



IMAGE RECOGNITION

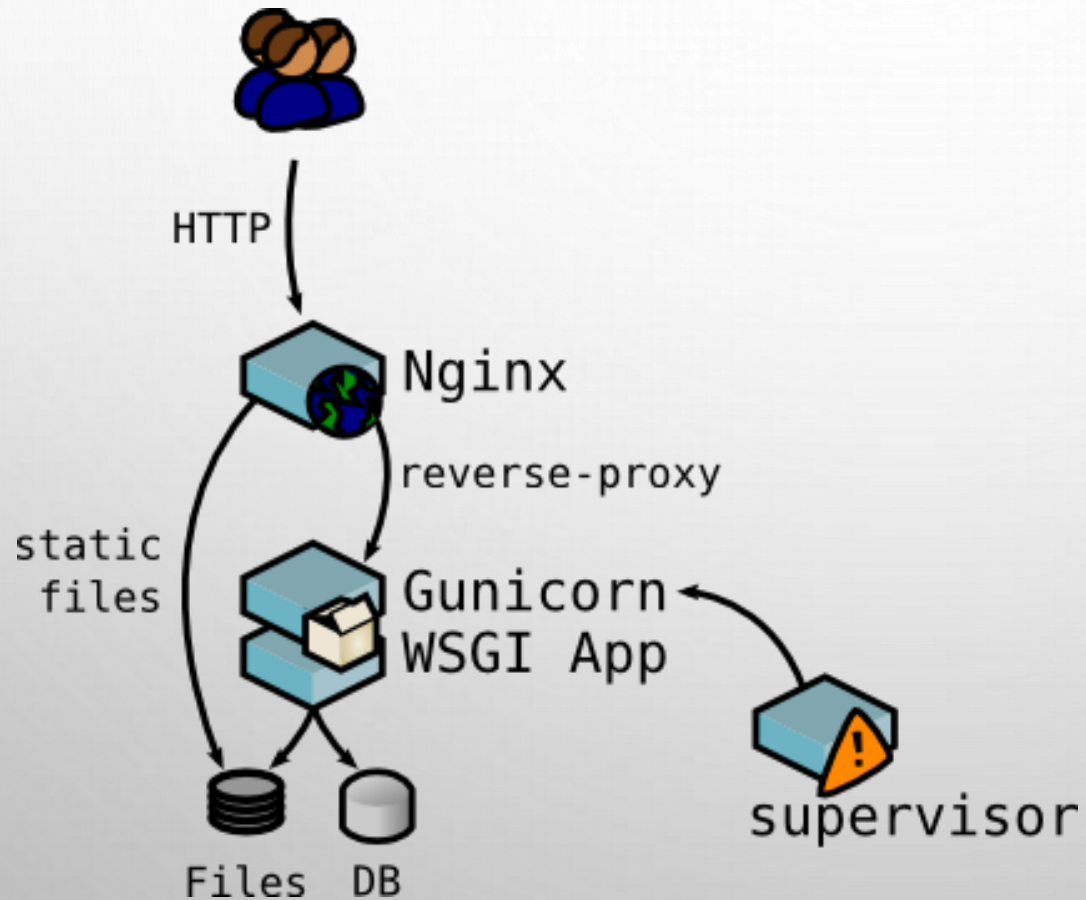
ON PRETRAINED MOBILENET_V2 MODEL

PENG LUO

TECH SELECTION

- BACKEND LANGUAGE: PYTHON
- FRAMEWORK: FLASK
- WEB SERVER NGINX
- HIGH PERFORMANCE WSGI SERVER: GUNICORN + GEVENT
- DATABASE: MONGODB
- API DOCUMENTATION: SWAGGER
- MACHINE LEARNING MODEL: MOBILENET_V2
- DEPLOYED ON: ALIYUN

SYSTEM STRUCTURE AND DATA FLOW



Ideally will be like what image shows

WHY NGINX?

- HIGH PERFORMANCE WEB SERVER
- REVERSE PROXY
- LOAD BALANCER
- STATIC FILES
- ...

WHY GUNICORN + GEVENT?



- HIGH PERFORMANCE
- MULTIPROCESSING SUPPORT
- HTTP TO WSGI
- COROUTINE

WHY MONGODB?

- KEY-VALUE NOSQL : SUITABLE FOR THE SCENARIO
- HIGH PERFORMANCE
- EASY TO USE

DOCKER?

```
Successfully built 0b2ac62bc9a6  
Successfully tagged ml-apple:latest  
'ml-apple Dockerfile: Dockerfile' has been deployed successfully.
```

Dependency management
Easier for deployment
Improve scalability

```
PS D:\develop> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3bc1076979b3	ml-apple	"gunicorn image_serv..."	5 seconds ago	Up 5 seconds	0.0.0.0:8080->80/tcp	docker_ml_apple
f3b70a8486e8	mongo	"docker-entrypoint.s..."	4 hours ago	Up 4 hours	0.0.0.0:27017->27017/tcp	docker_mongodb

DEMO

- HISTORY ENDPOINT: GET

`http://106.14.46.83/`

- PREDICTION ENDPOINT: POST

```
curl -k -X POST -F "image=@human.jpg"
"http://106.14.46.83/predict"
```

<input type="checkbox"/> 实例ID/名称	标签	监控	可用区 ▾	IP地址	状态 ▾	网络类型 ▾	配置	付费方式 ▾	
<input type="checkbox"/> i-uf6glr98hnlxukldg05b iZuf6glr98hnlxukldg05bZ	  		华东 2 可用区 G	106.14.46.83(公) 172.17.32.185 (私有)	 运行中	专有网络	1 vCPU 1 GiB (I/O优化) ecs.t5-lc1m1.small 1Mbps	包年包月 2020年9月13日 23:59 到期	管理

SWAGGER DOCUMENTATION

POST `/predict` predict a image

Parameters Try it out

Name	Description
body <small>★ required</small> object (body)	image that needs to be predicted
Example Value	Model
<pre>{}</pre>	
Parameter content type	<div>application/json</div>

Responses Response content type

application/json

Code	Description
200	success
Example Value	Model
<pre>{ "image_url": "20200802091119124Cat_07464.jpg", "prediction_result": [{ "class_name": "n02669723", "class_description": "academic_gown", "score": "0.041221227" }]}</pre>	
400	Bad Request

POST `/predict` predict a image

GET `/history` Returns all prediction result

Returns all prediction result

Parameters Try it out

No parameters

Responses Response content type

application/json

Code	Description
200	successful operation
Example Value	Model
<pre>[{ "_id": { "Oid": "5f2612b8bfe3763181e4e3d9" }, "image_url": "20200802091119124Cat_07464.jpg", "prediction_result": [{ "class_name": "n02669723", "class_description": "academic_gown", "score": "0.041221227" }] }]</pre>	

LIMITATIONS

- SECURITY DESIGN = NONE
- PERFORMANCE WONT BE SO GOOD SINCE EVERY IMAGE WILL BE STORED FOR DISPLAYING ON “HISTORY” WEBSITE.
- DEALING A SINGLE IMAGE AT A TIME WHILE KERAS MODEL CAN BATCH IMAGES
- ALIYUN SERVER IS FREE SO ITS PERFORMANCE IS POOR



HOW IT CAN BE IMPROVED

- REDIS AS QUEUE/MESSAGE BROKER PUT IN BETWEEN ML MODEL AND PREDICTION REQUESTS
 - DON'T STORE ACTUAL IMAGE TO REDUCE DISK IO FOR BETTER PERFORMANCE
 - SUPERVISOR FOR BETTER MONITORING
- 



Q&A?



The background is a light gray gradient. In the top-left and bottom-right corners, there are several realistic-looking water droplets of various sizes, some overlapping. A faint, circular, embossed-style logo is visible in the upper center of the page, above the text.

THANK U FOR WATCHING!