Rolling Planet

The Combination of Augmented Reality & Deep Learning

1. 摘要

眼睛是人的「靈魂之窗」，人類在蒐集資訊時，最強大的感官就是視覺。但是，一般正常人取得資訊的方式並不僅限於透過視覺，還可透過聽覺、嗅覺、味覺、觸覺等感官方式來接收資訊，並藉「靈魂」與「智慧」進一步解譯資訊。因此，人類接收與解譯資訊的運作模式，就成了AR系統發展的基礎。

本專案主要藉由智慧AR 的發展及人工智慧技術，結合軟體工程，來訓練孩子們的四則運算，提升學習效率，增添學習樂趣。

2. 緒論

2.1. 研究背景動機

自2016 年手機遊戲Pokemon GO 掀起全球抓寶旋風，讓當中的擴增實境（Augmented Reality，AR）技術再次受到矚目；隔年2017 年，Apple、Google 兩大開發者相繼投入後形成完整的AR 產業生態圈，隨著科技巨頭搭建的AR 平台逐漸成熟，所帶來不只是垂直領域的變革，更多人工智慧（Artificial Intelligence，AI）技術廠商開始加入AR 的競爭行列，技術的融合必然是趨勢所向。

從技術本身來看，AI 和AR 是兩種截然不同的技術，AI 技術主要是透過深度學習讓軟硬體優化，而AR 技術是透過顯示器呈現虛擬數位資訊與現實資訊之結合，其中在AR 的核心技術中，環境理解、交互理解和視覺定位追蹤都與深度學習有著緊密的聯繫，兩者技術可以互相滲透，這意味著AI 與AR 結合將創造無限可能。因此，一種新的概念已經發展，即智慧AR。

本專案主要藉由智慧AR 的發展及其相關技術，希望利用深度學習模型配合擴增實境的技巧，並將其移動至移動端，以利用人工智慧來訓練孩子們的四則運算，提升學習效率，增添學習樂趣。

2.2. 問題定義 - 5W1H

What

提高學齡兒童學習算術、識別及手寫數字的學習效率，並且增添學習意願與興趣。

Where

學校、家庭或是基礎教育、幼兒教育部門。

Who

正值學習四則運算的學齡兒童。

When

學習加法和減法其實是從學習計數開始，對於某些孩子來說，學習計數方法大概是從幼稚園中班到大班開始。

Why

幫助孩子更好地學習數學，甚至能夠協助學習四則運算比較困難的學童，更迅速的步上軌道。

How

通過擴增實境和人工智慧和邊緣運算的結合，實踐邊緣端推論的擴增實境人工智慧應用。

3. 文獻探討

3.1. 骰子偵測模型：YOLO 9000

Yolo 系列 (You only look once, Yolo) 是關於物件偵測 (object detection) 的類神經網路演算法，以小眾架構 darknet 實作，實作該架構的作者 Joseph Redmon 沒有用到任何著名深度學習框架，輕量、依賴少、演算法高效率，在工業應用領域很有價值，例如行人偵測、工業影像偵測等等。

而Yolo9000 也就是YOLO v2 針對 YoloV1 的缺點做了一些改進：

1. 引入 Faster RCNN 中的 anchor box，不再直接 mapping bounding box 的座標，而是預測相對於 anchor box 的參數，並使用 K-Means 求 anchor box 比例。
2. 去掉 Fully Connected layer，改成全部皆為 conv layer。
3. 每層加上 batch normalization，去掉 dropout。
4. 增加解析度：增加 ImageNet pretrain 的解析度，從 224×224 提升至 448×448。

而本專案因為要將兩個深度學習模型加上擴增實境實踐在IOS 作業系統上，為了維持軟體架構的良好，必須選擇輕巧、準確度足夠、推論速度快的模型架構，而 YOLO v3 的架構是以Darknet-53 作為Backbone，YOLO v2 則僅為Darknet-19，僅耗費0.2 MB 空間，兩者的Backbone 架構比較如下：

Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

Table 6: Darknet-19.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Table 1. **Darknet-53.**

3.2. 手寫偵測模型：CNN LeNet

LeNet5誕生於1994年，是最早的卷積神經網絡之一， 由Yann LeCun完成，推動了深度學習領域的發展。在那時候，沒有GPU幫助訓練模型，甚至CPU的速度也很慢，因此，LeNet5通過巧妙的設計，利用卷積、參數共享、池化等操作提取特徵，避免了大量的計算成本，最後再使用全連接神經網絡進行分類識別，這個網絡也是最近大量神經網絡架構的起點，給這個領域帶來了許多靈感，其網路架構如下。

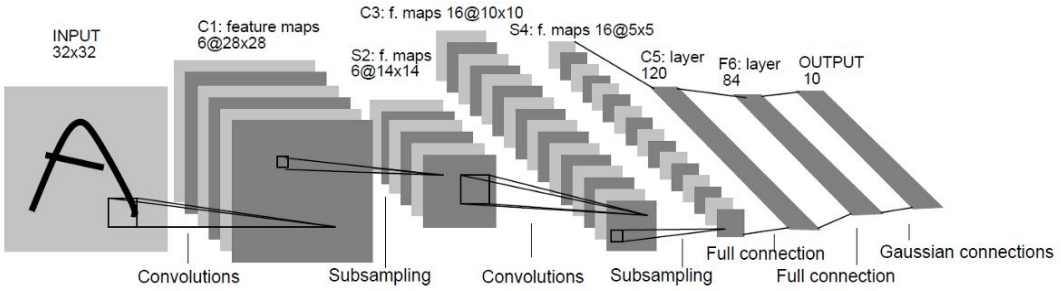


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeNet-5是Yann LeCun等人在多次研究後提出的最終卷積神經網絡結構，一般LeNet即指代LeNet-5。

LeNet-5包含七層，不包括輸入，每一層都包含可訓練參數（權重），當時使用的輸入數據是32*32像素的圖像。下面逐層介紹LeNet-5的結構，並且，卷積層將用Cx表示，子採樣層則被標記為Sx，完全連接層被標記為Fx，其中x是層索引。

層C1是具有六個5*5的捲積核的捲積層（convolution），特徵映射的大小為28*28，這樣可以防止輸入圖像的信息掉出卷積核邊界。C1包含156個可訓練參數和122304個連接。

層S2是輸出6個大小為14*14的特徵圖的子採樣層（subsampling/pooling）。每個特徵地圖中的每個單元連接到C1中的對應特徵地圖中的2*2個鄰域。S2中單位的四個輸入相加，然後乘以可訓練係數（權重），然後加到可訓練偏差（bias）。結果通過S形函數傳遞。由於2*2個感受域不重疊，因此S2中的特徵圖只有C1中的特徵圖的一半行數和列數。S2層有12個可訓練參數和5880個連接。

層C3是具有16個5-5的捲積核的捲積層。前六個C3特徵圖的輸入是S2中的三個特徵圖的每個連續子集，接下來的六個特徵圖的輸入則來自四個連續子集的輸入，接下來的三個特徵圖的輸入來自不連續的四個子集。最後，最後一個特徵圖的輸入來自S2所有特徵圖。C3層有1516個可訓練參數和156 000個連接。

層S4是與S2類似，大小為2*2，輸出為16個5*5的特徵圖。S4層有32個可訓練參數和2000個連接。

層C5是具有120個大小為5*5的捲積核的捲積層。每個單元連接到S4的所有16個特徵圖上的5*5鄰域。這裡，因為S4的特徵圖大小也是5*5，所以C5的輸出大小是1*1。因此S4和C5之間是完全連接的。C5被標記為卷積層，而不是完全連接的層，是因為如果LeNet-5輸入變得更大而其結構保持不變，則其輸出大小會大於1*1，即不是完全連接的層了。C5層有48120個可訓練連接。

F6層完全連接到C5，輸出84張特徵圖。它有10164個可訓練參數。這裡84與輸出層的設計有關。

3.3. 擴增實境（Augmented Reality）

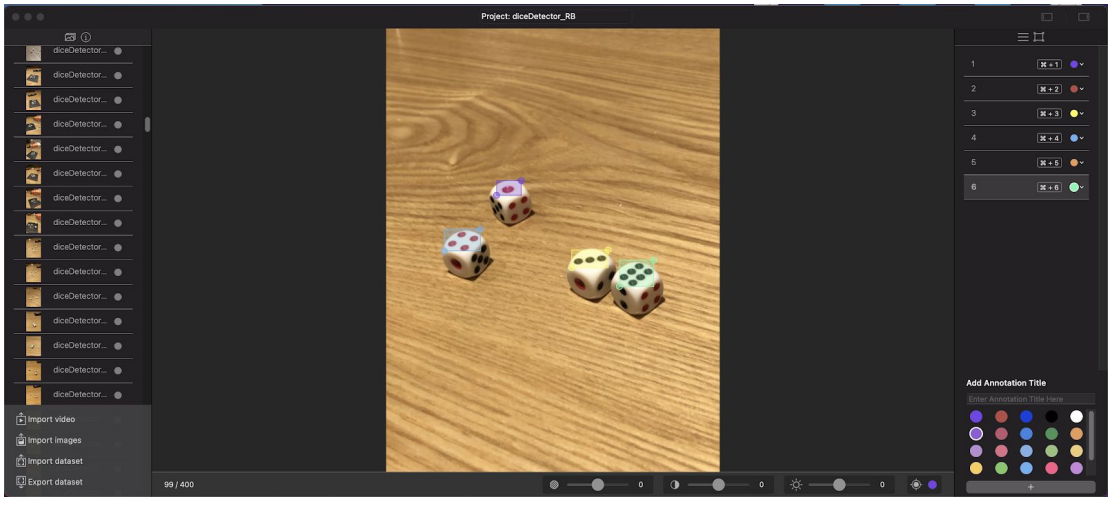
擴增實境，也有對應VR虛擬實境一詞的翻譯稱為實擬虛境或擴張現實，是指透過攝影機影像的位置及角度精算並加上圖像分析技術，讓螢幕上的虛擬世界能夠與現實世界場景進行結合與互動的技術。這種技術於1990年提出。隨著隨身電子產品運算能力的提升，擴增實境的用途也越來越廣。

4. 研究方法

4.1. 骰子辨識 - 資料處理

資料使用自己拍攝的440 張骰子實照，並利用Data augmentation 進行資料擴充，詳細處理流程如下：

1. 進行資料標注



2. 資料擴增

a. Random Crop

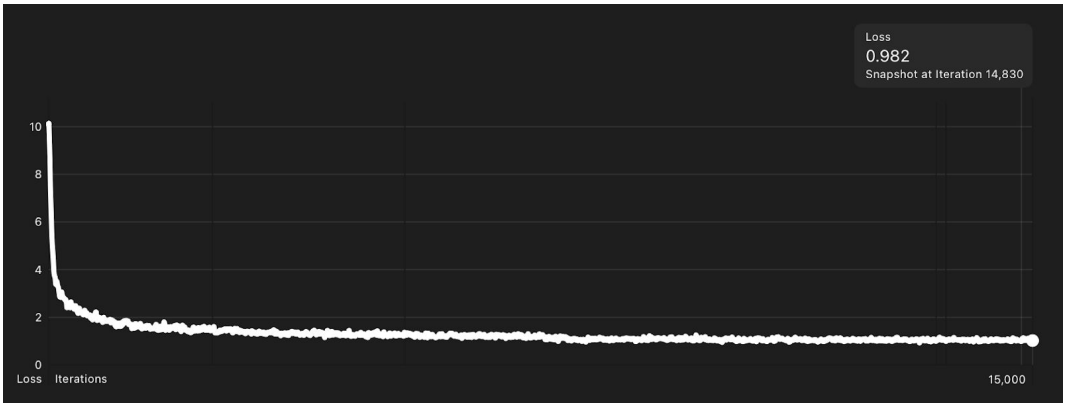
b. Color augmentation

3. 訓練資料分布

4.2. 骰子辨識 - 模型訓練

而本專案因為要將兩個深度學習模型加上擴增實境實踐在IOS 作業系統上，為了維持軟體架構的良好，必須選擇輕巧、準確度足夠、推論速度快的模型架構，而YOLO v3 的架構是以Darknet-53 作為Backbone，YOLO v2 則僅為Darknet-19，僅耗費0.2 MB 空間，所以這邊採用YOLO v2 去做模型訓練：

- Training set: 440 (Raw data) -> 20000 (Data Augmentation)
- Class number: 6
- Iterations: 15000
- Batch size: 32
- Validation Set: 1150 Items
- Loss: 0.982
- Training_Acc: 94%
- Validation_Acc: 92%



4.3. 骰子辨識 - 模型成果

經過足量的資料量訓練以及迭代次數，可以獲得不錯的辨識結果，Demo 如下：

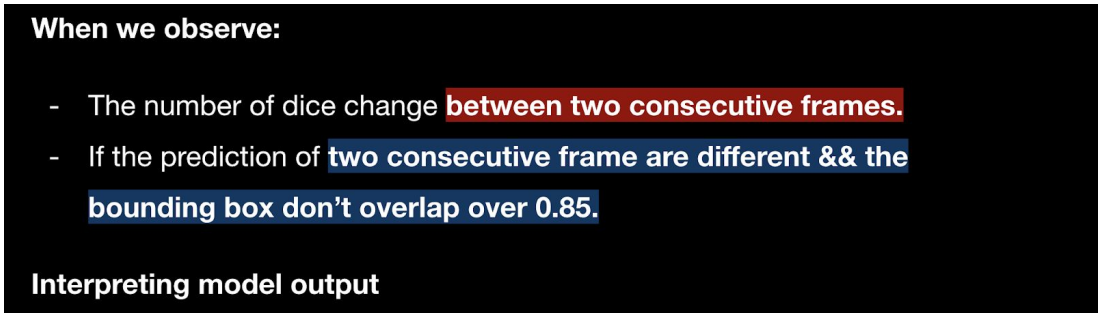
4.4. Real - time 即時推論資料處理

因為iPad 的相機輸入其實並非模型訓練時預處理過的Input 所以需要對相機輸入的照片作出處理：

由上圖所示，為了完整讀到骰子資訊，我們採用Scale fit 的方式處理。

4.5. 決定模型推論終止的時點

因為我們需要一個明確做出遊戲動作的始點，所以我們要決定畫面中，何時是骰子仍在動作，何時是已經有明確點數可以辨識，所以進行以下的程式邏輯辨識：



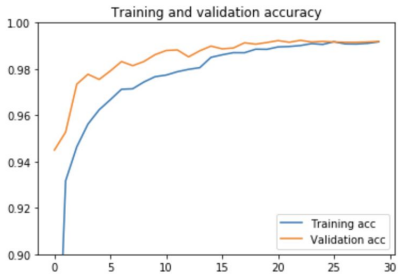
4.6. 手寫辨識模型 - LeNet + MNIST dataset

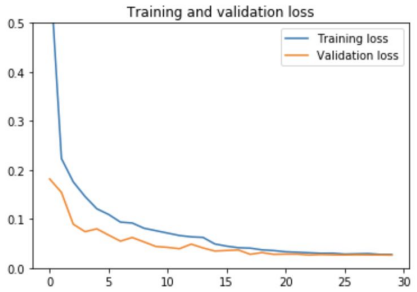
因為模型要求輕量簡便，所以使用了一個簡單LeNet 模型做結合，訓練資訊如下：

- Model Size: 0.2MB
- Training set: 60000
- Class number: 10
- Iterations: 30
- Batch size: 32
- Loss: 0.1441
- Training_Acc: 99.17 %
- Validation_Acc: 99.19 %

- Platform: Colab

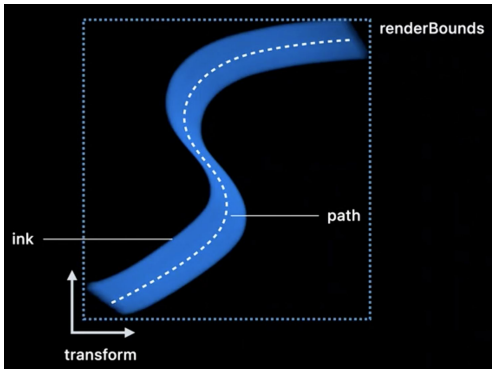
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 28, 28, 6)         156
_________________________________________________________________
average_pooling2d_1 (Average (None, 14, 14, 6)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 10, 10, 16)        2416
_________________________________________________________________
average_pooling2d_2 (Average (None, 5, 5, 16)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 400)               0
_________________________________________________________________
dense_1 (Dense)              (None, 120)               48120
_________________________________________________________________
dense_2 (Dense)              (None, 84)                10164
_________________________________________________________________
dense_3 (Dense)              (None, 10)                850
=================================================================
Total params: 61,706
Trainable params: 61,706
Non-trainable params: 0
_________________________________________________________________
```

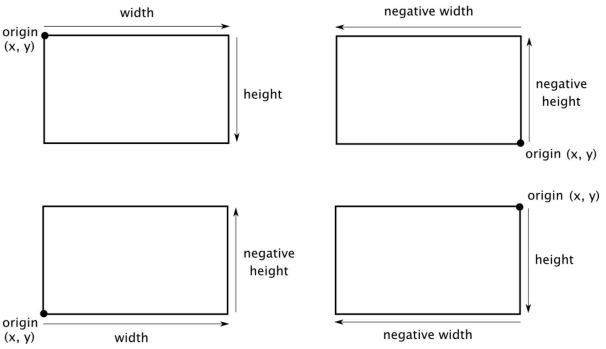




4.7. 手寫辨識模型 - 實現多筆畫多位數字即時推論

透過對輸入筆畫邊框的理解，我們可以用程式碼去實踐這個需求，因為這樣可以比實際訓練一個龐大的模型在邊緣裝置上更有效率：





While there exists any stroke:

1. Keep update the position of the final stroke
2. if **minX** of final stroke **>** **MaxX** in previous stroke:

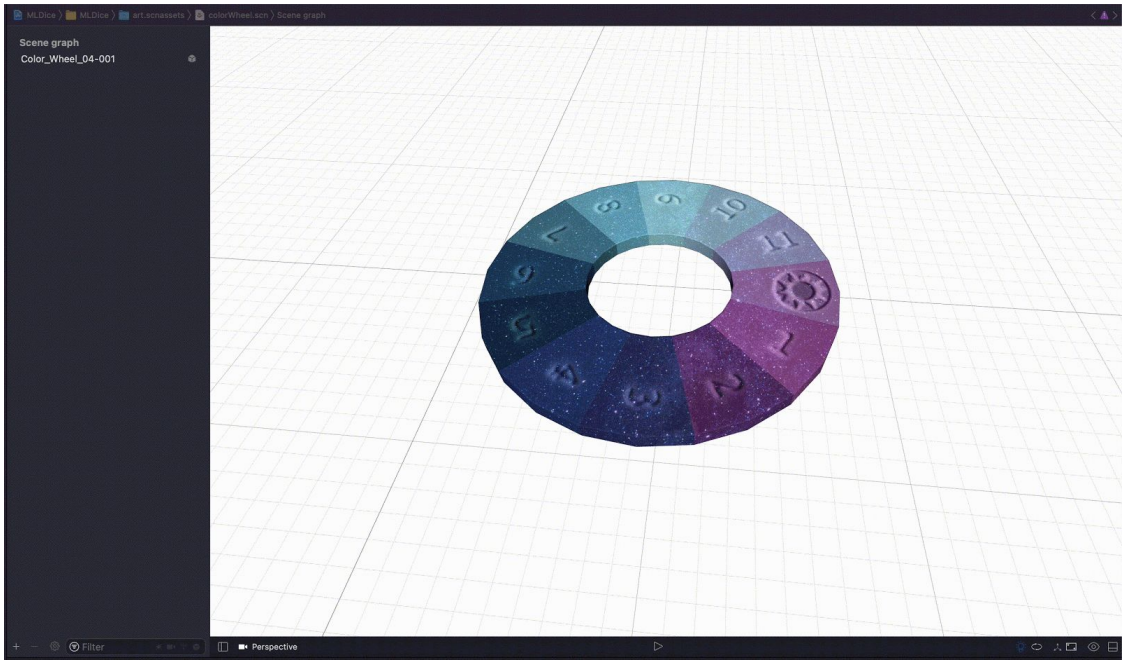
   New digit detected!

   Inference old digit and push into **queue**.

Comparision each digit with the result of objected detection.

4.8. 建構擴增實境模型 - 建構3D模型

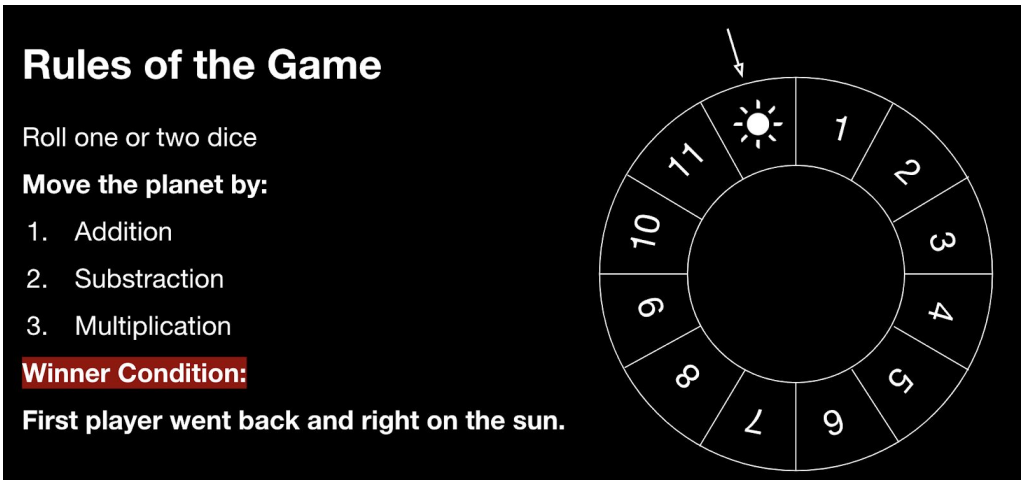
本專案利用軟體Blender 來建模，以星系為主視覺，建立土星、地球以及星盤模型如下所示：



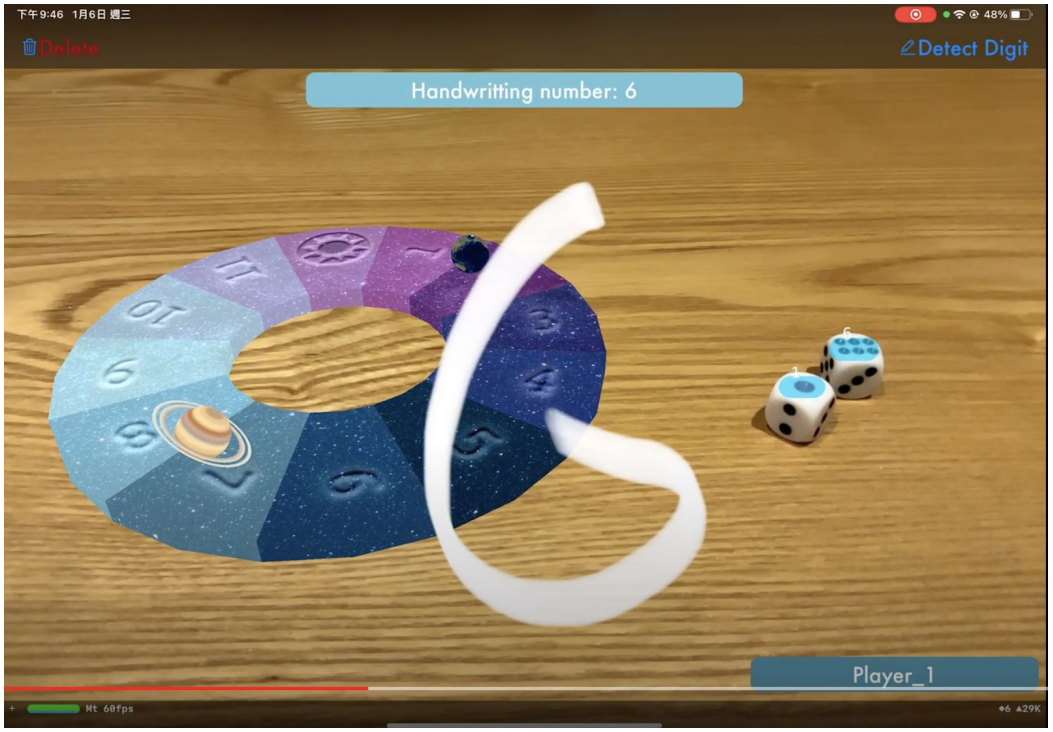
5. 專案成果

5.1. 遊戲規則

- 選擇丟擲一個或是兩個骰子
- 如果是單一骰子，則其點數就是使用者應該走的步數
- 如果是兩個骰子，可以選擇做兩數的加減乘作為步數
- 當先走回太陽並且剛好停在太陽標誌位置的獲勝



5.2. 遊戲結果Demo



https://www.youtube.com/watch?v=Bzlm9ouyLNA&feature=youtu.be

6. 結論

1. 本研究改進點：
    - 可以應用更多模型優化手法，像是量化模型或是剪枝等等，去針對龐大麼行做壓縮，以求將精準度更高的模型放入邊緣設備。
    - 針對邊是模型做出個別更細部的調整參數，使其精準度更高
    - 程式碼封裝的更加易用，並建立可以導出的API 接口
    - 辨識的邏輯部分可以作出優化，讓使用者功能更直覺，並可增加自然語言處理的API串接。
    - 支援更多種行動裝置及可穿戴設備
2. 未來展望：
    - 除了幼教遊戲之外，可以針對工業用的教育、品管等層面做出相關的AR+AI的邊緣裝置體驗。
    - 嘗試結合視覺、交互介面與人工智慧的核心技術研發，並應用於工業領域的AR，開發更多人機交互服務、挖掘工業深度數據、打造AR＋AI工業工具級的產品與服務。

7. 參考文獻

J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 6517-6525, doi: 10.1109/CVPR.2017.690.

WWDC Notes, *https://www.wwdcnotes.com/notes/wwdc20/10148/*

Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.