px solid rgba(0, 0, 0, 0.05);

transition: all 0.3s ease;

}

.file-input-group:hover {

transform: translateY(-2px);

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);

}

.file-input-group h3 {

margin: 0 0 1rem 0;

font-size: 1.1rem;

color: #1a1a1a;

font-weight: 600;

}

.selected-file {

margin-top: 0.8rem;

font-size: 0.95rem;

color: #4a5568;

padding: 0.5rem;

background: rgba(66, 185, 131, 0.1);

border-radius: 6px;

}

.action-button {

width: 100%;

padding: 1rem;

background: linear-gradient(135deg, #42b983 0%, #3aa876 100%);

color: white;

border: none;

border-radius: 8px;

cursor: pointer;

font-size: 1.1rem;

margin-bottom: 1.2rem;

transition: all 0.3s ease;

font-weight: 500;

letter-spacing: 0.5px;

box-shadow: 0 4px 12px rgba(66, 185, 131, 0.2);

}

.action-button:hover:not(:disabled) {

transform: translateY(-2px);

box-shadow: 0 6px 16px rgba(66, 185, 131, 0.3);

}

.action-button:disabled {

background: linear-gradient(135deg, #a8d5c2 0%, #9ecbb8 100%);

cursor: not-allowed;

box-shadow: none;

}

.process-button {

background: linear-gradient(135deg, #4a90e2 0%, #357abd 100%);

box-shadow: 0 4px 12px rgba(74, 144, 226, 0.2);

}

.process-button:hover:not(:disabled) {

box-shadow: 0 6px 16px rgba(74, 144, 226, 0.3);

}

.process-button:disabled {

background: linear-gradient(135deg, #a8c4e2 0%, #9ab8d9 100%);

}

.progress-container {

background: #f8fafc;

padding: 1.2rem;

border-radius: 10px;

box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.05);

}

.progress-label {

margin-bottom: 0.8rem;

color: #4a5568;

font-weight: 500;

}

.progress-bar {

height: 10px;

background: #e2e8f0;

border-radius: 6px;

overflow: hidden;

box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.05);

}

.progress-fill {

height: 100%;

background: linear-gradient(90deg, #42b983 0%, #3aa876 100%);

transition: width 0.4s ease;

box-shadow: 0 2px 4px rgba(66, 185, 131, 0.2);

}

.progress-text {

text-align: right;

font-size: 0.95rem;

color: #4a5568;

margin-top: 0.5rem;

font-weight: 500;

}

.processing-spinner {

width: 30px;

height: 30px;

border: 3px solid rgba(66, 185, 131, 0.1);

border-top: 3px solid #42b983;

border-radius: 50%;

animation: spin 1s linear infinite;

margin: 1rem auto;

}

.status-message {

padding: 1.2rem;

margin: 1.2rem 0;

border-radius: 10px;

text-align: center;

font-weight: 500;

animation: fadeIn 0.3s ease;

}

.success {

background: linear-gradient(135deg, #d4edda 0%, #c3e6cb 100%);

color: #155724;

box-shadow: 0 2px 8px rgba(21, 87, 36, 0.1);

}

.error {

background: linear-gradient(135deg, #f8d7da 0%, #f5c6cb 100%);

color: #721c24;

box-shadow: 0 2px 8px rgba(114, 28, 36, 0.1);

}

.result-section {

margin-top: 2rem;

padding: 1.5rem;

background: #f8fafc;

border-radius: 12px;

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);

display: block;

}

.log-header {

margin-bottom: 1rem;

font-weight: 500;

}

.log-content-wrapper {

margin-bottom: 1.2rem;

height: 100%;

}

.log-content {

max-height: 300px;

overflow-y: auto;

background: #ffffff;

padding: 1.2rem;

border-radius: 8px;

font-family: 'Monaco', 'Menlo', monospace;

line-height: 1.5;

border: 1px solid #e2e8f0;

}

.video-section {

background: #ffffff;

border-radius: 12px;

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);

height: fit-content;

position: sticky;

top: 2rem;

}

.video-player-container {

background: #ffffff;

border-radius: 12px;

overflow: hidden;

}

.video-wrapper {

aspect-ratio: 16/9;

background: #000000;

position: relative;

}

.video-player {

width: 100%;

max-width: 100%;

display: block;

border-radius: 8px;

}

.video-loading {

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background: rgba(0, 0, 0, 0.8);

color: white;

backdrop-filter: blur(4px);

}

.loading-spinner {

width: 48px;

height: 48px;

border: 4px solid rgba(255, 255, 255, 0.2);

border-top: 4px solid #ffffff;

border-radius: 50%;

animation: spin 1s linear infinite;

margin-bottom: 1.2rem;

}

.video-controls {

padding: 1rem;

border-top: 1px solid #e2e8f0;

background: #f8fafc;

}

.control-button {

display: inline-flex;

align-items: center;

gap: 0.8rem;

padding: 0.8rem 1.5rem;

border: none;

border-radius: 8px;

cursor: pointer;

font-size: 1rem;

transition: all 0.3s ease;

font-weight: 500;

letter-spacing: 0.5px;

}

.download-button {

background: linear-gradient(135deg, #42b983 0%, #3aa876 100%);

color: white;

text-decoration: none;

box-shadow: 0 4px 12px rgba(66, 185, 131, 0.2);

}

.download-button:hover {

transform: translateY(-2px);

box-shadow: 0 6px 16px rgba(66, 185, 131, 0.3);

}

.reload-button {

background: linear-gradient(135deg, #4a90e2 0%, #357abd 100%);

color: white;

box-shadow: 0 4px 12px rgba(74, 144, 226, 0.2);

}

.reload-button:hover {

transform: translateY(-2px);

box-shadow: 0 6px 16px rgba(74, 144, 226, 0.3);

}

.button-icon {

font-size: 1.3rem;

}

.video-error {

margin-top: 1.2rem;

padding: 1rem;

background: linear-gradient(135deg, #f8d7da 0%, #f5c6cb 100%);

border: 1px solid #f5c6cb;

border-radius: 8px;

color: #721c24;

text-align: center;

font-weight: 500;

box-shadow: 0 2px 8px rgba(114, 28, 36, 0.1);

}

.error-container {

margin: 1.2rem 0;

padding: 1.5rem;

background: linear-gradient(135deg, #fff3f3 0%, #ffe8e8 100%);

border: 1px solid #ffcdd2;

border-radius: 10px;

box-shadow: 0 4px 12px rgba(211, 47, 47, 0.1);

}

.error-message h4 {

color: #d32f2f;

margin: 0 0 1rem 0;

font-weight: 600;

}

.error-details {

background: #ffffff;

padding: 1.2rem;

border-radius: 8px;

font-size: 0.95rem;

margin: 0;

white-space: pre-wrap;

word-break: break-word;

box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.05);

font-family: 'Monaco', 'Menlo', monospace;

line-height: 1.5;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

@keyframes fadeIn {

from { opacity: 0; transform: translateY(-10px); }

to { opacity: 1; transform: translateY(0); }

}

.log-content::-webkit-scrollbar,

.error-details::-webkit-scrollbar {

width: 8px;

}

.log-content::-webkit-scrollbar-track,

.error-details::-webkit-scrollbar-track {

background: #f1f1f1;

border-radius: 4px;

}

.log-content::-webkit-scrollbar-thumb,

.error-details::-webkit-scrollbar-thumb {

background: #c1c1c1;

border-radius: 4px;

}

.log-content::-webkit-scrollbar-thumb:hover,

.error-details::-webkit-scrollbar-thumb:hover {

background: #a8a8a8;

}

input[type="file"] {

width: 100%;

padding: 0.8rem;

border: 2px dashed #e2e8f0;

border-radius: 8px;

cursor: pointer;

transition: all 0.3s ease;

}

input[type="file"]:hover {

border-color: #42b983;

background: rgba(66, 185, 131, 0.05);

}

input[type="file"]:disabled {

border-color: #e2e8f0;

background: #f8fafc;

cursor: not-allowed;

}

@media (max-width: 1024px) {

.main-content {

grid-template-columns: 1fr;

}

.video-section {

position: relative;

top: 0;

}

}

</style>

<template>

<div>

<FileUpload />

</div>

</template>

<script setup>

import FileUpload from '@/components/FileUpload.vue'

</script>

import argparse

import os

import os.path as osp

import numpy as np

import cv2

import torch

import gc

import sys

from pathlib import Path

os.environ["HYDRA\_FULL\_ERROR"] = "1"

current\_dir = os.path.dirname(os.path.abspath(\_\_file\_\_))

parent\_dir = os.path.dirname(current\_dir)

if parent\_dir not in sys.path:

sys.path.insert(0, parent\_dir)

sam2\_dir = os.path.join(parent\_dir, "sam2")

if sam2\_dir not in sys.path:

sys.path.insert(0, sam2\_dir)

sam2\_inner\_dir = os.path.join(sam2\_dir, "sam2")

if sam2\_inner\_dir not in sys.path:

sys.path.insert(0, sam2\_inner\_dir)

os.environ["HYDRA\_CONFIG\_PATH"] = os.path.join(parent\_dir, "sam2", "sam2", "configs")

from build\_sam import build\_sam2\_video\_predictor

color = [(255, 0, 0)]

def load\_txt(gt\_path):

with open(gt\_path, 'r') as f:

gt = f.readlines()

prompts = {}

for fid, line in enumerate(gt):

x, y, w, h = map(float, line.split(','))

x, y, w, h = int(x), int(y), int(w), int(h)

prompts[fid] = ((x, y, x + w, y + h), 0)

return prompts

def determine\_model\_cfg(model\_path):

config\_base = os.path.join(parent\_dir, "sam2", "sam2", "configs", "samurai")

if "large" in model\_path or "\_l" in model\_path:

return os.path.join(config\_base, "sam2.1\_hiera\_l.yaml")

elif "base\_plus" in model\_path or "\_b+" in model\_path:

return os.path.join(config\_base, "sam2.1\_hiera\_b+.yaml")

elif "base" in model\_path or "\_b" in model\_path:

return os.path.join(config\_base, "sam2.1\_hiera\_b+.yaml")

elif "small" in model\_path or "\_s" in model\_path:

return os.path.join(config\_base, "sam2.1\_hiera\_s.yaml")

elif "tiny" in model\_path or "\_t" in model\_path:

return os.path.join(config\_base, "sam2.1\_hiera\_t.yaml")

else:

return os.path.join(config\_base, "sam2.1\_hiera\_b+.yaml")

def prepare\_frames\_or\_path(video\_path):

if video\_path.endswith(".mp4") or osp.isdir(video\_path):

return video\_path

else:

raise ValueError("Invalid video\_path format. Should be .mp4 or a directory of jpg frames.")

def main(args):

output\_dir = os.path.join(os.path.dirname(os.path.dirname(os.path.dirname(\_\_file\_\_))),

'samurai\_django', 'myproject', 'media', 'processed\_videos')

os.makedirs(output\_dir, exist\_ok=True)

input\_filename = os.path.basename(args.video\_path)

output\_filename = os.path.splitext(input\_filename)[0] + "\_processed" + os.path.splitext(input\_filename)[1]

args.output\_path = os.path.join(output\_dir, output\_filename)

args.video\_output\_path = args.output\_path

if not os.path.isabs(args.model\_path):

args.model\_path = os.path.join(os.path.dirname(os.path.dirname(\_\_file\_\_)),

"sam2", "checkpoints", os.path.basename(args.model\_path))

if not os.path.exists(args.model\_path):

raise FileNotFoundError(f"模型文件不存在: {args.model\_path}")

model\_cfg = determine\_model\_cfg(args.model\_path)

predictor = build\_sam2\_video\_predictor(model\_cfg, args.model\_path, device="cuda:0")

frames\_or\_path = prepare\_frames\_or\_path(args.video\_path)

prompts = load\_txt(args.txt\_path)

frame\_rate = 30

if args.save\_to\_video:

if osp.isdir(args.video\_path):

frames = sorted([osp.join(args.video\_path, f) for f in os.listdir(args.video\_path) if f.endswith((".jpg", ".jpeg", ".JPG", ".JPEG"))])

loaded\_frames = [cv2.imread(frame\_path) for frame\_path in frames]

height, width = loaded\_frames[0].shape[:2]

else:

cap = cv2.VideoCapture(args.video\_path)

frame\_rate = cap.get(cv2.CAP\_PROP\_FPS)

loaded\_frames = []

while True:

ret, frame = cap.read()

if not ret:

break

loaded\_frames.append(frame)

cap.release()

height, width = loaded\_frames[0].shape[:2]

if len(loaded\_frames) == 0:

raise ValueError("No frames were loaded from the video.")

fourcc = cv2.VideoWriter\_fourcc(\*'avc1')

out = cv2.VideoWriter(args.video\_output\_path, fourcc, frame\_rate, (width, height))

with torch.inference\_mode(), torch.autocast("cuda", dtype=torch.float16):

state = predictor.init\_state(frames\_or\_path, offload\_video\_to\_cpu=True)

bbox, track\_label = prompts[0]

\_, \_, masks = predictor.add\_new\_points\_or\_box(state, box=bbox, frame\_idx=0, obj\_id=0)

for frame\_idx, object\_ids, masks in predictor.propagate\_in\_video(state):

mask\_to\_vis = {}

bbox\_to\_vis = {}

for obj\_id, mask in zip(object\_ids, masks):

mask = mask[0].cpu().numpy()

mask = mask > 0.0

non\_zero\_indices = np.argwhere(mask)

if len(non\_zero\_indices) == 0:

bbox = [0, 0, 0, 0]

else:

y\_min, x\_min = non\_zero\_indices.min(axis=0).tolist()

y\_max, x\_max = non\_zero\_indices.max(axis=0).tolist()

bbox = [x\_min, y\_min, x\_max - x\_min, y\_max - y\_min]

bbox\_to\_vis[obj\_id] = bbox

mask\_to\_vis[obj\_id] = mask

if args.save\_to\_video:

img = loaded\_frames[frame\_idx]

for obj\_id, mask in mask\_to\_vis.items():

mask\_img = np.zeros((height, width, 3), np.uint8)

mask\_img[mask] = color[(obj\_id + 1) % len(color)]

img = cv2.addWeighted(img, 1, mask\_img, 0.2, 0)

for obj\_id, bbox in bbox\_to\_vis.items():

cv2.rectangle(img, (bbox[0], bbox[1]), (bbox[0] + bbox[2], bbox[1] + bbox[3]), color[obj\_id % len(color)], 2)

out.write(img)

if args.save\_to\_video:

out.release()

del predictor, state

gc.collect()

torch.clear\_autocast\_cache()

torch.cuda.empty\_cache()

try:

if os.path.exists(args.output\_path):

print(f"处理完成，视频已保存到: {args.output\_path}")

relative\_path = os.path.relpath(args.output\_path,

os.path.join(os.path.dirname(os.path.dirname(os.path.dirname(\_\_file\_\_))),

'samurai\_django', 'media'))

return True, relative\_path

else:

print(f"处理失败：找不到输出文件 {args.output\_path}")

return False, None

except Exception as e:

print(f"处理过程中出错: {str(e)}")

return False, None

if \_\_name\_\_ == "\_\_main\_\_":

parser = argparse.ArgumentParser(description='视频处理脚本')

parser.add\_argument('--video\_path', type=str, required=True, help='输入视频路径')

parser.add\_argument('--txt\_path', type=str, required=True, help='输入文本路径')

default\_model\_path = os.path.join(os.path.dirname(os.path.dirname(\_\_file\_\_)),

"sam2", "checkpoints", "sam2.1\_hiera\_base\_plus.pt")

parser.add\_argument('--model\_path', type=str, default=default\_model\_path, help='模型路径')

parser.add\_argument("--save\_to\_video", default=True, help="Save results to a video.")

args = parser.parse\_args()

success, output\_path = main(args)

exit(0 if success else 1)

:root {

--vt-c-white: #ffffff;

--vt-c-white-soft: #f8f8f8;

--vt-c-white-mute: #f2f2f2;

--vt-c-black: #181818;

--vt-c-black-soft: #222222;

--vt-c-black-mute: #282828;

--vt-c-indigo: #2c3e50;

--vt-c-divider-light-1: rgba(60, 60, 60, 0.29);

--vt-c-divider-light-2: rgba(60, 60, 60, 0.12);

--vt-c-divider-dark-1: rgba(84, 84, 84, 0.65);

--vt-c-divider-dark-2: rgba(84, 84, 84, 0.48);

--vt-c-text-light-1: var(--vt-c-indigo);

--vt-c-text-light-2: rgba(60, 60, 60, 0.66);

--vt-c-text-dark-1: var(--vt-c-white);

--vt-c-text-dark-2: rgba(235, 235, 235, 0.64);

}

:root {

--color-background: var(--vt-c-white);

--color-background-soft: var(--vt-c-white-soft);

--color-background-mute: var(--vt-c-white-mute);

--color-border: var(--vt-c-divider-light-2);

--color-border-hover: var(--vt-c-divider-light-1);

--color-heading: var(--vt-c-text-light-1);

--color-text: var(--vt-c-text-light-1);

--section-gap: 160px;

}

@media (prefers-color-scheme: dark) {

:root {

--color-background: var(--vt-c-black);

--color-background-soft: var(--vt-c-black-soft);

--color-background-mute: var(--vt-c-black-mute);

--color-border: var(--vt-c-divider-dark-2);

--color-border-hover: var(--vt-c-divider-dark-1);

--color-heading: var(--vt-c-text-dark-1);

--color-text: var(--vt-c-text-dark-2);

}

}

\*,

\*::before,

\*::after {

box-sizing: border-box;

margin: 0;

font-weight: normal;

}

body {

min-height: 100vh;

color: var(--color-text);

background: var(--color-background);

transition:

color 0.5s,

background-color 0.5s;

line-height: 1.6;

font-family:

Inter,

-apple-system,

BlinkMacSystemFont,

'Segoe UI',

Roboto,

Oxygen,

Ubuntu,

Cantarell,

'Fira Sans',

'Droid Sans',

'Helvetica Neue',

sans-serif;

font-size: 15px;

text-rendering: optimizeLegibility;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

import torch

from torchvision.ops.boxes import box\_area

import numpy as np

def box\_cxcywh\_to\_xyxy(x):

x\_c, y\_c, w, h = x.unbind(-1)

b = [(x\_c - 0.5 \* w), (y\_c - 0.5 \* h),

(x\_c + 0.5 \* w), (y\_c + 0.5 \* h)]

return torch.stack(b, dim=-1)

def box\_xywh\_to\_xyxy(x):

x1, y1, w, h = x.unbind(-1)

b = [x1, y1, x1 + w, y1 + h]

return torch.stack(b, dim=-1)

def box\_xyxy\_to\_xywh(x):

x1, y1, x2, y2 = x.unbind(-1)

b = [x1, y1, x2 - x1, y2 - y1]

return torch.stack(b, dim=-1)

def box\_xyxy\_to\_cxcywh(x):

x0, y0, x1, y1 = x.unbind(-1)

b = [(x0 + x1) / 2, (y0 + y1) / 2,

(x1 - x0), (y1 - y0)]

return torch.stack(b, dim=-1)

def box\_iou(boxes1, boxes2):

area1 = box\_area(boxes1)

area2 = box\_area(boxes2)

lt = torch.max(boxes1[:, :2], boxes2[:, :2])

rb = torch.min(boxes1[:, 2:], boxes2[:, 2:])

wh = (rb - lt).clamp(min=0)

inter = wh[:, 0] \* wh[:, 1]

union = area1 + area2 - inter

iou = inter / union

return iou, union

def generalized\_box\_iou(boxes1, boxes2):

iou, union = box\_iou(boxes1, boxes2)

lt = torch.min(boxes1[:, :2], boxes2[:, :2])

rb = torch.max(boxes1[:, 2:], boxes2[:, 2:])

wh = (rb - lt).clamp(min=0)

area = wh[:, 0] \* wh[:, 1]

return iou - (area - union) / area, iou

def giou\_loss(boxes1, boxes2):

giou, iou = generalized\_box\_iou(boxes1, boxes2)

return (1 - giou).mean(), iou

def clip\_box(box: list, H, W, margin=0):

x1, y1, w, h = box

x2, y2 = x1 + w, y1 + h

x1 = min(max(0, x1), W-margin)

x2 = min(max(margin, x2), W)

y1 = min(max(0, y1), H-margin)

y2 = min(max(margin, y2), H)

w = max(margin, x2-x1)

h = max(margin, y2-y1)

return [x1, y1, w, h]

import math

import torch

import torch.nn.functional as F

def generate\_bbox\_mask(bbox\_mask, bbox):

b, h, w = bbox\_mask.shape

for i in range(b):

bbox\_i = bbox[i].cpu().tolist()

bbox\_mask[i, int(bbox\_i[1]):int(bbox\_i[1] + bbox\_i[3] - 1), int(bbox\_i[0]):int(bbox\_i[0] + bbox\_i[2] - 1)] = 1

return bbox\_mask

def generate\_mask\_cond(cfg, bs, device, gt\_bbox):

template\_size = cfg.DATA.TEMPLATE.SIZE

stride = cfg.MODEL.BACKBONE.STRIDE

template\_feat\_size = template\_size // stride

if cfg.MODEL.BACKBONE.CE\_TEMPLATE\_RANGE == 'ALL':

box\_mask\_z = None

elif cfg.MODEL.BACKBONE.CE\_TEMPLATE\_RANGE == 'CTR\_POINT':

if template\_feat\_size == 8:

index = slice(3, 4)

elif template\_feat\_size == 12:

index = slice(5, 6)

elif template\_feat\_size == 7:

index = slice(3, 4)

elif template\_feat\_size == 14:

index = slice(6, 7)

else:

raise NotImplementedError

box\_mask\_z = torch.zeros([bs, template\_feat\_size, template\_feat\_size], device=device)

box\_mask\_z[:, index, index] = 1

box\_mask\_z = box\_mask\_z.flatten(1).to(torch.bool)

elif cfg.MODEL.BACKBONE.CE\_TEMPLATE\_RANGE == 'CTR\_REC':

if template\_feat\_size == 8:

index = slice(3, 5)

elif template\_feat\_size == 12:

index = slice(5, 7)

elif template\_feat\_size == 7:

index = slice(3, 4)

else:

raise NotImplementedError

box\_mask\_z = torch.zeros([bs, template\_feat\_size, template\_feat\_size], device=device)

box\_mask\_z[:, index, index] = 1

box\_mask\_z = box\_mask\_z.flatten(1).to(torch.bool)

elif cfg.MODEL.BACKBONE.CE\_TEMPLATE\_RANGE == 'GT\_BOX':

box\_mask\_z = torch.zeros([bs, template\_size, template\_size], device=device)

box\_mask\_z = generate\_bbox\_mask(box\_mask\_z, gt\_bbox \* template\_size).unsqueeze(1).to(

torch.float)

box\_mask\_z = F.interpolate(box\_mask\_z, scale\_factor=1. / cfg.MODEL.BACKBONE.STRIDE, mode='bilinear',

align\_corners=False)

box\_mask\_z = box\_mask\_z.flatten(1).to(torch.bool)

else:

raise NotImplementedError

return box\_mask\_z

def adjust\_keep\_rate(epoch, warmup\_epochs, total\_epochs, ITERS\_PER\_EPOCH, base\_keep\_rate=0.5, max\_keep\_rate=1, iters=-1):

if epoch < warmup\_epochs:

return 1

if epoch >= total\_epochs:

return base\_keep\_rate

if iters == -1:

iters = epoch \* ITERS\_PER\_EPOCH

total\_iters = ITERS\_PER\_EPOCH \* (total\_epochs - warmup\_epochs)

iters = iters - ITERS\_PER\_EPOCH \* warmup\_epochs

keep\_rate = base\_keep\_rate + (max\_keep\_rate - base\_keep\_rate) \

\* (math.cos(iters / total\_iters \* math.pi) + 1) \* 0.5

return keep\_rate

@import './base.css';

#app {

max-width: 1280px;

margin: 0 auto;

padding: 2rem;

font-weight: normal;

}

a,

.green {

text-decoration: none;

color: hsla(160, 100%, 37%, 1);

transition: 0.4s;

padding: 3px;

}

@media (hover: hover) {

a:hover {

background-color: hsla(160, 100%, 37%, 0.2);

}

}

@media (min-width: 1024px) {

body {

display: flex;

place-items: center;

}

#app {

display: grid;

grid-template-columns: 1fr 1fr;

padding: 0 2rem;

}

}

from abc import ABC

import torch

import torch.nn as nn

import torch.nn.functional as F

class FocalLoss(nn.Module, ABC):

def \_\_init\_\_(self, alpha=2, beta=4):

super(FocalLoss, self).\_\_init\_\_()

self.alpha = alpha

self.beta = beta

def forward(self, prediction, target):

positive\_index = target.eq(1).float()

negative\_index = target.lt(1).float()

negative\_weights = torch.pow(1 - target, self.beta)

prediction = torch.clamp(prediction, 1e-12)

positive\_loss = torch.log(prediction) \* torch.pow(1 - prediction, self.alpha) \* positive\_index

negative\_loss = torch.log(1 - prediction) \* torch.pow(prediction,

self.alpha) \* negative\_weights \* negative\_index

num\_positive = positive\_index.float().sum()

positive\_loss = positive\_loss.sum()

negative\_loss = negative\_loss.sum()

if num\_positive == 0:

loss = -negative\_loss

else:

loss = -(positive\_loss + negative\_loss) / num\_positive

return loss

class LBHinge(nn.Module):

def \_\_init\_\_(self, error\_metric=nn.MSELoss(), threshold=None, clip=None):

super().\_\_init\_\_()

self.error\_metric = error\_metric

self.threshold = threshold if threshold is not None else -100

self.clip = clip

def forward(self, prediction, label, target\_bb=None):

negative\_mask = (label < self.threshold).float()

positive\_mask = (1.0 - negative\_mask)

prediction = negative\_mask \* F.relu(prediction) + positive\_mask \* prediction

loss = self.error\_metric(prediction, positive\_mask \* label)

if self.clip is not None:

loss = torch.min(loss, torch.tensor([self.clip], device=loss.device))

return loss

import argparse

import gc

import os

import os.path as osp

import pdb

import cv2

import numpy as np

import torch

from loguru import logger

from tqdm import tqdm

from sam2.build\_sam import build\_sam2\_video\_predictor

def load\_test\_video\_list(testing\_list\_path):

with open(testing\_list\_path, 'r') as f:

test\_videos = [line.strip() for line in f.readlines()]

return test\_videos

def load\_gt(gt\_path):

with open(gt\_path, 'r') as f:

gt = f.readlines()

prompts = {}

fid = 0

for line in gt:

x, y, w, h = map(int, line.split(','))

prompts[fid] = ((x, y, x+w, y+h), 0)

fid += 1

return prompts

def get\_ckpt\_and\_cfg(tracker\_name, model\_name):

assert tracker\_name in ["sam2.1", "samurai"], "Invalid tracker name"

assert model\_name in ["tiny", "small", "base\_plus", "large"], "Invalid model name"

model\_ckpt = f"sam2/checkpoints/sam2.1\_hiera\_{model\_name}.pt"

if model\_name == "base\_plus":

model\_cfg = f"configs/{tracker\_name}/sam2.1\_hiera\_b+.yaml"

else:

model\_cfg = f"configs/{tracker\_name}/sam2.1\_hiera\_{model\_name[0]}.yaml"

return model\_ckpt, model\_cfg

def split\_list(video\_list, num\_chunks):

chunk\_size = len(video\_list) // num\_chunks

return [video\_list[i:i+chunk\_size] for i in range(0, len(video\_list), chunk\_size)]

def inference\_chunk(dataset\_path, tracker\_name, model\_name, chunk\_videos, result\_folder):

exp\_name = "test"

model\_ckpt, model\_cfg = get\_ckpt\_and\_cfg(tracker\_name, model\_name)

for vid, video in enumerate(chunk\_videos):

cat\_name = video.split('-')[0]

cid\_name = video.split('-')[1]

video\_basename = video.strip()

frame\_folder = osp.join(dataset\_path, cat\_name, video.strip(), "img")

num\_frames = len(os.listdir(osp.join(dataset\_path, cat\_name, video.strip(), "img")))

height, width = cv2.imread(osp.join(frame\_folder, "00000001.jpg")).shape[:2]

logger.info(f"Running video [{vid+1}/{len(chunk\_videos)}]: {video} with {num\_frames} frames ({height}x{width})")

predictor = build\_sam2\_video\_predictor(model\_cfg, model\_ckpt, device="cuda:0")

predictions = []

with torch.inference\_mode(), torch.autocast("cuda", dtype=torch.float16):

state = predictor.init\_state(frame\_folder, offload\_video\_to\_cpu=True, offload\_state\_to\_cpu=True)

prompts = load\_gt(osp.join(dataset\_path, cat\_name, video.strip(), "groundtruth.txt"))

bbox, track\_label = prompts[0]

frame\_idx, object\_ids, masks = predictor.add\_new\_points\_or\_box(state, box=bbox, frame\_idx=0, obj\_id=0)

for frame\_idx, object\_ids, masks in predictor.propagate\_in\_video(state):

mask\_to\_vis = {}

bbox\_to\_vis = {}

assert len(masks) == 1 and len(object\_ids) == 1, "Only one object is supported right now"

for obj\_id, mask in zip(object\_ids, masks):

mask = mask[0].cpu().numpy()

mask = mask > 0.0

non\_zero\_indices = np.argwhere(mask)

if len(non\_zero\_indices) == 0:

bbox = [0, 0, 0, 0]

else:

y\_min, x\_min = non\_zero\_indices.min(axis=0).tolist()

y\_max, x\_max = non\_zero\_indices.max(axis=0).tolist()

bbox = [x\_min, y\_min, x\_max-x\_min, y\_max-y\_min]

bbox\_to\_vis[obj\_id] = bbox

mask\_to\_vis[obj\_id] = mask

predictions.append(bbox\_to\_vis)

os.makedirs(result\_folder, exist\_ok=True)

with open(osp.join(result\_folder, f'{video\_basename}.txt'), 'w') as f:

for pred in predictions:

x, y, w, h = pred[0]

f.write(f"{x},{y},{w},{h}\n")

del predictor

del state

gc.collect()

torch.clear\_autocast\_cache()

torch.cuda.empty\_cache()

def main():

parser = argparse.ArgumentParser()

parser.add\_argument("--dataset\_path", type=str, default="data/LaSOT-ext")

parser.add\_argument("--tracker\_name", type=str, default="samurai")

parser.add\_argument("--model\_name", type=str, default="large")

parser.add\_argument("--chunk\_idx", type=int, default=0)

parser.add\_argument("--num\_chunks", type=int, default=1)

parser.add\_argument("--exp\_name", type=str, default="test")

parser.add\_argument("--root\_result\_folder", type=str, default="results")

args = parser.parse\_args()

test\_videos = load\_test\_video\_list("data/LaSOT-ext/testing\_set.txt")

chunk\_video\_list = split\_list(test\_videos, args.num\_chunks)

chunk\_videos = chunk\_video\_list[args.chunk\_idx]

logger.info(f"Chunk ID: {args.chunk\_idx}, Number of videos: {len(chunk\_videos)} (from {chunk\_videos[0]} to {chunk\_videos[-1]})")

exp\_result\_folder = osp.join(args.root\_result\_folder, args.tracker\_name, f"{args.exp\_name}\_{args.model\_name}")

inference\_chunk(args.dataset\_path, args.tracker\_name, args.model\_name, chunk\_videos, exp\_result\_folder)

if \_\_name\_\_ == "\_\_main\_\_":

main()

import functools

import torch

import copy

from collections import OrderedDict

class TensorDict(OrderedDict):

def concat(self, other):

return TensorDict(self, \*\*other)

def copy(self):

return TensorDict(super(TensorDict, self).copy())

def \_\_deepcopy\_\_(self, memodict={}):

return TensorDict(copy.deepcopy(list(self), memodict))

def \_\_getattr\_\_(self, name):

if not hasattr(torch.Tensor, name):

raise AttributeError('\'TensorDict\' object has not attribute \'{}\''.format(name))

def apply\_attr(\*args, \*\*kwargs):

return TensorDict({n: getattr(e, name)(\*args, \*\*kwargs) if hasattr(e, name) else e for n, e in self.items()})

return apply\_attr

def attribute(self, attr: str, \*args):

return TensorDict({n: getattr(e, attr, \*args) for n, e in self.items()})

def apply(self, fn, \*args, \*\*kwargs):

return TensorDict({n: fn(e, \*args, \*\*kwargs) for n, e in self.items()})

@staticmethod

def \_iterable(a):

return isinstance(a, (TensorDict, list))

class TensorList(list):

def \_\_init\_\_(self, list\_of\_tensors = None):

if list\_of\_tensors is None:

list\_of\_tensors = list()

super(TensorList, self).\_\_init\_\_(list\_of\_tensors)

def \_\_deepcopy\_\_(self, memodict={}):

return TensorList(copy.deepcopy(list(self), memodict))

def \_\_getitem\_\_(self, item):

if isinstance(item, int):

return super(TensorList, self).\_\_getitem\_\_(item)

elif isinstance(item, (tuple, list)):

return TensorList([super(TensorList, self).\_\_getitem\_\_(i) for i in item])

else:

return TensorList(super(TensorList, self).\_\_getitem\_\_(item))

def \_\_add\_\_(self, other):

if TensorList.\_iterable(other):

return TensorList([e1 + e2 for e1, e2 in zip(self, other)])

return TensorList([e + other for e in self])

def \_\_radd\_\_(self, other):

if TensorList.\_iterable(other):

return TensorList([e2 + e1 for e1, e2 in zip(self, other)])

return TensorList([other + e for e in self])

def \_\_iadd\_\_(self, other):

if TensorList.\_iterable(other):

for i, e2 in enumerate(other):

self[i] += e2

else:

for i in range(len(self)):

self[i] += other

return self

def \_\_sub\_\_(self, other):

if TensorList.\_iterable(other):

return TensorList([e1 - e2 for e1, e2 in zip(self, other)])

return TensorList([e - other for e in self])

def \_\_rsub\_\_(self, other):

if TensorList.\_iterable(other):

return TensorList([e2 - e1 for e1, e2 in zip(self, other)])

return TensorList([other - e for e in self])

def \_\_isub\_\_(self, other):

if TensorList.\_iterable(other):

for i, e2 in enumerate(other):

self[i] -= e2

else:

for i in range(len(self)):

self[i] -= other

return self

def \_\_mul\_\_(self, other):

if TensorList.\_iterable(other):

return TensorList([e1 \* e2 for e1, e2 in zip(self, other)])

return TensorList([e \* other for e in self])

def \_\_rmul\_\_(self, other):

if TensorList.\_iterable(other):

return TensorList([e2 \* e1 for e1, e2 in zip(self, other)])

return TensorList([other \* e for e in self])

def \_\_imul\_\_(self, other):

if TensorList.\_iterable(other):

for i, e2 in enumerate(other):

self[i] \*= e2

else:

for i in range(len(self)):

self[i] \*= other

return self

def \_\_truediv\_\_(self, other):

if TensorList.\_iterable(other):

return TensorList([e1 / e2 for e1, e2 in zip(self, other)])

return TensorList([e / other for e in self])

def \_\_rtruediv\_\_(self, other):

{"name": "uav\_building1", "path": "data\_seq/UAV123/building1", "startFrame": 1, "endFrame": 469, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/building1.txt", "object\_class": "other"},

{"name": "uav\_building2", "path": "data\_seq/UAV123/building2", "startFrame": 1, "endFrame": 577, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/building2.txt", "object\_class": "other"},

{"name": "uav\_building3", "path": "data\_seq/UAV123/building3", "startFrame": 1, "endFrame": 829, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/building3.txt", "object\_class": "other"},

{"name": "uav\_building4", "path": "data\_seq/UAV123/building4", "startFrame": 1, "endFrame": 787, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/building4.txt", "object\_class": "other"},

{"name": "uav\_building5", "path": "data\_seq/UAV123/building5", "startFrame": 1, "endFrame": 481, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/building5.txt", "object\_class": "other"},

{"name": "uav\_car1\_1", "path": "data\_seq/UAV123/car1", "startFrame": 1, "endFrame": 751, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car1\_1.txt", "object\_class": "car"},

{"name": "uav\_car1\_2", "path": "data\_seq/UAV123/car1", "startFrame": 751, "endFrame": 1627, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car1\_2.txt", "object\_class": "car"},

{"name": "uav\_car1\_3", "path": "data\_seq/UAV123/car1", "startFrame": 1627, "endFrame": 2629, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car1\_3.txt", "object\_class": "car"},

{"name": "uav\_car10", "path": "data\_seq/UAV123/car10", "startFrame": 1, "endFrame": 1405, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car10.txt", "object\_class": "car"},

{"name": "uav\_car11", "path": "data\_seq/UAV123/car11", "startFrame": 1, "endFrame": 337, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car11.txt", "object\_class": "car"},

{"name": "uav\_car12", "path": "data\_seq/UAV123/car12", "startFrame": 1, "endFrame": 499, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car12.txt", "object\_class": "car"},

{"name": "uav\_car13", "path": "data\_seq/UAV123/car13", "startFrame": 1, "endFrame": 415, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car13.txt", "object\_class": "car"},

{"name": "uav\_car14", "path": "data\_seq/UAV123/car14", "startFrame": 1, "endFrame": 1327, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car14.txt", "object\_class": "car"},

{"name": "uav\_car15", "path": "data\_seq/UAV123/car15", "startFrame": 1, "endFrame": 469, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car15.txt", "object\_class": "car"},

{"name": "uav\_car16\_1", "path": "data\_seq/UAV123/car16", "startFrame": 1, "endFrame": 415, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car16\_1.txt", "object\_class": "car"},

{"name": "uav\_car16\_2", "path": "data\_seq/UAV123/car16", "startFrame": 415, "endFrame": 1993, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car16\_2.txt", "object\_class": "car"},

{"name": "uav\_car17", "path": "data\_seq/UAV123/car17", "startFrame": 1, "endFrame": 1057, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car17.txt", "object\_class": "car"},

{"name": "uav\_car18", "path": "data\_seq/UAV123/car18", "startFrame": 1, "endFrame": 1207, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car18.txt", "object\_class": "car"},

{"name": "uav\_car1\_s", "path": "data\_seq/UAV123/car1\_s", "startFrame": 1, "endFrame": 1475, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car1\_s.txt", "object\_class": "car"},

{"name": "uav\_car2", "path": "data\_seq/UAV123/car2", "startFrame": 1, "endFrame": 1321, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car2.txt", "object\_class": "car"},

{"name": "uav\_car2\_s", "path": "data\_seq/UAV123/car2\_s", "startFrame": 1, "endFrame": 320, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car2\_s.txt", "object\_class": "car"},

{"name": "uav\_car3", "path": "data\_seq/UAV123/car3", "startFrame": 1, "endFrame": 1717, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car3.txt", "object\_class": "car"},

{"name": "uav\_car3\_s", "path": "data\_seq/UAV123/car3\_s", "startFrame": 1, "endFrame": 1300, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car3\_s.txt", "object\_class": "car"},

{"name": "uav\_car4", "path": "data\_seq/UAV123/car4", "startFrame": 1, "endFrame": 1345, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car4.txt", "object\_class": "car"},

{"name": "uav\_car4\_s", "path": "data\_seq/UAV123/car4\_s", "startFrame": 1, "endFrame": 830, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car4\_s.txt", "object\_class": "car"},

{"name": "uav\_car5", "path": "data\_seq/UAV123/car5", "startFrame": 1, "endFrame": 745, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car5.txt", "object\_class": "car"},

{"name": "uav\_car6\_1", "path": "data\_seq/UAV123/car6", "startFrame": 1, "endFrame": 487, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car6\_1.txt", "object\_class": "car"},

{"name": "uav\_car6\_2", "path": "data\_seq/UAV123/car6", "startFrame": 487, "endFrame": 1807, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car6\_2.txt", "object\_class": "car"},

{"name": "uav\_car6\_3", "path": "data\_seq/UAV123/car6", "startFrame": 1807, "endFrame": 2953, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car6\_3.txt", "object\_class": "car"},

{"name": "uav\_car6\_4", "path": "data\_seq/UAV123/car6", "startFrame": 2953, "endFrame": 3925, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car6\_4.txt", "object\_class": "car"},

{"name": "uav\_car6\_5", "path": "data\_seq/UAV123/car6", "startFrame": 3925, "endFrame": 4861, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car6\_5.txt", "object\_class": "car"},

{"name": "uav\_car7", "path": "data\_seq/UAV123/car7", "startFrame": 1, "endFrame": 1033, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car7.txt", "object\_class": "car"},

{"name": "uav\_car8\_1", "path": "data\_seq/UAV123/car8", "startFrame": 1, "endFrame": 1357, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car8\_1.txt", "object\_class": "car"},

{"name": "uav\_car8\_2", "path": "data\_seq/UAV123/car8", "startFrame": 1357, "endFrame": 2575, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car8\_2.txt", "object\_class": "car"},

{"name": "uav\_car9", "path": "data\_seq/UAV123/car9", "startFrame": 1, "endFrame": 1879, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/car9.txt", "object\_class": "car"},

{"name": "uav\_group1\_1", "path": "data\_seq/UAV123/group1", "startFrame": 1, "endFrame": 1333, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/group1\_1.txt", "object\_class": "person"},

{"name": "uav\_group1\_2", "path": "data\_seq/UAV123/group1", "startFrame": 1333, "endFrame": 2515, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/group1\_2.txt", "object\_class": "person"},

{"name": "uav\_group1\_3", "path": "data\_seq/UAV123/group1", "startFrame": 2515, "endFrame": 3925, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/group1\_3.txt", "object\_class": "person"},

{"name": "uav\_group1\_4", "path": "data\_seq/UAV123/group1", "startFrame": 3925, "endFrame": 4873, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/group1\_4.txt", "object\_class": "person"},

{"name": "uav\_group2\_1", "path": "data\_seq/UAV123/group2", "startFrame": 1, "endFrame": 907, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/group2\_1.txt", "object\_class": "person"},

{"name": "uav\_group2\_2", "path": "data\_seq/UAV123/group2", "startFrame": 907, "endFrame": 1771, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/group2\_2.txt", "object\_class": "person"},

{"name": "uav\_group2\_3", "path": "data\_seq/UAV123/group2", "startFrame": 1771, "endFrame": 2683, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/group2\_3.txt", "object\_class": "person"},

{"name": "uav\_group3\_1", "path": "data\_seq/UAV123/group3", "startFrame": 1, "endFrame": 1567, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/group3\_1.txt", "object\_class": "person"},

{"name": "uav\_group3\_2", "path": "data\_seq/UAV123/group3", "startFrame": 1567, "endFrame": 2827, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/group3\_2.txt", "object\_class": "person"},

{"name": "uav\_group3\_3", "path": "data\_seq/UAV123/group3", "startFrame": 2827, "endFrame": 4369, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/group3\_3.txt", "object\_class": "person"},

{"name": "uav\_group3\_4", "path": "data\_seq/UAV123/group3", "startFrame": 4369, "endFrame": 5527, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/group3\_4.txt", "object\_class": "person"},

{"name": "uav\_person1", "path": "data\_seq/UAV123/person1", "startFrame": 1, "endFrame": 799, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person1.txt", "object\_class": "person"},

{"name": "uav\_person10", "path": "data\_seq/UAV123/person10", "startFrame": 1, "endFrame": 1021, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person10.txt", "object\_class": "person"},

{"name": "uav\_person11", "path": "data\_seq/UAV123/person11", "startFrame": 1, "endFrame": 721, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person11.txt", "object\_class": "person"},

{"name": "uav\_person12\_1", "path": "data\_seq/UAV123/person12", "startFrame": 1, "endFrame": 601, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person12\_1.txt", "object\_class": "person"},

{"name": "uav\_person12\_2", "path": "data\_seq/UAV123/person12", "startFrame": 601, "endFrame": 1621, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person12\_2.txt", "object\_class": "person"},

{"name": "uav\_person13", "path": "data\_seq/UAV123/person13", "startFrame": 1, "endFrame": 883, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person13.txt", "object\_class": "person"},

{"name": "uav\_person14\_1", "path": "data\_seq/UAV123/person14", "startFrame": 1, "endFrame": 847, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person14\_1.txt", "object\_class": "person"},

{"name": "uav\_person14\_2", "path": "data\_seq/UAV123/person14", "startFrame": 847, "endFrame": 1813, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person14\_2.txt", "object\_class": "person"},

{"name": "uav\_person14\_3", "path": "data\_seq/UAV123/person14", "startFrame": 1813, "endFrame": 2923,

"nz": 6, "ext": "jpg", "anno\_path": "anno/UAV123/person14\_3.txt", "object\_class": "person"},

{"name": "uav\_person15", "path": "data\_seq/UAV123/person15", "startFrame": 1, "endFrame": 1339, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person15.txt", "object\_class": "person"},

{"name": "uav\_person16", "path": "data\_seq/UAV123/person16", "startFrame": 1, "endFrame": 1147, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person16.txt", "object\_class": "person"},

{"name": "uav\_person17\_1", "path": "data\_seq/UAV123/person17", "startFrame": 1, "endFrame": 1501, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person17\_1.txt", "object\_class": "person"},

{"name": "uav\_person17\_2", "path": "data\_seq/UAV123/person17", "startFrame": 1501, "endFrame": 2347,

"nz": 6, "ext": "jpg", "anno\_path": "anno/UAV123/person17\_2.txt", "object\_class": "person"},

{"name": "uav\_person18", "path": "data\_seq/UAV123/person18", "startFrame": 1, "endFrame": 1393, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person18.txt", "object\_class": "person"},

{"name": "uav\_person19\_1", "path": "data\_seq/UAV123/person19", "startFrame": 1, "endFrame": 1243, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person19\_1.txt", "object\_class": "person"},

{"name": "uav\_person19\_2", "path": "data\_seq/UAV123/person19", "startFrame": 1243, "endFrame": 2791,

"nz": 6, "ext": "jpg", "anno\_path": "anno/UAV123/person19\_2.txt", "object\_class": "person"},

{"name": "uav\_person19\_3", "path": "data\_seq/UAV123/person19", "startFrame": 2791, "endFrame": 4357,

"nz": 6, "ext": "jpg", "anno\_path": "anno/UAV123/person19\_3.txt", "object\_class": "person"},

{"name": "uav\_person1\_s", "path": "data\_seq/UAV123/person1\_s", "startFrame": 1, "endFrame": 1600, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person1\_s.txt", "object\_class": "person"},

{"name": "uav\_person2\_1", "path": "data\_seq/UAV123/person2", "startFrame": 1, "endFrame": 1189, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person2\_1.txt", "object\_class": "person"},

{"name": "uav\_person2\_2", "path": "data\_seq/UAV123/person2", "startFrame": 1189, "endFrame": 2623, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person2\_2.txt", "object\_class": "person"},

{"name": "uav\_person20", "path": "data\_seq/UAV123/person20", "startFrame": 1, "endFrame": 1783, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person20.txt", "object\_class": "person"},

{"name": "uav\_person21", "path": "data\_seq/UAV123/person21", "startFrame": 1, "endFrame": 487, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person21.txt", "object\_class": "person"},

{"name": "uav\_person22", "path": "data\_seq/UAV123/person22", "startFrame": 1, "endFrame": 199, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person22.txt", "object\_class": "person"},

{"name": "uav\_person23", "path": "data\_seq/UAV123/person23", "startFrame": 1, "endFrame": 397, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person23.txt", "object\_class": "person"},

{"name": "uav\_person2\_s", "path": "data\_seq/UAV123/person2\_s", "startFrame": 1, "endFrame": 250, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person2\_s.txt", "object\_class": "person"},

{"name": "uav\_person3", "path": "data\_seq/UAV123/person3", "startFrame": 1, "endFrame": 643, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person3.txt", "object\_class": "person"},

{"name": "uav\_person3\_s", "path": "data\_seq/UAV123/person3\_s", "startFrame": 1, "endFrame": 505, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person3\_s.txt", "object\_class": "person"},

{"name": "uav\_person4\_1", "path": "data\_seq/UAV123/person4", "startFrame": 1, "endFrame": 1501, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person4\_1.txt", "object\_class": "person"},

{"name": "uav\_person4\_2", "path": "data\_seq/UAV123/person4", "startFrame": 1501, "endFrame": 2743, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person4\_2.txt", "object\_class": "person"},

{"name": "uav\_person5\_1", "path": "data\_seq/UAV123/person5", "startFrame": 1, "endFrame": 877, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person5\_1.txt", "object\_class": "person"},

{"name": "uav\_person5\_2", "path": "data\_seq/UAV123/person5", "startFrame": 877, "endFrame": 2101, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person5\_2.txt", "object\_class": "person"},

{"name": "uav\_person6", "path": "data\_seq/UAV123/person6", "startFrame": 1, "endFrame": 901, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person6.txt", "object\_class": "person"},

{"name": "uav\_person7\_1", "path": "data\_seq/UAV123/person7", "startFrame": 1, "endFrame": 1249, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person7\_1.txt", "object\_class": "person"},

{"name": "uav\_person7\_2", "path": "data\_seq/UAV123/person7", "startFrame": 1249, "endFrame": 2065, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person7\_2.txt", "object\_class": "person"},

{"name": "uav\_person8\_1", "path": "data\_seq/UAV123/person8", "startFrame": 1, "endFrame": 1075, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person8\_1.txt", "object\_class": "person"},

{"name": "uav\_person8\_2", "path": "data\_seq/UAV123/person8", "startFrame": 1075, "endFrame": 1525, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person8\_2.txt", "object\_class": "person"},

{"name": "uav\_person9", "path": "data\_seq/UAV123/person9", "startFrame": 1, "endFrame": 661, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/person9.txt", "object\_class": "person"},

{"name": "uav\_truck1", "path": "data\_seq/UAV123/truck1", "startFrame": 1, "endFrame": 463, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/truck1.txt", "object\_class": "truck"},

{"name": "uav\_truck2", "path": "data\_seq/UAV123/truck2", "startFrame": 1, "endFrame": 385, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/truck2.txt", "object\_class": "truck"},

{"name": "uav\_truck3", "path": "data\_seq/UAV123/truck3", "startFrame": 1, "endFrame": 535, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/truck3.txt", "object\_class": "truck"},

{"name": "uav\_truck4\_1", "path": "data\_seq/UAV123/truck4", "startFrame": 1, "endFrame": 577, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/truck4\_1.txt", "object\_class": "truck"},

{"name": "uav\_truck4\_2", "path": "data\_seq/UAV123/truck4", "startFrame": 577, "endFrame": 1261, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/truck4\_2.txt", "object\_class": "truck"},

{"name": "uav\_uav1\_1", "path": "data\_seq/UAV123/uav1", "startFrame": 1, "endFrame": 1555, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/uav1\_1.txt", "object\_class": "aircraft"},

{"name": "uav\_uav1\_2", "path": "data\_seq/UAV123/uav1", "startFrame": 1555, "endFrame": 2377, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/uav1\_2.txt", "object\_class": "aircraft"},

{"name": "uav\_uav1\_3", "path": "data\_seq/UAV123/uav1", "startFrame": 2473, "endFrame": 3469, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/uav1\_3.txt", "object\_class": "aircraft"},

{"name": "uav\_uav2", "path": "data\_seq/UAV123/uav2", "startFrame": 1, "endFrame": 133, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/uav2.txt", "object\_class": "aircraft"},

{"name": "uav\_uav3", "path": "data\_seq/UAV123/uav3", "startFrame": 1, "endFrame": 265, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/uav3.txt", "object\_class": "aircraft"},

{"name": "uav\_uav4", "path": "data\_seq/UAV123/uav4", "startFrame": 1, "endFrame": 157, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/uav4.txt", "object\_class": "aircraft"},

{"name": "uav\_uav5", "path": "data\_seq/UAV123/uav5", "startFrame": 1, "endFrame": 139, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/uav5.txt", "object\_class": "aircraft"},

{"name": "uav\_uav6", "path": "data\_seq/UAV123/uav6", "startFrame": 1, "endFrame": 109, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/uav6.txt", "object\_class": "aircraft"},

{"name": "uav\_uav7", "path": "data\_seq/UAV123/uav7", "startFrame": 1, "endFrame": 373, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/uav7.txt", "object\_class": "aircraft"},

{"name": "uav\_uav8", "path": "data\_seq/UAV123/uav8", "startFrame": 1, "endFrame": 301, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/uav8.txt", "object\_class": "aircraft"},

{"name": "uav\_wakeboard1", "path": "data\_seq/UAV123/wakeboard1", "startFrame": 1, "endFrame": 421, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/wakeboard1.txt", "object\_class": "person"},

{"name": "uav\_wakeboard10", "path": "data\_seq/UAV123/wakeboard10", "startFrame": 1, "endFrame": 469,

"nz": 6, "ext": "jpg", "anno\_path": "anno/UAV123/wakeboard10.txt", "object\_class": "person"},

{"name": "uav\_wakeboard2", "path": "data\_seq/UAV123/wakeboard2", "startFrame": 1, "endFrame": 733, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/wakeboard2.txt", "object\_class": "person"},

{"name": "uav\_wakeboard3", "path": "data\_seq/UAV123/wakeboard3", "startFrame": 1, "endFrame": 823, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/wakeboard3.txt", "object\_class": "person"},

{"name": "uav\_wakeboard4", "path": "data\_seq/UAV123/wakeboard4", "startFrame": 1, "endFrame": 697, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/wakeboard4.txt", "object\_class": "person"},

{"name": "uav\_wakeboard5", "path": "data\_seq/UAV123/wakeboard5", "startFrame": 1, "endFrame": 1675, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/wakeboard5.txt", "object\_class": "person"},

{"name": "uav\_wakeboard6", "path": "data\_seq/UAV123/wakeboard6", "startFrame": 1, "endFrame": 1165, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/wakeboard6.txt", "object\_class": "person"},

{"name": "uav\_wakeboard7", "path": "data\_seq/UAV123/wakeboard7", "startFrame": 1, "endFrame": 199, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/wakeboard7.txt", "object\_class": "person"},

{"name": "uav\_wakeboard8", "path": "data\_seq/UAV123/wakeboard8", "startFrame": 1, "endFrame": 1543, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/wakeboard8.txt", "object\_class": "person"},

{"name": "uav\_wakeboard9", "path": "data\_seq/UAV123/wakeboard9", "startFrame": 1, "endFrame": 355, "nz": 6,

"ext": "jpg", "anno\_path": "anno/UAV123/wakeboard9.txt", "object\_class": "person"}

]

return sequence\_info\_list

import numpy as np

from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList

import os

from lib.test.utils.load\_text import load\_text

class TrackingNetDataset(BaseDataset):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.base\_path = self.env\_settings.trackingnet\_path

sets = 'TEST'

if not isinstance(sets, (list, tuple)):

if sets == 'TEST':

sets = ['TEST']

elif sets == 'TRAIN':

sets = ['TRAIN\_{}'.format(i) for i in range(5)]

self.sequence\_list = self.\_list\_sequences(self.base\_path, sets)

def get\_sequence\_list(self):

return SequenceList([self.\_construct\_sequence(set, seq\_name) for set, seq\_name in self.sequence\_list])

def \_construct\_sequence(self, set, sequence\_name):

anno\_path = '{}/{}/anno/{}.txt'.format(self.base\_path, set, sequence\_name)

ground\_truth\_rect = load\_text(str(anno\_path), delimiter=',', dtype=np.float64, backend='numpy')

frames\_path = '{}/{}/frames/{}'.format(self.base\_path, set, sequence\_name)

frame\_list = [frame for frame in os.listdir(frames\_path) if frame.endswith(".jpg")]

frame\_list.sort(key=lambda f: int(f[:-4]))

frames\_list = [os.path.join(frames\_path, frame) for frame in frame\_list]

return Sequence(sequence\_name, frames\_list, 'trackingnet', ground\_truth\_rect.reshape(-1, 4))

def \_\_len\_\_(self):

return len(self.sequence\_list)

def \_list\_sequences(self, root, set\_ids):

sequence\_list = []

for s in set\_ids:

anno\_dir = os.path.join(root, s, "anno")

sequences\_cur\_set = [(s, os.path.splitext(f)[0]) for f in os.listdir(anno\_dir) if f.endswith('.txt')]

sequence\_list += sequences\_cur\_set

return sequence\_list

import importlib

import os

from collections import OrderedDict

from lib.test.evaluation.environment import env\_settings

import time

import cv2 as cv

from lib.utils.lmdb\_utils import decode\_img

from pathlib import Path

import numpy as np

def trackerlist(name: str, parameter\_name: str, dataset\_name: str, run\_ids = None, display\_name: str = None,

result\_only=False):

if run\_ids is None or isinstance(run\_ids, int):

run\_ids = [run\_ids]

return [Tracker(name, parameter\_name, dataset\_name, run\_id, display\_name, result\_only) for run\_id in run\_ids]

class Tracker:

def \_\_init\_\_(self, name: str, parameter\_name: str, dataset\_name: str, run\_id: int = None, display\_name: str = None,

result\_only=False):

assert run\_id is None or isinstance(run\_id, int)

self.name = name

self.parameter\_name = parameter\_name

self.dataset\_name = dataset\_name

self.run\_id = run\_id

self.display\_name = display\_name

env = env\_settings()

if self.run\_id is None:

self.results\_dir = '{}/{}/{}'.format(env.results\_path, self.name, self.parameter\_name)

else:

self.results\_dir = '{}/{}/{}\_{:03d}'.format(env.results\_path, self.name, self.parameter\_name, self.run\_id)

if result\_only:

self.results\_dir = '{}/{}'.format(env.results\_path, self.name)

tracker\_module\_abspath = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_),

'..', 'tracker', '%s.py' % self.name))

if os.path.isfile(tracker\_module\_abspath):

tracker\_module = importlib.import\_module('lib.test.tracker.{}'.format(self.name))

self.tracker\_class = tracker\_module.get\_tracker\_class()

else:

self.tracker\_class = None

def create\_tracker(self, params):

tracker = self.tracker\_class(params, self.dataset\_name)

return tracker

def run\_sequence(self, seq, debug=None):

params = self.get\_parameters()

debug\_ = debug

if debug is None:

debug\_ = getattr(params, 'debug', 0)

params.debug = debug\_

init\_info = seq.init\_info()

tracker = self.create\_tracker(params)

output = self.\_track\_sequence(tracker, seq, init\_info)

return output

def \_track\_sequence(self, tracker, seq, init\_info):

output = {'target\_bbox': [],

'time': []}

if tracker.params.save\_all\_boxes:

output['all\_boxes'] = []

output['all\_scores'] = []

def \_store\_outputs(tracker\_out: dict, defaults=None):

defaults = {} if defaults is None else defaults

for key in output.keys():

val = tracker\_out.get(key, defaults.get(key, None))

if key in tracker\_out or val is not None:

output[key].append(val)

image = self.\_read\_image(seq.frames[0])

start\_time = time.time()

out = tracker.initialize(image, init\_info)

if out is None:

out = {}

prev\_output = OrderedDict(out)

init\_default = {'target\_bbox': init\_info.get('init\_bbox'),

'time': time.time() - start\_time}

if tracker.params.save\_all\_boxes:

init\_default['all\_boxes'] = out['all\_boxes']

init\_default['all\_scores'] = out['all\_scores']

\_store\_outputs(out, init\_default)

for frame\_num, frame\_path in enumerate(seq.frames[1:], start=1):

image = self.\_read\_image(frame\_path)

start\_time = time.time()

info = seq.frame\_info(frame\_num)

info['previous\_output'] = prev\_output

if len(seq.ground\_truth\_rect) > 1:

info['gt\_bbox'] = seq.ground\_truth\_rect[frame\_num]

out = tracker.track(image, info)

prev\_output = OrderedDict(out)

\_store\_outputs(out, {'time': time.time() - start\_time})

for key in ['target\_bbox', 'all\_boxes', 'all\_scores']:

if key in output and len(output[key]) <= 1:

output.pop(key)

return output

def run\_video(self, videofilepath, optional\_box=None, debug=None, visdom\_info=None, save\_results=False):

params = self.get\_parameters()

debug\_ = debug

if debug is None:

debug\_ = getattr(params, 'debug', 0)

params.debug = debug\_

params.tracker\_name = self.name

params.param\_name = self.parameter\_name

multiobj\_mode = getattr(params, 'multiobj\_mode', getattr(self.tracker\_class, 'multiobj\_mode', 'default'))

if multiobj\_mode == 'default':

tracker = self.create\_tracker(params)

elif multiobj\_mode == 'parallel':

tracker = MultiObjectWrapper(self.tracker\_class, params, self.visdom, fast\_load=True)

else:

raise ValueError('Unknown multi object mode {}'.format(multiobj\_mode))

assert os.path.isfile(videofilepath), "Invalid param {}".format(videofilepath)

", videofilepath must be a valid videofile"

output\_boxes = []

cap = cv.VideoCapture(videofilepath)

display\_name = 'Display: ' + tracker.params.tracker\_name

cv.namedWindow(display\_name, cv.WINDOW\_NORMAL | cv.WINDOW\_KEEPRATIO)

cv.resizeWindow(display\_name, 960, 720)

success, frame = cap.read()

cv.imshow(display\_name, frame)

def \_build\_init\_info(box):

return {'init\_bbox': box}

if success is not True:

print("Read frame from {} failed.".format(videofilepath))

exit(-1)

if optional\_box is not None:

assert isinstance(optional\_box, (list, tuple))

assert len(optional\_box) == 4, "valid box's foramt is [x,y,w,h]"

tracker.initialize(frame, \_build\_init\_info(optional\_box))

output\_boxes.append(optional\_box)

else:

while True:

frame\_disp = frame.copy()

cv.putText(frame\_disp, 'Select target ROI and press ENTER', (20, 30), cv.FONT\_HERSHEY\_COMPLEX\_SMALL,

1.5, (0, 0, 0), 1)

x, y, w, h = cv.selectROI(display\_name, frame\_disp, fromCenter=False)

init\_state = [x, y, w, h]

tracker.initialize(frame, \_build\_init\_info(init\_state))

output\_boxes.append(init\_state)

break

while True:

ret, frame = cap.read()

if frame is None:

break

frame\_disp = frame.copy()

out = tracker.track(frame)

state = [int(s) for s in out['target\_bbox']]

output\_boxes.append(state)

cv.rectangle(frame\_disp, (state[0], state[1]), (state[2] + state[0], state[3] + state[1]),

(0, 255, 0), 5)

font\_color = (0, 0, 0)

cv.putText(frame\_disp, 'Tracking!', (20, 30), cv.FONT\_HERSHEY\_COMPLEX\_SMALL, 1,

font\_color, 1)

cv.putText(frame\_disp, 'Press r to reset', (20, 55), cv.FONT\_HERSHEY\_COMPLEX\_SMALL, 1,

font\_color, 1)

cv.putText(frame\_disp, 'Press q to quit', (20, 80), cv.FONT\_HERSHEY\_COMPLEX\_SMALL, 1,

font\_color, 1)

cv.imshow(display\_name, frame\_disp)

key = cv.waitKey(1)

if key == ord('q'):

break

elif key == ord('r'):

ret, frame = cap.read()

frame\_disp = frame.copy()

cv.putText(frame\_disp, 'Select target ROI and press ENTER', (20, 30), cv.FONT\_HERSHEY\_COMPLEX\_SMALL, 1.5,

(0, 0, 0), 1)

cv.imshow(display\_name, frame\_disp)

x, y, w, h = cv.selectROI(display\_name, frame\_disp, fromCenter=False)

init\_state = [x, y, w, h]

tracker.initialize(frame, \_build\_init\_info(init\_state))

output\_boxes.append(init\_state)

cap.release()

cv.destroyAllWindows()

if save\_results:

if not os.path.exists(self.results\_dir):

os.makedirs(self.results\_dir)

video\_name = Path(videofilepath).stem

base\_results\_path = os.path.join(self.results\_dir, 'video\_{}'.format(video\_name))

tracked\_bb = np.array(output\_boxes).astype(int)

bbox\_file = '{}.txt'.format(base\_results\_path)

np.savetxt(bbox\_file, tracked\_bb, delimiter='\t', fmt='%d')

def get\_parameters(self):

param\_module = importlib.import\_module('lib.test.parameter.{}'.format(self.name))

params = param\_module.parameters(self.parameter\_name)

return params

def \_read\_image(self, image\_file: str):

if isinstance(image\_file, str):

im = cv.imread(image\_file)

return cv.cvtColor(im, cv.COLOR\_BGR2RGB)

elif isinstance(image\_file, list) and len(image\_file) == 2:

return decode\_img(image\_file[0], image\_file[1])

else:

raise ValueError("type of image\_file should be str or list")

import os

import numpy as np

from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList

from lib.test.utils.load\_text import load\_text, load\_str

class TNL2kDataset(BaseDataset):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.base\_path = self.env\_settings.tnl2k\_path

self.sequence\_list = self.\_get\_sequence\_list()

def get\_sequence\_list(self):

return SequenceList([self.\_construct\_sequence(s) for s in self.sequence\_list])

def \_construct\_sequence(self, sequence\_name):

anno\_path = '{}/{}/groundtruth.txt'.format(self.base\_path, sequence\_name)

ground\_truth\_rect = load\_text(str(anno\_path), delimiter=',', dtype=np.float64)

text\_dsp\_path = '{}/{}/language.txt'.format(self.base\_path, sequence\_name)

text\_dsp = load\_str(text\_dsp\_path)

frames\_path = '{}/{}/imgs'.format(self.base\_path, sequence\_name)

frames\_list = [f for f in os.listdir(frames\_path)]

frames\_list = sorted(frames\_list)

frames\_list = ['{}/{}'.format(frames\_path, frame\_i) for frame\_i in frames\_list]

return Sequence(sequence\_name, frames\_list, 'tnl2k', ground\_truth\_rect.reshape(-1, 4), text\_dsp=text\_dsp)

def \_\_len\_\_(self):

return len(self.sequence\_list)

def \_get\_sequence\_list(self):

sequence\_list = []

for seq in os.listdir(self.base\_path):

if os.path.isdir(os.path.join(self.base\_path, seq)):

sequence\_list.append(seq)

return sequence\_list

import numpy as np

from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList

import os

import glob

import six

class TC128Dataset(BaseDataset):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.base\_path = self.env\_settings.tc128\_path

self.anno\_files = sorted(glob.glob(

os.path.join(self.base\_path, '\*/\*\_gt.txt')))

self.seq\_dirs = [os.path.dirname(f) for f in self.anno\_files]

self.seq\_names = [os.path.basename(d) for d in self.seq\_dirs]

self.range\_files = [glob.glob(os.path.join(d, '\*\_frames.txt'))[0] for d in self.seq\_dirs]

def get\_sequence\_list(self):

return SequenceList([self.\_construct\_sequence(s) for s in self.seq\_names])

def \_construct\_sequence(self, sequence\_name):

if isinstance(sequence\_name, six.string\_types):

if not sequence\_name in self.seq\_names:

raise Exception('Sequence {} not found.'.format(sequence\_name))

index = self.seq\_names.index(sequence\_name)

frames = np.loadtxt(self.range\_files[index], dtype=int, delimiter=',')

img\_files = [os.path.join(self.seq\_dirs[index], 'img/%04d.jpg' % f) for f in range(frames[0], frames[1] + 1)]

anno = np.loadtxt(self.anno\_files[index], delimiter=',')

assert len(img\_files) == len(anno)

assert anno.shape[1] == 4

return Sequence(sequence\_name, img\_files, 'tc128', anno.reshape(-1, 4))

def \_\_len\_\_(self):

return len(self.seq\_names)

import numpy as np

from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList

import os

import glob

import six

class TC128CEDataset(BaseDataset):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.base\_path = self.env\_settings.tc128\_path

self.anno\_files = sorted(glob.glob(

os.path.join(self.base\_path, '\*/\*\_gt.txt')))

self.anno\_files = [s for s in self.anno\_files if "\_ce" in s]

self.seq\_dirs = [os.path.dirname(f) for f in self.anno\_files]

self.seq\_names = [os.path.basename(d) for d in self.seq\_dirs]

self.range\_files = [glob.glob(os.path.join(d, '\*\_frames.txt'))[0] for d in self.seq\_dirs]

def get\_sequence\_list(self):

return SequenceList([self.\_construct\_sequence(s) for s in self.seq\_names])

def \_construct\_sequence(self, sequence\_name):

if isinstance(sequence\_name, six.string\_types):

if not sequence\_name in self.seq\_names:

raise Exception('Sequence {} not found.'.format(sequence\_name))

index = self.seq\_names.index(sequence\_name)

frames = np.loadtxt(self.range\_files[index], dtype=int, delimiter=',')

img\_files = [os.path.join(self.seq\_dirs[index], 'img/%04d.jpg' % f) for f in range(frames[0], frames[1] + 1)]

anno = np.loadtxt(self.anno\_files[index], delimiter=',')

assert len(img\_files) == len(anno)

assert anno.shape[1] == 4

return Sequence(sequence\_name, img\_files, 'tc128', anno.reshape(-1, 4))

def \_\_len\_\_(self):

return len(self.seq\_names)

import numpy as np

import multiprocessing

import os

import sys

from itertools import product

from collections import OrderedDict

from lib.test.evaluation import Sequence, Tracker

import torch

def \_save\_tracker\_output(seq: Sequence, tracker: Tracker, output: dict):

if not os.path.exists(tracker.results\_dir):

print("create tracking result dir:", tracker.results\_dir)

os.makedirs(tracker.results\_dir)

if seq.dataset in ['trackingnet', 'got10k']:

if not os.path.exists(os.path.join(tracker.results\_dir, seq.dataset)):

os.makedirs(os.path.join(tracker.results\_dir, seq.dataset))

if seq.dataset in ['trackingnet', 'got10k']:

base\_results\_path = os.path.join(tracker.results\_dir, seq.dataset, seq.name)

else:

base\_results\_path = os.path.join(tracker.results\_dir, seq.name)

def save\_bb(file, data):

tracked\_bb = np.array(data).astype(int)

np.savetxt(file, tracked\_bb, delimiter='\t', fmt='%d')

def save\_time(file, data):

exec\_times = np.array(data).astype(float)

np.savetxt(file, exec\_times, delimiter='\t', fmt='%f')

def save\_score(file, data):

scores = np.array(data).astype(float)

np.savetxt(file, scores, delimiter='\t', fmt='%.2f')

def run\_sequence(seq: Sequence, tracker: Tracker, debug=False, num\_gpu=8):

try:

worker\_name = multiprocessing.current\_process().name

worker\_id = int(worker\_name[worker\_name.find('-') + 1:]) - 1

gpu\_id = worker\_id % num\_gpu

torch.cuda.set\_device(gpu\_id)

except:

pass

import numpy as np

from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList

from lib.test.utils.load\_text import load\_text

class OTBDataset(BaseDataset):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.base\_path = self.env\_settings.otb\_path

self.sequence\_info\_list = self.\_get\_sequence\_info\_list()

def get\_sequence\_list(self):

return SequenceList([self.\_construct\_sequence(s) for s in self.sequence\_info\_list])

def \_construct\_sequence(self, sequence\_info):

sequence\_path = sequence\_info['path']

nz = sequence\_info['nz']

ext = sequence\_info['ext']

start\_frame = sequence\_info['startFrame']

end\_frame = sequence\_info['endFrame']

init\_omit = 0

if 'initOmit' in sequence\_info:

init\_omit = sequence\_info['initOmit']

frames = ['{base\_path}/{sequence\_path}/{frame:0{nz}}.{ext}'.format(base\_path=self.base\_path,

sequence\_path=sequence\_path, frame=frame\_num, nz=nz, ext=ext) for frame\_num in range(start\_frame+init\_omit, end\_frame+1)]

anno\_path = '{}/{}/groundtruth.txt'.format(self.base\_path, sequence\_info['name'])

ground\_truth\_rect = load\_text(str(anno\_path), delimiter=(',', None), dtype=np.float64, backend='numpy')

return Sequence(sequence\_info['name'], frames, 'otb', ground\_truth\_rect[init\_omit:,:],

object\_class=sequence\_info['object\_class'])

def \_\_len\_\_(self):

return len(self.sequence\_info\_list)

def \_get\_sequence\_info\_list(self):

sequence\_info\_list = [

{"name": "Basketball", "path": "Basketball/img", "startFrame": 1, "endFrame": 725, "nz": 4, "ext": "jpg", "anno\_path": "Basketball/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Biker", "path": "Biker/img", "startFrame": 1, "endFrame": 142, "nz": 4, "ext": "jpg", "anno\_path": "Biker/groundtruth\_rect.txt",

"object\_class": "person head"},

{"name": "Bird1", "path": "Bird1/img", "startFrame": 1, "endFrame": 408, "nz": 4, "ext": "jpg", "anno\_path": "Bird1/groundtruth\_rect.txt",

"object\_class": "bird"},

{"name": "Bird2", "path": "Bird2/img", "startFrame": 1, "endFrame": 99, "nz": 4, "ext": "jpg", "anno\_path": "Bird2/groundtruth\_rect.txt",

"object\_class": "bird"},

{"name": "BlurBody", "path": "BlurBody/img", "startFrame": 1, "endFrame": 334, "nz": 4, "ext": "jpg", "anno\_path": "BlurBody/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "BlurCar1", "path": "BlurCar1/img", "startFrame": 247, "endFrame": 988, "nz": 4, "ext": "jpg", "anno\_path": "BlurCar1/groundtruth\_rect.txt",

"object\_class": "car"},

{"name": "BlurCar2", "path": "BlurCar2/img", "startFrame": 1, "endFrame": 585, "nz": 4, "ext": "jpg", "anno\_path": "BlurCar2/groundtruth\_rect.txt",

"object\_class": "car"},

{"name": "BlurCar3", "path": "BlurCar3/img", "startFrame": 3, "endFrame": 359, "nz": 4, "ext": "jpg", "anno\_path": "BlurCar3/groundtruth\_rect.txt",

"object\_class": "car"},

{"name": "BlurCar4", "path": "BlurCar4/img", "startFrame": 18, "endFrame": 397, "nz": 4, "ext": "jpg", "anno\_path": "BlurCar4/groundtruth\_rect.txt",

"object\_class": "car"},

{"name": "BlurFace", "path": "BlurFace/img", "startFrame": 1, "endFrame": 493, "nz": 4, "ext": "jpg", "anno\_path": "BlurFace/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "BlurOwl", "path": "BlurOwl/img", "startFrame": 1, "endFrame": 631, "nz": 4, "ext": "jpg", "anno\_path": "BlurOwl/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Board", "path": "Board/img", "startFrame": 1, "endFrame": 698, "nz": 5, "ext": "jpg", "anno\_path": "Board/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Bolt", "path": "Bolt/img", "startFrame": 1, "endFrame": 350, "nz": 4, "ext": "jpg", "anno\_path": "Bolt/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Bolt2", "path": "Bolt2/img", "startFrame": 1, "endFrame": 293, "nz": 4, "ext": "jpg", "anno\_path": "Bolt2/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Box", "path": "Box/img", "startFrame": 1, "endFrame": 1161, "nz": 4, "ext": "jpg", "anno\_path": "Box/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Boy", "path": "Boy/img", "startFrame": 1, "endFrame": 602, "nz": 4, "ext": "jpg", "anno\_path": "Boy/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Car1", "path": "Car1/img", "startFrame": 1, "endFrame": 1020, "nz": 4, "ext": "jpg", "anno\_path": "Car1/groundtruth\_rect.txt",

"object\_class": "car"},

{"name": "Car2", "path": "Car2/img", "startFrame": 1, "endFrame": 913, "nz": 4, "ext": "jpg", "anno\_path": "Car2/groundtruth\_rect.txt",

"object\_class": "car"},

{"name": "Car24", "path": "Car24/img", "startFrame": 1, "endFrame": 3059, "nz": 4, "ext": "jpg", "anno\_path": "Car24/groundtruth\_rect.txt",

"object\_class": "car"},

{"name": "Car4", "path": "Car4/img", "startFrame": 1, "endFrame": 659, "nz": 4, "ext": "jpg", "anno\_path": "Car4/groundtruth\_rect.txt",

"object\_class": "car"},

{"name": "CarDark", "path": "CarDark/img", "startFrame": 1, "endFrame": 393, "nz": 4, "ext": "jpg", "anno\_path": "CarDark/groundtruth\_rect.txt",

"object\_class": "car"},

{"name": "CarScale", "path": "CarScale/img", "startFrame": 1, "endFrame": 252, "nz": 4, "ext": "jpg", "anno\_path": "CarScale/groundtruth\_rect.txt",

"object\_class": "car"},

{"name": "ClifBar", "path": "ClifBar/img", "startFrame": 1, "endFrame": 472, "nz": 4, "ext": "jpg", "anno\_path": "ClifBar/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Coke", "path": "Coke/img", "startFrame": 1, "endFrame": 291, "nz": 4, "ext": "jpg", "anno\_path": "Coke/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Couple", "path": "Couple/img", "startFrame": 1, "endFrame": 140, "nz": 4, "ext": "jpg", "anno\_path": "Couple/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Coupon", "path": "Coupon/img", "startFrame": 1, "endFrame": 327, "nz": 4, "ext": "jpg", "anno\_path": "Coupon/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Crossing", "path": "Crossing/img", "startFrame": 1, "endFrame": 120, "nz": 4, "ext": "jpg", "anno\_path": "Crossing/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Crowds", "path": "Crowds/img", "startFrame": 1, "endFrame": 347, "nz": 4, "ext": "jpg", "anno\_path": "Crowds/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Dancer", "path": "Dancer/img", "startFrame": 1, "endFrame": 225, "nz": 4, "ext": "jpg", "anno\_path": "Dancer/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Dancer2", "path": "Dancer2/img", "startFrame": 1, "endFrame": 150, "nz": 4, "ext": "jpg", "anno\_path": "Dancer2/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "David", "path": "David/img", "startFrame": 300, "endFrame": 770, "nz": 4, "ext": "jpg", "anno\_path": "David/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "David2", "path": "David2/img", "startFrame": 1, "endFrame": 537, "nz": 4, "ext": "jpg", "anno\_path": "David2/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "David3", "path": "David3/img", "startFrame": 1, "endFrame": 252, "nz": 4, "ext": "jpg", "anno\_path": "David3/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Deer", "path": "Deer/img", "startFrame": 1, "endFrame": 71, "nz": 4, "ext": "jpg", "anno\_path": "Deer/groundtruth\_rect.txt",

"object\_class": "mammal"},

{"name": "Diving", "path": "Diving/img", "startFrame": 1, "endFrame": 215, "nz": 4, "ext": "jpg", "anno\_path": "Diving/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Dog", "path": "Dog/img", "startFrame": 1, "endFrame": 127, "nz": 4, "ext": "jpg", "anno\_path": "Dog/groundtruth\_rect.txt",

"object\_class": "dog"},

{"name": "Dog1", "path": "Dog1/img", "startFrame": 1, "endFrame": 1350, "nz": 4, "ext": "jpg", "anno\_path": "Dog1/groundtruth\_rect.txt",

"object\_class": "dog"},

{"name": "Doll", "path": "Doll/img", "startFrame": 1, "endFrame": 3872, "nz": 4, "ext": "jpg", "anno\_path": "Doll/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "DragonBaby", "path": "DragonBaby/img", "startFrame": 1, "endFrame": 113, "nz": 4, "ext": "jpg", "anno\_path": "DragonBaby/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Dudek", "path": "Dudek/img", "startFrame": 1, "endFrame": 1145, "nz": 4, "ext": "jpg", "anno\_path": "Dudek/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "FaceOcc1", "path": "FaceOcc1/img", "startFrame": 1, "endFrame": 892, "nz": 4, "ext": "jpg", "anno\_path": "FaceOcc1/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "FaceOcc2", "path": "FaceOcc2/img", "startFrame": 1, "endFrame": 812, "nz": 4, "ext": "jpg", "anno\_path": "FaceOcc2/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Fish", "path": "Fish/img", "startFrame": 1, "endFrame": 476, "nz": 4, "ext": "jpg", "anno\_path": "Fish/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "FleetFace", "path": "FleetFace/img", "startFrame": 1, "endFrame": 707, "nz": 4, "ext": "jpg", "anno\_path": "FleetFace/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Football", "path": "Football/img", "startFrame": 1, "endFrame": 362, "nz": 4, "ext": "jpg", "anno\_path": "Football/groundtruth\_rect.txt",

"object\_class": "person head"},

{"name": "Football1", "path": "Football1/img", "startFrame": 1, "endFrame": 74, "nz": 4, "ext": "jpg", "anno\_path": "Football1/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Freeman1", "path": "Freeman1/img", "startFrame": 1, "endFrame": 326, "nz": 4, "ext": "jpg", "anno\_path": "Freeman1/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Freeman3", "path": "Freeman3/img", "startFrame": 1, "endFrame": 460, "nz": 4, "ext": "jpg", "anno\_path": "Freeman3/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Freeman4", "path": "Freeman4/img", "startFrame": 1, "endFrame": 283, "nz": 4, "ext": "jpg", "anno\_path": "Freeman4/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Girl", "path": "Girl/img", "startFrame": 1, "endFrame": 500, "nz": 4, "ext": "jpg", "anno\_path": "Girl/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Girl2", "path": "Girl2/img", "startFrame": 1, "endFrame": 1500, "nz": 4, "ext": "jpg", "anno\_path": "Girl2/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Gym", "path": "Gym/img", "startFrame": 1, "endFrame": 767, "nz": 4, "ext": "jpg", "anno\_path": "Gym/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Human2", "path": "Human2/img", "startFrame": 1, "endFrame": 1128, "nz": 4, "ext": "jpg", "anno\_path": "Human2/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Human3", "path": "Human3/img", "startFrame": 1, "endFrame": 1698, "nz": 4, "ext": "jpg", "anno\_path": "Human3/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Human4\_2", "path": "Human4/img", "startFrame": 1, "endFrame": 667, "nz": 4, "ext": "jpg", "anno\_path": "Human4/groundtruth\_rect.2.txt",

"object\_class": "person"},

{"name": "Human4", "path": "Human4/img", "startFrame": 1, "endFrame": 667, "nz": 4, "ext": "jpg", "anno\_path": "Human4/groundtruth\_rect.2.txt",

"object\_class": "person"},

{"name": "Human5", "path": "Human5/img", "startFrame": 1, "endFrame": 713, "nz": 4, "ext": "jpg", "anno\_path": "Human5/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Human6", "path": "Human6/img", "startFrame": 1, "endFrame": 792, "nz": 4, "ext": "jpg", "anno\_path": "Human6/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Human7", "path": "Human7/img", "startFrame": 1, "endFrame": 250, "nz": 4, "ext": "jpg", "anno\_path": "Human7/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Human8", "path": "Human8/img", "startFrame": 1, "endFrame": 128, "nz": 4, "ext": "jpg", "anno\_path": "Human8/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Human9", "path": "Human9/img", "startFrame": 1, "endFrame": 305, "nz": 4, "ext": "jpg", "anno\_path": "Human9/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Ironman", "path": "Ironman/img", "startFrame": 1, "endFrame": 166, "nz": 4, "ext": "jpg", "anno\_path": "Ironman/groundtruth\_rect.txt",

"object\_class": "person head"},

{"name": "Jogging", "path": "Jogging/img", "startFrame": 1, "endFrame": 307, "nz": 4, "ext": "jpg", "anno\_path": "Jogging/groundtruth\_rect.1.txt",

"object\_class": "person"},

{"name": "Jump", "path": "Jump/img", "startFrame": 1, "endFrame": 122, "nz": 4, "ext": "jpg", "anno\_path": "Jump/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Jumping", "path": "Jumping/img", "startFrame": 1, "endFrame": 313, "nz": 4, "ext": "jpg", "anno\_path": "Jumping/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "KiteSurf", "path": "KiteSurf/img", "startFrame": 1, "endFrame": 84, "nz": 4, "ext": "jpg", "anno\_path": "KiteSurf/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Lemming", "path": "Lemming/img", "startFrame": 1, "endFrame": 1336, "nz": 4, "ext": "jpg", "anno\_path": "Lemming/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Liquor", "path": "Liquor/img", "startFrame": 1, "endFrame": 1741, "nz": 4, "ext": "jpg", "anno\_path": "Liquor/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Man", "path": "Man/img", "startFrame": 1, "endFrame": 134, "nz": 4, "ext": "jpg", "anno\_path": "Man/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Matrix", "path": "Matrix/img", "startFrame": 1, "endFrame": 100, "nz": 4, "ext": "jpg", "anno\_path": "Matrix/groundtruth\_rect.txt",

"object\_class": "person head"},

{"name": "Mhyang", "path": "Mhyang/img", "startFrame": 1, "endFrame": 1490, "nz": 4, "ext": "jpg", "anno\_path": "Mhyang/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "MotorRolling", "path": "MotorRolling/img", "startFrame": 1, "endFrame": 164, "nz": 4, "ext": "jpg", "anno\_path": "MotorRolling/groundtruth\_rect.txt",

"object\_class": "vehicle"},

{"name": "MountainBike", "path": "MountainBike/img", "startFrame": 1, "endFrame": 228, "nz": 4, "ext": "jpg", "anno\_path": "MountainBike/groundtruth\_rect.txt",

"object\_class": "bicycle"},

{"name": "Panda", "path": "Panda/img", "startFrame": 1, "endFrame": 1000, "nz": 4, "ext": "jpg", "anno\_path": "Panda/groundtruth\_rect.txt",

"object\_class": "mammal"},

{"name": "RedTeam", "path": "RedTeam/img", "startFrame": 1, "endFrame": 1918, "nz": 4, "ext": "jpg", "anno\_path": "RedTeam/groundtruth\_rect.txt",

"object\_class": "vehicle"},

{"name": "Rubik", "path": "Rubik/img", "startFrame": 1, "endFrame": 1997, "nz": 4, "ext": "jpg", "anno\_path": "Rubik/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Shaking", "path": "Shaking/img", "startFrame": 1, "endFrame": 365, "nz": 4, "ext": "jpg", "anno\_path": "Shaking/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Singer1", "path": "Singer1/img", "startFrame": 1, "endFrame": 351, "nz": 4, "ext": "jpg", "anno\_path": "Singer1/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Singer2", "path": "Singer2/img", "startFrame": 1, "endFrame": 366, "nz": 4, "ext": "jpg", "anno\_path": "Singer2/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Skater", "path": "Skater/img", "startFrame": 1, "endFrame": 160, "nz": 4, "ext": "jpg", "anno\_path": "Skater/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Skater2", "path": "Skater2/img", "startFrame": 1, "endFrame": 435, "nz": 4, "ext": "jpg", "anno\_path": "Skater2/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Skating1", "path": "Skating1/img", "startFrame": 1, "endFrame": 400, "nz": 4, "ext": "jpg", "anno\_path": "Skating1/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Skating2", "path": "Skating2/img", "startFrame": 1, "endFrame": 473, "nz": 4, "ext": "jpg", "anno\_path": "Skating2/groundtruth\_rect.1.txt",

"object\_class": "person"},

{"name": "Skating2\_1", "path": "Skating2/img", "startFrame": 1, "endFrame": 473, "nz": 4, "ext": "jpg", "anno\_path": "Skating2/groundtruth\_rect.1.txt",

"object\_class": "person"},

{"name": "Skating2\_2", "path": "Skating2/img", "startFrame": 1, "endFrame": 473, "nz": 4, "ext": "jpg", "anno\_path": "Skating2/groundtruth\_rect.2.txt",

"object\_class": "person"},

{"name": "Skiing", "path": "Skiing/img", "startFrame": 1, "endFrame": 81, "nz": 4, "ext": "jpg", "anno\_path": "Skiing/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Soccer", "path": "Soccer/img", "startFrame": 1, "endFrame": 392, "nz": 4, "ext": "jpg", "anno\_path": "Soccer/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Subway", "path": "Subway/img", "startFrame": 1, "endFrame": 175, "nz": 4, "ext": "jpg", "anno\_path": "Subway/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Surfer", "path": "Surfer/img", "startFrame": 1, "endFrame": 376, "nz": 4, "ext": "jpg", "anno\_path": "Surfer/groundtruth\_rect.txt",

"object\_class": "person head"},

{"name": "Suv", "path": "Suv/img", "startFrame": 1, "endFrame": 945, "nz": 4, "ext": "jpg", "anno\_path": "Suv/groundtruth\_rect.txt",

"object\_class": "car"},

{"name": "Sylvester", "path": "Sylvester/img", "startFrame": 1, "endFrame": 1345, "nz": 4, "ext": "jpg", "anno\_path": "Sylvester/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Tiger1", "path": "Tiger1/img", "startFrame": 1, "endFrame": 354, "nz": 4, "ext": "jpg", "anno\_path": "Tiger1/groundtruth\_rect.txt", "initOmit": 5,

"object\_class": "other"},

{"name": "Tiger2", "path": "Tiger2/img", "startFrame": 1, "endFrame": 365, "nz": 4, "ext": "jpg", "anno\_path": "Tiger2/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Toy", "path": "Toy/img", "startFrame": 1, "endFrame": 271, "nz": 4, "ext": "jpg", "anno\_path": "Toy/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Trans", "path": "Trans/img", "startFrame": 1, "endFrame": 124, "nz": 4, "ext": "jpg", "anno\_path": "Trans/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Trellis", "path": "Trellis/img", "startFrame": 1, "endFrame": 569, "nz": 4, "ext": "jpg", "anno\_path": "Trellis/groundtruth\_rect.txt",

"object\_class": "face"},

{"name": "Twinnings", "path": "Twinnings/img", "startFrame": 1, "endFrame": 472, "nz": 4, "ext": "jpg", "anno\_path": "Twinnings/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Vase", "path": "Vase/img", "startFrame": 1, "endFrame": 271, "nz": 4, "ext": "jpg", "anno\_path": "Vase/groundtruth\_rect.txt",

"object\_class": "other"},

{"name": "Walking", "path": "Walking/img", "startFrame": 1, "endFrame": 412, "nz": 4, "ext": "jpg", "anno\_path": "Walking/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Walking2", "path": "Walking2/img", "startFrame": 1, "endFrame": 500, "nz": 4, "ext": "jpg", "anno\_path": "Walking2/groundtruth\_rect.txt",

"object\_class": "person"},

{"name": "Woman", "path": "Woman/img", "startFrame": 1, "endFrame": 597, "nz": 4, "ext": "jpg", "anno\_path": "Woman/groundtruth\_rect.txt",

"object\_class": "person"}

]

return sequence\_info\_list

import numpy as np

from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList

from lib.test.utils.load\_text import load\_text

class NFSDataset(BaseDataset):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.base\_path = self.env\_settings.nfs\_path

self.sequence\_info\_list = self.\_get\_sequence\_info\_list()

def get\_sequence\_list(self):

return SequenceList([self.\_construct\_sequence(s) for s in self.sequence\_info\_list])

def \_construct\_sequence(self, sequence\_info):

sequence\_path = sequence\_info['path']

nz = sequence\_info['nz']

ext = sequence\_info['ext']

start\_frame = sequence\_info['startFrame']

end\_frame = sequence\_info['endFrame']

init\_omit = 0

if 'initOmit' in sequence\_info:

init\_omit = sequence\_info['initOmit']

frames = ['{base\_path}/{sequence\_path}/{frame:0{nz}}.{ext}'.format(base\_path=self.base\_path,

sequence\_path=sequence\_path, frame=frame\_num, nz=nz, ext=ext) for frame\_num in range(start\_frame+init\_omit, end\_frame+1)]

anno\_path = f"{self.base\_path}/{sequence\_info['name'][4:]}/30/groundtruth.txt"

ground\_truth\_rect = load\_text(str(anno\_path), delimiter=',', dtype=np.float64)

return Sequence(sequence\_info['name'][4:], frames, 'nfs', ground\_truth\_rect[init\_omit:,:],

object\_class=sequence\_info['object\_class'])

def \_\_len\_\_(self):

return len(self.sequence\_info\_list)

def \_get\_sequence\_info\_list(self):

sequence\_info\_list = [

{"name": "nfs\_Gymnastics", "path": "sequences/Gymnastics", "startFrame": 1, "endFrame": 368, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_Gymnastics.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_MachLoop\_jet", "path": "sequences/MachLoop\_jet", "startFrame": 1, "endFrame": 99, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_MachLoop\_jet.txt", "object\_class": "aircraft", 'occlusion': False},

{"name": "nfs\_Skiing\_red", "path": "sequences/Skiing\_red", "startFrame": 1, "endFrame": 69, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_Skiing\_red.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_Skydiving", "path": "sequences/Skydiving", "startFrame": 1, "endFrame": 196, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_Skydiving.txt", "object\_class": "person", 'occlusion': True},

{"name": "nfs\_airboard\_1", "path": "sequences/airboard\_1", "startFrame": 1, "endFrame": 425, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_airboard\_1.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_airplane\_landing", "path": "sequences/airplane\_landing", "startFrame": 1, "endFrame": 81, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_airplane\_landing.txt", "object\_class": "aircraft", 'occlusion': False},

{"name": "nfs\_airtable\_3", "path": "sequences/airtable\_3", "startFrame": 1, "endFrame": 482, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_airtable\_3.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_basketball\_1", "path": "sequences/basketball\_1", "startFrame": 1, "endFrame": 282, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_basketball\_1.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_basketball\_2", "path": "sequences/basketball\_2", "startFrame": 1, "endFrame": 102, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_basketball\_2.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_basketball\_3", "path": "sequences/basketball\_3", "startFrame": 1, "endFrame": 421, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_basketball\_3.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_basketball\_6", "path": "sequences/basketball\_6", "startFrame": 1, "endFrame": 224, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_basketball\_6.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_basketball\_7", "path": "sequences/basketball\_7", "startFrame": 1, "endFrame": 240, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_basketball\_7.txt", "object\_class": "person", 'occlusion': True},

{"name": "nfs\_basketball\_player", "path": "sequences/basketball\_player", "startFrame": 1, "endFrame": 369, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_basketball\_player.txt", "object\_class": "person", 'occlusion': True},

{"name": "nfs\_basketball\_player\_2", "path": "sequences/basketball\_player\_2", "startFrame": 1, "endFrame": 437, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_basketball\_player\_2.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_beach\_flipback\_person", "path": "sequences/beach\_flipback\_person", "startFrame": 1, "endFrame": 61, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_beach\_flipback\_person.txt", "object\_class": "person head", 'occlusion': False},

{"name": "nfs\_bee", "path": "sequences/bee", "startFrame": 1, "endFrame": 45, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_bee.txt", "object\_class": "insect", 'occlusion': False},

{"name": "nfs\_biker\_acrobat", "path": "sequences/biker\_acrobat", "startFrame": 1, "endFrame": 128, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_biker\_acrobat.txt", "object\_class": "bicycle", 'occlusion': False},

{"name": "nfs\_biker\_all\_1", "path": "sequences/biker\_all\_1", "startFrame": 1, "endFrame": 113, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_biker\_all\_1.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_biker\_head\_2", "path": "sequences/biker\_head\_2", "startFrame": 1, "endFrame": 132, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_biker\_head\_2.txt", "object\_class": "person head", 'occlusion': False},

{"name": "nfs\_biker\_head\_3", "path": "sequences/biker\_head\_3", "startFrame": 1, "endFrame": 254, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_biker\_head\_3.txt", "object\_class": "person head", 'occlusion': False},

{"name": "nfs\_biker\_upper\_body", "path": "sequences/biker\_upper\_body", "startFrame": 1, "endFrame": 194, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_biker\_upper\_body.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_biker\_whole\_body", "path": "sequences/biker\_whole\_body", "startFrame": 1, "endFrame": 572, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_biker\_whole\_body.txt", "object\_class": "person", 'occlusion': True},

{"name": "nfs\_billiard\_2", "path": "sequences/billiard\_2", "startFrame": 1, "endFrame": 604, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_billiard\_2.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_billiard\_3", "path": "sequences/billiard\_3", "startFrame": 1, "endFrame": 698, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_billiard\_3.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_billiard\_6", "path": "sequences/billiard\_6", "startFrame": 1, "endFrame": 771, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_billiard\_6.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_billiard\_7", "path": "sequences/billiard\_7", "startFrame": 1, "endFrame": 724, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_billiard\_7.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_billiard\_8", "path": "sequences/billiard\_8", "startFrame": 1, "endFrame": 778, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_billiard\_8.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_bird\_2", "path": "sequences/bird\_2", "startFrame": 1, "endFrame": 476, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_bird\_2.txt", "object\_class": "bird", 'occlusion': False},

{"name": "nfs\_book", "path": "sequences/book", "startFrame": 1, "endFrame": 288, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_book.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_bottle", "path": "sequences/bottle", "startFrame": 1, "endFrame": 2103, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_bottle.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_bowling\_1", "path": "sequences/bowling\_1", "startFrame": 1, "endFrame": 303, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_bowling\_1.txt", "object\_class": "ball", 'occlusion': True},

{"name": "nfs\_bowling\_2", "path": "sequences/bowling\_2", "startFrame": 1, "endFrame": 710, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_bowling\_2.txt", "object\_class": "ball", 'occlusion': True},

{"name": "nfs\_bowling\_3", "path": "sequences/bowling\_3", "startFrame": 1, "endFrame": 271, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_bowling\_3.txt", "object\_class": "ball", 'occlusion': True},

{"name": "nfs\_bowling\_6", "path": "sequences/bowling\_6", "startFrame": 1, "endFrame": 260, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_bowling\_6.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_bowling\_ball", "path": "sequences/bowling\_ball", "startFrame": 1, "endFrame": 275, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_bowling\_ball.txt", "object\_class": "ball", 'occlusion': True},

{"name": "nfs\_bunny", "path": "sequences/bunny", "startFrame": 1, "endFrame": 705, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_bunny.txt", "object\_class": "mammal", 'occlusion': False},

{"name": "nfs\_car", "path": "sequences/car", "startFrame": 1, "endFrame": 2020, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_car.txt", "object\_class": "car", 'occlusion': True},

{"name": "nfs\_car\_camaro", "path": "sequences/car\_camaro", "startFrame": 1, "endFrame": 36, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_car\_camaro.txt", "object\_class": "car", 'occlusion': False},

{"name": "nfs\_car\_drifting", "path": "sequences/car\_drifting", "startFrame": 1, "endFrame": 173, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_car\_drifting.txt", "object\_class": "car", 'occlusion': False},

{"name": "nfs\_car\_jumping", "path": "sequences/car\_jumping", "startFrame": 1, "endFrame": 22, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_car\_jumping.txt", "object\_class": "car", 'occlusion': False},

{"name": "nfs\_car\_rc\_rolling", "path": "sequences/car\_rc\_rolling", "startFrame": 1, "endFrame": 62, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_car\_rc\_rolling.txt", "object\_class": "car", 'occlusion': False},

{"name": "nfs\_car\_rc\_rotating", "path": "sequences/car\_rc\_rotating", "startFrame": 1, "endFrame": 80, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_car\_rc\_rotating.txt", "object\_class": "car", 'occlusion': False},

{"name": "nfs\_car\_side", "path": "sequences/car\_side", "startFrame": 1, "endFrame": 108, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_car\_side.txt", "object\_class": "car", 'occlusion': False},

{"name": "nfs\_car\_white", "path": "sequences/car\_white", "startFrame": 1, "endFrame": 2063, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_car\_white.txt", "object\_class": "car", 'occlusion': False},

{"name": "nfs\_cheetah", "path": "sequences/cheetah", "startFrame": 1, "endFrame": 167, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_cheetah.txt", "object\_class": "mammal", 'occlusion': True},

{"name": "nfs\_cup", "path": "sequences/cup", "startFrame": 1, "endFrame": 1281, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_cup.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_cup\_2", "path": "sequences/cup\_2", "startFrame": 1, "endFrame": 182, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_cup\_2.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_dog", "path": "sequences/dog", "startFrame": 1, "endFrame": 1030, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_dog.txt", "object\_class": "dog", 'occlusion': True},

{"name": "nfs\_dog\_1", "path": "sequences/dog\_1", "startFrame": 1, "endFrame": 168, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_dog\_1.txt", "object\_class": "dog", 'occlusion': False},

{"name": "nfs\_dog\_3", "path": "sequences/dog\_3", "startFrame": 1, "endFrame": 200, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_dog\_3.txt", "object\_class": "dog", 'occlusion': False},

{"name": "nfs\_dogs", "path": "sequences/dogs", "startFrame": 1, "endFrame": 198, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_dogs.txt", "object\_class": "dog", 'occlusion': True},

{"name": "nfs\_dollar", "path": "sequences/dollar", "startFrame": 1, "endFrame": 1426, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_dollar.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_drone", "path": "sequences/drone", "startFrame": 1, "endFrame": 70, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_drone.txt", "object\_class": "aircraft", 'occlusion': False},

{"name": "nfs\_ducks\_lake", "path": "sequences/ducks\_lake", "startFrame": 1, "endFrame": 107, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_ducks\_lake.txt", "object\_class": "bird", 'occlusion': False},

{"name": "nfs\_exit", "path": "sequences/exit", "startFrame": 1, "endFrame": 359, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_exit.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_first", "path": "sequences/first", "startFrame": 1, "endFrame": 435, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_first.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_flower", "path": "sequences/flower", "startFrame": 1, "endFrame": 448, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_flower.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_footbal\_skill", "path": "sequences/footbal\_skill", "startFrame": 1, "endFrame": 131, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_footbal\_skill.txt", "object\_class": "ball", 'occlusion': True},

{"name": "nfs\_helicopter", "path": "sequences/helicopter", "startFrame": 1, "endFrame": 310, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_helicopter.txt", "object\_class": "aircraft", 'occlusion': False},

{"name": "nfs\_horse\_jumping", "path": "sequences/horse\_jumping", "startFrame": 1, "endFrame": 117, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_horse\_jumping.txt", "object\_class": "horse", 'occlusion': True},

{"name": "nfs\_horse\_running", "path": "sequences/horse\_running", "startFrame": 1, "endFrame": 139, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_horse\_running.txt", "object\_class": "horse", 'occlusion': False},

{"name": "nfs\_iceskating\_6", "path": "sequences/iceskating\_6", "startFrame": 1, "endFrame": 603, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_iceskating\_6.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_jellyfish\_5", "path": "sequences/jellyfish\_5", "startFrame": 1, "endFrame": 746, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_jellyfish\_5.txt", "object\_class": "invertebrate", 'occlusion': False},

{"name": "nfs\_kid\_swing", "path": "sequences/kid\_swing", "startFrame": 1, "endFrame": 169, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_kid\_swing.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_motorcross", "path": "sequences/motorcross", "startFrame": 1, "endFrame": 39, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_motorcross.txt", "object\_class": "vehicle", 'occlusion': True},

{"name": "nfs\_motorcross\_kawasaki", "path": "sequences/motorcross\_kawasaki", "startFrame": 1, "endFrame": 65, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_motorcross\_kawasaki.txt", "object\_class": "vehicle", 'occlusion': False},

{"name": "nfs\_parkour", "path": "sequences/parkour", "startFrame": 1, "endFrame": 58, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_parkour.txt", "object\_class": "person head", 'occlusion': False},

{"name": "nfs\_person\_scooter", "path": "sequences/person\_scooter", "startFrame": 1, "endFrame": 413, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_person\_scooter.txt", "object\_class": "person", 'occlusion': True},

{"name": "nfs\_pingpong\_2", "path": "sequences/pingpong\_2", "startFrame": 1, "endFrame": 1277, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_pingpong\_2.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_pingpong\_7", "path": "sequences/pingpong\_7", "startFrame": 1, "endFrame": 1290, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_pingpong\_7.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_pingpong\_8", "path": "sequences/pingpong\_8", "startFrame": 1, "endFrame": 296, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_pingpong\_8.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_purse", "path": "sequences/purse", "startFrame": 1, "endFrame": 968, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_purse.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_rubber", "path": "sequences/rubber", "startFrame": 1, "endFrame": 1328, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_rubber.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_running", "path": "sequences/running", "startFrame": 1, "endFrame": 677, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_running.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_running\_100\_m", "path": "sequences/running\_100\_m", "startFrame": 1, "endFrame": 313, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_running\_100\_m.txt", "object\_class": "person", 'occlusion': True},

{"name": "nfs\_running\_100\_m\_2", "path": "sequences/running\_100\_m\_2", "startFrame": 1, "endFrame": 337, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_running\_100\_m\_2.txt", "object\_class": "person", 'occlusion': True},

{"name": "nfs\_running\_2", "path": "sequences/running\_2", "startFrame": 1, "endFrame": 363, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_running\_2.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_shuffleboard\_1", "path": "sequences/shuffleboard\_1", "startFrame": 1, "endFrame": 42, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_shuffleboard\_1.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_shuffleboard\_2", "path": "sequences/shuffleboard\_2", "startFrame": 1, "endFrame": 41, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_shuffleboard\_2.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_shuffleboard\_4", "path": "sequences/shuffleboard\_4", "startFrame": 1, "endFrame": 62, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_shuffleboard\_4.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_shuffleboard\_5", "path": "sequences/shuffleboard\_5", "startFrame": 1, "endFrame": 32, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_shuffleboard\_5.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_shuffleboard\_6", "path": "sequences/shuffleboard\_6", "startFrame": 1, "endFrame": 52, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_shuffleboard\_6.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_shuffletable\_2", "path": "sequences/shuffletable\_2", "startFrame": 1, "endFrame": 372, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_shuffletable\_2.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_shuffletable\_3", "path": "sequences/shuffletable\_3", "startFrame": 1, "endFrame": 368, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_shuffletable\_3.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_shuffletable\_4", "path": "sequences/shuffletable\_4", "startFrame": 1, "endFrame": 101, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_shuffletable\_4.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_ski\_long", "path": "sequences/ski\_long", "startFrame": 1, "endFrame": 274, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_ski\_long.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_soccer\_ball", "path": "sequences/soccer\_ball", "startFrame": 1, "endFrame": 163, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_soccer\_ball.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_soccer\_ball\_2", "path": "sequences/soccer\_ball\_2", "startFrame": 1, "endFrame": 1934, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_soccer\_ball\_2.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_soccer\_ball\_3", "path": "sequences/soccer\_ball\_3", "startFrame": 1, "endFrame": 1381, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_soccer\_ball\_3.txt", "object\_class": "ball", 'occlusion': False},

{"name": "nfs\_soccer\_player\_2", "path": "sequences/soccer\_player\_2", "startFrame": 1, "endFrame": 475, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_soccer\_player\_2.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_soccer\_player\_3", "path": "sequences/soccer\_player\_3", "startFrame": 1, "endFrame": 319, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_soccer\_player\_3.txt", "object\_class": "person", 'occlusion': True},

{"name": "nfs\_stop\_sign", "path": "sequences/stop\_sign", "startFrame": 1, "endFrame": 302, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_stop\_sign.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_suv", "path": "sequences/suv", "startFrame": 1, "endFrame": 2584, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_suv.txt", "object\_class": "car", 'occlusion': False},

{"name": "nfs\_tiger", "path": "sequences/tiger", "startFrame": 1, "endFrame": 1556, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_tiger.txt", "object\_class": "mammal", 'occlusion': False},

{"name": "nfs\_walking", "path": "sequences/walking", "startFrame": 1, "endFrame": 555, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_walking.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_walking\_3", "path": "sequences/walking\_3", "startFrame": 1, "endFrame": 1427, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_walking\_3.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_water\_ski\_2", "path": "sequences/water\_ski\_2", "startFrame": 1, "endFrame": 47, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_water\_ski\_2.txt", "object\_class": "person", 'occlusion': False},

{"name": "nfs\_yoyo", "path": "sequences/yoyo", "startFrame": 1, "endFrame": 67, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_yoyo.txt", "object\_class": "other", 'occlusion': False},

{"name": "nfs\_zebra\_fish", "path": "sequences/zebra\_fish", "startFrame": 1, "endFrame": 671, "nz": 5, "ext": "jpg", "anno\_path": "anno/nfs\_zebra\_fish.txt", "object\_class": "fish", 'occlusion': False},

]

return sequence\_info\_list

from lib.test.evaluation.environment import EnvSettings

from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList

from lib.utils.lmdb\_utils import \*

class LaSOTlmdbDataset(BaseDataset):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.base\_path = self.env\_settings.lasot\_lmdb\_path

self.sequence\_list = self.\_get\_sequence\_list()

self.clean\_list = self.clean\_seq\_list()

def clean\_seq\_list(self):

clean\_lst = []

for i in range(len(self.sequence\_list)):

cls, \_ = self.sequence\_list[i].split('-')

clean\_lst.append(cls)

return clean\_lst

def get\_sequence\_list(self):

return SequenceList([self.\_construct\_sequence(s) for s in self.sequence\_list])

def \_\_len\_\_(self):

return len(self.sequence\_list)

def \_get\_sequence\_list(self):

import numpy as np

from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList

from lib.test.utils.load\_text import load\_text

class LaSOTExtensionSubsetDataset(BaseDataset):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.base\_path = self.env\_settings.lasot\_extension\_subset\_path

self.sequence\_list = self.\_get\_sequence\_list()

self.clean\_list = self.clean\_seq\_list()

def clean\_seq\_list(self):

clean\_lst = []

for i in range(len(self.sequence\_list)):

cls, \_ = self.sequence\_list[i].split('-')

clean\_lst.append(cls)

return clean\_lst

def get\_sequence\_list(self):

return SequenceList([self.\_construct\_sequence(s) for s in self.sequence\_list])

def \_construct\_sequence(self, sequence\_name):

class\_name = sequence\_name.split('-')[0]

anno\_path = '{}/{}/{}/groundtruth.txt'.format(self.base\_path, class\_name, sequence\_name)

ground\_truth\_rect = load\_text(str(anno\_path), delimiter=',', dtype=np.float64)

occlusion\_label\_path = '{}/{}/{}/full\_occlusion.txt'.format(self.base\_path, class\_name, sequence\_name)

full\_occlusion = load\_text(str(occlusion\_label\_path), delimiter=',', dtype=np.float64, backend='numpy')

out\_of\_view\_label\_path = '{}/{}/{}/out\_of\_view.txt'.format(self.base\_path, class\_name, sequence\_name)

out\_of\_view = load\_text(str(out\_of\_view\_label\_path), delimiter=',', dtype=np.float64, backend='numpy')

target\_visible = np.logical\_and(full\_occlusion == 0, out\_of\_view == 0)

frames\_path = '{}/{}/{}/img'.format(self.base\_path, class\_name, sequence\_name)

frames\_list = ['{}/{:08d}.jpg'.format(frames\_path, frame\_number) for frame\_number in range(1, ground\_truth\_rect.shape[0] + 1)]

target\_class = class\_name

return Sequence(sequence\_name, frames\_list, 'lasot\_extension\_subset', ground\_truth\_rect.reshape(-1, 4),

object\_class=target\_class, target\_visible=target\_visible)

def \_\_len\_\_(self):

return len(self.sequence\_list)

def \_get\_sequence\_list(self):

return sequence\_list

import numpy as np

from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList

from lib.test.utils.load\_text import load\_text

class LaSOTDataset(BaseDataset):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.base\_path = self.env\_settings.lasot\_path

self.sequence\_list = self.\_get\_sequence\_list()

self.clean\_list = self.clean\_seq\_list()

def clean\_seq\_list(self):

clean\_lst = []

for i in range(len(self.sequence\_list)):

cls, \_ = self.sequence\_list[i].split('-')

clean\_lst.append(cls)

return clean\_lst

def get\_sequence\_list(self):

return SequenceList([self.\_construct\_sequence(s) for s in self.sequence\_list])

def \_\_len\_\_(self):

return len(self.sequence\_list)

import numpy as np

from lib.test.evaluation.data import Sequence, BaseDataset, SequenceList

from lib.test.utils.load\_text import load\_text

import os

class ITBDataset(BaseDataset):

def \_\_len\_\_(self):

return len(self.sequence\_info\_list)

def get\_fileNames(self, rootdir):

fs = []

fs\_all = []

for root, dirs, files in os.walk(rootdir, topdown=True):

files.sort()

files.sort(key=len)

if files is not None:

for name in files:

\_, ending = os.path.splitext(name)

if ending == ".jpg":

\_, root\_ = os.path.split(root)

fs.append(os.path.join(root\_, name))

fs\_all.append(os.path.join(root, name))

return fs\_all, fs

def \_get\_sequence\_info\_list(self, base\_path):

sequence\_info\_list = []

for scene in os.listdir(base\_path):

if '.' in scene:

continue

videos = os.listdir(os.path.join(base\_path, scene))

for video in videos:

\_, fs = self.get\_fileNames(os.path.join(base\_path, scene, video))

video\_tmp = {"name": video, "path": scene + '/' + video, "startFrame": 1, "endFrame": len(fs),

"nz": len(fs[0].split('/')[-1].split('.')[0]), "ext": "jpg",

"anno\_path": scene + '/' + video + "/groundtruth.txt",

"object\_class": "unknown"}

sequence\_info\_list.append(video\_tmp)

return sequence\_info\_list