

白皮書

AI用於影像分析

基於機器學習和深度學習之分析的考量要點

3月2021

目錄

1	摘要	3
2	簡介	4
3	AI、機器學習和深度學習	4
	3.1 機器學習	4
	3.2 深度學習	5
	3.3 傳統機器學習和深度學習	6
4	機器學習的階段	6
	4.1 資料收集和資料註解	7
	4.2 訓練	7
	4.3 測試	8
	4.4 建置	8
5	邊際分析	8
6	硬體加速	8
7	AI仍然在初期發展階段	9
8	最佳分析效能的考量要點	9
	8.1 影像可用性	9
	8.2 偵測距離	10
	8.3 警報與錄影設定	10
	8.4 維護	11
9	隱私與個人誠信	11
10	附錄	13
	10.1 神經網路	13
	10.2 卷積神經網路(CNN)	13

1 摘要

AI影像分析是影像監控產業中討論最熱烈的主題之一。部分應用可大幅加速資料分析和自動化重複任務。但現在的AI解決方案無法取代人類操作者的經驗和決策技能。實際的優勢在於組合：利用AI解決方案改進並提升人類效率。

AI概念涵蓋機器學習演算法和深度學習演算法。這兩者都可自動建立數學模型，使用大量範例資料(訓練資料)，不需具體程式設計就可獲得計算結果的能力。AI演算法是透過迭代過程開發，其中會循環重複收集訓練資料、標示訓練資料、使用標示資料訓練演算法和測試訓練後演算法，直到達到所需的品質等級為止。之後演算法就可用於分析應用程式，可供採購並建置在監控地點。此時所有訓練已完成，且應用程式將不再學到任何新東西。

AI影像分析的典型任務是視覺偵測影像串流中的人車，並區分這兩者。機器學習演算法學習定義這些物件的視覺特徵組合。深度學習演算法更精細，並且若專門訓練，可以偵測更複雜的物件。但也需要投入更大量資源到開發和訓練中，且使用完成的應用程式時，需要更多的運算資源。對於明確指定的監控需求，應該考慮專屬的最佳化機器學習應用程式是否就已足夠。

演算法開發和攝影機的處理能力日益提升，已經可以在攝影機上(基於邊際)直接執行進階AI影像分析，而不需要在伺服器上(基於伺服器)進行計算。這可提供更好的即時功能，因為應用程式能夠立即存取未壓縮的影像資料。透過攝影機內專用的硬體加速器，例如MLPU (機器學習處理單元)和DLPU (深度學習處理單元)，能夠以較CPU或GPU (圖形處理單元)更節能的方式實現邊際分析。

安裝AI影像分析應用程式之前，必須仔細研讀並遵循製造商基於已知前置條件和限制的建議。每個監控安裝設置都是獨特的，且應在每個監控地點評估應用程式的效能。若發現品質低於預期，應進行全面性調查，而非僅著重於分析應用程式本身。影像分析的效能取決於許多因素，涉及攝影機硬體、攝影機設定、影像畫質、情境動態和照明。在許多情況下，知道這些因素的影像並相應最佳化，能夠提升安裝設置的影像分析效能。

由於AI應用至監控的情況日益普遍，必須審慎討論何時何地應用技術，以便在運行效率的優勢與新使用案例之間取得平衡。

2 簡介

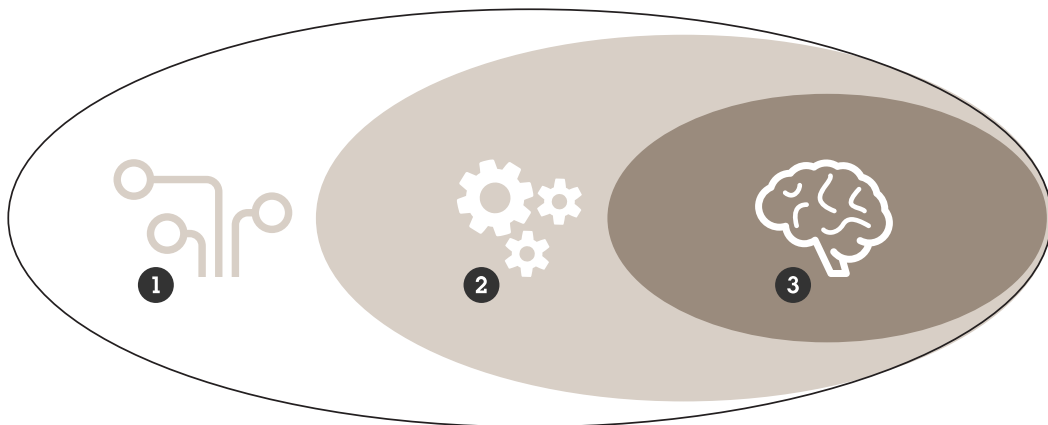
自從發明出第一台電腦，AI人工智慧就開始發展，且一直是爭議的焦點。雖然最具革命性的應用還未到來，目前AI技術已廣泛用於進行應用中定義明確的任務，例如語音辨識、搜尋引擎和虛擬助理。AI在醫療上的應用也日益普及，可在例如X光診斷和視網膜掃描分析中提供寶貴的資訊。

AI影像分析是影像監控產業中討論最熱烈的主題之一，且大家的期望非常高。市場上已經有應用程式使用AI演算法，成功加速資料分析和自動化重複任務。但在更廣泛的監控情境下，僅應將現在及日後短時間內的AI，視為打造準確解決方案過程中的多項要素之一。

本白皮書提供機器學習和深度學習演算法的技術背景，以及如何開發並應用至影像分析。包括簡短說明AI加速硬體，以及在邊際上相較於在伺服器上，運行AI分析的優缺點。本白皮書也檢視最佳化AI影像分析效能的前置條件，將多種因素納入考量。

3 AI、機器學習和深度學習

人工智慧(AI)是與可解決複雜任務，同時似乎可展現出智慧特徵之機器相關的廣泛概念。深度學習和機器學習是AI的子集合。



- 1 人工智慧
- 2 機器學習
- 3 深度學習

3.1 機器學習

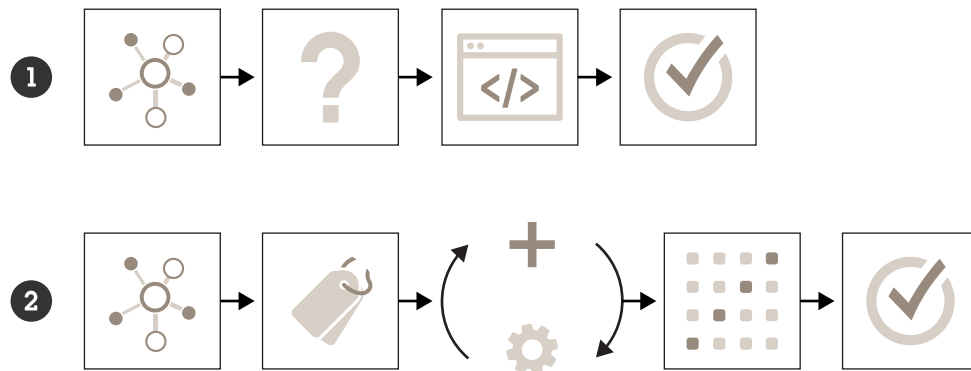
機器學習是AI中的一個子集合，使用統計學習演算法打造有能力在訓練期間自動學習和改進，而不用具體編寫程式的系統。

我們在本節就電腦視覺情境區分傳統程式設計和機器學習，這個領域是透過分析影像或影片，讓電腦理解場景中發生的情況。

傳統程式設計電腦視覺依據的方法為，計算影像的*特徵*，例如由電腦程式尋明顯邊緣和角落點。這些特徵需要由知道影像資料中哪些部分重要的演算法開發人員手動定義。開發人員接著結合這些特徵，讓演算法能夠推斷在場景中找到什麼。

機器學習演算法使用大量範例資料(訓練資料)建立數學模型，不需具體設計程式，就可獲得透過計算結果做出決定的能力。特徵仍然為人工開發，不過這些特徵的組合方式，由演算法自行透過接觸大量標示或註解的訓練資料而學習得到。在本文中，我們將這種在學習組合中使用人工開發特徵的技術，稱為傳統機器學習。

換句話說，針對機器學習應用程式，我們需要訓練電腦以取得我們想要的程式。會由人類收集並註解資料，有時可能由伺服器電腦協助進行預先註解。結果會餵入系統中並持續這個過程，直到應用程式學到足以偵測我們想要的東西，例如特定類型的車輛為止。訓練過的模型會成為程式。請注意在程式完成時，系統不會繼續學習任何新東西。



- 1 傳統程式設計：
收集資料。定義程式條件。編寫程式代碼(由人類進行)。完成。
- 2 機器學習：
收集資料。標示資料。模型進行迭代訓練過程。完成的訓練模型會成為程式。完成。

建立電腦視覺程式時，AI優於傳統程式設計的優點是有能力處理大量資料。電腦能夠瀏覽數千張影像而不會轉移注意力，而人類程式設計師在一段時間後就會疲累且無法專注。透過這種方式，AI能夠打造更準確的應用程式。不過應用程式越複雜，機器越難產生想要的結果。

3.2 深度學習

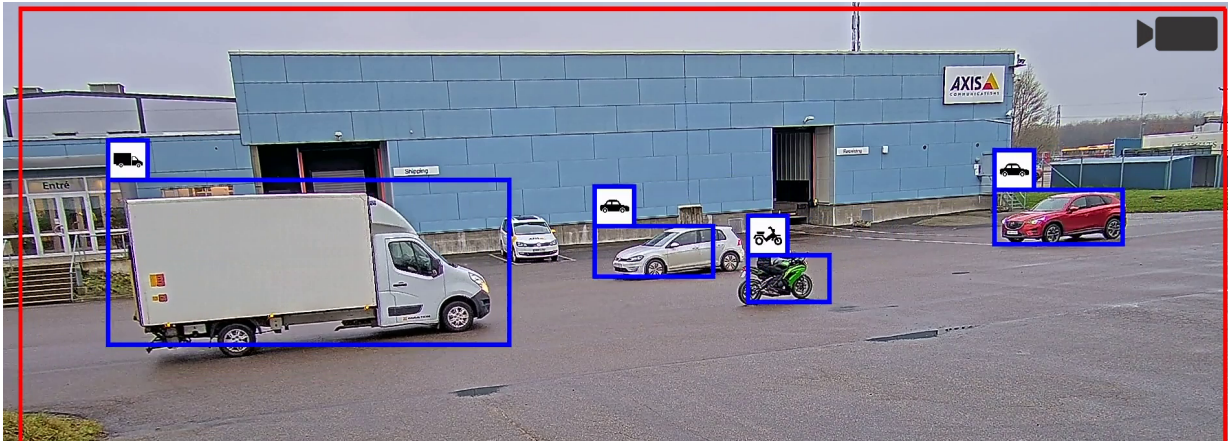
深度學習是機器學習的細化版本，在產生輸出的深層規則結構中，以資料導向方式學習特徵抽取和如何結合這些特徵。演算法可自動定義要找尋訓練資料中的哪些特徵。也可以學習非常深層的特徵鏈結組合結構。

深度學習中使用的演算法，核心理念源自神經元的運作方式，以及大腦如何透過以鏈結規則的深度階層或網路結合神經元輸出，以使用這些形成高階知識。大腦是一個系統，其中組合本身也由神經元構成，去除特徵抽取和特徵組合之間的差異，讓這兩者在某些方面等同。這些結構由研究人員模擬成為所謂的類神經網路，這是深度學習中最廣泛使用的演算法類型。有關神經網路的簡短概述，請參見本文件的附錄。

使用深度學習演算法可以建立精細的視覺偵測器，並自動訓練它們偵測非常複雜的物件，不會受到大小、旋轉和其他變因影響。

達成這個彈性的原因是，深度學習系統比起傳統機器學習系統，可以從更大量的資料以及差異更大的資料學習。在大部分情況下，表現將顯著優於人工開發的電腦視覺演算法。這

讓深度學習特別適合人類專家無法輕易建立特徵組合的複雜問題，例如影像分類、語言處理和物件偵測。



依據深度學習的物件偵測可以分類複雜的物件。在這個範例中，分析應用程式不僅能夠偵測車輛，也能夠分類車輛的類型。

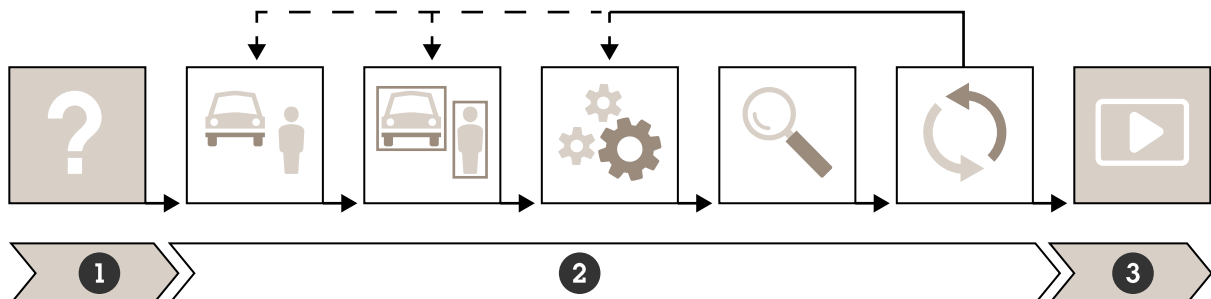
3.3 傳統機器學習和深度學習

雖然演算法類型相似，但深度學習演算法使用的學習特徵組合資料集，通常遠大於傳統機器學習演算法。這表示深度學習分析能夠更有彈性，且若接受過訓練，可以學習進行更複雜的任務。

不過對於特定監控分析，專用的最佳化傳統機器學習演算法可能就已足夠。在明確指定的範圍內，可以提供和深度學習相近的結果，同時所需的數學操作較少，因此更具成本效益且使用時消耗的電力較少。進一步來說，需要的訓練資料更少，而這可以大幅減少開發時投入的資源。

4 機器學習的階段

機器學習演算法的開發遵循一系列的步驟和迭代，大致如下圖所示，之後完成的分析應用程式才能夠建置安裝。分析應用程式的核心是一種或多種演算法，例如物件偵測器。就深度學習應用程式而言，演算法的核心是深度學習模型。



- 1 準備：定義應用程式的目的。
- 2 訓練：收集訓練資料。註解資料。訓練模型。測試模型。若品質不如預期，會在迭代改進循環中，再次進行先前的步驟。

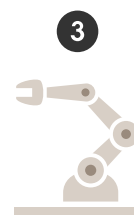
3 建置：安裝和使用完成的應用程式。

4.1 資料收集和資料註解

為了開發AI分析應用程式，您需要收集大量資料。在影像監控中，這些資料通常包含人類和車輛或其他關注物件的影像和短片。為了讓資料可由機器或電腦辨識，需要一個資料註解過程，其中會將相關物件分類和標示。資料註解主要為手動且需要大量人力的任務。準備的資料需要涵蓋，與將使用的分析應用程式情境相關，且足夠多樣化的範例。

4.2 訓練

訓練或學習是將註解資料餵給模型，且使用訓練框架迭代修改並改進模型，直到達到所需的品質為止。換句話說，會最佳化模型以解決定義的任務。可依據三種主要方法之一進行訓練。



- 1 監督式學習：模型學習進行準確預測
- 2 非監督式學習：模型學習識別叢集
- 3 強化學習：模型從錯誤中學習

4.2.1 監督式學習

監督式學習是目前機器學習中最常用的方法。可以描述為依據範例學習。訓練資料會明確註解，表示輸入資料已經和所需的輸出結果配對。

監督式學習通常需要非常大的註解資料，且訓練演算法的效能與訓練資料的品質直接相關。最重要的品質層面，是使用代表實際建置情況中，所有可能輸入資料的資料集。對於物件偵測器，開發人員必須確保以多種不同影像訓練演算法，這些影像應涵蓋不同物件實例、方向、大小、光線情況、背景和干擾事項。僅在訓練資料可代表預定的使用案例，最終分析應用程式在處理訓練階段從未見過的新資料時，才能進行準確預測。

4.2.2 非監督式學習

非監督式學習使用演算法分析未標示資料集並分群。這並非監控產業常用的訓練方法，因為模型需要大量校正和測試，而品質可能仍然無法預測。

資料集必須和分析應用程式相關，但不必明確標示或標記。不需要手動註解工作，但訓練所需的影像或影片，必須大幅增加好幾個數量級。訓練階段期間，待訓練模型會由訓練框架輔助，識別資料集中的共同特徵。這可在建置階段期間依據樣式將資料分群，同時也允許偵測不符合任何學習群組的異常。

4.2.3 強化學習

強化學習會用於例如機器人、產業自動化和商業策略規劃，但由於需要大量回饋，此方法在目前監控中的用途有限。強化學習是在特定情境中採取適當行動，以達到可能的最大獎

勵，模型做出正確選擇時，獎勵幅度會增大。演算法並未使用資料/標示配對進行訓練，而改由透過與環境互動，測試其決定同時測量獎勵，以最佳化演算法。演算法的目標是學習將協助達成最大獎勵的行動政策。

4.3 測試

模型訓練完成後，需要徹底測試。這個步驟通常包含一個自動化部分，由真實建置情境中的廣泛測試互補。

在自動化部分中，會以模型訓練期間未接觸過的新資料集，評估應用程式的效能。若這些評估結果並非預期結果，此過程會再次從頭開始：收集新訓練資料、建立或修正註解，並重新訓練模型。

達到想要的品質程度後，會開始現場測試。在這項測試中，應用程式會暴露到真實情境。數量和差異取決於應用程式的範圍。範圍越窄，需要測試的差異越少。範圍越廣，需要越多測試。

結果會再次比較並評估。接著此步驟會再次導致過程重頭開始。另一個可能的結果是定義前置條件，說明應用程式不建議或僅部分建議使用的已知情境。

4.4 建置

建置階段也稱為推論或預測階段。*推論*或*預測*是執行訓練過的機器學習模型的過程。演算法使用訓練階段期間學到的內容，產生其所需的輸出。在監控分析情境中，推論階段是由在監控系統上執行的應用程式，監控真實場景。

為了在針對音訊或影像輸入資料執行機器學習演算法時達成即時效能，通常需要特定硬體加速。

5 邊際分析

高效能影像分析過去採用伺服器進行，因為需要遠超出攝影機可提供的更多功率和冷卻。但近年來的演算法開發和邊際設備處理能力日益增加，已經能夠在邊際上執行進階AI影像分析。

邊際分析應用程式有明顯的優勢：可用極低延遲存取未壓縮的影像資料，能夠達成即時應用，同時避免將資料搬移到雲端計算的額外成本和複雜度。邊際分析也具有較低硬體及建置成本，因為監控系統中所需的伺服器資源較少。

部分應用可能適用邊際和伺服器處理的組合，在攝影機上預處理，然後在伺服器上進行後續處理。此類混合系統可透過同時處理數個攝影機串流，協助以具成本效益的方式擴充分析應用程式。

6 硬體加速

雖然通常可在數種類型的平台上執行特定分析應用程式，但使用專用的硬體加速，可在電力有限時達到更高的效能。硬體加速器可用節能方式實現分析應用程式。適當時可由伺服器和雲端運算資源補強。

- GPU (圖形處理單元)。GPU主要開發用於圖形處理應用，但也在伺服器和雲端平台上用於加速AI。雖然有時候也會用於嵌入式系統(邊際)，但從節能角度看來，GPU並非用於機器學習推論任務的最佳選擇。

- MLPU (機器學習處理單元)。MLPU可加速特定傳統機器學習演算法的推論，以極高能源效率解決電腦視覺任務。設計可同時針對有限數量的物件類型進行即時物件偵測，例如人類和車輛。
- DLPU (深度學習處理單元)。具有內建DLPU的攝影機，可用高能源效率加速通用深度學習演算法推論，以進行更精細的物件分類。

7 AI仍然在初期發展階段

大家都想比較AI解決方案的潛力，和人類能夠達成的結果。人類影像監控操作者只能短時間集中注意，但電腦能夠極快速處理大量資料而不會疲倦。不過假設AI解決方案會取代人類操作者，是天大的誤解。真正的優勢來自於實際的組合：利用AI解決方案的優點改進並提升人類操作者的效率。

一般會將機器學習或深度學習解決方案描述為，有能力自動學習或透過經驗改進。但現在的AI系統無法在建置後自動學習新技能，也無法記住發生過的特定事件。若要改進系統的效能，需要在監督式學習過程中，以更好更準確的資料重新訓練。非監督式學習通常需要大量資料才能產生叢集，因此不會用於影像監控應用程式。現在非監督式學習主要用於分析大資料集以找出異常，例如金融交易。影像監控中大部分宣稱「自我學習」的方法是依據統計資料分析，而並非實際重新訓練深度學習模型。

就監控目的而言，人類經驗仍然可以打敗許多AI分析。特別是進行非常通泛的任務，且理解前因後果非常關鍵的情況下。機器學習應用程式如果專門接受過訓練，可能成功偵測到「奔跑的人」，但不像人類能夠分析前因後果，應用程式無法理解為什麼這個人正在奔跑，是要趕公車還是要逃離附近奔跑的警察？即使將AI套用到監控分析應用的公司信誓旦旦，這些應用程式還是無法理解在影像上看到的内容，遠遠不及人類能夠提供的見解。

基於相同的理由，AI分析應用程式也可能觸發假警報或漏掉警報。這是包含大量位移的複雜環境中常見的情況。但也可能因為一個人攜帶大型物件，阻礙應用程式辨識人類特徵，因此較無法正確分類。

現在的AI分析應該用來做為輔助手段，例如先大致決定事件相關性，再警示人類操作者以決定如何反應。如此可使用AI達成擴充性，而由人類操作者評估潛在事件。

8 最佳分析效能的考量要點

為了瞭解對於AI分析應用程式的品質期望，建議仔細研究並瞭解已知的前置條件和限制，這些通常會列在應用程式的說明文件中。

每個監控安裝設置都是獨特的，且應在每個監控地點評估應用程式的效能。若品質沒有達到期望或預期的程度，強烈建議不要僅將調查著重於應用程式本身。所有調查都應該以全面方式進行，因為分析應用程式的效能取決於許多因素，若我們知道這些因素的影響，就能夠最佳化其中絕大多數因素。這些因素包括例如攝影機硬體、影像畫質、場景動態、照明等級，以及攝影機設定、位置和方向。

8.1 影像可用性

影像畫質通常取決於攝影機的高解析度和高光線靈敏度。雖然這些因素的重要性毋庸置疑，但當然還有其他因素也會影響影像或影片的實際可用性。例如，如果場景在夜間照明不足、攝影機轉向，或系統連線中斷，即使監控攝影機再昂貴、影像串流品質再好，也都毫無用武之處。

建置前應審慎考量攝影機的放置位置。為了讓影像分析達到預期表現，攝影機需要放在能夠獲得適用場景之無遮蔽清楚視野的位置。

影像可用性也可能取決於使用案例。人眼看起來良好的影像，畫質可能不足以讓影像分析應用程式發揮作用。事實上，許多經常用於增強人類觀看影像感受的影像處理方法，不建議在使用影像分析時使用。這可能包括例如套用減噪方法、寬動態範圍方法或自動曝光演算法。

現在的影像攝影機經常搭配整合式紅外線照明，令其可在完全黑暗中運作。好處是可將攝影機放在光線條件不佳的位置，並減少安裝額外照明的需求。然而如果預期監控地點下大雨或下雪，則強烈建議不要仰賴來自攝影機或來自極接近攝影機位置的光線。過多光線可能會被雨滴和雪花直接反射回到攝影機，導致分析無法進行。另一方面，使用環境光線時，即使在惡劣的氣候中，分析也較可能提供某些結果。

8.2 偵測距離

很難決定AI分析應用程式的最大偵測距離，資料表上精確的公尺或英尺數值，無法完全反映實際情況。影像畫質、場景特性、天氣狀況，和顏色及亮度等級等物件特性，對於偵測距離都有顯著的影響。例如，晴天深色背景中的明亮物件，相較於雨天的深色物件，顯然可在更遠的距離下以可見光偵測到。

偵測距離也取決於待偵測物件的速度。為了達到準確結果，影像分析應用程式需要在夠長的一段時間內「看到」物件。需要多長的時間取決於平台的處理效能(影格速率)：處理效能越低，偵測物件所需的可見時間越長。若攝影機的快門時間和物件速度沒有適當匹配，影像中出現的位移殘影也可能降低偵測準確度。

快速物件若在較接近攝影機的距離通過，可能較容易被漏掉。在遠離攝影機位置奔跑的人可清楚檢測，而以相同速度在非常靠近攝影機位置奔跑的人，可能太快進出視野而無法觸發警報。

在依據位移偵測的分析中，直接朝向或遠離攝影機移動的物件，會帶來另一項挑戰。對於緩慢移動的物件，偵測特別困難，因為相較於橫越場景的移動，只會在影像中造成非常小的改變。

較高解析度的攝影機通常不會提供更長的偵測距離。執行機器學習演算法所需的處理能力，與輸入資料的大小成正比。這表示分析4K攝影機完整解析度所需的處理能力，比1080p攝影機至少高出四倍。因為攝影機的處理能力限制，通常會針對比攝影機或串流更低的解析度執行AI應用程式。

8.3 警報與錄影設定

因為套用不同等級的篩選條件，物件分析產生的假警報非常少。但僅在滿足其所有前置條件時，物件分析才能達成應有的表現。在其他情況下，可能反而會漏掉重要事件。

若無法絕對確定隨時都可滿足所有條件，則建議採取保守方法，設定系統不要將特定物件分類做為唯一的警報觸發條件。這會造成較多假警報，但也會降低漏掉重要事件的風險。警報或觸發條件直接進入警報監控中心時，每項假警報的成本都會變得很高。顯然需要可靠的物件分類過濾掉不想要的警報。但錄影解決方案仍然可以且應設定為不會只仰賴物件分類。若漏掉真正的警報，此設定可讓您從錄影評估漏掉警報的原因，然後改進整體安裝和設定。

若事件搜尋期間在伺服器上進行物件分類，建議設定系統持續錄影，且完全不要過濾初始錄影。持續錄影會耗費大量儲存空間，不過這點在某種程度上，可由Zipstream等現代壓縮演算法補償。

8.4 維護

監控安裝應該定期維護。建議進行實體檢查，而不要僅透過VMS介面檢視影像，以發現並去除可能干擾或阻礙視野的任何事物。這對標準的單純錄影安裝也很重要，不過使用分析時更為重要。

在基本影像位移偵測的情境中，典型的障礙物，例如隨風飄揚的蜘蛛網，可能會增加警報數量，導致消耗比必要更多的儲存空間。使用物件分析時，蜘蛛網會在偵測區內建立一個不包含區域。絲線會阻擋物件並大幅降低偵測和分類的機率。



蜘蛛網可能會干擾監控攝影機的視野。

攝影機前方玻璃上的灰塵或氣泡，在白天不至於造成問題。但在弱光條件下，從側邊射入髒氣泡的光線，例如來自汽車頭燈的光，可能會造成意外反射而降低偵測準確度。

場景相關維護和攝影機維護同樣重要。在攝影機的使用期間，監控的場景中可能會發生許多改變。簡單的前後影像比較就可顯示出潛在問題。攝影機建置時的場景是什麼樣子，而現在是什麼樣子？是否需要調整偵測區域？是否應調整攝影機的視野，或者是否應該將攝影機移到不同位置？

9 隱私與個人誠信

安防與監控工作需要在個人隱私權及個人誠信之間取得平衡，且企圖透過防止犯罪或進行鑑識調查以提升安全性。在特定安裝與使用案例中，需要審慎倫理考量，且理解及應用當地法律。也會要求解決方案能夠例如確保網路安全及預防意外存取影像資料。在此同時，邊際分析和針對統計目的產生的詮釋資料，若僅傳送匿名資料供日後處理，可能增進隱私保護。

隨著自動化分析在監控系統中的應用日益增加，也必須將一些新層面納入考量。因為分析應用程式具有錯誤偵測風險，決策過程也必須納入有經驗的操作者或使用者的經驗。這通常稱為保留「人機迴圈」。還有，也必須瞭解人類決定可能會受到警報產生和呈現的方式影響。若沒有經過適當訓練並知道分析解決方案的功能，可能會推導出錯誤的結論。

其他疑慮可能來自深度學習演算法的開發方式，而對於某些使用案例，應用此技術時需要採用謹慎的做法。這些演算法的品質在本質上與用於訓練演算法的資料集相關，也就是影片和影像。測試已顯示若沒有精心挑選資料，某些AI系統可能會在偵測時出現種族和性別偏見。此問題已促成開放討論，並引發法律限制和活動，確保系統開發期間解決此類問題。

由於AI應用至監控的情況日益普遍，透過審慎討論何時何地應用技術，以便在運行效率的優勢與新潛在使用案例之間取得平衡，非常重要。

10 附錄

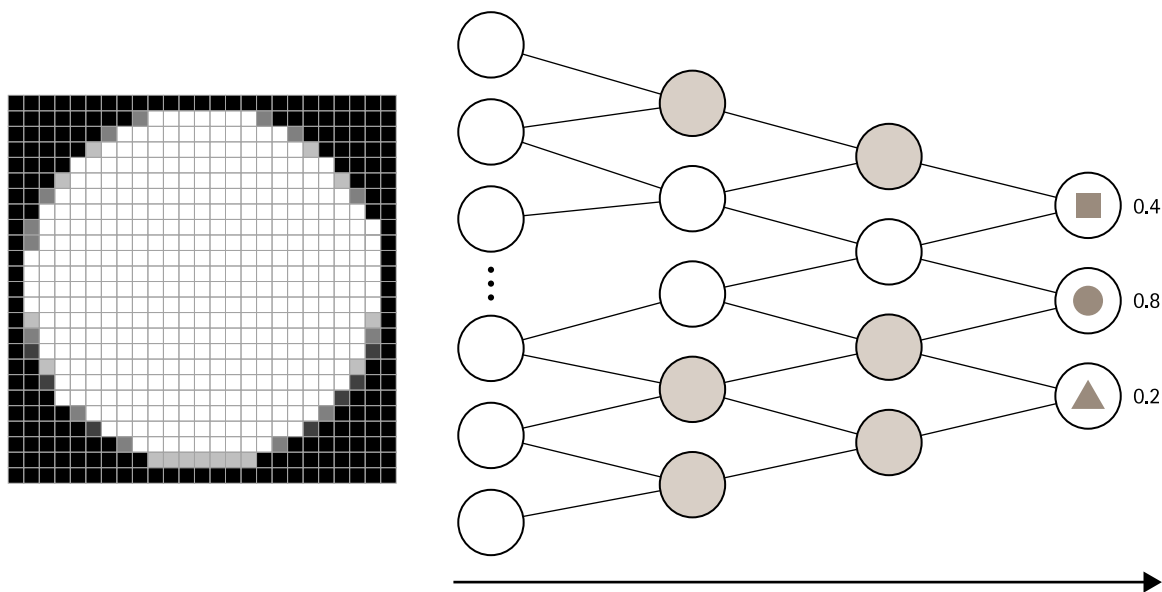
此附錄提供有關構成深度學習基礎之類神經網路的背景資訊。

10.1 神經網路

神經網路是一大類演算法，可透過與人類大腦運作方式類似的過程，辨識資料集內的關係。神經網路包含節點或神經元互連的多層階層結構，資訊會通過這些連結，從輸入層通過網路到輸出層。

神經網路發揮作用的假設為，輸入資料範例可縮減為有限的特徵集，並可良好表示輸入資料。這些特徵接著可結合，且將協助分類輸入資料，例如描述影像的內容。

下方例圖顯示的範例，使用神經網路識別輸入資料屬於哪個類別。影像中的每個畫素都由一個輸入節點表示。所有輸入節點連接到第一層內的節點。這些會產生輸出值，然後做為輸入值傳到第二層，以此類推。在每一層中，也會有加權函數、偏差值和啟動函數參與這個過程。



輸入影像(左)和神經網路(右)範例。達到輸出層時，網路也會為每個可能類別(方塊、圓圈或三角形)推論出機率。具有最高機率值的類別，會是輸入影像的最可能形狀。

此過程稱為**正向傳播**。若正向傳播結果不符，會透過**反向傳播**略微修改網路參數。在這個迭代訓練過程中，會逐漸改進網路的效能。

建置後，神經網路通常不會具有先前正向傳遞的記憶。這表示不會逐漸改進，且只能偵測訓練過的物件類型或解決訓練過的任務類型。

10.2 卷積神經網路(CNN)

卷積神經網路(CNN)是類神經網路的一種子型，已證實特別適合電腦視覺任務，且為快速進展之深度學習的核心。就電腦視覺而言，會訓練網路自動找尋獨特的影像特徵，類似於邊緣、角落和顏色差異，以有效識別影像中的物件形狀。

達成此目的之主要操作是稱為 卷積的數學操作。這是一項非常有效率的操作，因為每個個別節點的輸出僅仰賴周圍由前一層產生的有限輸入資料，而非使用完整輸入資料量。換句話說：在CNN中，每個節點並未連接到前一層的每個節點，而僅連接到一個小型子集。卷積可由縮減資料大小的其他操作互補，同時保留最有用的資訊。如同標準類神經網路，在網路中越深層的資料會變得越來越抽象。

在訓練階段期間，CNN會學習套用各層的最佳方式。也就是卷積應如何結合來自前一層的特徵，讓網路輸出盡可能符合訓練資料的註解。在推論期間，訓練好的卷積神經網路進而套用訓練導致的卷積層。

關於 Axis Communications

Axis 透過建立網路解決方案，提供改善安全的洞察力和執行業務的新方式，以打造一個更智慧、更安全的世界。Axis 做為網路影像的產業領導者，提供用於影像監控和分析、存取控制、對講和音訊系統的產品和服務。Axis 在 50 多個國家/地區擁有超過 3,800 名敬業員工，並與全球合作夥伴合作提供客戶解決方案。Axis 成立於 1984 年，總部位於瑞典隆德。

如需 Axis 詳細資訊，請瀏覽本公司網站 axis.com。