STYLE.CSS

body {

background-color: #1B1919;

}

.no-display {

display: none;

}

.center {

/\* Espaciado desde el objeto hacia afuera \*/

margin: auto;

/\* espacio del recuadro hacia adentro (vertical horizontal)\*/

padding: 10px 50px;

text-align: center;

font-size: 14px;

color: #F1EFFA;

}

.title {

font-size: 30px;

margin-top: 1em;

margin-bottom: 1em;

color: #F1EFFA;

}

.content {

margin-top: 10em;

}

.analyze {

margin-top: 5em;

}

.upload-label {

padding: 10px;

font-size: 12px;

}

.result-label {

margin-top: 0.5em;

padding: 10px;

font-size: 13px;

}

button.choose-file-button {

width: 200px;

height: 40px;

border-radius: 10px;

background-color: #ffffff;

border: solid 1px #2000A2;

font-size: 13px;

color: #3300FF;

}

button.analyze-button {

width: 200px;

height: 40px;

border: solid 1px #7052CB;

border-radius: 2px;

background-color: #3300FF;

font-size: 13px;

color: #ffffff;

}

button:focus {

outline: 0;

}

button:hover {

box-shadow: 1px 1px 10px #888;

}

INDEX.HTML

<html lang='en'>

<head>

<meta charset='utf-8'>

<link rel='stylesheet' href='../static/style.css'>

<script src='../static/client.js'></script>

</head>

<body>

<div>

<div class='center'>

<div class='title'>Pneumonia Detection <img src="https://prod-images-static.radiopaedia.org/images/25074603/4994014ef5c834e4803541aa1dc874\_jumbo.jpeg" height="30px" style="display:inline" /></div>

<p>

Submit <Strong>chest X-ray images </strong> so we could analize if you have, or have not, <strong>Pneumonia </strong>

and if you do have it, we will tell you the probable cause of your disseas. Here, we analize <Strong> Stress-Smoking </strong>,

<Strong> Stress-Smoking </strong>, <Strong> COVID-19 </strong>, <Strong> SARS </strong>, <Strong> Streptococcus</strong> or other

<Strong> Virus</strong> or <Strong> Bacteria </strong>

</p>

<div class='content'>

<div class='no-display'>

<input id='file-input'

class='no-display'

type='file'

name='file'

accept='image/\*'

onchange='showPicked(this)'>

</div>

<button class='choose-file-button' type='button' onclick='showPicker()'>Select Image</button>

<div class='upload-label'>

<label id='upload-label'>No file chosen</label>

</div>

<div>

<img id='image-picked' class='no-display' alt='Chosen Image' height='200'>

</div>

<div class='analyze'>

<button id='analyze-button' class='analyze-button' type='button' onclick='analyze()'>Analyze</button>

</div>

<div class='result-label'>

<label id='result-label'></label>

</div>

</div>

</div>

</div>

</body>

</html>

SERVER.PY

import aiohttp

import asyncio

import uvicorn

from fastai import \*

from fastai.vision import \*

from io import BytesIO

from starlette.applications import Starlette

from starlette.middleware.cors import CORSMiddleware

from starlette.responses import HTMLResponse, JSONResponse

from starlette.staticfiles import StaticFiles

export\_file\_url = 'https://drive.google.com/uc?export=download&id=1X9bxZBZ8HhaPAN0CF1WsNSXLLKnph6Cw'

export\_file\_name = 'xray\_covi19.pkl'

classes = ['', 'ARDS', 'COVID19', 'Normal', 'Pnemonia', 'SARS', 'Streptococcus', 'StressSmoking', 'Virus', 'bacteria']

path = Path(\_\_file\_\_).parent

app = Starlette()

app.add\_middleware(CORSMiddleware, allow\_origins=['\*'], allow\_headers=['X-Requested-With', 'Content-Type'])

app.mount('/static', StaticFiles(directory='app/static'))

async def download\_file(url, dest):

if dest.exists(): return

async with aiohttp.ClientSession() as session:

async with session.get(url) as response:

data = await response.read()

with open(dest, 'wb') as f:

f.write(data)

async def setup\_learner():

await download\_file(export\_file\_url, path / export\_file\_name)

try:

learn = load\_learner(path, export\_file\_name)

return learn

except RuntimeError as e:

if len(e.args) > 0 and 'CPU-only machine' in e.args[0]:

print(e)

message = "\n\nThis model was trained with an old version of fastai and will not work in a CPU environment.\n\nPlease update the fastai library in your training environment and export your model again.\n\nSee instructions for 'Returning to work' at https://course.fast.ai."

raise RuntimeError(message)

else:

raise

loop = asyncio.get\_event\_loop()

tasks = [asyncio.ensure\_future(setup\_learner())]

learn = loop.run\_until\_complete(asyncio.gather(\*tasks))[0]

loop.close()

@app.route('/')

async def homepage(request):

html\_file = path / 'view' / 'index.html'

return HTMLResponse(html\_file.open().read())

@app.route('/analyze', methods=['POST'])

async def analyze(request):

img\_data = await request.form()

img\_bytes = await (img\_data['file'].read())

img = open\_image(BytesIO(img\_bytes))

prediction = learn.predict(img)[0]

return JSONResponse({'result': str(prediction)})

if \_\_name\_\_ == '\_\_main\_\_':

if 'serve' in sys.argv:

uvicorn.run(app=app, host='0.0.0.0', port=5000, log\_level="info")