Improved Fast Intra Mode Decision in H.266/Versatile Video Coding (VVC) Based on Deep Learning

Chi-Chou Kao[1]*, Mei-Yu Lai[2]

[1,2]Department of Computer Science and Information Engineering

National University of Tainan

Tainan, Taiwan

ABSTACT

H.266/VVC is ultra-high-definition video over 4K, and can be applied in High Dynamic Range Imaging (HDR) and wide color gamut (WCG). However, it has high coding computational complexity based on the coding unit (CU) structure of a quadtree plus binary tree (QTBT). This plan first proposes a fast coding unit spatial features decision method to reduce the coding complexity in H.266/VVC such that the H.266/VVC coding can be speed up. Another important contribution of this plan is to combine video coding with Convolutional Neural Networks (CNNs) in H.266/VVC in-frame coding mode prediction decision. It can be shown that the proposed methods can achieve better encoding performance than the original encoding method (JEM7.0).

Keywords: H.266, VVC, CNN, 3-step search

---

* Corresponding author: cckao@mail.nutn.edu.tw
DOI：10.53106/222344892022041201004

基於深度學習之改良式多功能影像編碼快速畫面內模式決策研究

高啟洲*, 賴美妤

國立臺大學資訊工程學系碩士班

摘要

H.266/Versatile Video Coding (VVC) 是針對 4K 以上的超高畫質影片，且能適用在高動態範圍(High Dynamic Range Imaging, HDR)及廣色域(wide color gamut, WCG)中，但基於四元樹加二元樹(Quadtree plus Binary Tree, QTBT)的編碼單元(Coding Unit, CU)結構增加了 H.266/VVC 編碼的計算複雜性。本論文提出了一種基於深度學習之改良式多功能影像編碼快速畫面內模式決策方法，減少 H.266/VVC 內編碼複雜性以加快 H.266/VVC 的編碼速度，並將畫面內影像編碼結合卷積神經網路(Convolutional Neural Networks, CNN)在 H.266/VVC 畫面內編碼的模式預測決策，以達到比原始編碼方式(JEM7.0)更好的編碼效能。

關鍵字：影像編碼，卷積神經網路，三步搜尋

1. 緒論

影像編碼是一種用於計算機圖像處理計算機視覺和視覺傳達的基礎技術，最早從1960 年代開始了影像編碼的研究及開發。並從 1969 年，著名的國際論壇圖片編碼研討會開始專門致力於影像編碼的發展。從此之後，學術界和工業界都為此做出許多努力。隨著影像編碼的發展並為了影像編碼的互操作性，在過去三十年逐漸開始制訂一系列有關影像編碼的標準。在國際標準化組織 ISO/IEC 中兩個專家小組：Joint Photographic Experts Group (JPEG)、Moving Picture Experts Group (MPEG)，以及 ITU-T 的 Video Coding Experts Group(VCEG)，這些組織已經發布了幾個著名且被廣泛採用的標準，例如：JPEG、JPEG 200、H.262(MPEG-2 Part 2)、H.264(MPEG-4 Part 10 or AVC)、H.265(MPEG-H Part 2 or HEVC)等等。

目前 2013 年正式發布的高效能影像編碼(High Efficiency Video Coding, HEVC, H.265)是最新的影像編碼技術，壓縮效能能達到 H.264 的兩倍，當時被認定是能替代高級影像編碼(Advanced Video Coding, AVC, H.264)的正式版本。隨著影像技術的進步、設備的提升、超高畫質(Ultra High Definition, UHD)影片越來越普及，進一步提升影像編碼壓縮效率的技術以將 UHD 影片容納在有限的儲存空間和有限的傳輸頻寬中也越重要。因此，在 H.265/HEVC 之後， MPEG 和 VCEG 組成了聯合影片專家小組(Joint Video Exploration Team, JVET)，希望開發更先進的影片編碼技術，該小組開發了聯合探索模型來進行研究。至此從 2018 年以來，JVET 團隊研究出一種新的影片編碼標準稱之為多功能影像編碼(Versatile Video Coding, VVC, H.266)，作為繼承 HEVC 的新一代標準。

H.266/VVC 預期和 H.265/HEVC 相比，特別是對於 UHD 影片， H.266/VVC 可以通過節省約 50% 的位元率(Bitrate)來提高壓縮效率，同時還能保持相同的影像品質，但 H.266/VVC 的改進提升了乘法編碼和解碼的複雜性。

H.266/VVC 增加及修改許多先進的編碼技術以增加編碼效能及減少位元率，例如在編碼單元(Coding Unit, CU)中，H.265/HEVC 採用四元樹(QuadTree, QT)架構來進行分割，CU 大小為最小 8x8 到最大 64x64 的正方形編碼區塊；而 H.266/VVC 則是採用四元樹加二元樹(QuadTree plus Binary Tree, QTBT)架構來分割，支援 CU 從最小 8x8 到最大 128x128 的編碼區塊，且依據畫面的紋理特性不僅提供正方形還提供矩形的編碼區塊，比起 H.265/HEVC 更能適應各種區域特徵。這些增加或修改的先進技術整體在 H.266/VVC 的參考軟體 JEM7.0 之編碼器中雖然有很好的編碼效能，但需要消耗的複雜度比 H.265/ HEVC 的參考軟體 HM16.6 提高許多。

因此本論文提出了一種改良式畫面內模式決策研究，將影像編碼結合人工智慧系統(Artificial Intelligence, AI)，基於深度學習(Deep Learning)中卷積類神經網路(Convolutional Neural Network, CNN)並結合改良式搜尋及分類法在 H.266/VVC 畫面內編碼的模式預測，進一步提升 H.266/VVC 畫面內預測模式，預期在不損失太多影像畫質的同時可以減少時間複雜度。

2. 系統架構

研究方法的系統架構如圖 1 ，H.266/VVC 編碼模式預測基礎可分為：前處理階段(Pre-processing Stage)、訓練階段(Training Stage)和測試階段(Testing Stage)。在進入模型的訓練階段之前，需要預先對將送進去訓練的輸入資料進行一些前處理，因此在前處理階段我們先探討要拿什麼資訊當訓練資料及如何取得這些訓練資料，並且

要對欲訓練的資料庫預處理，以利訓練資料庫在訓練階段能夠發揮較佳的功效。此外，為了能夠在往後將本論文所提出的模式預測方法結合至編碼軟體 JEM7.0 中，且為了能方便從編碼軟體 JEM7.0 中直接獲得送入預測模型的測試資料，我們透過模擬影像壓縮的編碼流程，讓前處理階段與編碼流程一致以利我們取得進入訓練的資料。因此我們透過使用 H.266/VVC 壓縮標準 JEM7.0 事先進行一遍完整的編碼，讓訓練資料的準確性和效率都能夠提升，並參考 JEM7.0 中畫面內編碼預測的快速演算法，其中有兩個部分，一是較低複雜度的預測殘差絕對值總和(Sum of Absolute Transformed Differences, SATD)，二是最可能模式(Most Probable Modes, MPM)。

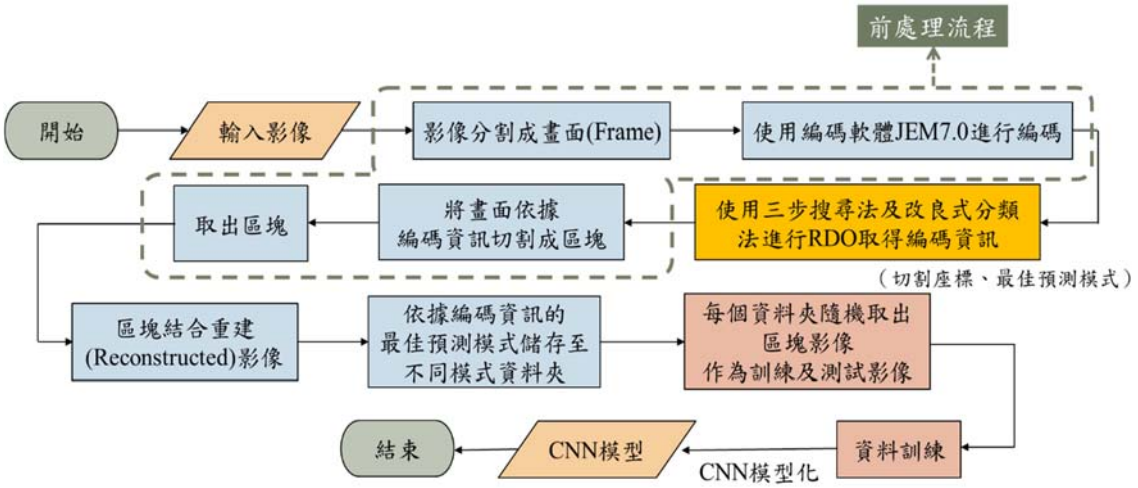

圖 1. 系統架構

前處理流程首先會將各種不同解析度的影像序列(Video Sequence)分割成一張一張畫面(Frames)，透過 H.266/VVC 編碼軟體 JEM7.0 進行一次原始編碼後，使用本篇論文所提出的 H.266/VVC 快速畫面內模式決策中的三步搜尋法及改良式分類法進行 RDO 後取得編碼資訊，編碼資訊中又包含了依據畫面切割區塊的座標大小、切割的寬高以及經過改良式搜尋法所產生的最佳預測模式(Mode)，將上一步驟中取出的畫面區塊結合重建(Reconstructed)影像後儲存至該最佳預測模式的資料夾中，如圖 2，區塊所對應的模式資料夾就是 CNN 訓練時每一筆訓練資料的標籤(Label)，每個資料夾中會有不同數量的區塊圖片，其中較常出現的模式數量就會較多，反之，較不常出現的模式數量就會比較少，在本論文中採用最少量的模式來決定每個模式最少要取幾張區塊圖片來做訓練資料。

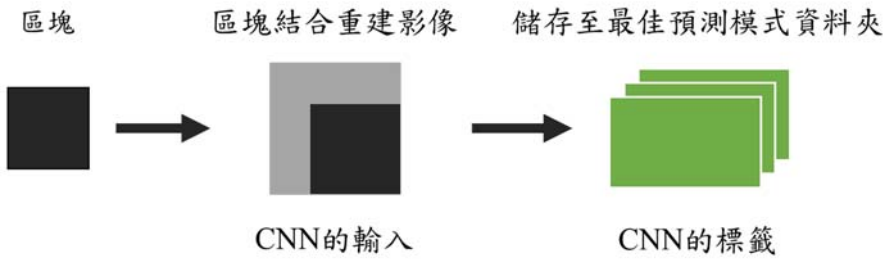

圖 2. CNN 模型的訓練及測試影像

最後，從每個模式資料夾中的隨機取出 500 張區塊圖片，把前 400 張區塊圖片作訓練模型的訓練資料庫，訓練資料的數量總共有 400 × 80 = 32,000 筆做為訓

片作訓練模型的訓練資料庫，訓練資料的數量總共有 400 × 80 = 32,000 筆做為訓練圖片，而另外 100 張區塊圖片做為測試模型的測試資料庫，總共有 100 × 80 = 8,000 筆測試圖片，接著進行 CNN 模型化產出 CNN 模型後再執行系統測試。

編碼資訊本論文將由原本的全域搜尋(Global Search)取代為使用由 Koga 在 1981 年提出的三步搜尋法(Three-Step Search)來取得，為一種有效減少搜尋計算量的快速演算法，三步搜尋與全域搜尋相較之下，三步搜尋的優勢在於可以找到具有最小成本函數的搜尋結果，並且與全域搜尋相比可以大幅減少計算時間和搜尋點的數量。以搜尋 255 個的圖像子區塊為例，如圖 3~4 為全域搜尋及三步搜尋的計算量，全域搜尋就必須遍歷所有圖像子區塊的 225 個搜尋點，但三步搜尋相較於全域搜尋只需 1/9 的計算量(9+8+8 個搜尋點)。

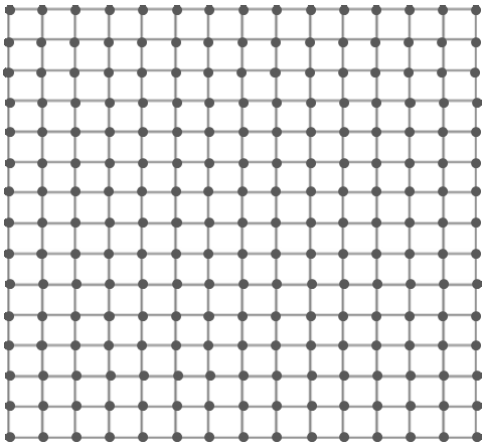

圖 3. 全域搜尋計算量

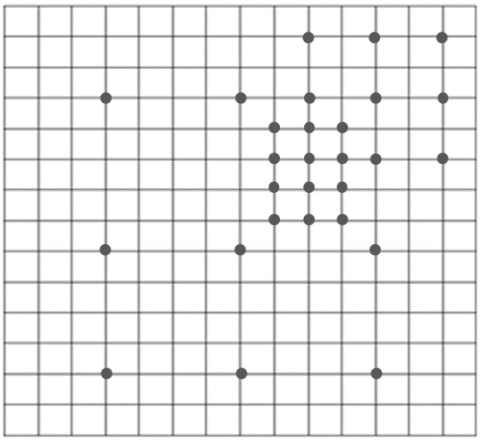

圖 4. 三步搜尋計算量

三步搜尋與全域搜尋的影像測試結果如表 1，峰值訊噪比(Peak Signal-to-Noise Ratio, PSNR)的數值越高代表影像的壓縮品質越好，雖然三步搜尋在 PSNR 上較全域搜尋降低了 16.81%，但在計算搜尋時間上三步搜尋比全域搜尋大幅節省了 90.8%的搜尋時間。如圖 5 所示。三步搜尋的搜尋步驟如下：第一步以中心為原點開始，步長(Step) 設為 S = 4 ，搜尋參數值設為 P = 7， 向外延伸 4 格相等距離(步長正負 4 的搜尋範圍)的 8 個位置點，在 SATD 值中找到最小區塊誤差(Minimum Block Distortion, MBD)得到最小成本函數(Cost Function)的位置成為新的搜尋點也就是第二步的位置。接著重複上述步驟，再次減少第二步距離的一半，取得第三步後得到最後的搜尋結果，並獲取圖像編碼資訊。

表 1. TSS 與全域搜尋的效能比較

參數	峰值訊噪比 PSNR(db)	計算搜尋時間 CST
全域搜尋 Full Search	100%	100%
三步搜尋 3-Step Search	83.19%	9.2%

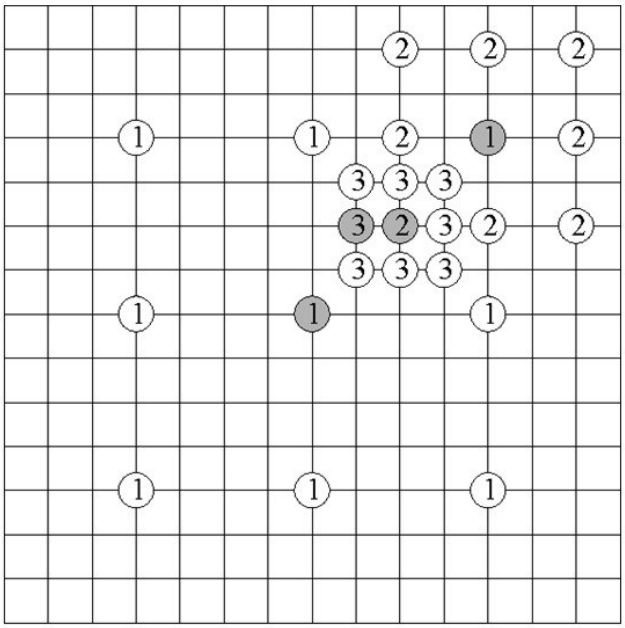

圖 5. 三步搜尋演算法

為了更有效提升編碼效能，在取得編碼資訊獲得最佳預測模式時提出了一種改良式分類法，目的在於減少畫面內預測模式的模式數量，藉此來減少畫面內預測密集的進行 RDO 所需的時間。提出的改良方法針對 67 種畫面內預測模式中的 9 種角度模式如圖 6，並分為 7 個群組：(2, 10, 18)、(10, 18, 26)、(18, 26, 34)、(26, 34, 42)、(34, 42, 50)、(42, 50, 58)、(50, 58, 65)、(26, 34, 42)、(26, 34, 42)，接著進行以下步驟：

1. 計算 7 個群組內三個模式的總 RDcost。
2. 選出總 RDcost 最小的群組後計算群組內 17 個模式的 RDcost。
3. 比較 17 個模式與模式 0 和 1 的 RDcost，選出最低的 3 個模式。
4. 進行 RDO 篩選出一個最佳預測模式。

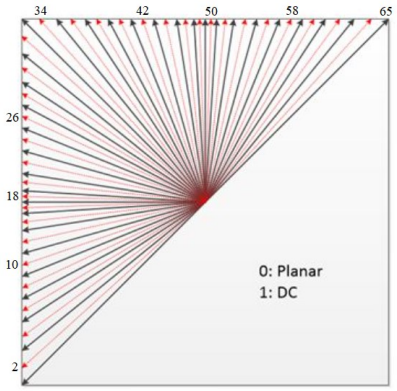

圖 6. H.266/VVC 畫面內預測模式

由於 H.266/VVC 的畫面內預測模式相較 H.265/HEVC 的大幅增加，而原始的編碼方式會對每個預測模式都進行一次 RDO 的過程，因此造成時間複雜度的提升，

因此透過改良式分類法將預測模式事先分類為群組後分別計算 RDcost 再進行篩選，可減少大約 1/9 RDO 原本在畫面內預測模式時需在每個預測模式上進行 RDO 所花費的時間。

3. 實務開發

H.266/VVC 畫面內編碼模式的環境設定如表 2 所示，範圍從 0 到 1023 ，每個像素值以 10 Bit 為單位，除去了 H.265/HEVC 以 8 bit 的低複雜度配置並且每 8 張畫面編碼一次，因此其真實位元率為計算結果位元率乘以 8 。

表 2. H.266/VVC 原始環境設定

設定檔	Intra_Main10
Bit Depth	10
TemporalSubsampleRatio	8
CTU Size	128
MinQT	8
MaxBTDepth	3
Intra Prediction	DC, Planner, 65 Angles
Inter Prediction	ATMVP, AFFINE, OBMC, FRUC, BIO
Transform	AMT, ST, SDT
In-loop Filter	De-blocking, SA
Entropy Coding	CABAC

H.266/VVC 樣本從原有的 Class B~F 增加了 4K 解析度的 Class A1 和 Class A2 測試序列，表 3 為所有測試序列的設定，表 4 則為各測試序列的影像解析度。

實驗環境設置主要由兩大部分，第一個部份為訓練 CNN 之預測模型架構，圖片識別中，神經網路架構採用監督式學習(Supervised Learning)的網路架構，訓練過程中提供輸入的圖片，也會提供標籤(Label)的輸出資料，使模型可以從中學習輸入與輸出資料的關係。另外一部份為結合訓練好的模型及參數至影像壓縮的參考軟體 JEM7.0 中並執行編碼測試，這兩部份的實驗在環境設置上我們使用不同的軟硬體規格及工具。我們用來訓練 CNN 預測模型架構的實驗環境設置如表 5 所示，使用的深度學習相關工具為 Tensorflow 函式庫，並且採用 GPU 版本來進行訓練。

表 3. H.266/FVC 測試序列的測試設定

Class	Name	Frame Count	Frame Rate	Bit Depth
A1	Tango	294	60 fps	10
A1	Drums	300	100 fps	10
A1	Campfire	300	30 fps	10
A1	ToddlerFountain	300	60 fps	10
A2	CatRobot	300	60 fps	10
A2	TrafficFlow	300	30 fps	10
A2	DaylightRoad	300	60 fps	10
A2	Rollercoaster	300	60 fps	10
B	Kimono	240	24 fps	8
B	ParkScene	240	24 fps	8

B	Cactus	500	50 fps	8
B	BQTerrace	600	60 fps	8
B	BasketballDrive	500	50 fps	8
C	RaceHorses	300	30 fps	8
C	BQMall	600	60 fps	8
C	PartyScene	500	50 fps	8
C	BasketballDrill	500	50 fps	8
D	RaceHorses	300	30 fps	8
D	BQSquare	600	60 fps	8
D	BlowingBubbles	500	50 fps	8
D	BasketballPass	500	50 fps	8
E	FourPeople	600	60 fps	8
E	Johnny	600	60 fps	8
E	KristenAndSara	600	60 fps	8

表 4. H.266/VVC 測試序列的影像解析度大小

A1	3840×2160
	4096×2160
A2	3840×2160
	4096×2160
B	1920×1080
C	832×480
D	416×240
E	1280×720

表 5. 訓練 CNN 預測模型架構的環境設置

硬體設備	
CPU	Intel Xeon E7-4809 v4
RAM	128GB DDR4-3600
GPU	TUF-GTX1660S
OS	Ubuntu 18.04 x64

在本篇所使用的方法採用 CNN 作為特徵的訓練模型，架構有三層卷積層(Convolutional Layer)、三層池化層(Pooling Layer)及一層全連接層(Fully-Connected Layer)共同組成，如圖 7 所示。表 6 為各個架構參數設定，並加入了線性整流元來將卷積過後造成的負值去掉變成 0，每張原始輸入影像皆可以用一個 1024 維的特徵向量來表示。

在訓練階段使用的幾個參數設置如下：

- 每次訓練數據量(batch size):1000(blocks)
- 迭代次數:10000(epochs)
- 學習率(learning rate):0.025
- Dropout:0.6

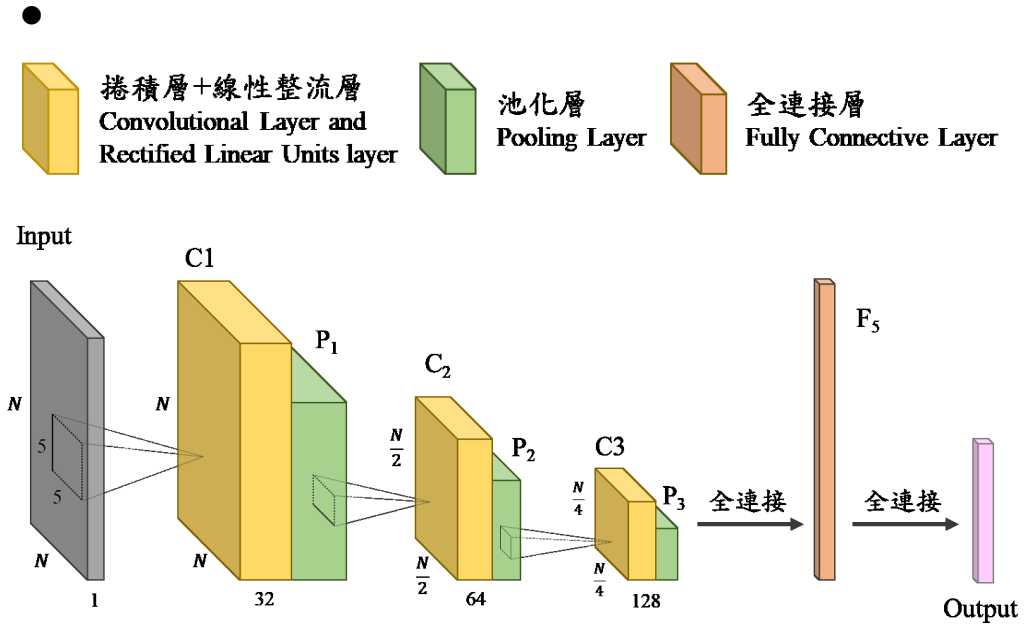

圖 7. 神經網路訓練模型

表 6. CNN 參數設定

Layer	卷積核大小	卷積數量	備註
C_1	5×5	32	ReLU
C_2	4×4	64	ReLU
C_3	3×3	128	ReLU
P_1	2×2		Max Pooling
P_2	2×2		Max Pooling
P_3	2×2		Max Pooling
F_5		1024	

結合 CNN 模型執行影像編碼之實驗環境配置如表 7 所示，此系統實作的程式語言為 C/C++，但過程中會調用我們的訓練模型，而在編碼時所使用的版本為 CPU 版本。此實驗我們將 ClassB 至 ClassE 這些不同影像特性及解析度做為測試影像。

表 7. CNN 模型執行影像編碼環境配置

硬體設備	
CPU	Intel(R) Core(TM) i7-4790 CPU @3.60Ghz
RAM	16 GB
GPU	GTX1660 OC 6G
軟體配置	
編碼器	JEM 7.0
編碼工具	Visual Studio 2019, Tensorflow, Python
設定檔	Intra_main 10
QP 值設定	22、27、32、37

測試序列		
ClassA1	Tnago	CampfireParty
ClassA2	CatRobot	TrafficFlow
ClassB	Kimono	BasketballDrive
	Catus	ParkScene
ClassC	BasketballDrill	RaceHorsesC
ClassE	FourPeople	Johnny

實驗結果則是使用 Bjøntegaard Delta Bit Rate(BD-rate)和時間減少率(ΔT)做為比較依據，BD-rate 是在測試序列中編碼時，使用相同的量化參數作為比較不同編碼方法下所節省的位元速率(Bitrate)。時間減少率的公式定義如式(1)，T_{Prop}及$T_{JEM7.0}$分別代表本論文提出的決策的編碼時間和影像編碼標準 JEM7.0 的編碼時間。

$$\Delta T = \frac{T_{JEM7.0} - T_{Prop}}{T_{JEM7.0}} \qquad (1)$$

在實驗結果中，我們以文獻[6]、[7]及本論文所提出的方法進行性能的比較。評斷指標包含了 BD-rate 及時間減少率ΔT，實驗結果如表 8~9 。變異數決策[6]的平均時間減少率為-45.199%，平均 BD-rate 為-1.164%，傳統深度學習決策[7]的平均時間減少率為 26.443%，平均 BD-rate 為-1.042%，本論文提出的方法平均時間減少率為 35.125%，在 BD-rate 方面則是-1.294%。

表 8. 本論文與文獻[6]比較的實驗結果

測試序列		本文決策		變異數決策[6]	
		BD-rate	ΔT	BD-rate	ΔT
ClassA1	Tnago	-1.082%	-30.752%	-0.521%	-45.301%
	CampfireParty	-1.934%	-32.683%	-1.904%	-38.84%
ClassA2	CatRobot	-0.867%	-30.591%	-1.093%	-31.301%
	TrafficFlow	-0.885%	-39.13%	-1.925%	-41.594%
ClassB	Kimono	-0.671%	-37.742%	-0.466%	-55.842%
	Catus	-1.959%	-37.662%	-0.871%	-56.622%
	BasketballDrive	-0.898%	-33.589%	-1.585%	-52.05%
	ParkScene	-0.543%	-34.511%	-0.332%	-59.353%
ClassC	BasketballDrill	-1.705%	-35.504%	-1.891%	-56.954%
	RaceHorsesC	-1.294%	-36.432%	-1.152%	-33.115%
ClassE	FourPeople	-0.726%	-38.408%	-1.446%	-40.889%
	Johnny	-0.827%	-34.499%	-0.792%	-30.523%
Average		-1.116%	-35.125%	-1.164%	-45.199%

表 9. 本論文與文獻[7]比較的實驗結果

測試序列		本文決策		傳統深度學習決策[7]	
		BD-rate	ΔT(%)	BD-rate	ΔT(%)
ClassA1	Tnago	-1.082%	-20.752	-1.424%	-22.602
	CampfireParty	-1.934%	-22.683	-1.623%	-30.354
ClassA2	CatRobot	-0.867%	-30.591	-0.941%	-29.455
	TrafficFlow	-0.885%	-29.13	-0.891%	-23.264
ClassB	Kimono	-0.671%	-27.742	-0.633%	-27.354
	Catus	-1.959%	-27.662	-1.827%	-30.333
	BasketballDrive	-0.898%	-23.589	-0.436%	-24.244
	ParkScene	-0.543%	-24.511	-0.445%	-26.939
ClassC	BasketballDrill	-1.705%	-25.504	-1.784%	-24.661
	RaceHorsesC	-1.294%	-26.432	-1.426%	-26.252
ClassE	FourPeople	-0.726%	-28.408	-0.672%	-25.416
	Johnny	-0.827%	-24.499	-0.409%	-26.263
Average		-1.116%	-25.959	-1.042%	-26.443

在 BD-rate 的方面，本論文所提出的決策方法與傳統深度學習方法相比損失較多一點，但在時間減少率上可明顯減少許多，與變異數決策相較之下在畫質損失的方面本論文的表現會比變異數決策還要來的少。本文決策透過有效的減少了原始 H.266/VVC 中不必要的 RDO 以及 CNN 的模型架構，因此能夠在維持影像畫質的同時節省編碼的時間。

4. 參考文獻

[1] K. Zhang, W. Zuo, Y. Chen, D. Mendfefefg, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[2] Sookyung Ryu and Je-Won Kang "Machine Learning-Based Fast Angular Prediction Mode Decision Technique in Video Coding" in IEEE Transactions on Image Processing ,Vol. 27, Issue: 11, Nov. 2018.

[3] R. D. Dony and S. Haykin, "Neural network approaches to image compression," *Proceedings of the IEEE*, vol. 83, no. 2, pp. 288–303, 1995.

[4] I.Mrazova, Kukacka, "Hybrid convolutional neural networks", Industrial Informatics *INDIN2008. 6th IEEE International Conference*, 2008.

[5] Y.Lecun, et al., "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol.86, no. 11, pp.2278-2324, 1998

[6] T. Fu, H. Zhang, F. Mu and H. Chen, "Fast CU Partitioning Algorithm for H.266/VVC Intra-Frame Coding," *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 2019, pp. 55-60, doi: 10.1109/ICME.2019.00018

[7] Huang Jingya,” Prediction of in-screen mode of H.266 / FVC video coding based on convolutional neural network”, June 2017.