

# THÔNG TIN CHUNG CỦA NHÓM

- Link YouTube video của báo cáo (tối đa 5 phút):  
[CS2205.FEB2025---TwinLiteNetPLus](#)
- Link slides (dạng .pdf đặt trên Github của nhóm):  
[Slide pdf](#)
- *Mỗi thành viên của nhóm điền thông tin vào một dòng theo mẫu bên dưới*
- *Sau đó điền vào Đề cương nghiên cứu (tối đa 5 trang), rồi chọn Turn in*
- *Lớp Cao học, mỗi nhóm một thành viên*

- Họ và Tên: Hoàng Công  
Danh
- MSSV: 240101039



- Lớp: CS2205.FEB2025
- Tự đánh giá (điểm tổng kết môn): 8.5/10
- Số buổi vắng: 0
- Số câu hỏi QT cá nhân: 1
- Số câu hỏi QT của cả nhóm: 0
- Link Github: [GitHub](#)

# ĐỀ CƯƠNG NGHIÊN CỨU

## TÊN ĐỀ TÀI (IN HOA)

TWINLITENET+: MỘT MÔ HÌNH MẠNH MẼ HƠN CHO PHÂN ĐOẠN LÀN ĐƯỜNG VÀ VÙNG CÓ THỂ LÁI THEO THỜI GIAN THỰC

## TÊN ĐỀ TÀI TIẾNG ANH (IN HOA)

TWINLITENET+ A STRONGER MODEL FOR REAL-TIME DRIVABLE AREA AND LANE SEGMENTATION

## TÓM TẮT (Tối đa 400 từ)

Trong thế giới hiện đại ngày nay, để có thể tạo ra được một hệ thống xe tự hành thì cần rất nhiều tác vụ khác nhau và 2 tác vụ quan trọng nhất - để có thể giúp phương tiện xác định được chính xác khu vực an toàn và vị trí làn đường để điều hướng - là: **Phân đoạn vùng có thể lái** và **Phân đoạn làn đường**. Tuy nhiên, các mô hình hiện nay có thể giải quyết tác vụ này như **YOLOP**, **Hybrid Nets** hay **YOLOP v2** thường đòi hỏi tài nguyên tính toán lớn và khó triển khai trên các thiết bị nhúng. Chính vì thế **TwinLiteNet** được đề xuất với kiến trúc **lightweight CNN** (Mô hình có trọng số rất ít), tốc độ xử lý rất nhanh, độ chính xác lại ngang ngửa với các mô hình SOTA. Và sau này được cải tiến lên phiên bản tốt hơn là **TwinLiteNet+**.

TwinLiteNet+ sử dụng encoder mới với các lớp **Depthwise Dilated Separable Convolution (DESP)**, kết hợp với cơ chế chú ý **Partial Class Activation Attention (PCAA)**, và **decoder riêng biệt** cho từng nhiệm vụ. Mô hình được xây dựng dưới bốn biến thể: **Nano**, **Small**, **Medium** và **Large**, tương ứng với khả năng đáp ứng của từng loại thiết bị phần cứng. Các kỹ thuật như Re-parameterization, Tversky Loss, Focal Loss sẽ được áp dụng để tăng cường độ chính xác và tốc độ suy luận. Biến thể lớn nhất đạt độ chính xác cao nhất với **92.9% mIoU** và **34.2% IoU**, nhưng vẫn có **FLOPs thấp hơn** so với các mô hình SOTA.

Mục tiêu đề tài là triển khai lại TwinLiteNet+ dựa theo bài báo gốc, huấn luyện trên tập dữ liệu **BDD100K**, đánh giá các cấu hình mô hình về tốc độ, độ chính xác, mức

tiêu thụ tài nguyên, và kiểm thử thực tế trên thiết bị nhúng.

## **GIỚI THIỆU** (*Tối đa 1 trang A4*)

Việc hiểu và nhận diện được môi trường xung quanh là yếu tố cực kỳ quan trọng và thiết yếu của hệ thống vận hành xe tự động. Và trong vô vàn tác vụ quan trọng đó thì phân đoạn vùng có thể lái được (**Drivable Area Segmentation**) và phát hiện làn đường (**Lane Detection**) là 2 bài toán quan trọng nhất của mô hình trên. Trong đó, yêu cầu đầu ra không chỉ dừng lại ở việc có được độ chính xác cao trong tính toán mà còn là khả năng xử lý trong thời gian thực - điều mà đang được thực hiện trên thiết bị giới hạn tài nguyên tính toán.

**TwinLiteNet+** là phiên bản nâng cấp từ TwinLiteNet, được đề xuất bởi nhóm nghiên cứu UIT – Đại học Quốc gia TP.HCM. Mô hình này ứng dụng các kỹ thuật nhẹ như:

- Depthwise Dilated Separable Convolution thay cho convolution truyền thống nhằm giảm chi phí tính toán.
- Partial Class Activation Attention (PCAA) để tăng cường trọng tâm vào vùng quan trọng trong ảnh.
- Thiết kế hai decoder riêng biệt cho từng tác vụ giúp tối ưu hóa độc lập từng nhiệm vụ.
- Cấu trúc mô-đun dễ triển khai, có 4 phiên bản mô hình với số tham số từ 0.03M đến 1.94M, tương ứng với mức FLOPs từ 0.57G đến 17.58G.

Kết quả thực nghiệm trên tập **BDD100K** cho thấy TwinLiteNet+ vượt trội về cả độ chính xác và tốc độ xử lý so với các mô hình hiện có. Ví dụ, bản Large đạt 92.9% mIoU (Drivable Area) và 34.2% IoU (Lane), vượt cả SegFormer và YOLOP, trong khi nhẹ hơn tới 11 lần về FLOPs.

Với các ưu điểm này, TwinLiteNet+ là ứng viên sáng giá để triển khai trong các ứng dụng xe tự hành thực tế.

## **MỤC TIÊU** (*Viết trong vòng 3 mục tiêu*)

1. Phân tích và tái hiện được cấu trúc của TwinLiteNet+ dựa theo bài báo.
2. Huấn luyện mô hình trên tập dữ liệu BDD100K cho hai tác vụ phân đoạn

chính.

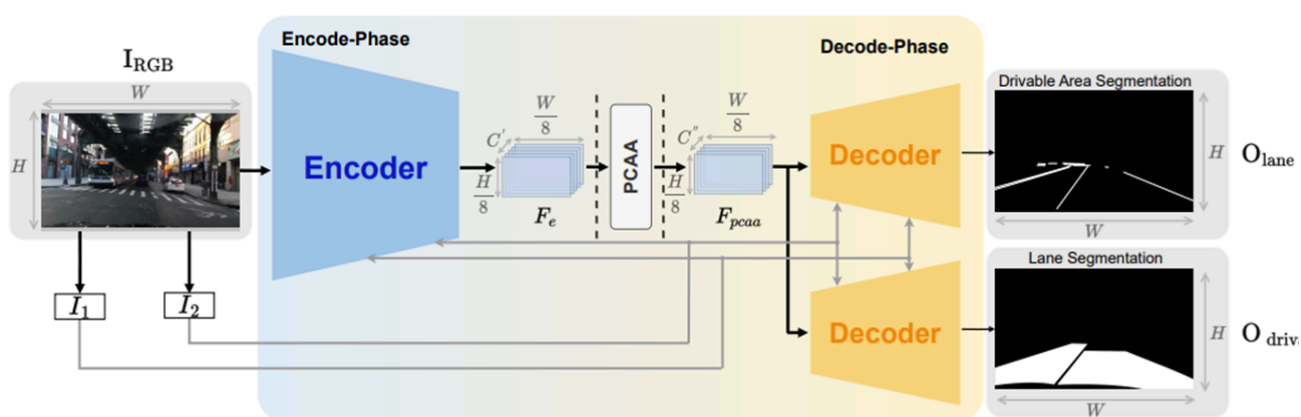
3. Triển khai và đánh giá qua thời gian thực của mô hình thông qua thực nghiệm đã làm được.

## NỘI DUNG VÀ PHƯƠNG PHÁP

### NỘI DUNG

1. Nghiên cứu 2 bài toán trong mô hình xe tự hành: Drivable Area Segmentation và Lane Detection.
2. Phân tích về kiến trúc của TwinLiteNet+: Encoder, Decoder, PCAA.
3. Tiền xử lý dữ liệu BDD100K với 100k frames và gán nhãn cho 10 task
4. Huấn luyện mô hình và áp dụng 2 hàm Focal Loss, Tversky Loss.
5. Đánh giá mô hình bằng các chỉ số: mIoU cho Drivable Area Segmentation, IoU cho Lane detection và Latency của mô hình.
6. So sánh TwinLiteNet+ với mô hình trước

### PHƯƠNG PHÁP



Hình 1: Mô tả cách hoạt động của mô hình TwinLiteNet+

#### 1. Phân rã bài toán

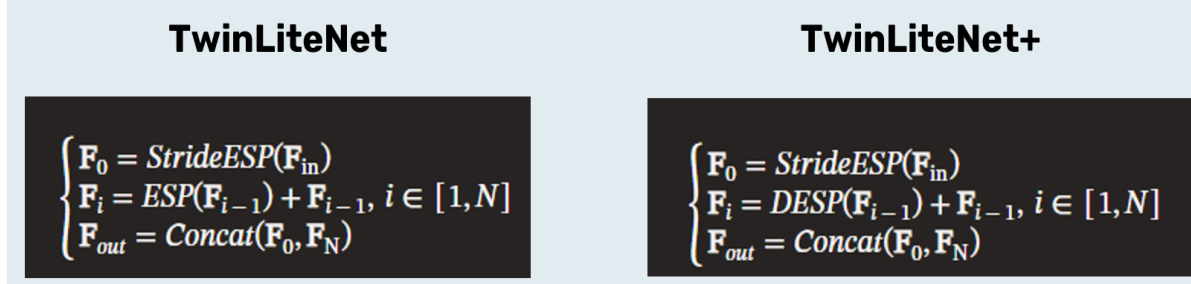
Bài toán trong xe tự hành được chia thành 2 tác vụ chính là:

- **Phân đoạn vùng có thể lái (Drivable Area Segmentation):** xác định khu vực có thể lái được trên đường.
- **Phát hiện làn đường (Lane Detection):** xác định vị trí các vạch kẻ đường.

Mỗi tác vụ sẽ có đầu ra riêng biệt → mô hình được thiết kế theo kiến trúc đa nhánh (multi-head) để xử lý độc lập.

## 2. Trích xuất đặc trưng ảnh đầu vào

### 2.1 Encoder (DESP – Depthwise Dilated Separable Convolution)



Hình 2: Mô tả 2 phương pháp TwinLiteNet và TwinLiteNet+

Encoder của TwinLiteNet sử dụng ý tưởng từ module ESP của ESPNet với ý tưởng ở đây là: Reduce (Giảm), Split (Chia), Transform (Biến đổi), Merge (Kết hợp).

Với TwinLiteNet (tạm gọi là TwinLiteNetv1) tác giả chỉ sử dụng module ESP truyền thống từ ESPNet. Nhưng với **TwinLiteNet+** tác giả đã chuyển sang sử dụng **DESP (Depthwise ESP)**, module này giúp giảm tải nguyên tính toán đi rất nhiều so với module ESP truyền thống trước đó

### 2.2 PCAA module (Partial Class Activation Attention)

Chú ý vào các khu vực đặc trưng quan trọng trong ảnh.

Tăng khả năng tập trung vào khu vực làn đường và vùng có thể lái, giúp giảm nhiễu từ background.

## 3. Decoder

Sau khi ảnh đầu vào được đưa qua Encoder và PCAA module để trích xuất đặc trưng, features map lúc này sẽ được đưa qua decoder để đưa về kích thước ảnh ban đầu với kết quả là làn đường và đường đi được ở 2 khối decoder riêng biệt

Sử dụng kiến trúc phân tầng giúp mô hình học tốt các đặc trưng đa cấp:

- Đặc trưng biên, texture (tầng dưới).
- Hình dạng đối tượng như làn đường, vùng đường (tầng giữa và trên).

Skip connection từ encoder → decoder để phục hồi không gian chính xác trong

ảnh đầu ra.

#### 4. Loss function

Sử dụng tổ hợp **Focal Loss** và **Tversky Loss** để tối ưu phân đoạn trong điều kiện mất cân bằng lớp.

### KẾT QUẢ MONG ĐỢI

- Mô hình TwinLiteNet+ được huấn luyện hoàn chỉnh đạt  $mIoU \geq 92\%$  và  $IoU \geq 34\%$  trên tập validation.
- Latency của mô hình nhanh: Vì TwinLiteNet+ được đề xuất nhằm không trade off giữa độ chính xác và thời gian inference của mô hình nên chúng ta cũng sẽ có bảng so sánh thời gian inference của mô hình với từng batch dữ liệu, mô hình được đo trên A5000.
- Mô hình sau cải tiến kỳ vọng đạt độ chính xác cao hơn hoặc tốc độ suy luận tốt hơn.
- Bổ sung khả năng mở rộng cho các hệ thống perception thực tế.

	Paper(Epoch 100)		Our(Epoch 25)	
Model	Drivable Area	Lane	Drivable Area	Lane
	mIoU (%)	IoU (%)	mIoU (%)	IoU (%)
<b>TwinLiteNet+Nano</b>	87.3	23.3	87.0	23.0
<b>TwinLiteNet+Small</b>	90.6	29.3	89.0	27.0
<b>TwinLiteNet+Medium</b>	92.0	32.3	90.0	30.0
<b>TwinLiteNet+Large</b>	92.9	34.2	91.0	31.0

Bảng 1: Bảng so sánh kết quả từ bài báo và dự đoán

### TÀI LIỆU THAM KHẢO (Định dạng DBLP)

[1] Che, Quang-Huy; Le, Duc-Tri; Pham, Minh-Quan; Nguyen, Vinh-Tiep; Lam, Duc-Khai. TwinLiteNet+: A Stronger Model for Real-time Drivable Area and Lane Segmentation.

- [2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” 2018.
- [3] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” 2021.
- [4] D. Wu, M.-W. Liao, W.-T. Zhang, X.-G. Wang, X. Bai, W.-Q. Cheng, and W.-Y. Liu, “Yolop: You only look once for panoptic driving perception,” Machine Intelligence Research, pp. 1–13, 2022.
- [5] Q.-H. Che, D.-P. Nguyen, M.-Q. Pham, and D.-K. Lam, “Twinlitenet: An efficient and lightweight model for driveable area and lane segmentation in self-driving cars,” in 2023 International Conference on Multimedia Analysis and Pattern Recognition (MAPR), pp. 1–6, 2023.
- [6] S.-A. Liu, H. Xie, H. Xu, Y. Zhang, and Q. Tian, “Partial class activation attention for semantic segmentation,” in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16815–16824, 2022.
- [7] A. C. L. S. Sachin Mehta, Mohammad Rastegari and H. Hajishirzi, “Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation,” in ECCV, 2018.
- [8] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations, p. 240–248. Springer International Publishing, 2017.