# **摘要**

隨著數位技術的持續進展，手寫圖形與文字的向量化在人工智慧、圖形辨識等領域中扮演關鍵角色。然而，現行方法常受限於解析度與像素誤差，導致轉換精度不足。本研究提出一套基於貝茲曲線的向量化系統，藉由自動曲線分段演算法，有效提升手寫圖形的轉換效率與資料精簡度。本系統採用網頁作為互動介面，使用者可直接繪製手寫圖形，並由伺服器端執行自行開發之線段切割演算法「線段向量與曲率特徵處理（Segment Vector and Curvature Feature Processing, SVCFP）」，自動萃取節點後，透過最小平方法進行最佳貝茲曲線擬合。實驗結果顯示，該系統不僅有效降低擬合誤差與儲存成本，亦顯著提升轉換準確度，適用於數位手寫輸入、字型設計與手寫辨識等應用場景，為人工智慧與圖形處理技術提供具前瞻性的解決方案。

**壹、前言**

## 一、 **研究動機**

在數位繪圖與手寫輸入的領域中，將手繪圖形準確地轉換為數位格式一直是一大挑戰。當我們使用觸控螢幕或數位畫板進行手繪時，筆劃是由離散的像素點組成，這些像素點的排列可能會受到解析度或設備規格的影響，導致筆劃邊緣不夠平滑，或在放大縮小時出現鋸齒狀失真，並且在旋轉時會極大程度的破壞原有圖像。

在現有的方法中，使用者普遍使用鋼筆工具，普遍出現以下幾個問題:

1. 貝茲曲線節點過多，造成資料冗餘與計算負擔

* 現有軟體（如 Adobe Illustrator、Inkscape）通常透過大量貝茲節點來提高擬合精度，然而這種方法導致資料量急劇增加，使向量圖儲存與運算效率降低。

1. 無法適應手繪筆劃的細微變化，影響轉換效果

* 傳統鋼筆工具仰賴使用者手動調整控制點，難以精確擬合手繪筆劃中的細節，尤其在曲率變化劇烈的區域（例如手寫字的筆鋒或書法筆觸）表現更為不足。此外，當圖像結構過於複雜時，向量化處理所需時間顯著增加，降低整體使用效率。

1. 缺乏自動化能力，使用門檻較高

* 目前大多數向量化工具需要使用者手動調整貝茲曲線的控制點，這不僅耗時，還需要較高的技術門檻，對於非專業使用者來說不夠友善。

這些問題讓筆者思考，是否能有一種方法，不僅能夠將手繪圖形轉換為數位格式，還能保持手繪的流暢性與細節，並且異於市面上方法。經過一番搜尋，筆者接觸到了貝茲曲線（Bézier curve），它是一種數學模型，可以用來表示曲線，其中包含許多表示方法，這裡選擇最常見的三次貝茲曲線介紹，其公式如公式1-1所示，其中P0及P3代表貝茲節點、P1及P2則是貝茲控制點，由四個點的座標控制曲線的形狀。

|  |  |
| --- | --- |
| B(*t*)=P0(1-*t*)­­3+ P1*t*2(1-*t*)­­2+P2*t*2(1-*t*)­­1+P3*t*­­3 | (公式1-1) |

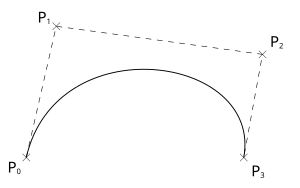
產生結果如(圖1-2):

圖1-2.貝茲曲線表達方式(由作者親自繪製)

貝茲曲線的特性在於能夠通過少量的控制點來描述光滑的曲線，並且能在不同解析度下保持穩定，貝茲曲線在許多領域已有廣泛應用，例如字型設計、數位藝術、動畫製作等，並且因為其向量化特性，可以有效減少資料量並提高畫面呈現的效果，同時不受到解析度影響。然而，儘管貝茲曲線在理論上具有很大優勢，但直接使用貝茲曲線來擬合手寫筆劃有幾項技術難點:

* 如何自動選擇最佳的控制點？
  + 如果控制點選擇不當，擬合出的曲線可能會與使用者手繪圖形產生偏差
  + 傳統的擬合方式可能會忽略使用者手繪的細微變化，導致手繪特徵失真。
* 如何在降低資料量的同時保持手繪的完整性？
  + 向量化的過程需要在減少計算負擔與保持手繪特徵之間取得平衡。
  + 特徵保留愈多，雖可提升圖像還原度，卻也將導致檔案容量顯著增加，影響儲存與傳輸效率。

上述挑戰限制了貝茲曲線在手繪圖形數位轉換中的應用潛力。因此，本研究致力於克服這些技術障礙，開發一套具備高效率、精確度與自動化能力的向量化方法，使使用者的手繪輸入能更準確地轉換為電腦可處理的貝茲曲線，進而提升數位手繪技術的實用性與應用價值。

**二、研究目的**

本研究旨在開發一套手繪圖形向量化方法，運用貝茲曲線精確描述手繪筆劃之形狀輪廓。系統核心為自創切割演算法「線段向量與曲率特徵處理（Segment Vector and Curvature Feature Processing, SVCFP）」，可對使用者於網頁介面繪製之筆劃序列進行特徵提取與曲線分段，進而達成高擬合度的貝茲曲線生成。

此外，為提升擬合精度與相似度，本研究亦設計圖形比對損失函數，以輔助貝茲控制點的最佳化，使擬合曲線更貼近原始筆跡的細節特徵。綜合而言，研究目標包括：

1. 改善傳統貝茲曲線擬合節點過多問題
   * 本研究的方法透過貝茲節點的自動優化，能夠以更少的節點達到相同甚至更高的擬合精度，從而減少計算負擔與儲存成本，並可調整閾值，選擇使用者所需要的精確度。
2. 無法適應手繪筆劃的細微變化，影響轉換效果
   * 本研究方法能夠根據筆劃長度與曲率動態，並使用SVCFP，自動調整貝茲節點的數量與位置，使得轉換後的向量曲線更加貼合原始筆跡，保留細微特徵。
3. 缺乏自動化能力，使用門檻較高
   * 本研究透過演算法自動化選擇貝茲節點與擬合曲線，使使用者無需進行手動調整，即可獲得高品質的向量化結果，提升使用體驗與轉換效率。

綜合以上研究目標，本研究的技術可望在提高手繪圖形轉數位格式的精度與效率的同時，思考其在各種數位內容創作領域中的應用潛力，並促進人工智慧技術在手寫辨識和數位繪圖中的發展。

## **三、文獻回顧**

本研究將貝茲曲線擬合分為幾個關鍵字: 貝茲曲線、手繪線條向量化、演算法，並依序找了幾個論文:

1. 貝茲曲線
   1. 作者在其論文《Defining a curve as a Bézier curve》[3]中提出了一種新的方法，能夠確定任何一條曲線是否可以被定義為貝茲曲線。文章介紹了創建矩陣的概念，並證明了該矩陣的反向矩陣可用於精確求解貝茲曲線的控制點，其優點在於提供了一種精確而非近似的方法，適用於多領域如計算機輔助幾何設計，但缺點是該方法依賴於矩陣反推，可能對於計算資源的需求較高，並且對於高階多項式的處理較為複雜，對於已被破壞之圖像難以反推回數學函式，難以對其進行利用。
   2. 作者在其論文《Bézier Curve Fitting》[4]中，研究了如何使用總體最小二乘法（Total Least Squares, TLS）來擬合貝茲曲線，使其能夠同時最小化水平方向與垂直方向的殘差。他詳細探討了伯恩施坦多項式與貝茲曲線的數學基礎，並提出了一種基於高斯-牛頓法(Gauss-Newton Method）的優化演算法，以求得最佳的控制點與節點。該研究的優點在於提供了一種更準確的擬合方法，適用於數據分佈不均的情境，並具有較高的幾何穩定性。然而，該方法在計算上較為複雜，尤其是在處理高維數據或實時應用時，計算成本可能較高。此外，論文主要聚焦於數學方法的發展，對於具體的工程應用討論較少，故較難依其理論進行實作應用。
   3. 作者在《Curve Fitting Using Generalized Fractional Bézier Curve》[5]一文中，探討了如何利用廣義分數貝茲曲線進行曲線擬合。該研究指出，傳統貝茲曲線在靈活性和可調整性方面存在侷限，因為其形狀無法在不改變控制點的情況下進行調整，而為了解決這一問題，作者引入了具有形狀參數和分數參數的廣義分數貝茲曲線，這些參數允許在不改變控制點的情況下調整曲線形狀，從而提高了曲線擬合過程的靈活性和效率，此外，該方法還利用了分數連續性，允許在曲線的公共點或連接處進行位置調整，進一步增強了曲線擬合的靈活性，然而，該研究主要集中在數學模型的構建與理論分析，對於實際應用中的性能和效果尚需進一步驗證。
2. 手繪線條向量化
   1. 在《基於擴散曲線之點陣圖自動向量化》[6]一文中，作者提出了一種利用擴散曲線技術，將點陣圖自動轉換為向量圖的方法。該方法從點陣圖中提取輪廓、顏色和模糊象素點，並將其轉換為擴散曲線所需的幾何元素，使生成的向量圖能夠接近原始圖像，且易於編輯和動畫製作。然而，該方法可能在處理高細節圖像時，面臨精度下降的挑戰，需要進一步優化以提高對此類圖像的適應性。
   2. 在《Using Machine Learning to Improve Automatic Vectorization》[7]一文中，作者探討了如何利用機器學習技術提升自動向量化的效果。該研究提出了一種新穎的方法，將機器學習應用於向量化過程中，以提高生成向量圖的品質和效率。然而，該方法可能需要大量的訓練數據和計算資源，對於資源有限的環境存在挑戰。
   3. 在《Systematic Comparison of Vectorization Methods in Classification Context》[8]一文中，作者對多種向量化方法在分類任務中的性能進行了系統性比較。研究評估了不同向量化技術在處理各類資料集時的準確性和效率，旨在為選擇適當的向量化方法提供實證依據。結果顯示，某些方法在特定情境下表現出色，但在其他情境中可能效果不佳，這強調了根據具體應用場景選擇向量化技術的重要性。然而，該研究可能受限於所選資料集的多樣性，未來研究可考慮引入更多元的資料集以驗證結果的普遍性。
3. 演算法
   1. 在《Learning with Genetic Algorithms: An Overview》[9]一文中，作者探討了遺傳演算法（Genetic Algorithms, GAS）[1]在機器學習中的應用。該研究強調，GAS[1]作為一種自適應搜尋技術，能有效解決全域函數最佳化和 NP-hard 問題，且不需要高度領域專業知識。然而，作者也指出，GAS[1]的效能可能受限於問題的複雜性和搜尋空間的大小，對於特定應用可能需要結合其他方法以提升效率。
   2. 在《遺傳演算法應用於三維地下水數值模型之參數優選》[10]一文中，作者探討了如何利用遺傳演算法（Genetic Algorithms, GAS）[1]對三維地下水數值模型的參數進行優化。該研究旨在提高模型的準確性和效率，透過 GAS[1]的全域搜尋能力，尋找最佳參數組合，以更準確地模擬地下水流動。結果顯示，與傳統方法相比，GAS[1]能有效避免局部最優解，並在複雜的地下水系統中提供更可靠的參數估計。然而，該方法需要較高的計算資源，且對於大型或更複雜的模型，計算時間將會顯著增加，需要對此進一步優化。
   3. Ramer (1972) 與 Douglas & Peucker (1973)[11][12] 提出的多邊形逼近演算法，皆致力於簡化曲線表示，減少資料點數量以提升處理效率。其優點在於能有效保留原始曲線的形狀特徵，並廣泛應用於圖像處理與地理資訊系統。然而，這些方法對於曲線細節的保留仍有侷限，尤其在高曲率區域可能產生失真。此外，演算法參數需依資料特性調整，對自動化處理構成挑戰。

# **貳、研究設備與器材**

1. 程式環境
   1. **Anaconda**：專為資料科學與機器學習設計的 Python 發行版，可使用多種常用函式庫如 NumPy、SciPy、TensorFlow 與 OpenCV，並搭載 Conda 套件管理系統，方便快速切換套件版本與管理開發環境，提升開發效率與穩定性。
   2. **Inkscape**：一款向量圖形編輯工具，能將圖形匯出為 SVG 格式。在本研究中，主要用於與本程式進行擬合效果之比較。
2. 前端開發技術

為提供直觀且易於操作的手繪畫布，本研究網頁前端使用以下技術：

* 1. HTML5：負責建立網頁結構與畫布元素。
  2. CSS3：進行視覺設計與畫面排版，提升使用體驗。
  3. JavaScript：處理使用者手繪輸入，並將筆劃資料傳送至伺服器端。
  4. Canvas API：在瀏覽器中實現即時手寫繪圖與動態筆劃渲染。

三、伺服器端技術

伺服器端負責接收前端筆劃資料並進行貝茲曲線擬合計算，主要採用：

* 1. Python 3.9：作為後端主要開發語言，進行資料運算處理。
  2. Flask：輕量級 Web 框架，負責建立 API 供前端傳送與接收資料。
  3. NumPy：進行筆劃資料的數值運算，加速處理流程。
  4. OpenCV：負責影像處理與曲線擬合前的預處理步驟。
  5. TensorFlow/Keras：建構與訓練 CNN 模型，用於增強筆劃特徵提取能力。
  6. SciPy：用於貝茲曲線的數學優化與數值逼近。
  7. Matplotlib：視覺化擬合結果，協助分析模型表現。

四、優化與演算法技術

為提升貝茲曲線擬合的準確性與效率，採用以下演算法輔助開發：

* 1. **遺傳演算法（**GAS**）**[1]：最佳化貝茲曲線控制點，使擬合結果更精準。
  2. **Dijkstra 演算法[14]**：用於尋找筆劃與曲線之最佳對應路徑，提高擬合準確性。
  3. **Hausdorff 距離**[2]：作為損失函數，評估原始筆劃與曲線之間的誤差。
  4. **Douglas–Peucker（RDP）演算法**：用於抽取線段特徵，協助節點判斷與曲線簡化。

五、卷積神經網路技術（CNN）

本研究亦嘗試使用 CNN 模型進行筆劃特徵擷取，並探討其在曲線擬合任務中的應用潛力與限制，為未來進一步整合深度學習技術提供基礎。

# **參、研究過程與方法**

(圖3-1)為本研究系統架構流程圖，主要分為五個步驟，分別為:

1. 使用者輸入手繪曲線/圖片
2. 線段點位預測
3. 切割線段
4. 貝茲擬合
5. 使用者取得貝茲曲線

以下將逐步驟介紹詳細的研究過程及方法。

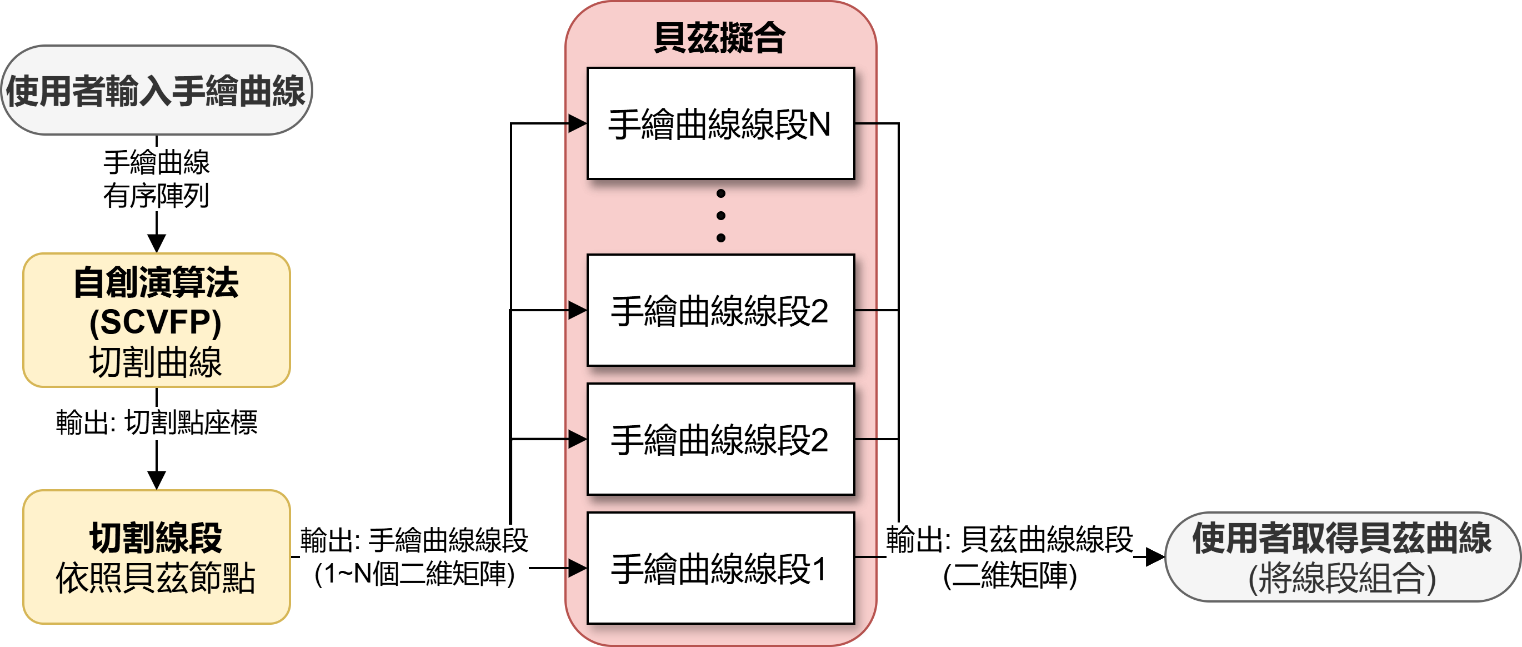


圖3-1. 系統架構流程圖(由作者親自繪製)

## **使用者輸入手繪曲線**

本研究採用Python提供的Flask套件作為網頁框架，搭建後端運算伺服器，並透過HTML建立網站架構供使用者操作，在使用者端加上Canva套件追蹤滑鼠動作，取得使用者的滑鼠位置及當前狀態，達成網頁畫布的效果，儲存成路徑後傳入伺服器端進行運算，網頁版面配置如圖3-2所示。



圖3-2. 本程式網頁設計(由作者親自繪製)

1. 線段點位預測

為了有效地對線段進行切割和特徵提取，本研究嘗試了兩種不同的方法：(1) 基於卷積神經網路（CNN）的線段切割、(2) 線段向量和曲率特徵處理(Segment Vector and Curvature Feature Processing,SVCFP)進行線段切割。以下將逐項詳細介紹這兩種方法：

* 1. CNN線段切割:

參考VGG-16[13]模型架構，自行進行修改及精簡後得到的新模型VGG-16(ours)，此模型的目的是預測出線段切割點，將手繪線條切割成數個貝茲三次曲線，模型架構如圖3-3所示，將使用者手繪曲線輸入模型預測後，將得到的輸出標於曲線上如圖3-4所示，會產生4個切割點供後續使用，但遇到更複雜線條時，輸出結果將會限制於四個點，難以完整表達更複雜的曲線。

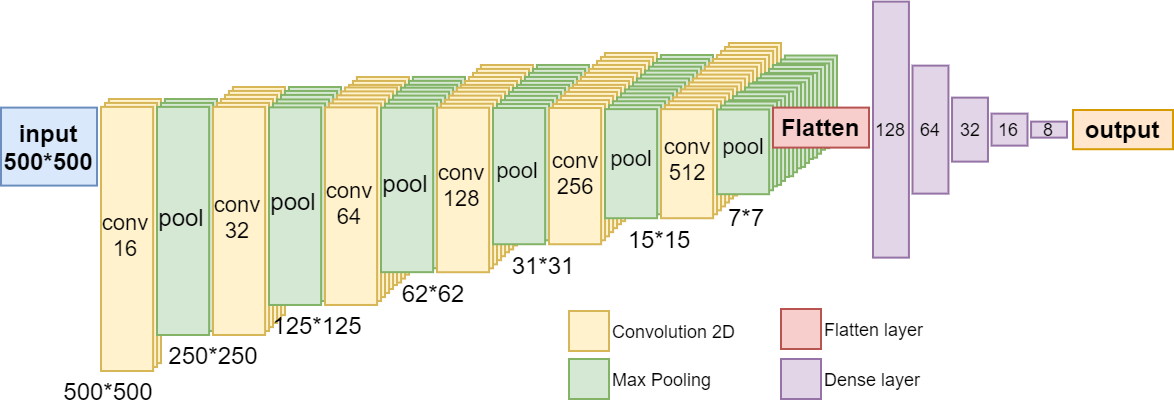


圖3-3. VGG-16(ours)模型架構圖(由作者親自繪製)

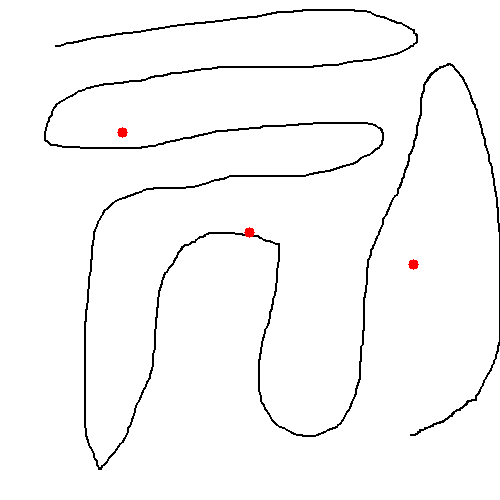
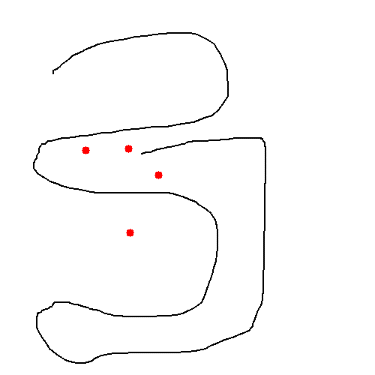
 

圖3-4. VGG-16(ours)模型輸出預測切割點(紅點) (由作者親自繪製)

* 1. 線段向量和曲率特徵處理(Segment Vector and Curvature Feature Processing-SVCFP)

為了解決CNN點位不足等諸多限制，我們發明了一套嶄新的演算法，透過幾何特徵分析，能夠有效提取曲線的關鍵特徵點，並進行有序線段的精確判斷，以下是此演算法的架構流程:

* + 1. Ramer-Douglas-Peucker（RDP）演算法簡化路徑

為了有效加速運算並精確提取曲線特徵，我們在演算法中整合了RDP演算法。RDP演算法透過設定容忍誤差值，遞迴地檢查線段並移除不重要的點，從而在保持曲線形狀的同時，顯著減少路徑點數，並保留像轉折點和極值點，如 圖3-5所示，為後續的有序線段判斷提供了重要的資訊。

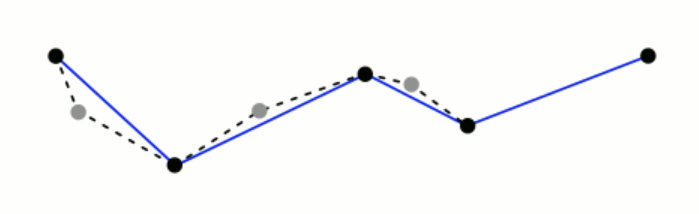


圖3-5.RDP. 線條示意圖(由作者親自繪製)

* + 1. 向量判斷

針對接收到的 RDP 陣列，我們採用一種基於多尺度向量預測的簡化點分析方法，此方法的原理是透過分析簡化點在不同尺度下的向量變化，並結合統計分析和閾值判斷，來更準確地識別出簡化點的顯著性。

首先，對每個簡化點，我們在其鄰域內（初始為較小的範圍）進行向量預測，並記錄該範圍內的最大值。向量預測的具體做法是將簡化點左右兩側的線段視為向量，並計算這些向量的平均值，以評估該點附近的曲線走向和變化程度。接著，我們逐步擴大鄰域範圍，重複進行向量預測，並持續更新最大值。透過迭代式擴展鄰域，我們可以捕捉不同尺度下的向量變化，從而更全面地了解簡化點的特徵。

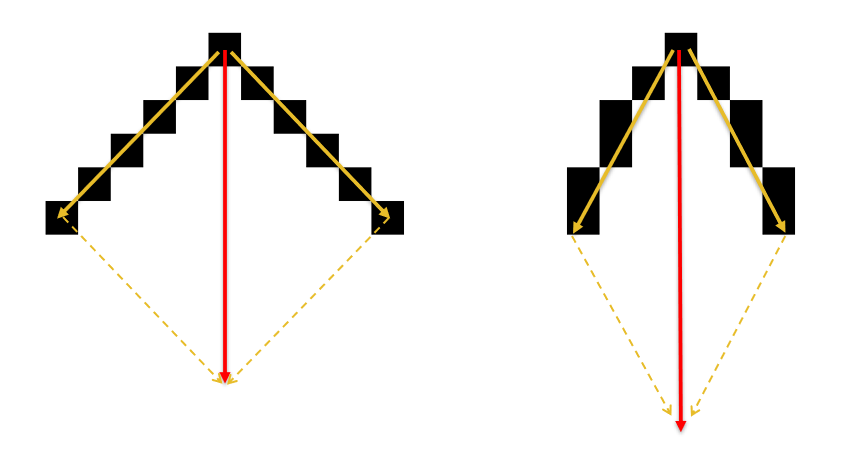
這種由小到大擴展鄰域的策略，有助於避免特殊曲線（例如 ω 形曲線）對分析結果的干擾。在 ω 形曲線中，中間的突出點容易受到左右兩側下降線段的影響，導致向量計算結果偏小，如圖3-6所示。透過記錄每個尺度下的最大值，我們能夠更準確地識別出簡化點的真實突出程度。

圖3-6. 向量判斷原理(由作者親自繪製)

完成所有尺度的向量計算後，我們對每個簡化點所記錄的最大值進行統計分析，計算其平均值和標準差，接著，我們根據這些統計量與結合預先定義的權重，為每個簡化點進行評分，以量化其顯著程度，透過這種多尺度向量預測和自適應閾值判斷相結合的方法，我們能夠更有效地分析 RDP 陣列中的簡化點，並提取出具有重要意義的特徵。

* + 1. 角度變化

針對接收到的 RDP 陣列，本研究對每個點進行角度變化分析，以識別曲線中的顯著轉折。具體而言，我們首先計算每個點與其前後相鄰點之間的單位向量，然後計算這兩個單位向量的外積(圖3-7)，並將其轉換為角度，由此得到每個點的角度變化量。

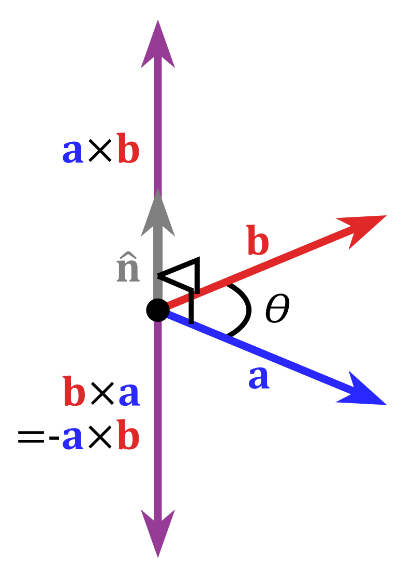


圖3-7. 外積示意圖(由作者親自繪製)

若某個點的角度變化量超過預先設定的閾值，則將該點標記為轉折點，並增加其在評分計算中的權重，以強化該點在曲線中的影響力。然而，當偵測到曲線呈現直線狀態時，我們則會減少該點的評分權重，以降低其對整體評分的影響。透過這種動態調整權重的方式，我們能夠更有效地識別曲線中的關鍵特徵，並提高後續分析的準確性。

在完成上述角度變化分析和權重調整後，對每個點的評分進行閾值判斷。具體而言，我們提取評分超過預先設定閾值的點位，並將其標記為特徵點進行下個步驟的計算。

## **切割線段**

傳統上，當使用卷積神經網路（CNN）處理圖像資料時，由於圖像資料並非有序陣列，因此在進行線條運算時，通常採用**Dijkstra 演算法[14]**來尋找最短路徑，然而Dijkstra 演算法**[14]**的運算量較大，會增加整體運算時間。

為了解決這個問題，本研究將上步驟所得節點替有序陣列切個成多線段。並將每個線段的端點轉換為指標。透過這些指標，我們可以直接存取曲線中任意兩點之間的所有路徑點，包括起點、終點以及路徑點。

相較於 **Dijkstra 演算法[14]**，我們的SVCFP具有以下優勢：

* 運算量更小： 由於直接透過指標存取路徑點，避免了**Dijkstra 演算法[14]**中複雜的搜尋過程，因此運算量顯著降低。
* 運算時間更短： 運算量的減少直接帶來運算時間的縮短，提高了整體運算效率。
* 更適合處理有序曲線： 本演算法專為處理有序曲線而設計，能夠更有效地提取曲線特徵。透過使用我們的SVCFP，我們能夠在 CNN 運算中更高效地進行線條運算，從而提高整體運算效能。

## **貝茲擬合**

在取得各曲線線段後，進入了整個流程中最關鍵的步驟——尋找最適合的貝茲曲線控制點進行擬合，為了達成準確的擬合結果，有兩大部分需要精心設計，其一是為擬合結果評分的損失函數，其二則是進行擬合的演算法。

### 損失函數(loss function)，優秀的損失函數能夠協助擬合程式更快速精準的得到理想的結果，在本研究中嘗試了數種損失函數，包含:(1)圖形比對法、(2)輪廓封閉空間計算以及(3)hausdorff演算法[2]，接下來逐項進行介紹。

#### 圖形比對法:

此方法利用 Python 的 OpenCV 套件提供的陣列差異比對技術來進行計算，具體作法是將手繪曲線所切割出的線段 A 與擬合後的線段 B 進行重疊，並計算兩張圖片中不重疊區域的像素數量，將其視為損失值，再除以整張圖的總像素數，如公式 3-8所示。然而，此函數存在一個顯著問題：即使僅移動一個像素，也可能導致極大的誤差，使得該方法無法有效擬合出最相似的貝茲曲線，因此不適合作為損失函數使用。

公式3-8

此外，由於分母固定，該計算方式會使準確度表現過高，進而降低鑑別度，使得難以獲得最佳的擬合結果

#### 輪廓封閉空間計算:

此方法同樣使用Python之Opencv套件提供之方法，使用contour輪廓計算方法為主體，將手繪曲線切割出的線段A及擬合出的線段B進行重疊，並進行輪廓計算，得到的輪廓面積經標準化過後代表兩線段AB的損失，在這裡使用輪廓面積除以整張圖的總像素作為損失，如公式3-9所示，此方法計算效果準確，但當兩條線完全沒交集時會導致無封閉空間，輪廓面積過小而視為異常準確而輸出，且可能會出現面積極小，但線條極長的情況(如圖3-10所示)，故不適合使用。

公式3-9

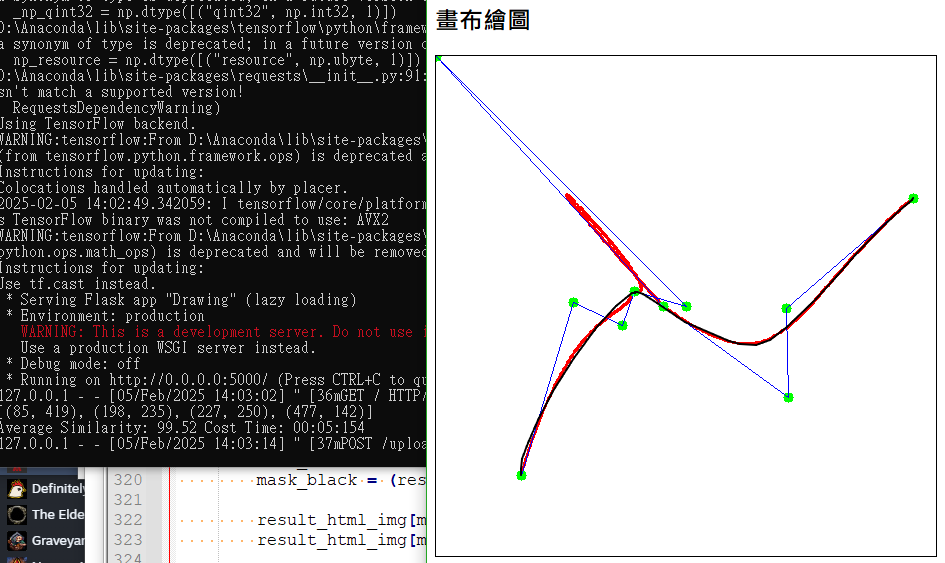


圖3-10. 空間計算之謬誤(由作者親自繪製)

#### Hausdorff距離（Hausdorff Distance）[2]

Hausdorff距離[2]是一種用來衡量兩個點集之間的相似度的度量方式，而本研究將其用來比對手繪曲線切割出的線段A及擬合出的線段B的座標相似度，計算兩曲線最長距離並標準化作為損失函數，如公式3-11所示:

公式3-11

經過多次的測試與嘗試，最終發現Hausdorff距離[2]做為損失函數同時兼顧了速度與準確度，故後續的實驗都以Hausdorff距離[2]做為主要測試對象。

### 擬合演算法，進行擬合的演算法需要考慮到計算時間、精準度，且要能避免區域最佳解導致無法得出全域最佳解的狀況，此部份本研究亦測試了數種演算法來進行擬合，其中包含:(1)暴力演算法、(2)九宮格法以及(3)遺傳演算法(GAS) [1]，以下分別介紹優缺點。

#### 暴力演算法（Brute Force Algorithm）

即對所有可能的貝茲曲線控制點組合進行窮舉，並將每條生成的貝茲曲線與原圖進行比較，但經過幾次嘗試，本研究發現暴力演算法的計算量非常巨大，由於本次實驗設定的畫布為500×500像素，每個三次貝茲曲線需要2個控制點，以一次輸入有5段貝茲曲線需要計算的前提下，並且這些控制點可能在畫布的任何一個位置中出現，得出計算量公式如公式3-12。

= 公式3-12

此公式顯示，可能的計算最大次數會非常多，會導致花費時間過長，無法實際應用於使用者端。

#### **自創擬合演算法：九宮格遞迴縮小範圍法**

為了解決暴力法的計算量問題，本研究嘗試設計了一種**自創擬合演算法**，通過遞迴縮小控制點的搜尋範圍來大幅降低運算量。

首先將整個畫布劃分為九個區塊（九宮格），初始的控制點位於每個區塊的中心點，進行損失計算，接著對表現最好的區塊再一次進行細分，生成新的九宮格，並將範圍縮小到該區域內的像素點。當範圍縮小到一定程度時，將候選點設為最終的貝茲曲線控制點，並使用這些點生成最終的貝茲曲線。

遞迴縮小範圍的方式大幅減少了候選控制點的數量，相較於暴力法的O(N4)，自創演算法的計算量接近於O(k\*logN)，其中 k為遞迴深度， N為畫布像素點數量。但在經過我們數百次的實驗後，發現它存在會陷入局部最優解的情況，導致其無法正確算出控制點，故最終放棄此演算法。

#### **遺傳演算法[2](Genetic Algorithm,GAS)** [1]

遺傳演算法**[2]**（Genetic Algorithm, GAS）[1]是一種模仿自然演化過程的最佳化方法，透過選擇（Selection）、交配（Crossover）、突變（Mutation）等機制，從隨機種群中逐步篩選出最適解，以尋找問題的最優解。本研究將遺傳演算法**[2]**應用於貝茲曲線的控制點優化，盡可能的使曲線的擬合接近手繪曲線。

在傳統的貝茲曲線擬合方法中，常見的數學模型可能受限於局部最佳解，導致擬合效果不理想，而遺傳演算法**[2]**能夠透過隨機變異與選擇機制，讓控制點在搜尋空間中進行全局搜尋，並逐步逼近最優解。本研究具體的演化流程(圖3-11)如下：

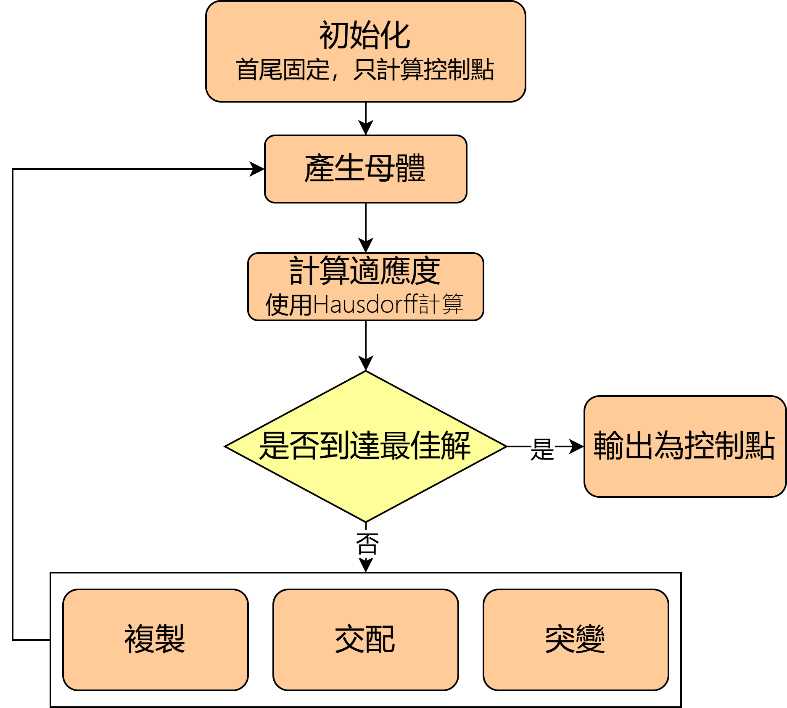


圖3-13. GAS演化流程(由作者親自繪製)

* + - 1. 初始化種群（Initialization）

首先，隨機生成一組貝茲曲線控制點作為初始種群，這些控制點的數量與位置各不相同，每組控制點的配置將影響曲線擬合的準確度。並為了加快收斂，本研究固定了起點（P₁）與終點（P₄），僅優化中間控制點（P₂,、P₃），使搜尋範圍更聚焦，避免非必要的自由度影響計算效能。這些初始個體會隨機分佈在一定範圍內，並以Hausdorff 距離[2]作為損失函數，評估其與目標曲線的相似度。

(2) 選擇（Selection）

採用輪盤選擇（Roulette Wheel Selection）或錦標賽選擇（Tournament Selection）機制，而本次使用的是Hausdorff距離[2]作為損失函數，選擇損失較低的個體進入下一代。

* + - 1. 交配（Crossover）

在選擇出的個體之間進行交配，以交換部分基因資訊，產生新的後代。本研究採用單點交叉（Single-point Crossover），具體方法如下：

隨機選擇交配點，將控制點 P₂ 和 P₃ 進行部分交換。

確保交換後的控制點仍符合貝茲曲線的合理分佈，避免不連續或極端值。

新個體繼承雙親的部分特徵，確保下一代在保留優勢基因的同時，擁有更多變異機會。

這種交叉機制可以有效擴展搜尋空間，提高基因多樣性，避免過早收斂到局部最優解。

(4) 突變（Mutation）

以一定機率對控制點的位置進行隨機擾動，確保演化過程不會陷入局部最佳解，這可以幫助探索新的解空間，提高收斂效果。本研究的突變策略主要應用於 P₂ 和 P₃，確保在較優解的基礎上仍有探索新解的能力。

(5) 迭代演化（Iteration）

重複上述步驟，直到滿足收斂條件。

遺傳演算法**[2]**在本研究中的最大特色是可以跳脫區域最佳解，同時藉由調整種群大小及演化次數提升計算速度及準確度，在本研究中屢次取得最佳的結果，搭配上Hausdorff距離[2]作為損失函數同時兼顧了速度與準確率，故後續實驗大多以這兩者為主進行實驗。

## **使用者取得貝茲曲線**

在上一步驟中會得到許多的三次貝茲曲線，透過堆疊的方式將其重疊至一張圖片上，最後透過flask伺服器傳送到網頁中供使用者下載成SVG。

# **肆**、**研究結果**

本研究使用線段向量和曲率特徵處理(Segment Vector and Curvature Feature Processing-SVCFP)進行切割，在後續進行貝茲擬合的演算法則使用遺傳演算法[2]為主體，並輔以Hausdorff距離[2]所做成之損失函數進行運算，並且在本研究中我們使用Inkscape做為比較對象，Inkscape作為免費軟體擁有方便的向量圖編輯功能，同時也支援直接輸入預製好的點陣圖轉換成向量圖型，故本研究將其作為比較對象。

市面上大多數的向量轉換應用，追求的主要是準確率，而本研究發現這種方法的最大缺點便是大量的節點，大量的節點會導致表達複雜，對於非專業繪圖人士也較難進行調整，所以本研究中將目標放在減少貝茲節點，讓使用者能夠在手繪曲線數位化以及向量化的同時保持簡單的描述，但不會因此而失真。本研究在測試上優先考慮節點數量，其次才考慮到準確率。

為了重現實際使用者的應用場景，本研究測試方法是先架設伺服器，而後在網頁上直接繪圖，並將繪圖的原始輸入、計算過程以及最終結果儲存起來，而為了比對實驗效果，會將手繪圖檔輸入Inkscape進行向量化，最後將兩者的輸出結果進行比對。

在本研究中，我們選擇使用 NSS（Normalized Scanpath Saliency） 作為主要評估指標，以量化擬合曲線與原始手繪圖形之間的相似程度。NSS 能有效衡量預測圖與參考圖像在視覺焦點上的重疊性，並反映其在關鍵結構區域的對齊程度，特別適用於評估人為描繪與機器生成圖形之間的匹配表現。

相較之下，傳統常用的 RMSE（均方根誤差）雖能反映數值誤差，但無法充分捕捉人眼感知層面的差異，因此在圖形擬合任務中，其解釋力相對有限。相較而言，NSS 更具感知相關性與應用價值，適合作為本研究之品質衡量核心指標。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 指標名稱 | 全名 | 數值範圍 | 優點 | 缺點 | 適用情境 |
| MSE | Mean Squared Error 平均平方誤差 | ≥ 0（越小越好） | 計算簡單、實作容易 | 對人眼不敏感，無法反映結構性差異 | 初步誤差分析，需搭配其他指標 |
| RMSE | Root Mean Squared Error 均方根誤差 | ≥ 0（越小越好） | 與MSE 相似，具單位意義（與像素範圍同） | 同樣無法反映結構，易受極端值影響 | 圖像壓縮、數值誤差統計 |
| PSNR | Peak Signal-to-Noise Ratio 峰值信噪比 | 通常介於 20~50 dB（越大越好） | 常用於圖像壓縮評估，對數表現使變化更平滑 | 仍無法捕捉視覺結構、邊緣資訊 | 視訊壓縮、影像傳輸品質評估 |
| SSIM | Structural Similarity Index 結構相似性指標 | -1 ~ 1（通常介於 0~1，越接近 1 越好） | 反映人眼感知、可評估亮度、對比與結構 | 計算較複雜，對部分圖像類型表現有限 | 視覺相似度分析、圖像品質主觀評估 |
| NSS | |  | | --- | |  |  |  | | --- | | Normalized Scanpath Saliency 標準化掃視顯著性 | |  |  |  |  |

表4-1. 評估依據比較表(由作者親自繪製)

SSIM 是一種專門用來評估兩張圖片在結構上是否相似的指標，它綜合考慮了亮度、對比度與結構三個面向。這代表它不僅僅是計算像素間的數值差距，而是從更貼近人類視覺的角度出發，去衡量圖像的整體相似度。其數值範圍介於 -1 到 1 之間，越接近 1 代表兩張圖片越相似。

因此，在處理如手寫字、曲線這類結構複雜的圖形時，使用 SSIM 能更有效捕捉到關鍵的結構資訊。以本研究中的「萬」字為例，即便SVCFP使用的點數遠少於 Inkscape，但在 SSIM 上仍維持在 0.92 以上，說明其在視覺上與原圖仍高度相似。

總結來說，SSIM 更符合本研究對「形狀還原」與「視覺準確度」的評估需求，能夠在精簡點數的同時，合理量化擬合結果與原圖的結構差異，因此成為本研究中最具代表性的評估指標。

|  |  |  |  |
| --- | --- | --- | --- |
| 原圖 | CNN切割 | SVCFP | Inkscape |
|  | 4個點 SSIM=0.9909 | 3個點SSIM=0.9923 | 15個點 SSIM=0.9958 |
|  | 5個點SSIM=0.9761 | 8個點SSIM=0.9853 | 41個點SSIM=0.9922 |
|  | 2個點SSIM=0.9658 | 8個點SSIM=0.9794 | 29個點SSIM=0.9896 |
|  | 4個點SSIM=0.9668 | 9個點SSIM=0.9708 | 29個點SSIM=0.9895 |
|  | 4個點SSIM=0.9629 | 8個點SSIM=0.9672 | 43個點SSIM=0.9886 |

表4-2. 第一組，初階手繪曲線(由作者親自繪製)

表4-3. 第二組，高階手繪曲線(由作者親自繪製)

|  |  |  |  |
| --- | --- | --- | --- |
| 原圖 | CNN切割 | SVCFP | Inkscape |
|  | 4個點SSIM=0.9411 | 9個點SSIM=0.9464 | 59個點SSIM=0.9836 |
|  | 4個點SSIM=0.9488 | 15個點SSIM=0.9575 | 61個點SSIM=0.9859 |
|  | 5個點SSIM=0.9311 | 14個點SSIM=0.9415 | 74個點SSIM=0.9805 |
|  | 6個點SSIM=0.9270 | 14個點SSIM=0.9533 | 99個點SSIM=0.9542 |
|  | 3個點SSIM=0.9418 | 15個點SSIM=0.9529 | 92個點SSIM=0.9835 |

表4-4. 第三組，簡易手寫國字(由作者親自繪製)

|  |  |  |  |
| --- | --- | --- | --- |
| 原圖 | CNN切割 | SVCFP | Inkscape |
| 國字:田 | 3個點SSIM=0.9582 | 10個點SSIM=0.9767 | 91個點SSIM=0.9893 |
| 國字:王 | 5個點SSIM=0.9692 | 7個點SSIM=0.9734 | 58個點SSIM=0.9908 |
| 國字:孔 | 4個點SSIM=0.9607 | 9個點SSIM=0.9617 | 59個點SSIM=0.9897 |
| 國字:元 | 6個點SSIM=0.9533 | 9個點SSIM=0.9716 | 83個點SSIM=0.9875 |
| 國字:何 | 5個點SSIM=0.9533 | 12個點SSIM=0.9726 | 81個點SSIM=0.9875 |

表4-5. 第四組，複雜手寫國字(由作者親自繪製)

|  |  |  |  |
| --- | --- | --- | --- |
| 原圖 | CNN切割 | SVCFP | Inkscape |
| 國字:朱 | 6個點SSIM=0.9329 | 12個點SSIM=0.9386 | 105個點SSIM=0.9834 |
| 國字:吳 | 7個點SSIM=0.9175 | 15個點SSIM=0.9331 | 104個點SSIM=0.9811 |
| 國字:金 | 5個點SSIM=0.9295 | 12個點SSIM=0.9419 | 111個點SSIM=0.9824 |
| 國字:車 | 4個點SSIM=0.9178 | 14個點SSIM=0.9163 | 134個點SSIM=0.9782 |
| 國字:萬 | 6個點SSIM=0.9145 | 19個點SSIM=0.9217 | 156個點SSIM=0.9807 |

# **伍、討論**

依照表4-2~4-5的結果顯示出，自創切割演算法在點位數與擬合程度之間取得了相當不錯的平衡。以第一組圖中的某些例子來看，雖然SVCFP在點數上略高於CNN切割（例如由5點增加至9點），但其SSIM值也顯著提升（例如從SSIM=0.9629提升至0.9672）。時我們考慮到手寫中文字的向量化，所以對於一些常見的手寫字進行測試，如田、王、元，而複雜的字如車、萬，SVCFP雖然使用了14至19個點，但在結構保留與視覺上都更加貼近原圖，SSIM也接近Inkscape的0.98水準。

此外，Inkscape雖然達到了最高的SSIM值（多數在0.98以上），但其代價是大量的節點使用，常見超過100點，如「車」使用了134點，「萬」甚至達到156點，這在實際應用中將會大幅增加儲存與運算成本，顯然不利於效能最佳化。而CNN切割法雖然點數最少（幾乎每例都小於6個點），但其擬合程度普遍偏低，SSIM常見落在0.91～0.96之間，無法有效保留筆畫的彎曲結構與字形特徵，尤其在筆劃變化劇烈的字元中效果特別差。

綜合來看，SVCFP在節點數量與視覺擬合之間達到較理想的折衷。例如在字元「孔」、「元」、「何」等中，SVCFP比CNN多出數個點，但其SSIM提升幅度明顯，且遠低於Inkscape的節點數，表現出優異的效率。更進一步，在抽象線條測試中，如圓形、蛇形、8字型等，SVCFP仍能有效擬合曲線，SSIM皆維持在0.94以上，顯示其不僅適用於字元向量化，也具備良好的通用性。

因此，從這些結果可以推論：在圖形結構保存與節點效率的綜合表現上，SVCFP相較於CNN切割法與Inkscape，提供了一種高效、實用且視覺效果良好的中間解法，非常適合作為向量化前處理的演算法核心。若後續結合動態調整節點數與局部曲率分析，SVCFP潛力進一步逼近Inkscape等級的精度，同時保持節點控制上的高度彈性。

# **陸、結論**

本研究旨在提供一種高效且準確地向量化解決方案，因此開發了一套基於貝茲曲線的手寫圖形向量化轉換系統，整合了線段向量和曲率特徵處理(Segment Vector and Curvature Feature Processing-SVCFP)、遺傳演算法（Genetic Algorithm, GAS）[1]以及 Hausdorff 距離[2]作為損失函數的優化架構。實驗結果顯示，該系統不僅大幅降低了計算成本，還能有效維持筆劃擬合的高品質，充分展現了本研究的三大核心貢獻。

首先，本研究針對有序筆劃的特性，進行了準確的貝茲曲線擬合，透過筆劃結構的分析與分段切割，結合貝茲曲線對每段筆劃進行最佳擬合，使曲線能夠精細還原手寫軌跡中的細節與特徵，為進一步提升系統的計算效率，本系統導入**Dijkstra 演算法[14]**於筆劃點之間的最佳匹配路徑搜尋，強化筆劃結構的一致性與向量化邏輯的合理性，同時，應用GAS[1]優化貝茲曲線的控制點分佈，有效降低擬合誤差約 20%，並讓整體運算時間維持在線性增長範圍內，適合應用於即時手寫輸入處理。

在準確度與視覺效果方面，系統透過 Hausdorff 距離[2]來量化曲線與原始筆劃之間的相似度，實驗結果顯示本方法與市面上的應用程式的平均誤差相差無幾，擁有相近的表現效果，此外，視覺化測試亦顯示，在複雜筆劃與筆劃交錯的情境下，本系統仍能產生流暢且精確的曲線轉換結果，特別值得一提的是，系統在使用最少節點的情況下，依然能維持高準確度與良好的幾何對應，達成「高準確度、強適應性、最簡節點」的目標。

綜合而言，本研究所提出的方法為手繪圖形的向量化提供了一種高效、準確且具有實用價值的解決方案，其潛在應用涵蓋數位手寫輸入、字體設計、手寫資料集生成等領域，展現出良好的擴展性與推廣潛力，未來若能進一步結合更多輸入設備與應用場景，預期可為手繪圖形處理技術帶來新的突破與發展，亦為人工智慧與數位筆跡分析領域提供更多可能性。  
本研究已經實現了良好的向量化效果，但仍有幾個方向可進一步改進與拓展：

本研究目前主要使用全局優化的方式來進行貝茲曲線的擬合，但在實際應用中，手寫筆劃常常具有明顯的局部特徵，例如鋒利的轉角或筆劃收尾的特殊變化。因此，未來可進一步提升曲線在局部的擬合能力，嘗試結合局部曲率分析或向量場調整技術，將轉折變化最劇烈之處作為節點，讓整體曲線能更細緻地對應原始筆劃的微妙變化。

此外，目前透過遺傳演算法搜尋最佳貝茲曲線雖可獲得不錯的結果，但運算時間花費較多，對即時處理需求仍有侷限。未來可嘗試導入強化學習（Reinforcement Learning, RL）或結合其他混合式最佳化技術，以加速搜尋收斂過程，提升整體向量化效率與即時性，讓系統更適用於互動式手寫場景。

在應用層面，目前系統主要設計於網頁端的滑鼠或觸控輸入，未來則可擴展至支援更多元的裝置，包括觸控筆、繪圖板、行動裝置等，並探索其在數位簽名、手寫驗證、AR/VR 空間手寫等場域中的潛力與應用價值。

最後，雖然本研究使用 Hausdorff 距離[2]作為損失函數，取得了不錯的擬合效果，但該距離指標在處理複雜筆劃時仍可能忽略局部對應的細節。未來可進一步導入動態時間規劃（Dynamic Time Warping, DTW）等更適合曲線對齊的損失計算方法，以提升曲線擬合的準確度與穩定性。

綜合以上幾點，本研究所提出的技術未來在準確性、效率與通用性方面均具高度可擴展性，將有機會推動人工智慧在手寫圖形處理與數位人機互動領域的進一步發展。

# **柒、參考文獻**

[1]Holland, J. H. (1975). Adaptation in natural and artificial systems. University of Michigan Press.

[2]Hausdorff, F. (1914). Grundzüge der Mengenlehre. Veit.

[3]Baydas, S., & Karakas, B. (2019). Defining a curve as a Bezier curve. Journal of Taibah University for Science, 13(1), 522-528.

[4]Pastva, T. A. (1998). Bezier curve fitting (Doctoral dissertation, Monterey, California. Naval Postgraduate School).

[5]Mad, S. A. A. A. S., Zain, M. Y. M., & Miura, K. T. (2023). Curve fitting using generalized fractional Bézier curve.

[6]池品軒, 林文杰, & 莊榮宏. (2013). 基於擴散曲線之點陣圖自動向量化 (Doctoral dissertation).

[7]Stock, K., Pouchet, L. N., & Sadayappan, P. (2012). Using machine learning to improve automatic vectorization. ACM Transactions on Architecture and Code Optimization (TACO), 8(4), 1-23.

[8]Krzeszewska, U., Poniszewska-Marańda, A., & Ochelska-Mierzejewska, J. (2022). Systematic comparison of vectorization methods in classification context. Applied Sciences, 12(10), 5119.

[9]De Jong, K. (1988). Learning with genetic algorithms: An overview. Machine learning, 3, 121-138.

[10]明杰, & 張良正. 遺傳基因演算法應用於擬三維地下水數值模式之參數優選.

[11]Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. Computer graphics and image processing, 1(3), 244-256.

[12]Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: the international journal for geographic information and geovisualization, 10(2), 112-122.

[13]Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[14]Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik, 1, 269–271.