**(二) 研究計畫之背景與目的**

1. **背景**

現今雲端運算技術發展成熟，已經成為許多企業佈署應用服務的主要方法。隨著晶片製造科技微型化，單一使用者可以同時擁有多個裝置聯網存取服務。也因為網路技術的普及，用戶可以來自世界各地與伺服器連結使用，大程度的帶來便利性。然而，大量的用戶導致伺服器發生意外的故障與停機，影響用戶的使用體驗，尤其是高風險應用更為顯著。因此，正需要導入智慧化的故障預測系統防止無預期的事件發生，減少意外帶來的高額成本及損耗。

伺服器的狀態監控可以藉由溫度、硬體資源使用率、系統日誌等紀錄來觀測。雖然現今已有許多基於深度學習的方法來辨認異常狀態，但仍然無法有效的解決此問題。一般的深度類神經網路模型會受限於樣本不均衡的問題，使得模型傾向於預測樣本比例大的類別來得到最好的損失。半監督學習方法使用自動編碼器學習對於正常狀態建模，在實際使用時判定與正常分布的差距來辨別異常。而正常模式卻會頻繁的變換導致模型失準，需要時常離線重新學習來維持準確度。為了能讓雲端應用服務維持正常營運，減少因故障帶來不好的經濟損失，我們為台灣恩智浦半導體股份有限公司(NXP)量身訂做一套雲端應用服務異常檢測系統。半導體工廠在對晶片加工與測試時，對於一片複雜設計的晶片會產生大約數萬筆參數。因此，資料的上傳與存取都必須要確保伺服器隨時處於正常狀態，並在問題發生時能儘快啟動相關機制防止用戶無法及時使用資料。在大公司的分佈式系統中，為了確保每項服務彈性佈署的特性，管理者會建立多項虛擬機維持營運，並視情況執行擴充或遷移。然而，一個異常檢測系統需要監測如此多樣且複雜的系統是具有挑戰性的。我們必需要考慮到異常檢測模型的適應性，設計一套對於隨時間變動的規則，能夠以最少的資源重新訓練與適應，大程度的減少維護的複雜流程的全方位系統健康檢測系統。系統可以透過內部的檢測機制判定模型可否維持在線服務，在必要時也能輔以人工判讀。本計畫提出「基於數據驅動元學習和強化學習的計算資源異常偵測與預測之研究」乙案，分三年期程，逐年分別提出 (1) 基於強化學習的伺服器異常檢測、(2) 元強化學習的自適應流程和(3) 基於元策略流程的主動異常檢測系統等三項的基於數據驅動元學習和強化學習的計算資源異常偵測與預測之研究解決方案，以三年期程解決台灣恩智浦公司在生產過程所遭遇的棘手問題。

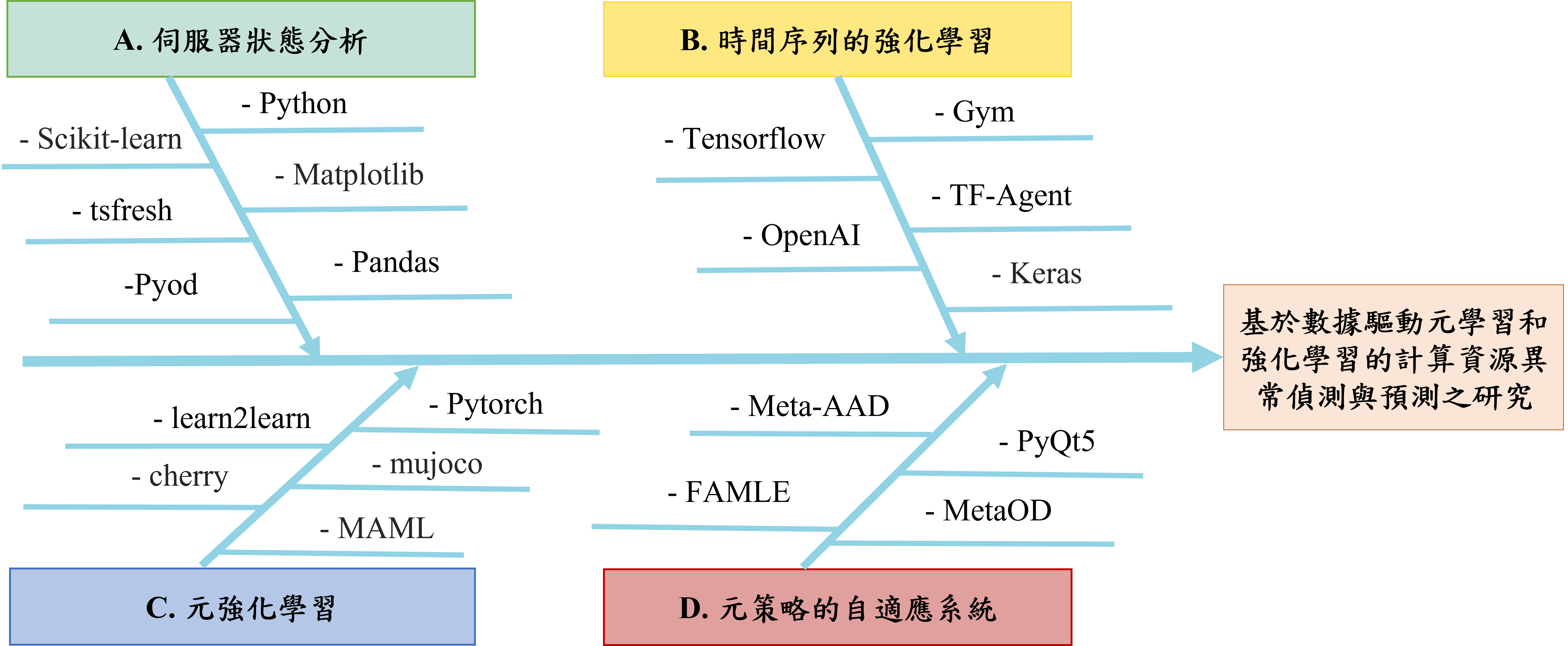


Fig. . 基於數據驅動元學習和強化學習的計算資源異常偵測與預測之研究

如Fig. 1所示，基於數據驅動元學習和強化學習的計算資源異常偵測與預測之研究所涵蓋技術包含了(A)伺服器狀態分析、(B)時間序列的強化學習、(C)元強化學習和(D)元策略的自適應系統。本計畫提出「基於數據驅動元學習和強化學習的計算資源異常偵測與預測之研究」乙案，將會針對(C)和(D)的部份聚焦於主動式的元策略異常檢測流程的優化。如此一來本計畫便能解決因為雲端環境變化而發生模型失準以及減少重新佈署模型之成本。

本案預計與台灣恩智浦公司合作進行三年期改善的解決方案。首先該公司在僅使用Zabbix Server監控公司雲端服務的健康狀態，並且使用靜態閥值作為系統發生警訊的依據。此方法因為對於某些時常處於高度使用的應用服務會時常收到警示郵件，但經檢查後卻是處於正常狀態。然而，當故障發生時卻是無預警且沒有任何徵兆。因此，為了讓公司能夠對於系統的故障能夠及早的準備，計畫第一年以深度強化學習演算法進行雲端應用服務的異常檢測，大幅減少管理者因為假警報或是無預警的故障帶來的損失；其次台灣恩智浦公司的多個應用服務性質不同，需要為每個應用服務制定一個預測模型。然而，多個模型訓練的成本將難以估計。所以第二年計畫我們將強化學習中加入MAML的訓練流程，減少不同應用服務之間更新模型的複雜性；最後為了維持檢測系統的持續性，我們期望能以最少的人工干預來維護在線的檢測模型。第三年計畫以Meta-AAD的檢測流程建立一個全自動化的系統，再加入FAMLE執行快速適應主動檢測無預期的故障，提升台灣恩智浦公司雲端服務的穩定性。

1. **目的**

本計畫的目的是實現在數位化的時代下以資料驅動結合人工智慧的力量，建立資料中心集群服務異常檢測系統，減少非預期的故障帶來的經濟損失。因此提出「基於數據驅動元學習和強化學習的計算資源異常偵測與預測之研究」乙案並分三年期程執行，逐年分別提出(1) 基於強化學習的伺服器異常檢測、(2) 元強化學習的自適應流程和(3) 基於元策略流程的主動異常檢測系統等三項基於數據驅動元學習和強化學習的計算資源異常偵測與預測之研究的解決方案。如Fig. 2所示，第一年期程預計導入基於強化學習的伺服器異常檢測，做為台灣恩智浦公司伺服器故障分析與檢測的解決方案，運用強化學習演算法辨識複雜的雲端故障現象。如Fig. 3所示，第二年期程預計導入元強化學習的自適應流程，可為台灣恩智浦公司維護在線異常檢測系統，並導入Meta Learning自適應方法加速模型更新流程。如Fig. 4所示，第三年期程預計導入智動化晶圓側面刮痕檢測系統，為台灣恩智浦公司檢視晶圓進行切割後的橫切面是否因為機台切割不確實造成晶圓側面有刮痕產生，且利用改良模型中表現最佳的物件偵測演算法GSEH-YOLOv5[8][9][10][11]執行智動化晶圓側面刮痕檢測。



Fig. . 基於強化學習的伺服器異常檢測之流程

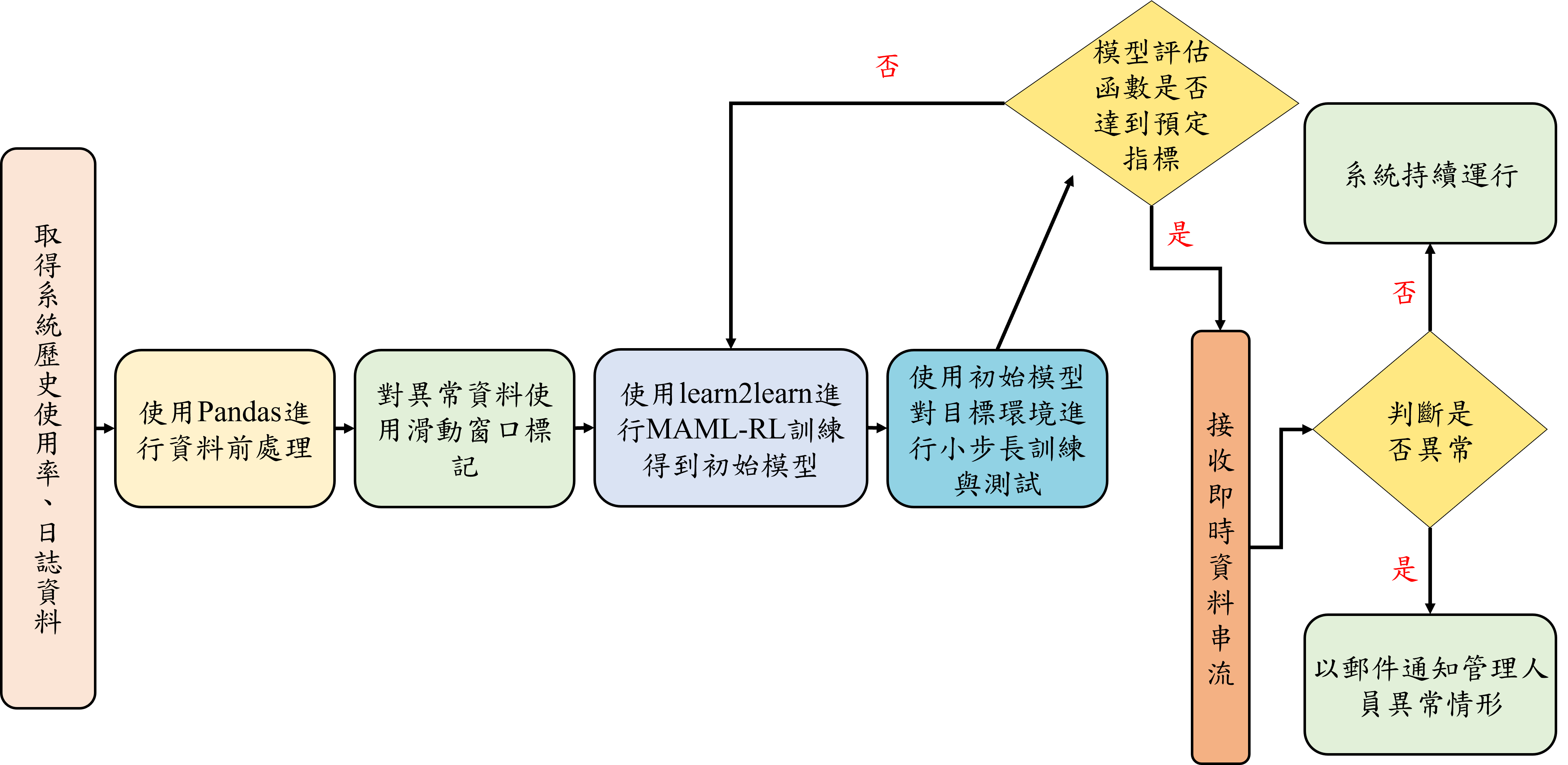


Fig. . 元強化學習的自適應之流程

1. **重要性**

如Fig. 5所示，本計畫的重要性是實現在台灣恩智浦公司的晶圓製程及晶片封裝之產品品質檢測是採取目前最為先進的物件偵測演算法YOLOv5，使用YOLO系列的最新版本，可提供台灣恩智浦公司最快速與最準確的智動化晶片及晶圓的品質檢測。其中針對傳統YOLOv5架構進行改良，可以大幅提升檢測效能。

Fig. 5. 基於數據驅動元學習和強化學習的計算資源異常偵測與預測之研究之整體架構說明

近年來因為工業4.0的推動，工廠逐漸以智慧製造為目標，使生產流程能更加的智慧化。然而，智慧化的設備雖然大幅提升生產的效率，不過若產品製造的過程需要高精密度。機台的動作精準度將會格外重要，只要有些微的誤差則容易造成產品的損失。讓生產流程更加智慧化的做法主要是將需要使用到人力的部分能夠盡量以機器做為取代，一方面能帶來有效率的工作，另一方面能夠提升品質檢測時的穩定性。然而，智慧化的品檢設備確實帶來生產流程的效率，不過產品製造的過程仍需要高精密度。隨著時代更迭，人工智慧的市場需求水漲船高。也因為市場的需求，許多科學家及工程師紛紛投入人工智慧的行列，於是各式新技術也應運而生。YOLOv1演算法[12]在發表的當時即造成轟動，因為該演算法是以單階段物件偵測的模式進行影像辨識及物件分類，因此不但可以即時地偵測物件，在當時的準確性表現也相當傑出。於隔年由YOLOv1演算法改良的YOLOv2演算法[13]，其改良目的在於提升即時性及準確性，因此透過批次標準化及錨框機制達到兩者的提升。時至今日，因YOLOv2演算法的模型以較為輕巧的架構組成，於是YOLOv2演算法在即時性的效果仍是YOLO演算法系列中不可動搖的地位。於是我們初步嘗試使用YOLOv2演算法應用於本計畫的研究內容。透過各種調整參數的方式，反覆測試YOLOv2演算法的模型是否具有良好的準確性。在使用晶片製程的影片測試時，造成漏檢的情形相當嚴重。經評估後僅有約五成的準確度，將無法達到台灣恩智浦公司期望的準確度。因此我們將焦點轉移至以YOLOv2演算法為基礎進行改良而誕生的YOLOv3演算法[14]。

YOLOv2演算法與YOLOv3演算法之間最大的區別在於YOLOv3演算法加入殘差網路[15]的概念。殘差網路可說是深度學習的最大助力，在沒有殘差網路的理論出現時，深度學習在建立模型時容易因為梯度消失的問題導致模型深度無法加深。在殘差網路出現後，紛紛有人透過將殘差網路的概念引進深度學習中。來自世界各地的學者經過實驗後發現殘差網路當中殘差連接的概念對於深度學習是有相當大的影響力，透過殘差連接的方式使梯度能夠不斷向下傳遞而不容易消失。因此現今許多有名的深度學習模型都使用到殘差網路的概念。透過相同影片進行測試後發現，YOLOv3演算法的模型進行評估後可得到約八成的準確度，與YOLOv2演算法的模型相比之下提升約三成的準確度。不過以實際進行產線的品質檢測而言，八成的準確度仍是差強人意的。我們同樣測試YOLOv4演算法[16]的效果，僅針對台灣恩智浦公司所提供的資料而言，會因為影像品質與內容的背景複雜程度等因素造成一定程度的影響。雖然對於檢測物件的種類信心程度極高，不過將正常晶片判斷成損壞晶片的頻率偏高，在誤報率的考量下將不利於產線的品質檢測。

最後我們決定使用於2020年問世的YOLOv5演算法。YOLOv5演算法與YOLOv4演算法之間的差異在於YOLOv5演算法於網路結構一開始使用Focus結構。目的與特徵壓縮相同，只是該做法是以拼接的方式將特徵圖拆分為四等分，並將四個等分串接在一起，實際上特徵的訊息並未受到影響，而以壓縮的方式則會失去一些影像的訊息。另外自動錨框計算則是針對不同資料集進行訓練時可為資料集設定初始錨框大小，使錨框大小適合於使用的資料集。最後是自動影像縮放的技巧，進行影像推論的過程，主要會將影像設定為長寬等長的大小。如果影像來源長寬比例較大時，則根據縮放係數將短邊形成的填充影像大幅減少，減少推論所需的計算量，增加推論的效率。以相同影片進行測試時，發現YOLOv5演算法的模型能夠達到零誤報的表現，在晶片製程的測試影片能夠達到100% 的準確度。不過由於在Workstation上執行物件偵測可能會造成接收即時影像時的影像延遲。於是我們提出採用較新版本的YOLOv5演算法加上嵌入式平台NVIDIA Jetson Nano與嵌入式平台NVIDIA Jetson Xavier NX的部署[17]來嘗試解決台灣恩智浦公司對於目前問題所遇到的瓶頸。

1. **國內外相關研究**
2. 金大開發高粱酒酒標檢測防偽技術 辨識正確率高達95%，2021-8-6。
3. YOLOv5 is Here: State-of-the-Art Object Detection at 140 FPS，2020-6-10。
4. Face Mask Detection using YOLOv5 for COVID-19，2020-11-24。
5. Augmented Reality Maintenance Assistant Using YOLOv5，2021-5-22。
6. Mold Detection on Food Surfaces Using YOLOv5，2021-5-27。
7. Short Communication: Detecting Heavy Goods Vehicles in RestAreas in Winter Conditions Using YOLOv5，2021-3-31。
8. Safety Helmet Detection Based on YOLOv5，2021-3-1。
9. Face Mask Recognition System with YOLOV5 Based on Image Recognition，2021-2-12。
10. Ship detection in optical sensing images based on YOLOv5，2021-1-27。
11. A Comparative Study of Underwater Marine Products Detection based on YOLOv5 and Underwater Image Enhancement，2021-5-1。
12. Design of Barcode Recognition System Based on YOLOV5，2021-8-13。
13. Using YOLOv5 Algorithm to Detect and Recognize American Sign Language，2021-7-26。
14. Improved YOLOv5 Network Model and Application in Safety Helmet Detection，2021-5-10。
15. Combined Channel and Spatial Attention for YOLOv5 during Target Detection，2021-8-26。
16. Deep Learning-based Face Mask Detection Using YoloV5，2021-5-31。
17. Smoking Behavior Detection Based On Improved YOLOv5s Algorithm，2021-7-26。
18. Detection of Flying Birds in Airport Monitoring Based on Improved YOLOv5，2021-4-26。
19. **重要參考文獻**
20. C. Wang, H. Mark Liao, Y. Wu, P. Chen, J. Hsieh and I. Yeh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2020, pp. 1571-1580.
21. T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 936-944.
22. S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, "Path Aggregation Network for Instance Segmentation," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 8759-8768.
23. X. Chen et al., "A Light-Weighted CNN Model for Wafer Structural Defect Detection," IEEE Access, vol. 8, pp. 24006-24018, 2020.
24. T. Tsai and Y. Lee, "Wafer Map Defect Classification with Depthwise Separable Convolutions," 2020 IEEE International Conference on Consumer Electronics (ICCE), 2020.
25. T. -H. Tsai and Y. -C. Lee, "A Light-Weight Neural Network for Wafer Map Classification Based on Data Augmentation," in IEEE Transactions on Semiconductor Manufacturing, vol. 33, no. 4, pp. 663-672, Nov. 2020.
26. N. Yu, Q. Xu and H. Wang, "Wafer Defect Pattern Recognition and Analysis Based on Convolutional Neural Network," in IEEE Transactions on Semiconductor Manufacturing, vol. 32, no. 4, pp. 566-573, Nov. 2019.
27. F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 1800-1807.
28. K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu and C. Xu, "GhostNet: More Features From Cheap Operations," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 1577-1586.
29. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, "Attention Is All You Need," arXiv:1706.03762 [cs], Jun. 2017.
30. J. Hu, L. Shen and G. Sun, "Squeeze-and-Excitation Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 7132-7141.
31. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788, 2016.
32. J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," IEEE Conference on Computer Vision and Pattern Recognition, pp. 6517-6525, 2017.
33. J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement,". arXiv:1804.02767, 2018.
34. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.
35. A. Bochkovskiy, C.-Y. Wang and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection", arXiv:2004.10934, 2020.
36. Y. Sun, C. Wang and L. Qu, "An Object Detection Network for Embedded System," 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), Shenyang, China, 2019, pp. 506-512.
37. M. Saqlain, Q. Abbas and J. Y. Lee, "A Deep Convolutional Neural Network for Wafer Defect Identification on an Imbalanced Dataset in Semiconductor Manufacturing Processes," IEEE Transactions on Semiconductor Manufacturing, vol. 33, no. 3, pp. 436-444, 2020.
38. J. Cha, S. Oh, D. Kim and J. Jeong, "A Defect Detection Model for Imbalanced Wafer Image Data Using CAE and Xception," 2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA), 2020.
39. N. Ma, X. Zhang, H.T. Zheng, J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," arXiv:1707.01083 [cs.CV], Jul. 2017.
40. N. Ma, X. Zhang, H.T. Zheng, J. Sun, "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture

Design," arXiv:1807.11164 [cs.CV], Jul. 2018.

1. A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861 [cs], Apr. 2017.
2. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520.
3. A. Howard et al., "Searching for MobileNetV3," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1314-1324.

**(三) 研究方法、進行步驟及執行進度**

1. **研究方法與原因**

**研究方法**

**第一年期程**

**（1）異常資料標記**

首先我們使用台灣恩智浦半導體股份公司用於半導體封測生產線資料的上傳、搜尋與運算的應用服務Oplus，資料集為每三分鐘接收一次資料。其中，包含了CPU、Memory、Disk Queue等資料以及不同編號的虛擬機(代號1V、2V、3V、4V......等)。我們主要觀察系統的CPU與Memory使用率資料作為主要特徵，如Fig. 4所示。

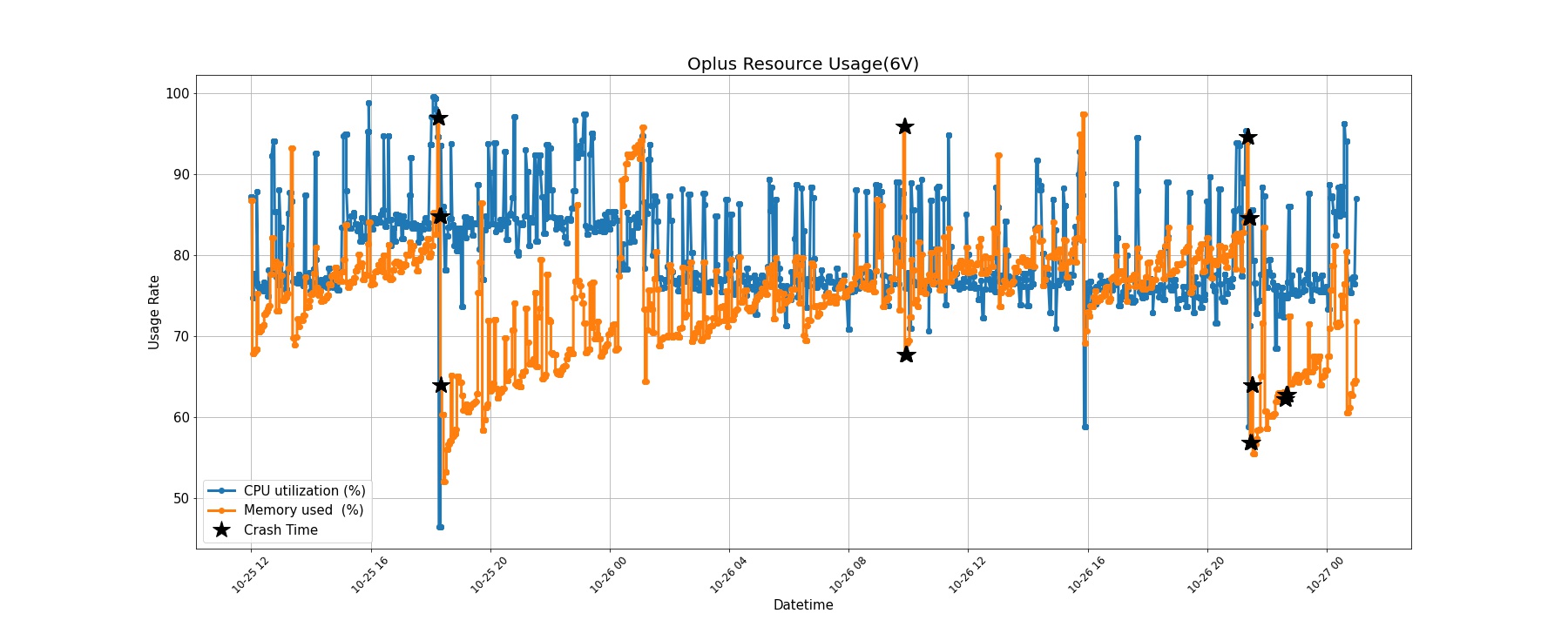


Fig. . Oplus資源使用率與異常時間標記

對於真實的異常時間是以一固定寬度的窗口標記。在實際的二元數字標記上我們希望當檢測模型遇到異常的前兆時發出警告。因此，資料集在異常區間內都標記為1，其餘正常資料我們標記為0，如此一來窗口內只要出現一個異常點的話即可認定此窗口可能出現異常。標記狀況如Fig. 5所示。

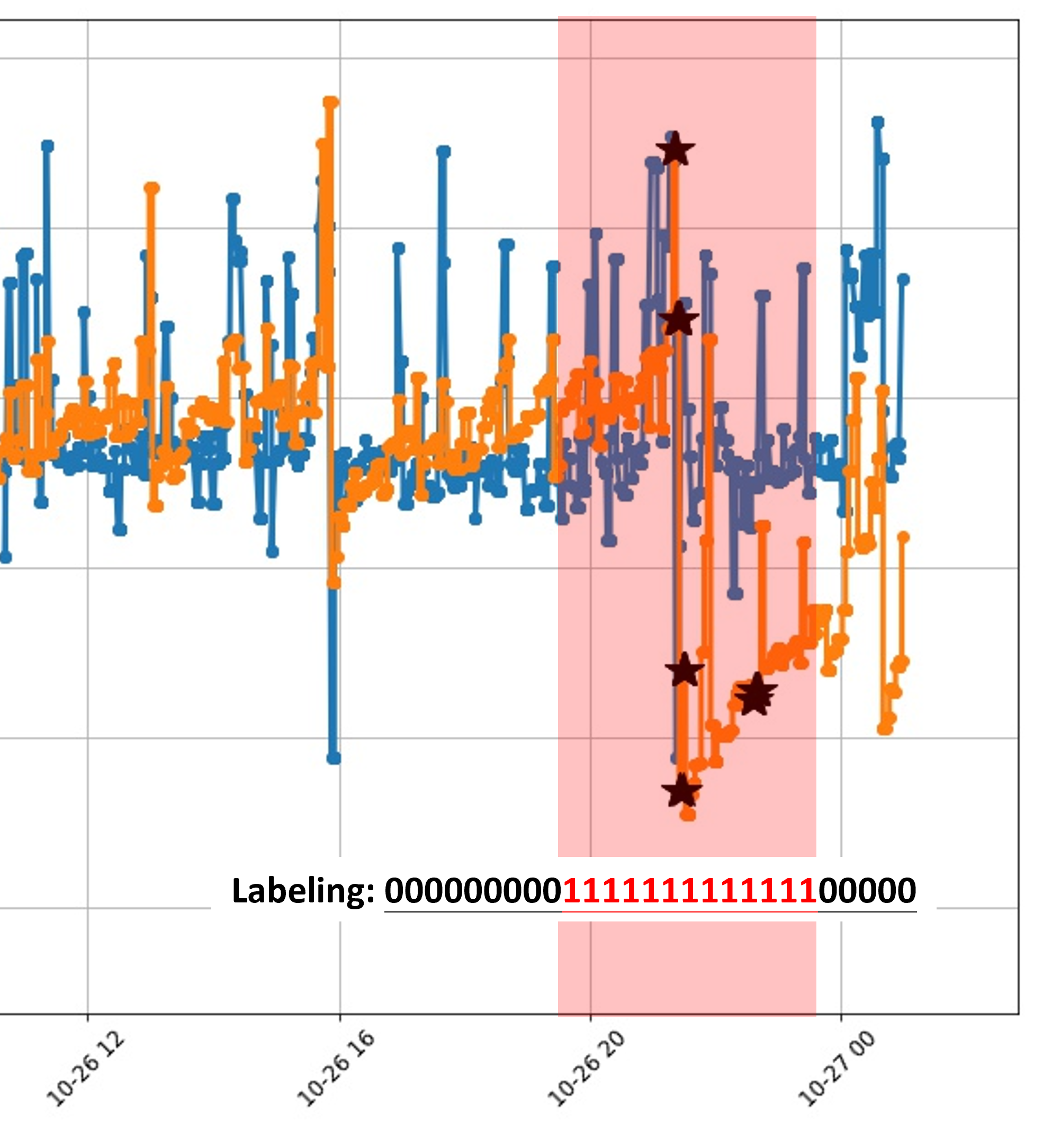


Fig. . 異常窗口的標記方法

**（2）建立深度強化學習環境**

本計畫使用RLAD[]方法建立強化學習環境，首先，我們將時間序列使用固定的滑動窗口擷取資料作為環境狀態(State)，並將動作空間(Action Space)限制為兩個離散狀態:0(正常)、1(異常)。將環境狀態作為輸入資料輸入策略網路(Policy Network)輸出代理人的決策，並且有一個概率決定代理是否遵從策略網路提出的建議。輸出的決策會與資料集的標籤計算回饋(Reward)，在此我們設定回饋函式為。最後，會有一個記憶模組(Replay Memory)儲存此轉移狀態，如Fig. 6所示。

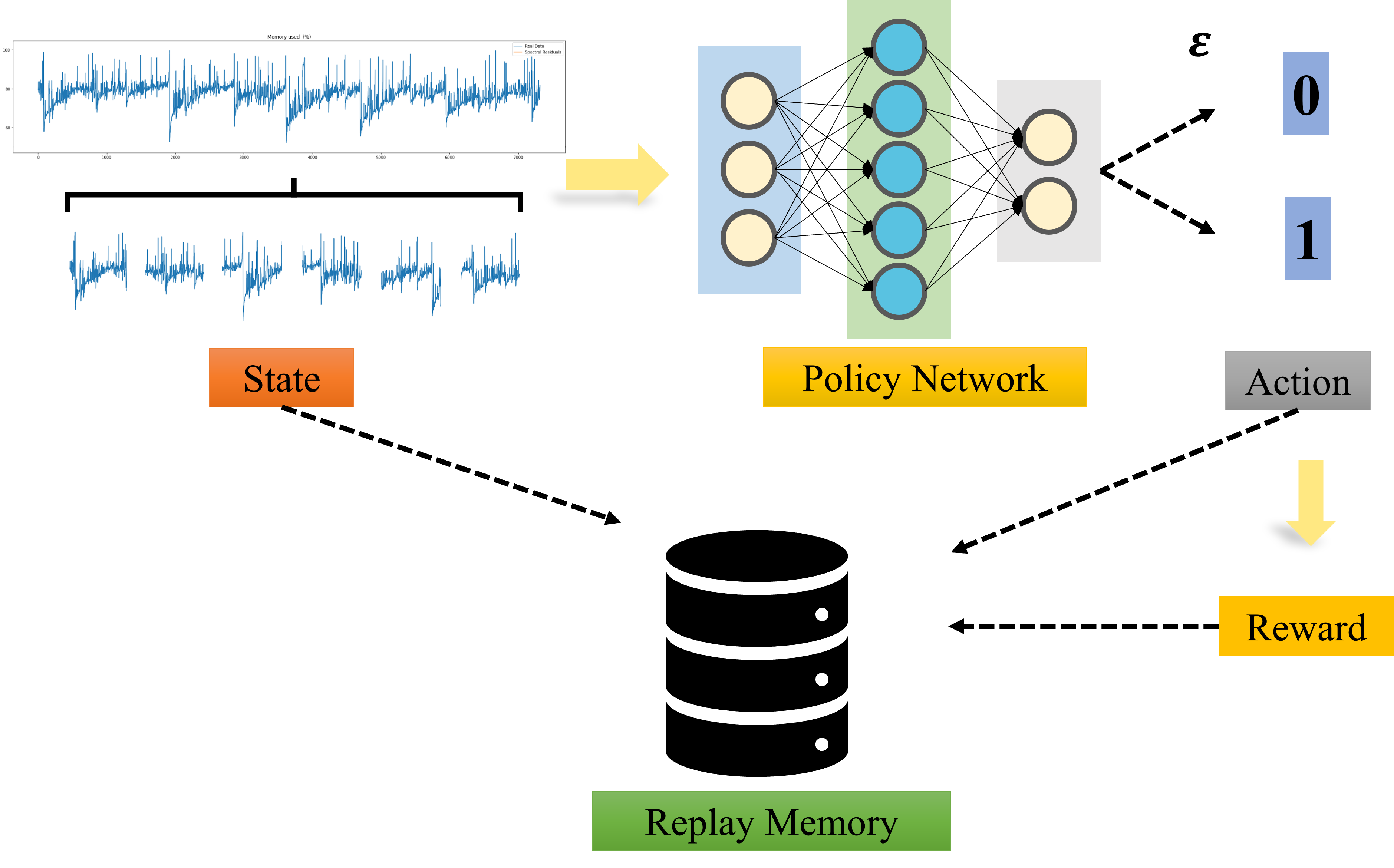


Fig. . 強化學習流程圖

我們額外加入一個隨機抽取記憶中的狀態與本身的輸出值執行梯度下降法更新參數，並在間隔固定的迭代次數後複製模型參數作為後進行下一次的回合訓練，如Fig. 7所示。

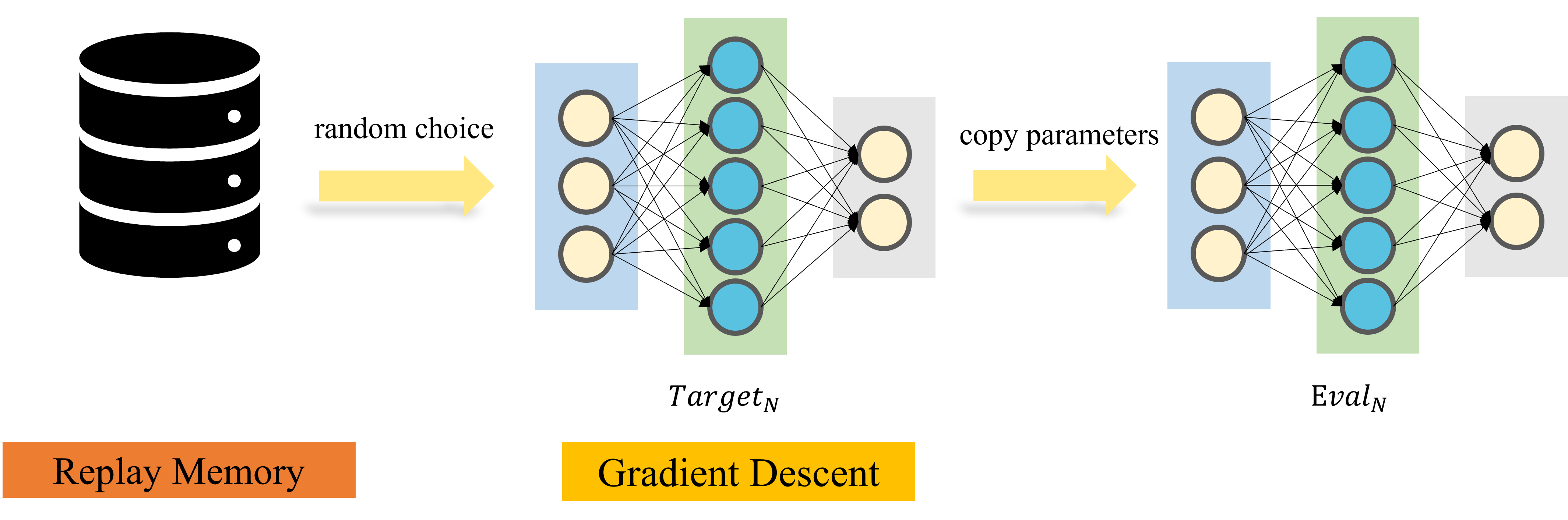


Fig. . 策略網路更新流程

**（3）策略模型評估**

本計畫以PyTorch做為訓練框架，過程會觀察每一個迭代累加的Reward數值，並透過觀測趨勢修改環境設定的獎勵與懲罰機制，如Fig. 8所示。



Fig. . 強化學習訓練過程之Reward趨勢

我們使用Precision、Recall、F1-score三項指標評估模型是否達到可以佈署的程度，公式如下。

|  |  |
| --- | --- |
|  | (1) |

|  |  |
| --- | --- |
|  | (2) |

|  |  |
| --- | --- |
|  | (3) |

**（4）佈署模型執行在線預測**

經過評估可使用之模型將會佈署至台灣恩智浦半導體股份公司的運算集群。該公司目前是使用Hadoop與Spark建立集群系統，使用Zabbix Server監控各個虛擬機的資源使用率。系統設定固定時間會將資料匯入HDFS(Hadoop Distributed File System)。本計畫使用Spark streaming將資料轉換成DStream(discretized stream)形式輸入在線預測模型進行故障預測。

**第二年期程**

**（1）建立元學習子任務**

本計畫融合了強化學習與Meta Learning演算法，提升強化學習對於不同的環境下的適應性。Meta Learning將資料集拆分為不同的小型分類任務並分別執行訓練與分類，並得到一個初始化模型去適應當前的任務。對於雲端系統的不同的應用服務下需要不同模型分別專注於分類當下環境的異常，若使用一般機器學習方法勢必會造成過多的時間成本。因此，我們運用該演算法的優勢減少管理人員新增或修改某項服務的檢測系統。

對於Meta Learning的強化學習，我們將Environment視作一個小型的分類任務，並小幅度修改強化學習環境作為元學習的子任務。原本作為靜態的強化學習環境修改為僅有一次的互動，在透過多執行緒分別建立多個環境訓練。Action為強化學習中的Agent判定是否異常，並將Reward設定為(TP, TN, -FP, -FN)。我們採用DNN建立作為Policy Network，Policy Network會接收環境狀態並輸出機率分布，再由分布中採樣一個Action。每次Task運作結束後會將State、Action、Reward資料儲存至Replay Memory儲存。(多執行緒->環境)

**（2）建立MAML演算法**

MAML(Model-Agnostic Meta-Learning)是一種與模型本身無關的學習演算法，目標在於多種任務中學習並從中得到任務的本質。其中，包含了Inner Loop和Outer Loop。Inner Loop負責產生多個Task，每個Task會採樣一小部份的時間序列訓練並保存經過梯度下降的參數。再來，Outer Loop將訓練後的參數進行測試，記錄損失並使用適合的優化函式更新最終網路參數。

本研究使用learn2learn套件與python語言開發。首先，我們的Inner Loop建立多個執行緒用於同時在多個強化學習環境下執行，並為每個環境建立一個Task。再來，將每個環境產生的State、Action、Reward、Loss儲存至Replay memory。在Outer Loop部分，我們選擇TRPO找出最佳策略。TRPO演算法重複使用策略的Trajectory增加樣本的使用率，也能確保強化學習在訓練的過程不會因為策略的改變影響模型的學習效果，該演算法也透過劃定一個可信任的策略學習區域，保證策略學習的穩定性和有效性。流程圖如Fig. 9所示。

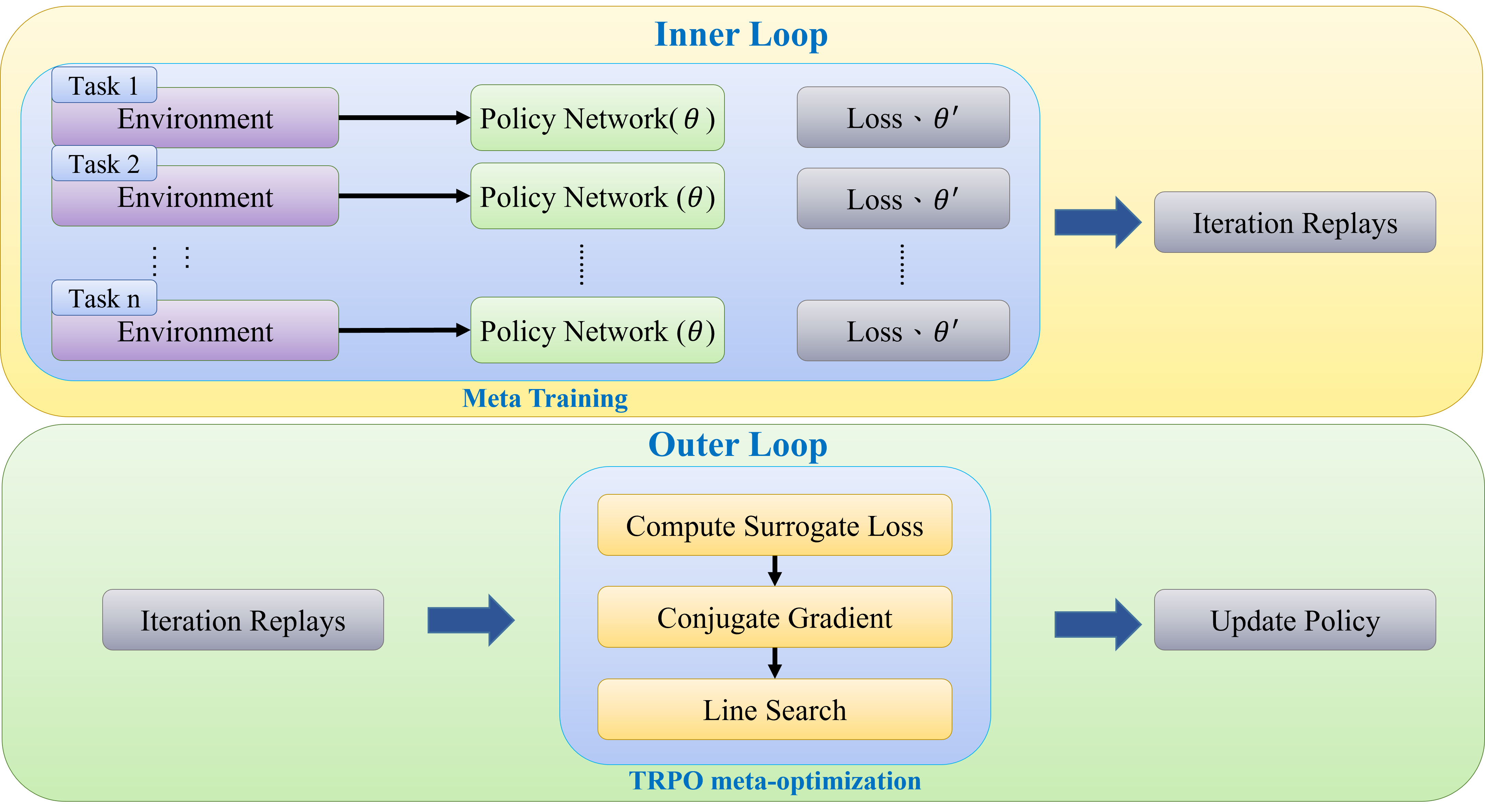


Fig. . MAML-RL訓練流程圖

**（3）訓練元強化模型**

本步驟首先使用OpenAI gym套件註冊資源使用率環境作為本實驗的Environment，並設置Inner Loop之超參數如下表。

|  |  |
| --- | --- |
| 項目 | 數值 |
| Hidden layer | [128,128,128] |
| Adapt learning rate | 0.5 |
| Number of iterations | 100 |
| Meta batch size | 10 |
| Number of workers | 10 |
| cuda | 1 |

每個Iteration設置記錄檔紀錄Reward與目前最終模型的準確度，並設置Tensorboard觀察訓練過程趨勢，如Fig. 10所示。



Fig. . Tensorboard觀察訓練過程

**（4）模型自適應**

上步驟的初始化模型

**第三年期程**

**（1）資料集準備**

本期程主要使用台灣恩智浦公司所提供的晶圓側面刮痕的資料集，此資料集具有許多在實際案例中常見的刮痕外型，如Fig. 28所示。首先因為這些資料已經透過灰階化處理，所以在一些較為不明顯的刮痕會與晶圓本身具有的紋路相似。於是需要專家先進行資料的判斷，決定刮痕的真實位置，避免刮痕與紋路混淆的問題。

|  |  |
| --- | --- |
|  |  |
| (a) | (b) |
|  |  |
| (c) | (d) |
| Fig. 28. 晶圓側面刮痕型態 (a) 局部型 (b) 直線型 (c) 不規則型(d) 短塊型 | |

**（2）人工資料標記**

經過專家確認真實的刮痕位置後則可對目前的資料進行刮痕位置的標記。我們使用的標記工具為LabelImg，如Fig. 29所示。標記方式則是遵循專家提供的指示將每筆資料一一進行標記，標記後產生的標記檔案對應一筆資料可得到的資訊。標記的檔案中包含五項資訊如Fig. 30所示。第一項數值代表的是類別的編號，從零開始編號，根據使用者定義的類別數量而有相對應的編號。第二及第三項為x及y，分別對應各自的標記框之中心座標與圖片寬高的比值。第四及第五項為w及h，分別對應各自的標記框之寬高與圖片寬高的比值。

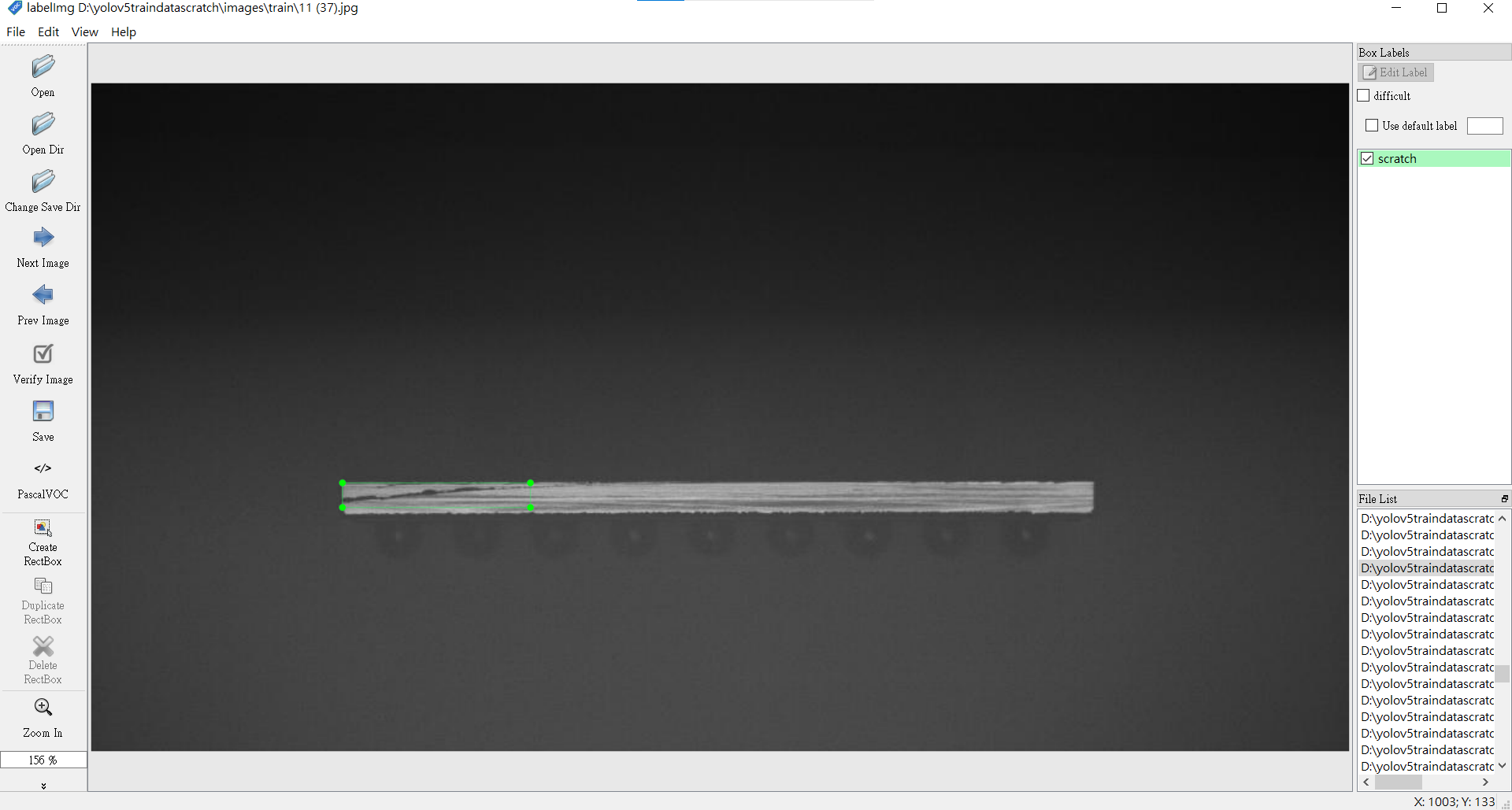


Fig. 29. 以LabelImg進行人工標記晶圓側面的刮痕

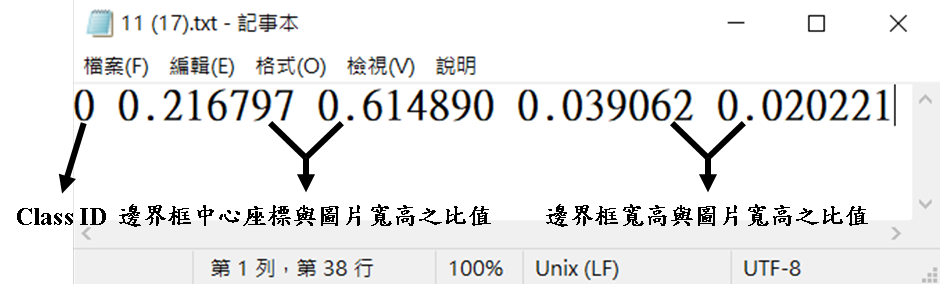


Fig. 30. 標記檔案之內容截圖

**（3）YOLOv5模型訓練**

本期程以基於PyTorch做為訓練框架的YOLOv5使用於晶圓側面刮痕檢測，利用標記完成的晶圓側面資料集訓練適合於分辨晶圓側面是否有刮痕的模型。實際訓練過程的進度截圖如Fig. 31所示。在物件偵測領域當中，訓練完成的模型通常會以mAP (mean average precision) 做為評估模型表現的依據，如Fig. 32所示。當線下面積數值越接近一時，表示訓練好的模型對於目前的情境能具有健全的表現。若模型的表現不夠理想，則需考量參數調整以及資料集的標記品質。

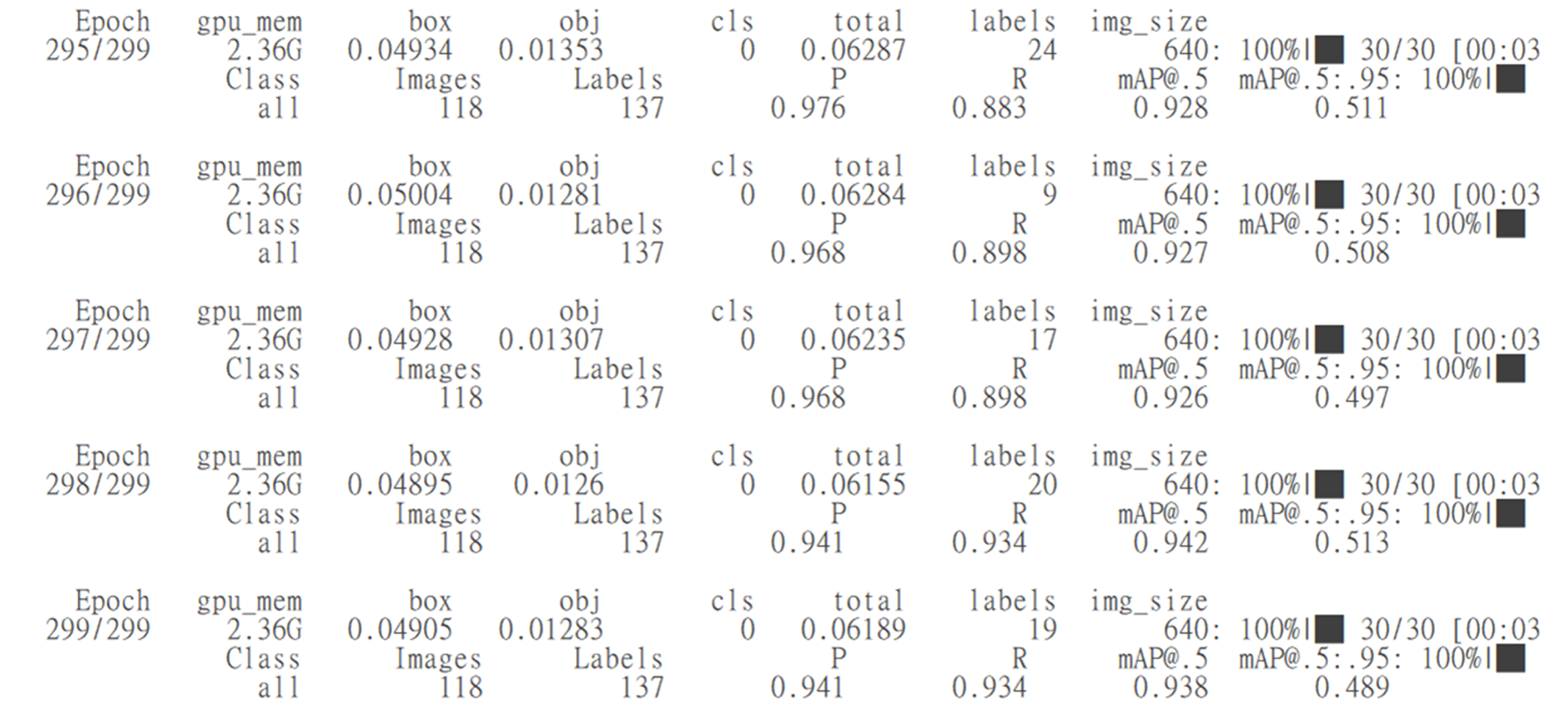


Fig. 31. 實際訓練過程進度截圖

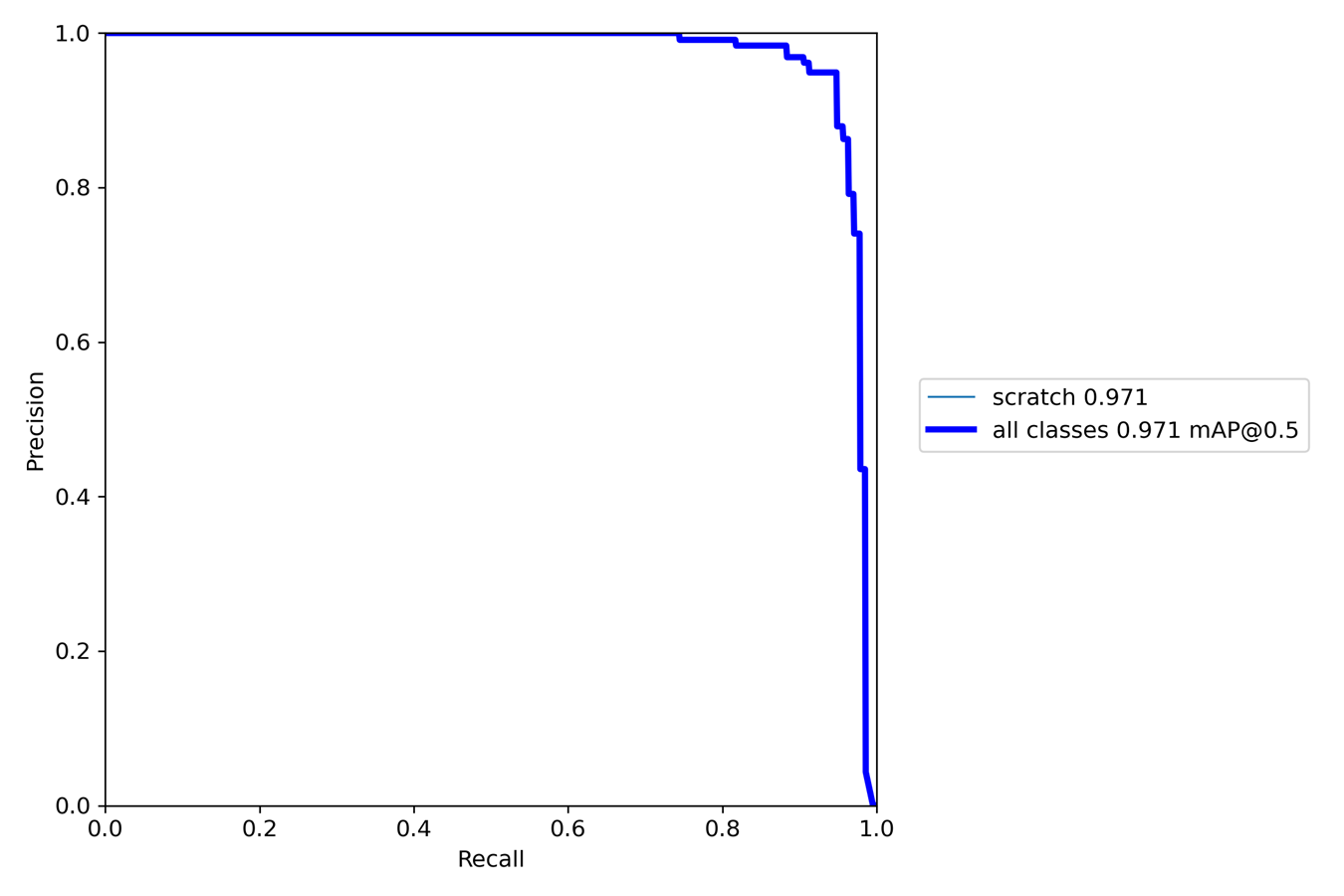


Fig. 32.晶圓側面刮痕檢測的平均精度均值折線圖

**（4）嵌入式平台進行推論**

本期程中我們將使用嵌入式平台NVIDIA Jetson Xavier NX做為推論流程的環境部署。此嵌入式平台具有相當優異的運算性能以及與Jetson Nano具有相同大小的尺寸，相當適合進行需要高計算量的本地端部署，如Fig. 33所示。訓練完成適合於資料集的模型後則可利用Jetson Xavier NX嵌入式平台進行物件偵測顯示辨識的成果，如Fig. 34所示。



Fig. 33. 使用嵌入式平台Jetson Xavier NX在本地端部署

|  |  |
| --- | --- |
|  |  |
| (a) | (b) |
|  |  |
| (c) | (d) |
| Fig. 34. GSEH-YOLOv5物件偵測模型的辨識結果 (a) 局部型 (b) 直線型 (c) 不規則型 (d) 短塊型 | |

**（5）不同架構的YOLOv5模型比較**

起初在Jetson Xavier NX嵌入式平台上使用傳統YOLOv5架構的模型測試該模型在測試資料集的表現，不過由於評估指標不夠理想，於是我們決定調整傳統YOLOv5的架構，目的是優於傳統YOLOv5架構的表現。最後，我們將三種論文所提出的架構取代於傳統YOLOv5的架構當中，分別是ShuffleNet[20][21]、MobileNet[22][23][24]以及GhostNet。以三種不同的特徵擷取方式與傳統YOLOv5的架構訓練適合於目前資料集的模型，如Fig. 35所示。實驗結果表明，以三種不同架構搭建的YOLOv5模型與傳統的YOLOv5模型相比，四種模型皆有其優勢之處，如Table 3所示。若以速度的觀點比較四種模型的表現，由GhostNet做為特徵擷取的架構能夠在最短的時間內得到推論的結果，其次是使用MobileNet的架構，再來是使用傳統的CSPNet架構，最後是使用ShuffleNet的架構。若以準確度的觀點比較四種模型的表現，由GhostNet做為特徵擷取的架構能夠在本期程的測試資料得到最好的準確度，其次是使用傳統的CSPNet架構，再來是使用ShuffleNet的架構，最後是使用MobileNet的架構。若以參數量的觀點比較四種模型的表現，由ShuffleNet做為特徵擷取的架構能夠使模型的參數量最少，其次是使用MobileNet的架構，再來是使用GhostNet的架構，最後是使用傳統的CSPNet架構。

若以計算量的觀點比較四種模型的表現，由ShuffleNet做為特徵擷取的架構能夠使模型的計算量最少，其次是使用MobileNet的架構，再來是使用GhostNet的架構，最後是使用傳統的CSPNet架構。綜合以上觀點，GhostNet做為特徵擷取架構的YOLOv5模型能夠在Jetson Xavier NX嵌入式平台上以最快的速度以及最高的準確度進行推論過程。

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| (a) | (b) | (c) | (d) |
| Fig. 35. 四種不同YOLOv5物件偵測模型的架構 (a) CSP架構 (b) GhostNet架構 (c) ShuffleNet架構 (d) MobileNet架構 | | | |

Table 3. 傳統與三種不同模型的速度及準確度於嵌入式平台Jetson Xavier NX之表現

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | YOLOv5 | GSEH-YOLOv5 | ShuffleNet-YOLOv5 | MobileNet-YOLOv5 |
| Speed (fps) | 27.8 | 41.8 | 25.8 | 37.7 |
| Precision (%) | 96.4 | 97.1 | 93.9 | 91.8 |

**（6）智動化檢測與分析**

利用智動化晶圓側面刮痕檢測可以準確將有刮痕的晶圓揀選出來。此外，因為能夠得到刮痕的位置，於是我們會將刮痕的位置分成左右兩側。透過兩邊的比例可以比較容易讓專家研判切割機在進行切割時的狀態，造成刮痕較多的一側會有比較高的機會是因為切割機本身下刀時有些微的偏差導致刮痕的產生。

**進行步驟**

**第一年期程**

(1) 以原始影像轉換為晶片資料集。(2) 將晶片資料集進行晶片位置標記。(3) 晶片資料集分成訓練、驗證及測試資料。(4) PyTorch訓練環境的建置。(5) YOLOv5模型訓練。(6) 將推論環境與模型部署於嵌入式平台。(7)

全自動晶片輪廓檢測。(8) 出現損壞晶片時通知使用者。

**第二年期程**

(1) 以卷積型自動編碼器產生各種不同類別的晶圓圖資料集。(2) 將晶圓圖資料集進行晶圓瑕疵種類標記。(3) 晶圓圖資料集分成訓練、驗證及測試資料。(4) YOLOv5模型訓練。(5) 將推論環境與模型部署於嵌入式平台。(6) 全自動晶圓圖瑕疵型態檢測。(7) 紀錄出現瑕疵型態的時間戳記。

**第三年期程**

(1) 影像處理過濾部分資料訊息。(2) 將晶圓側面刮痕資料集進行刮痕位置標記。(3) 晶圓側面刮痕資料集分成訓練、驗證及測試資料。(4) YOLOv5模型訓練。(5) 將推論環境與模型部署於嵌入式平台。(6) 不同架構的YOLOv5模型之效能比較。(7) 全自動晶圓側面刮痕檢測。(8) 將有刮痕與無刮痕進行分群。(9) 計算刮痕於兩側出現的比例。(10) 針對合作廠商進行技術轉移。

**執行進度**

**第一年期程**

(1) 訓練資料與驗證資料的影像資料及標記資料設置。(2) 訓練物件偵測模型的參數設置。(3) 訓練過程之設定參數有效性測試。(4) 部署嵌入式平台可執行推論環境設置。(5) 嵌入式平台進行推論時的準確率測試。(6) 嵌入式平台進行推論時的速度測試。(7) 改良傳統模型於嵌入式平台進行推論時的準確率測試。(8) 改良傳統模型於嵌入式平台進行推論時的速度測試。(9) 紀錄損壞晶片發生位置與晶片槽的編號設置。

**第二年期程**

(1) 利用自動編碼器生成晶圓圖資料與原始資料之整合適用性測試。(2) 訓練資料與驗證資料的影像資料及標記資料設置。(3) 訓練物件偵測模型的參數設置。(4) 訓練過程之設定參數有效性測試。(5) 部署嵌入式平台可執行推論環境設置。(6) 嵌入式平台進行推論時的準確率測試。(7) 嵌入式平台進行推論時的速度測試。(8) 改良傳統模型於嵌入式平台進行推論時的準確率測試。(9) 改良傳統模型於嵌入式平台進行推論時的速度測試。(10) 辨識結果之刮痕位置其比例關係之設置。

**第三年期程**

(1) 利用影像處理技術於訓練模型之適宜性測試。(2) 訓練資料與驗證資料的影像資料及標記資料設置。(3) 訓練物件偵測模型的參數設置。(4) 訓練過程之設定參數有效性測試。(5) 部署嵌入式平台可執行推論環境設置。(6) 嵌入式平台進行推論時的準確率測試。(7) 嵌入式平台進行推論時的速度測試。(8) 改良傳統模型於嵌入式平台進行推論時的準確率測試。(9) 改良傳統模型於嵌入式平台進行推論時的速度測試。(10) 瑕疵晶圓圖各種瑕疵型態對應發生時間點設置。(11) 針對合作廠商進行技術轉移。

**二、預計可能遭遇之困難及解決途徑**

**第一年期程**

(1) 由於晶片的位置經過加工貼模，以原先的影像資料擷取的訓練資料集可能會對目前的情境造成檢測不利因素。因此需要將使用的資料集透過加工的方式，使其符合目前情境的畫面。

(2) 因為物件偵測演算法YOLOv5目前並沒有發表論文提供使用者閱讀，再深入瞭解詳細做法會需要花費相當多時間。因此透過網路資源解析該演算法可減少研讀時間，且更容易理解使用這些做法的用意。

(3) 使用物件偵測時通常需要GPU以應付龐大的運算量，因此使用具有GPU運算能力的嵌入式平台以避免過低的運算能力導致檢測能力受限。

(4) 若傳統YOLOv5模型的成效不彰，考慮改採用成效較佳的改良型YOLOv5模型取代之。

(5) 合作廠商接收智動化晶片輪廓檢測系統初期，會存在成本與需求上的考量。在軟體開發以及人員培訓上對於廠商而言可能需要多次規劃，因此需要和合作廠商長期提供意見以及支援協助。

**第二年期程**

(1) 由於使用資料集有各類別資料數不平衡的問題，導致訓練模型時的困難。因此使用資料增強技術，將數量較少的資料增加以保持各類別資料數平衡。

(2) 因為物件偵測演算法YOLOv5目前並沒有發表論文提供使用者閱讀，再深入瞭解詳細做法會需要花費相當多時間。因此透過網路資源解析該演算法可減少研讀時間，且更容易理解使用這些做法的用意。

(3) 使用物件偵測時通常需要GPU以應付龐大的運算量，因此使用具有GPU運算能力的嵌入式平台以避免過低的運算能力導致檢測能力受限。

(4) 若傳統YOLOv5模型的成效不彰，考慮改採用成效較佳的改良型YOLOv5模型取代之。

(5) 合作廠商接收智動化晶圓圖瑕疵型態檢測系統初期，會存在成本與需求上的考量。在軟體開發以及人員培訓上對於廠商而言可能需要多次規劃，因此需要和合作廠商長期提供意見以及支援協助。

**第三年期程**

(1) 由於使用資料集有機密資訊不宜公開，因此必須透過影像處理的方式將含有機密資訊的資料消除以避免機密外洩。

(2) 因為物件偵測演算法YOLOv5目前並沒有發表論文提供使用者閱讀，再深入瞭解詳細做法會需要花費相當多時間。因此透過網路資源解析該演算法可減少研讀時間，且更容易理解使用這些做法的用意。

(3) 使用物件偵測時通常需要GPU以應付龐大的運算量，因此使用具有GPU運算能力的嵌入式平台以避免過低的運算能力導致檢測能力受限。

(4) 若傳統YOLOv5模型的成效不彰，考慮改採用成效較佳的改良型YOLOv5模型取代之。

(5) 合作廠商接收智動化晶圓側面刮痕檢測系統初期，會存在成本與需求上的考量。在軟體開發以及人員培訓上對於廠商而言可能需要多次規劃，因此需要和合作廠商長期提供意見以及支援協助。

**(四) 預期完成之工作項目及成果**

**ㄧ、本研究計畫預期完成之項目**

本計畫三年期程的甘特圖，如Table 4所示。

Table 4. 三年期程甘特圖Gantt Chart

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 月數  工作 | 三年計畫 | | | | | | | | | | | |
| 1-3 | 4-6 | 7-9 | 10-12 | 13-15 | 16-18 | 19-21 | 22-24 | 25-27 | 28-30 | 31-33 | 34-36 |
| 月. | 月. | 月. | 月 | 月. | 月 | 月. | 月 | 月 | 月 | 月. | 月 |
| 基於強化學習的伺服器異常檢測 |  |  |  |  |  |  |  |  |  |  |  |  |
| 第一期各項效能測試與人員訓練 |  |  |  |  |  |  |  |  |  |  |  |  |
| 元強化學習的自適應流程 |  |  |  |  |  |  |  |  |  |  |  |  |
| 第二期各項效能測試與人員訓練 |  |  |  |  |  |  |  |  |  |  |  |  |
| 基於元策略流程的主動異常檢測系統 |  |  |  |  |  |  |  |  |  |  |  |  |
| 第三期各項效能測試與人員訓練 |  |  |  |  |  |  |  |  |  |  |  |  |
| 針對合作廠商進行技術轉移 |  |  |  |  |  |  |  |  |  |  |  |  |
| 撰寫研究報告和公佈研究結果 |  |  |  |  |  |  |  |  |  |  |  |  |
| 預期的進度累進比例 | **15** | **35** | **40** | **50** | **60** | **70** | **75** | **80** | **85** | **90** | **95** | **100** |

**第一年期程**

(1) 完成訓練資料與驗證資料的影像資料及標記資料設置。(2) 完成訓練物件偵測模型的參數設置。(3) 完成訓練過程之設定參數有效性測試。(4) 完成部署嵌入式平台可執行推論環境設置。(5) 完成嵌入式平台進行推論時的準確率測試。(6) 完成嵌入式平台進行推論時的速度測試。(7) 完成改良傳統模型於嵌入式平台進行推論時的準確率測試。(8) 完成改良傳統模型於嵌入式平台進行推論時的速度測試。(9) 完成紀錄損壞晶片發生位置與晶片槽的編號設置。

**第二年期程**

(1) 完成利用自動編碼器生成晶圓圖資料與原始資料之整合適用性測試。(2) 完成訓練資料與驗證資料的影像資料及標記資料設置。(3) 完成訓練物件偵測模型的參數設置。(4) 完成訓練過程之設定參數有效性測試。(5) 完成部署嵌入式平台可執行推論環境設置。(6) 完成嵌入式平台進行推論時的準確率測試。(7) 完成嵌入式平台進行推論時的速度測試。(8) 完成改良傳統模型於嵌入式平台進行推論時的準確率測試。(9) 完成改良傳統模型於嵌入式平台進行推論時的速度測試。(10) 完成辨識結果之刮痕位置其比例關係之設置。

**第三年期程**

(1) 完成利用影像處理技術於訓練模型之適宜性測試。(2) 完成訓練資料與驗證資料的影像資料及標記資料設置。(3) 完成訓練物件偵測模型的參數設置。(4) 完成訓練過程之設定參數有效性測試。(5) 完成部署嵌入式平台可執行推論環境設置。(6) 完成嵌入式平台進行推論時的準確率測試。(7) 完成嵌入式平台進行推論時的速度測試。(8) 完成改良傳統模型於嵌入式平台進行推論時的準確率測試。(9) 完成改良傳統模型於嵌入式平台進行推論時的速度測試。(10) 完成瑕疵晶圓圖各種瑕疵型態對應發生時間點設置。(11) 完成針對合作廠商進行技術轉移。

**二、對於學術研究、國家發展及其他應用方面預期之貢獻**

**第一年期程**

(1) 學習標記資料的使用技術。(2) 學習標記檔案轉換格式的使用技術。(3) 獲得部署推論環境至嵌入式平台之相關技術。(4) 熟知開源軟體的使用技術。(5) 熟知並運用高效能的YOLOv5物件偵測演算法。(6) 學習改良YOLOv5物件偵測模型之相關技術。(7) 有效運用智動化晶片輪廓檢測系統解決效率問題。(8) 提升晶片的產能以及加速品質監控。

**第二年期程**

(1) 解決資料數量不平衡的相關技術。(2) 學習標記資料的使用技術。(3) 獲得部署推論環境至嵌入式平台之相關技術。(4) 熟知開源軟體的使用技術。(5) 熟知並運用高效能的YOLOv5物件偵測演算法。(6) 學習改良YOLOv5物件偵測模型之相關技術。(7) 有效運用智動化晶圓圖瑕疵型態分類檢測系統解決效率問題。(8) 提升晶圓製程的效率以及加速品質監控。

**第三年期程**

(1) 解決機密資訊外洩的影像處理使用技術。(2) 學習標記資料的使用技術。(3) 獲得部署推論環境至嵌入式平台之相關技術。(4) 熟知開源軟體的使用技術。(5) 熟知並運用高效能的YOLOv5物件偵測演算法。(6) 學習改良YOLOv5物件偵測模型之相關技術。(7) 有效運用智動化晶圓側面刮痕檢測系統解決效率問題。(8) 提升晶圓製程的效率以及加速品質監控。

**三、參與之工作人員，預期可獲之訓練**

**第一年期程**

(1) 熟知開源軟體開發與使用技術。(2) 獲得軟體工程相關概念與技術。(3) 智動化檢測的相關概念與使用技術。(4) 學習影像辨識的相關概念與技術。(5) 理解閉路電視的概念與相關技術。(6) 學習通訊協定的相關概念。(7) 學習深度學習的概念與技術。

**第二年期程**

(1) 熟知開源軟體開發與使用技術。(2) 獲得軟體工程相關概念與技術。(3) 智動化檢測的相關概念與使用技術。(4) 學習影像辨識的相關概念與技術。(5) 學習深度學習的概念與技術。(6) 學習資料增強的概念與技術。

**第三年期程**

(1) 熟知開源軟體開發與使用技術。(2) 獲得軟體工程相關概念與技術。(3) 智動化檢測的相關概念與使用技術。(4) 學習影像辨識的相關概念與技術。(5) 學習深度學習的概念與技術。