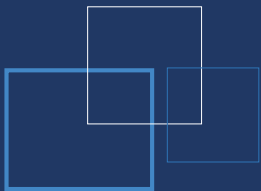
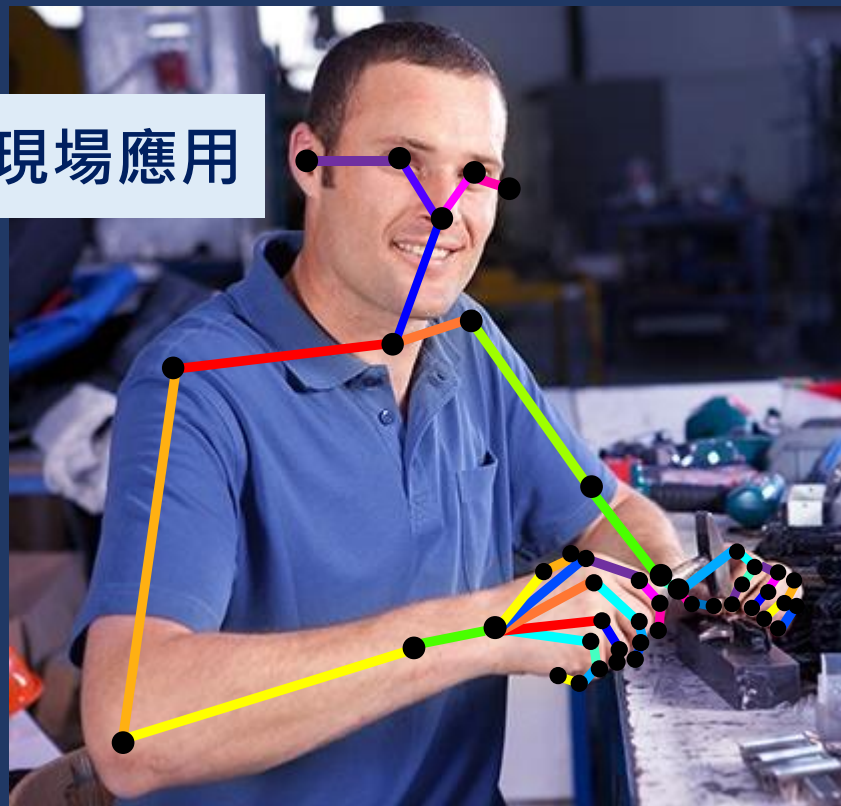


人體動作辨識結合語音助理於組裝現場應用

Group 7 110034538 許淨嵐





報告摘要

01

簡介

02

研究方法

03

實驗流程與結果

04

結論

05

參考資料



全球化

客製化

人機互動

簡介

研究目的

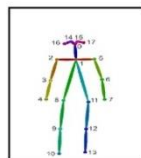
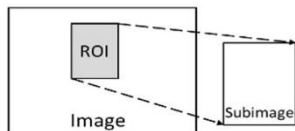


必須完成多種裝配的多能工

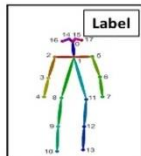


人腦記憶有限

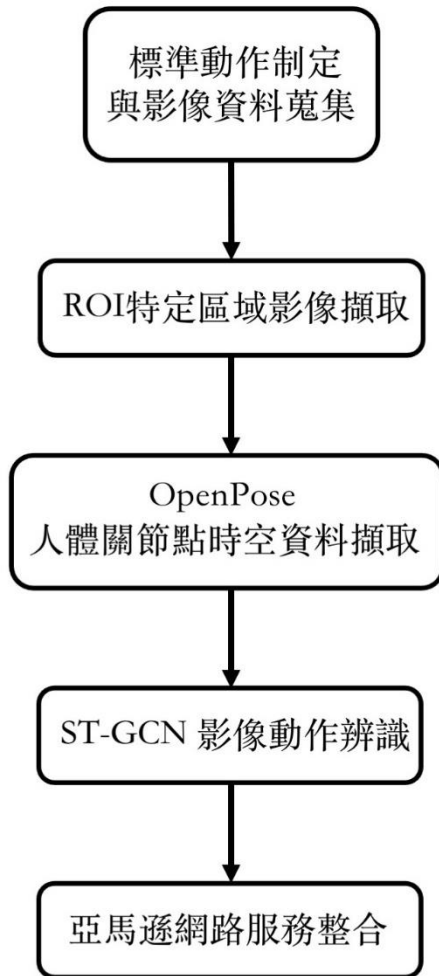
雙手可能不方便操作



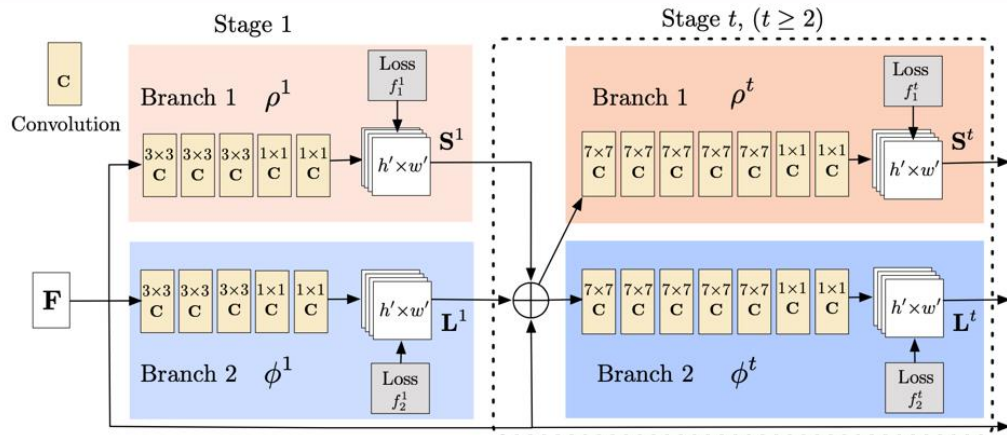
投入：人員作業影像
產出：人體關節點位座標
(時空資料)



投入：人體關節點位座標
(時空資料)
產出：影像動作預測結果

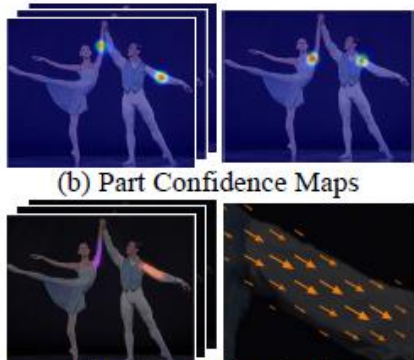


1. 辨識方式為由下而上 (Bottom-up)
2. 兩分支多階段的CNN
3. 第一個分支中的每個階段都預測信心圖 (Confidence map)
4. 第二個分支中的每個階段都預測局部關聯向量場 (Part affinity field)





(a) Input Image



(b) Part Confidence Maps

(c) Part Affinity Fields

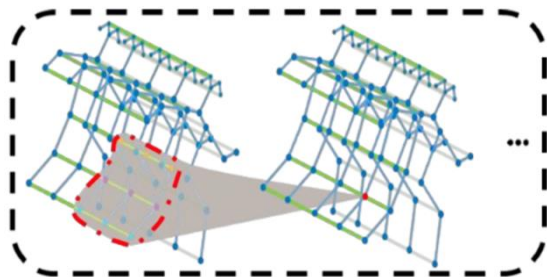


(d) Bipartite Matching

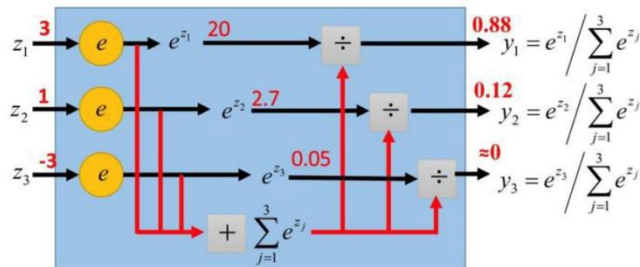


(e) Parsing Results

1. 別於傳統卷積神經網路，ST-GCN擷取輸入影片之「時間」、「空間」數據
2. 透過圖卷積計算影片之坐標網絡，針對影片之每一幀建立時空關聯
3. 計算出圖卷積之座標網路後，使用Softmax計算標籤分數並分類動作



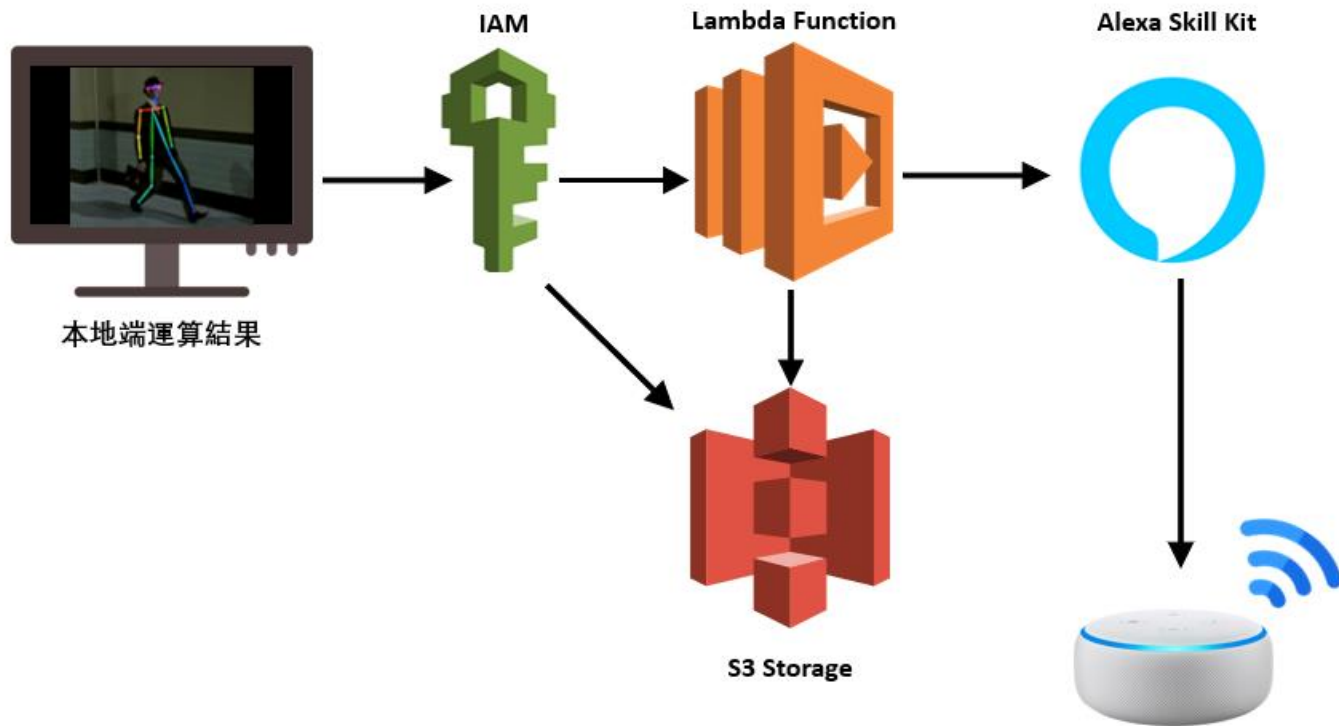
ST-GCN



Softmax 分類器

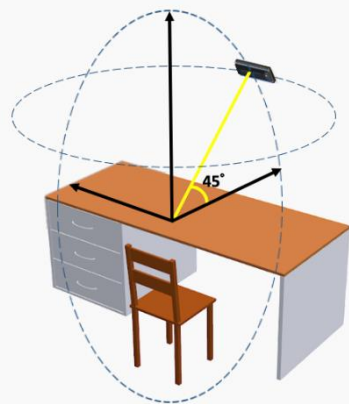
1. Batch size 設置為硬體可負荷之最大值7
2. Epoch設置為100
3. 調整 Dropout Rate、Base Learning Rate、Step

#	Dropout	Base_lr	Step	Top1	Top5	Mean_loss
1	0.1	0.1	[80,90,100]	80.58%	100.00%	0.4127432756
2	0.3	0.1	[80,90,100]	76.70%	100.00%	0.4480777343
3	0.3	0.1	[80,90,95,100]	84.47%	100.00%	0.3942492694
4	0.1	0.1	[80,90,95,100]	73.79%	100.00%	0.5054964473
5	0.1	0.01	[80,90,95,100]	99.03%	100.00%	0.0215713551
6	0	0.1	[80,90,95,100]	100.00%	100.00%	0.0115661198

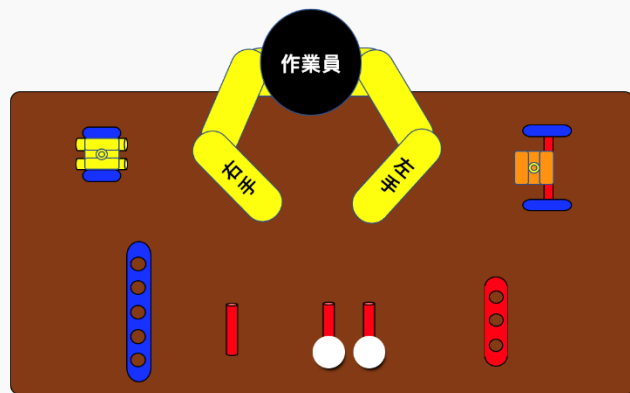


標準動作	組裝說明
1. 拿取紅色長板	左手拿取紅色短板，右手穩定綠色長板，並將紅色短板對準綠色長板中心孔後疊在其上。
2. 組裝眼睛-1	右手穩定半成品，左手拿取眼睛1對準紅色短板最左側的孔組裝。
3. 組裝眼睛-2	左手穩定半成品，右手拿取眼睛2對準紅色短板最右側的孔組裝，並雙手將組裝完眼睛後之半成品翻面向下壓緊。
4. 組裝紅色棍棒	左手穩定半成品，右手拿取紅色棍棒對準綠色長板中心孔組裝。
5. 組裝藍色短板	左手穩定半成品，右手拿取藍色長板並將其中心孔對準紅色棍棒組裝。
6. 組裝前輪	左手穩定半成品，右手拿取前輪零件對準藍色長板最前方的孔組裝。
7. 組裝後輪	右手穩定半成品，左手拿取後輪零件對準藍色長板最後方的孔組裝，並翻面完成一成品。





01 攝影機相對位置



02 工作站設計圖



03 工作站架設成果



1

訓練集 5

2

驗證集 1

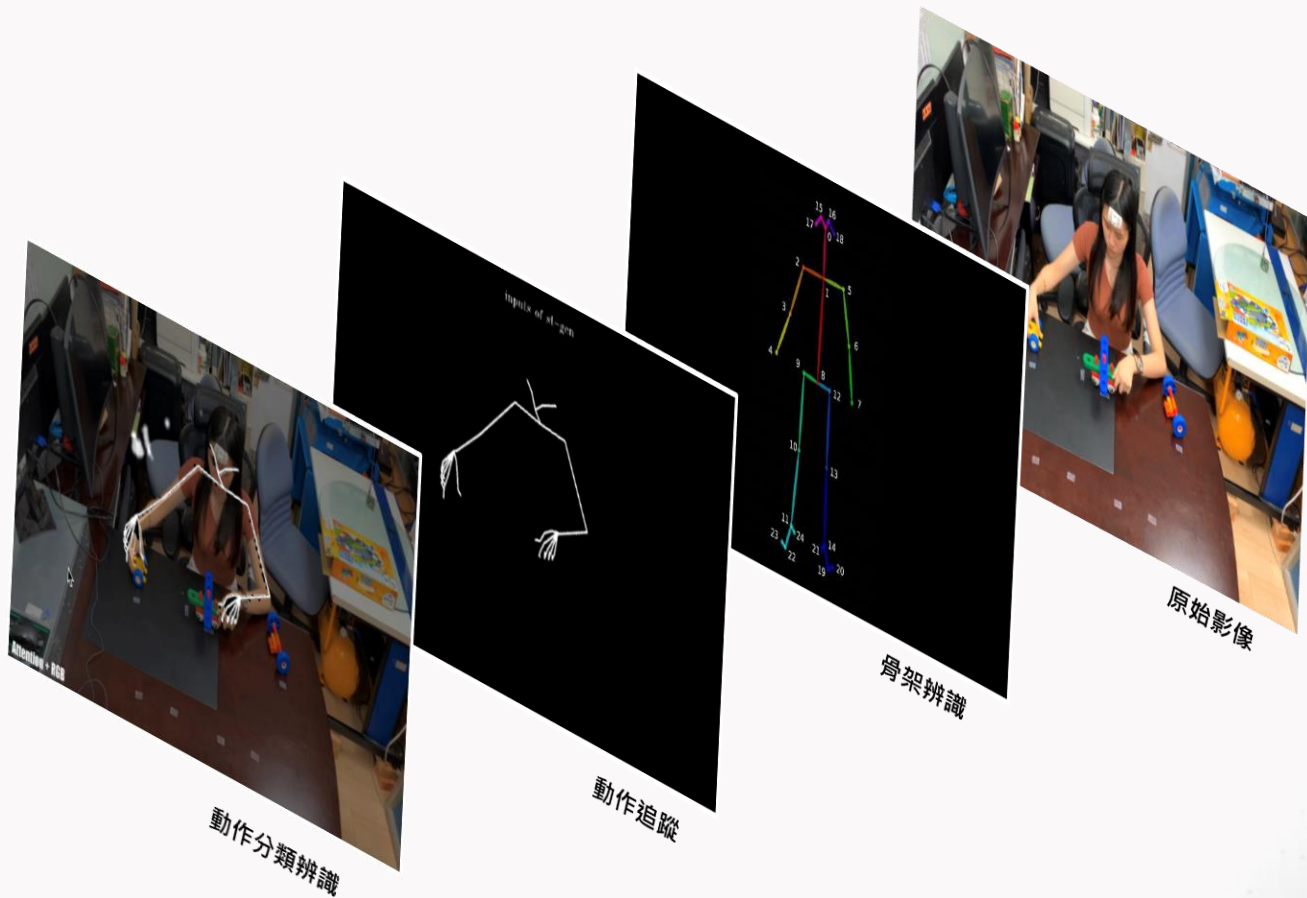
3

測試集 1

4

泛化用 1

(每人共8部影片)



Pose肢體辨識套件

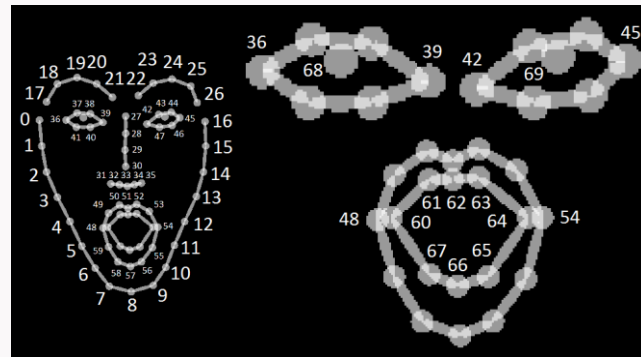
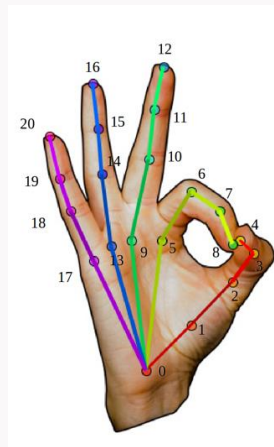
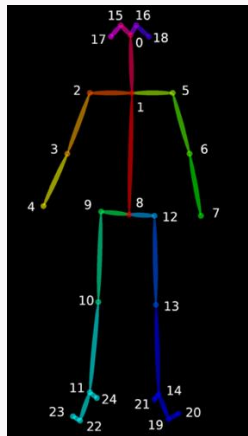
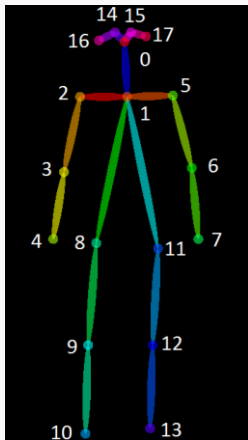
```
for dname in /home/simslab/CMU/openpose/Video_forSTGCN/Vic_Cut/*;
do
for filename in $dname/* ;
dd
d=$(dirname $filename | cut -d '/' -f 8 );
echo $d;
if [ -d "output/$d" ]; then
mkdir output/$d;
fi
name=$(basename "$filename" | cut -f 1 -d '.');
mkdir output/$d/$name;
./build/examples/openpose/openpose.bin --model_pose COCO --video $filename --hand --write_json ./output/$d/$name;
done;
done;
```

關鍵程式碼

```
#!/bin/bash
for dname in /home/simslab/CMU/openpose/Video_forSTGCN/Vic_Cut/*;
do
for filename in $dname/* ;
dd
d=$(dirname $filename | cut -d '/' -f 8 );
echo $d;
if [ -d "output/$d" ]; then
mkdir output/$d;
fi
name=$(basename "$filename" | cut -f 1 -d '.');
mkdir output/$d/$name;
./build/examples/openpose/openpose.bin --model_pose COCO --video $filename --hand --write_json ./output/$d/$name;
done;
done;
```

步驟說明

在指定路徑下，對所有的影片以 COCO 模式使用 OpenPose 進行關節點辨識運算，並將辨識資料以 JSON 檔案寫出。



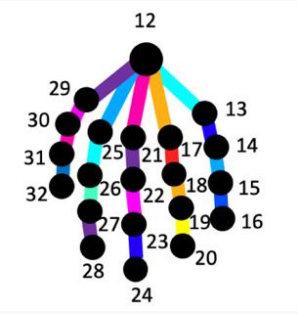
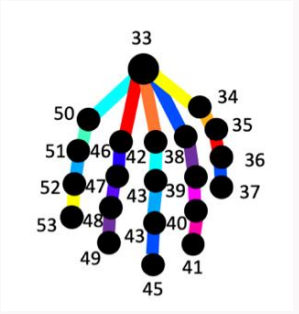
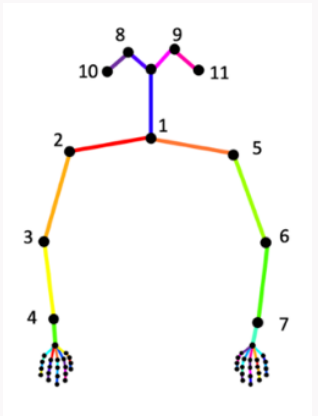
```

layout='prettyALenawithHands':
self.num_node=54
self.link=[(i,i)for i in range(self.num_node)]
neighbor_link=[(0,1),(8,0),(9,0),(10,8),(9,11),(1,2),(2,3),(3,4),(1,5),(5,6),(6,7),
(7,12),(12,13),(13,14),(14,15),(15,16),
(12,17),(17,18),(18,19),(19,20),
(12,21),(21,22),(22,23),(23,24),
(12,25),(25,26),(26,27),(27,28),
(12,29),(29,30),(30,31),(31,32),
(4,33),(33,34),(34,35),(35,36),(36,37),
(33,38),(38,39),(39,40),(40,41),
(33,42),(42,43),(43,44),(44,45),
(33,46),(46,47),(47,48),(48,49),
(33,50),(50,51),(51,52),(52,53)]
self.edge = self_link+neighbor_link
self.center=1

```

Hand Pose 肢體辨識套件

代碼	步驟說明
<pre> elif layout=='prettyALenawithHands': self.num_node=54 self.link=[(i,i)for i in range(self.num_node)] neighbor_link=[(0,1),(8,0),(9,0),(10,8),(9,11),(1,2),(2,3),(3,4),(1,5),(5,6),(6,7), (7,12),(12,13),(13,14),(14,15),(15,16), (12,17),(17,18),(18,19),(19,20), (12,21),(21,22),(22,23),(23,24), (12,25),(25,26),(26,27),(27,28), (12,29),(29,30),(30,31),(31,32), (4,33),(33,34),(34,35),(35,36),(36,37), (33,38),(38,39),(39,40),(40,41), (33,42),(42,43),(43,44),(44,45), (33,46),(46,47),(47,48),(48,49), (33,50),(50,51),(51,52),(52,53)] self.edge = self_link+neighbor_link self.center=1 </pre>	<p>將資料點數量設定為54個並在定義每個點的自我連接後定義空間邊的連接。</p>




```

self.layout='prettyAlenawithHands':
self.num_node=54
self.link=[(i,i)for i in range(self.num_node)]
neighbor_link=[(0,1),(8,0),(9,0),(10,8),(9,11),(1,2),(2,3),(3,4),(1,5),(5,6),(6,7),
(7,12),(12,13),(13,14),(14,15),(15,16),
(12,17),(17,18),(18,19),(19,20),
(12,21),(21,22),(22,23),(23,24),
(12,25),(25,26),(26,27),(27,28),
(12,29),(29,30),(30,31),(31,32),
(4,33),(33,34),(34,35),(35,36),(36,37),
(33,38),(38,39),(39,40),(40,41),
(33,42),(42,43),(43,44),(44,45),
(33,46),(46,47),(47,48),(48,49),
(33,50),(50,51),(51,52),(52,53)]
self.edge = self.link+neighbor_link
self.center=1

```

模型訓練與驗證結果

		1	2	3	4	5	6	7
Actual	1	80.129%	0.000%	0.000%	8.682%	0.000%	0.000%	31.190%
	2	9.605%	85.429%	12.712%	4.802%	1.977%	1.695%	6.780%
	3	11.416%	0.000%	75.572%	10.163%	0.907%	0.000%	31.942%
	4	5.674%	1.064%	6.738%	84.794%	15.957%	1.418%	0.355%
	5	8.333%	0.000%	1.087%	4.710%	88.783%	1.087%	0.000%
	6	2.644%	5.048%	3.846%	20.933%	0.962%	80.231%	10.337%
	7	12.596%	0.641%	4.327%	8.013%	0.000%	0.481%	77.942%

#	Recall	Precision	F1-Score
1	80.129%	35.283%	70.008%
2	85.429%	88.755%	87.060%
3	75.572%	64.052%	69.336%
4	84.794%	39.271%	63.120%
5	88.783%	79.322%	83.786%
6	80.231%	92.000%	85.713%
7	77.942%	53.919%	63.742%

$$Accuracy = \frac{\sum_i \sum_j (N_{ij} \times k)}{\sum_i \sum_j N_{ij}}, \text{ 其中 } k = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

準確率 **84.5%**



A collage of several overlapping screenshots from the AWS and Alexa Developer consoles. The screenshots show various configuration pages for an Alexa skill named 'Simla_Lambda'. The visible elements include:

- Alexa Developer console:** The 'Invocation Model' section, showing the 'Invocation Name' field set to 'say-hello' and a warning about 'invocation name requirements'.
- AWS Lambda console:** The 'Simla_Lambda' function configuration page, showing the 'Runtime' as 'Python 3.6' and the 'Handler' as 'lambda.handler.lambda_handler'.
- AWS IAM console:** The 'Identity and Access Management (IAM)' console, showing a policy named 'AlexaSkillPolicy'.
- AWS S3 console:** The 'AWS S3' console, showing a bucket named 'simla-lambda-dev'.
- Terminal:** A terminal window showing the execution of a command: `aws lambda update-function-code --function-name simla-lambda-dev --zip-file fileb://code.zip`.

AWS provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis.



AWS provides more than 175 services in the platform.

AWS IAM enables users to securely control access to AWS services and resources. Using IAM, users can create and manage AWS services and use permissions to allow and deny their permissions to AWS resources.

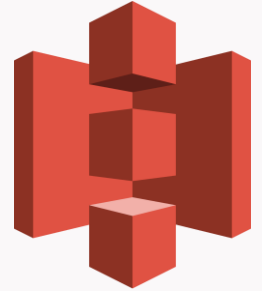


S3 provides a repository of data, allows users to access reliable, fast and inexpensive data storage infrastructure.

Bucket: File folders, store objects, which consist of data and its descriptive metadata

Key: File, data

AWS Lambda lets users run code without provisioning or managing servers. Users can run code for virtually any type of application or backend service - all with zero administration.





Identity and Access Management (IAM)

儀表板

▼ 存取管理

群組

使用者

角色

政策

身份供應商

帳戶設定

▼ 存取報告

存取分析器

存權規則

分析器詳細資訊

登入資料報告

組織活動

服務控制政策 (SCP)

角色 ARN [arn:aws:iam::328046982668:role/happyVic](#)角色說明 [For Alexa | 編輯](#)執行個體描述檔 ARN [🔗](#)

路徑 /

建立時間 2019-10-19 11:21 UTC+0800

上次活動 Not accessed in the tracking period

CLI/API 工作階段持續時間上限 1 小時 [編輯](#)

將此連結提供給可在主控台中切換角色
的使用者 <https://signin.aws.amazon.com/switchrole?roleName=happyVic&account=328046982668> [🔗](#)

許可

信任關係

標籤

存取顧問

撤銷工作階段

▼ Permissions policies (6 套用的多個政策)

[連接政策](#)
[+ 新增內嵌政策](#)

政策名稱	政策類型	
▶ AlexaForBusinessGatewayExecution	AWS 受管政策	✕
▶ AlexaForBusinessReadOnlyAccess	AWS 受管政策	✕
▶ AlexaForBusinessFullAccess	AWS 受管政策	✕
▶ AlexaForBusinessDeviceSetup	AWS 受管政策	✕
▶ AWSMarketplaceManageSubscriptions	AWS 受管政策	✕



happy_lambda

調節 合格者 ▼ 操作 ▼ 選擇測試事件 ▼ 測試 儲存 ⓘ

```
48 # 問數量
49 def get_amount_response():
50     data = s3.get_object(Bucket='happy1a', Key='amount.txt')
51     contents = data['Body'].read()
52     contents=str(contents, encoding = "utf-8")
53     session_attributes = {}
54     card_title = "amount"
55     speech_output = "You have done"+contents
56     reprompt_text = "You never responded to the first test message. Sending another one."
57     should_end_session = True
58     return build_response(session_attributes, build_speechlet_response(
59         card_title, speech_output, reprompt_text, should_end_session))
60
61
62 # 問下一步
63 def get_nextStep_response():
64     data = s3.get_object(Bucket='happy1a', Key='nextStep.txt')
65     contents = data['Body'].read()
66     contents=str(contents, encoding = "utf-8")
67     session_attributes = {}
68     card_title = "nextStep"
69     speech_output = contents
70     reprompt_text = "You never responded to the first test message. Sending another one."
71     should_end_session = True
72     return build_response(session_attributes, build_speechlet_response(
73         card_title, speech_output, reprompt_text, should_end_session))
74
```

48:1 Python Spaces: 4 ⚙

環境變數 (0) 編輯

意見回饋 中文(繁體)

© 2008 - 2020, Amazon Web Services, Inc. 或其合作夥伴。保留所有權利。 隱私權政策 使用條款



Alexa provides a set of built-in capabilities, referred to as skills. ASK lets users teach Alexa new skills. Customers can access these new abilities by asking Alexa questions or making requests.

Invocation: A keyword of waking up Alexa Skill.

Intent: Represent the user's request to look up tide information for a particular coastal city.



Intents / amount

Sample Utterances (2) ⓘ

[Bulk Edit](#) [Export](#)

What might a user say to invoke this intent?



how many products



How many products have i done



Intents / nextStep

Sample Utterances (3) ⓘ

[Bulk Edit](#) [Export](#)

What might a user say to invoke this intent?



next step

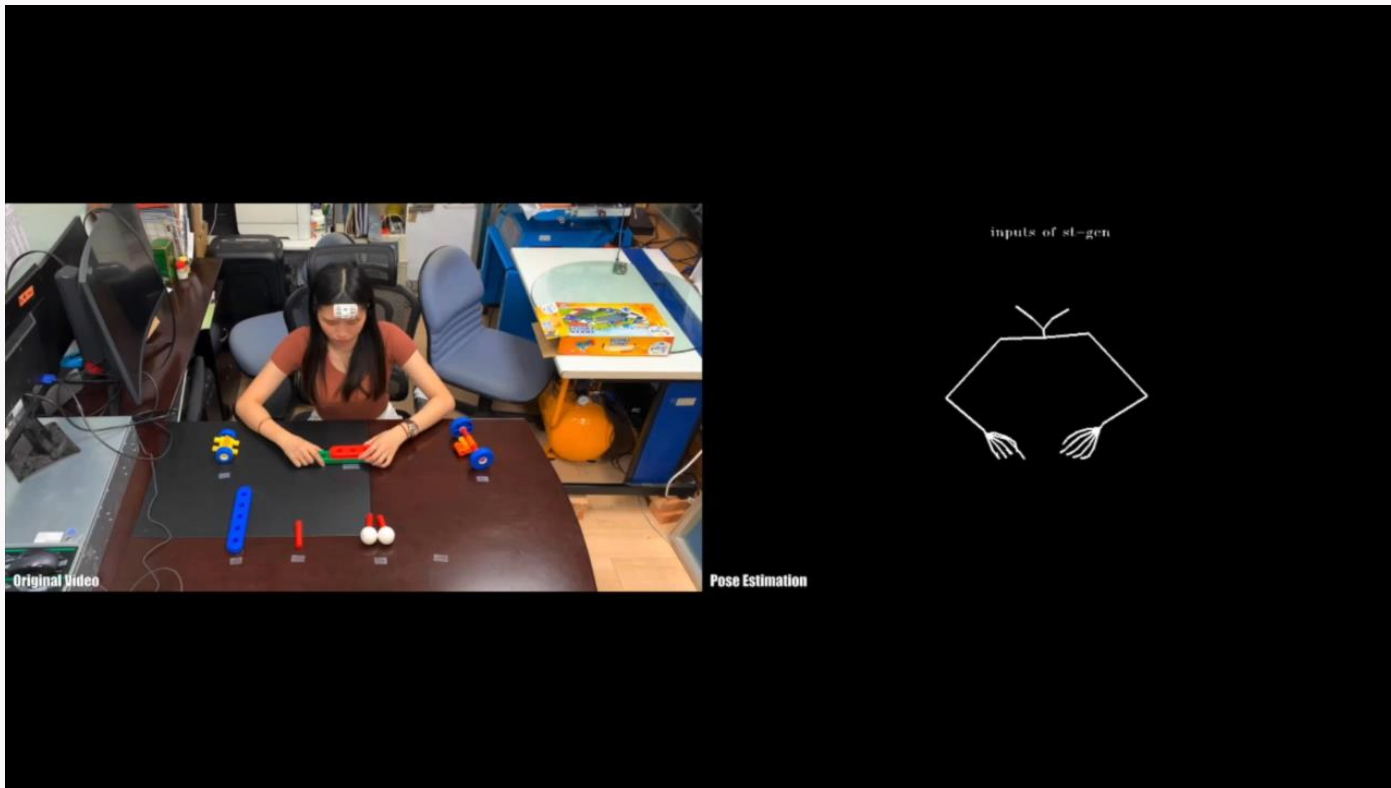


what should i do now




what's the next step





Skill testing is enabled in: Development ▾

Alexa Simulator Manual JSON Voice & Tone


English (US) ▾ Type or click and hold the mic 

ask happy toy what's the next step

take the blue board and install the red stick to its middle hole.

Skill testing is enabled in: Development ▾

Alexa Simulator Manual JSON Voice & Tone

English (US) ▾ Type or click and hold the mic 

ask happy toy how many products have i done

You have done 4

結論



模擬組裝站
整合完成



可再提升準確率

1

整體貢獻

提升工作效能，取得組裝相關資訊

2

研究限制

AWS 付費機制、VRAM限制

3

應用方面

組裝員之輔助工具、管理者之教育訓練工具

4

未來展望

多角度影像、雲端串聯即時性



感謝指教

- [1] H.-Y. Huang. "GCN(Graph Convolutional Network)的理解."
https://purelyvivid.github.io/2019/07/07/GCN_1/ (accessed 2021.01)
- [2] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii. USA, July 22 - 25, 2017 2017, pp. 7291-7299.
- [3] OpenCV. "OpenCV." <https://opencv.org/> (accessed 2021.01)
- [4] Amazon. "Amazon AWS." <https://aws.amazon.com/tw/> (accessed.
- [5] CMU-Perceptual-Computing-Lab. "openpose." <https://github.com/CMU-Perceptual-Computing-Lab/openpose> (accessed.
- [6] S. Yan, Y. Xiong, and D. Lin, "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition," presented at the Thirty-Second AAAI Conference on Artificial Intelligence, Hilton New Orleans Riverside, NewOrleans, Louisiana, USA, February 2 - 7, 2018.