Declaration of Originality

Unit Name: Advanced Artificial Intelligence Research Topics COMP6011

I/We declare that:

- The above information is complete and accurate.
- The work I am/We are submitting is entirely my/our own, except where clearly indicated otherwise and correctly referenced.
- I/We have taken (and will continue to take) all reasonable steps to ensure my/our work is not accessible to any other students who may gain unfair advantage from it.
- I/We have not previously submitted this work for any other unit, whether at Curtin University or elsewhere, or for prior attempts at this unit, except where clearly indicated otherwise.

I/We understand that:

- Plagiarism and collusion are dishonest and unfair to all other students.
- Detection of plagiarism and collusion may be done manually or by using tools (such as Turnitin).
- If I/We plagiarise or collude, I/We risk failing the unit with a grade of ANN ("Result Annulled due to Academic Misconduct"), which will remain permanently on my academic record. I/We also risk termination from my/our course and other penalties.
- Even with correct referencing, my/our submission will only be marked according to what I/we have done myself/ourselves, specifically for this assessment. I/We cannot re-use the work of others, or my/our own previously submitted work, in order to fulfil the assessment requirements.
- It is my/our responsibility to ensure that the submission is complete, correct, and not corrupted.

Date: April 4, 2025

Jain	**Hardik Ashish**	**21657791**	

REPORT 1: Boosting freight driver safety in Australia with advanced computer vision

Hardik Ashish Jain, *Master of Computing, 21567791*

Abstract—Recent advancements in semantic segmentation have led to the development of transformer-based architectures that surpass traditional convolutional models in both accuracy and generalisability. In this study, we benchmark the performance of six prominent segmentation models: BiSeNet, STDC, DDRNet, PIDNet, SegFormer and Mask2Former on three diverse datasets to evaluate their adaptability and robustness across varied visual domains. The first four models are state of the art in real time inference benchmarks and the last two provide better performance and overall generalisation capabilites, ideal to serve as foundation models. Our evaluation focuses on quantitative metrics (mIoU and mean pixel accuracy), measures computational efficiency (in FLOPS and FPS) and highlights qualitative observations of segmentation fidelity. While Mask2Former achieves the highest mIoU at 83.6% on average based on quantative performance, SegFormer demonstrates greater consistency across all datasets as demonstrated in qualatative analysis, making it a compelling candidate for real-world deployment. DDRNet achieves extremely fast inference at 108 frames per second. Additionally, we highlight the architectural trade-offs and data efficiency characteristics of each model, providing insights into their suitability for different application scenarios. The findings offer valuable guidance for researchers and practitioners seeking effective segmentation solutions under varying constraints. We also highlight that all models perform poorly when passed data from a different distribution such as Dark Zurich (accuracy drops to a quarter) and SydneyScapes (accuracy drops by third). They also are less accurate in identifying many important classes (such as rider, truck, motorcycle) which signifies more efforts are required for practical usability.

Abstract—An abstract in a research report is a concise summary that outlines the key aspects of your study, typically structured into several essential components. Begin with a clear problem statement that explains the research question or gap your work addresses. Next, briefly describe your methodology, including the techniques and approaches you used to gather and analyze data. Follow this with a summary of your main findings or results, highlighting the most significant insights or discoveries from your study. Conclude with a brief statement on the implications or contributions of your research, outlining its significance and potential impact. This structured format ensures that readers can quickly understand the purpose, methods, results, and importance of your research.
Convey the main points professionally. Final candidates identified.

Index Terms—Semantic segmentation, Vision transformers, Driver safety, real-time inference.

1 INTRODUCTION

THE Australian freight and logistics market is projected to grow from USD 89.78 billion in 2023 to USD 136.91 billion by 2032, exhibiting a growth rate (CAGR) of 4.50% as per a market research report published by IMARC[1]. There has also been a steady year-on-year rise in the volume of goods transported by the Australian freight business using roads, growing from 181.48 billion tonne kilometres (btkm) in 2010 to 222.94 btkm in 2020 as per Bureau of Infrastructure, Transport and Research Economics (BITRE)[2]. Comparatively, there has been negligible growth in freight business carried via air transport and through ports in the same decade, underlying the importance of land based transportation for future growth. However, BITRE freight performance report also showcases a worrying trajectory in the number of road fatalities, rising from 1,094 in 2020 to 1,128 in 2021[3]. The success of future freight business can thus be enhanced by not only targeting higher volume and revenue growth but also improving driver safety. Since settlements in Australia are mostly concentrated in few urban and regional hubs, freight drivers often travel long inter-state journeys for several hundreds of kilometres with little to no settlements. Challenges include: poor vision when travelling overnight or in poorly lit conditions [1], fatigue/ sleepiness/ exhaustion [2], sparse rest stops, poorly maintained infrastructure [3] and unforeseen weather conditions [4] to name a few.

While it would be difficult to deal with the challenges of infrastructure, enhancing safety with the use of modern computer vision methods is very much achievable and has been the topic of study for over two decades now [5], [6]. The initial studies mostly explored using Kalman filters for motion tracking purposes due to its lower computational requirement and better interpretability compared to artificial neural networks. Other approaches included local edge and corner feature detection algorithms such as SIFT, Canny, etc. It wasn't until 2012 that the use of GPUs for parallel computation enabled AlexNet, a deep convolutional neural network [7] to achieve state-of-the-art (SOTA) performance in the ImageNet Large Scale Visual Recognition Challenge (an image classification task). Subsequent improvements in computer vision model was achieved by VGGNet [8] which demonstrated benefits of

- *Hardik Jain is a Master of Computing, AI Major at Curtin University. E-mail: h.jain2@student.curtin.edu.au*

Report started March 1; submitted April 4.

1. https://www.imarcgroup.com/australia-freight-logistics-market
2. https://www.freightaustralia.gov.au/a-closer-look/freight-performance-dashboard
3. Data after 2021 hasn't been published by BITRE

uniform architecture and consistent filter size for deeper networks, ResNet which introduced skip connections and mitigated vanishing gradient problem [9], DenseNet which demonstrated feature reuse and reduced number of model parameters required [10] and Vision Transformers (ViTs) [11] which applied the transformer architecture originally developed for natural language processing tasks to image classification tasks, enhancing scalability and model generalisation. The achievements of these architecture is not just limited to classification task but has also been proven to be just as effective in object detection and semantic segmentation tasks. Fully Convolutional Network (FCN) introduced by Long et al. in 2015 [12] were groundbreaking because they adapted existing classification networks (like AlexNet, VGGNet) into fully convolutional networks that could produce pixel-wise predictions. Before FCNs, traditional methods for image segmentation often relied on sliding window approaches or patch-based methods, which were less efficient and less accurate. FCNs introduced the idea of upsampling layers (deconvolution layers) to produce dense predictions, which significantly improved the performance and efficiency of semantic segmentation tasks.

In this project, we would use three datasets: Cityscape test, Sydneyscape, and Dark Zurich to benchmark state of the art semantic segmentation models published in recent years. The models' performance would be evaluated using the mean Intersection over Union metric (mIoU, also called as Jaccard's index). For practical usage, besides accuracy, real-time performance on a video feed is also important. Thus, their computational complexity (in GFLOPs) and realtime-performance (in FPS) would also be reported.

2 Literature Review

2.1 Semantic Segmentation

Image segmentation is an extension of image classification from image level to pixel level that has three broad categories: instance, semantic, and panoptic segmentation (Figure 1). *Instance segmentation* involves identifying and delineating each object instance in an image, evaluated by metrics such as Average Precision (AP) and Intersection over Union (IoU). *Semantic segmentation* classifies each pixel into a predefined category without distinguishing between object instances, and is typically evaluated using mean Intersection over Union (mIoU) and mean Pixel Accuracy. Panoptic segmentation combines both instance and semantic segmentation, assigning each pixel to a specific object instance or a semantic class, and is assessed using the Panoptic Quality (PQ) metric, which considers both segmentation quality and recognition quality. In this project our focus would be on semantic segmentation where classification of pixel into a predefined category is the objective, allowing many downstream applications such as enhancing driver safety (for our case). The first widely adopted neural network approach for semantic segmentation was the Fully Convolutional Network (FCN) [12], which performs pixel-to-pixel classification in an end-to-end manner. Subsequent research aimed to enhance FCN by various means, such as: 1) expanding the receptive field [13], [14], [15], 2) improving contextual information [16], [17], [18], 3) incorporating boundary information [19], [20], [21], 4) developing diverse

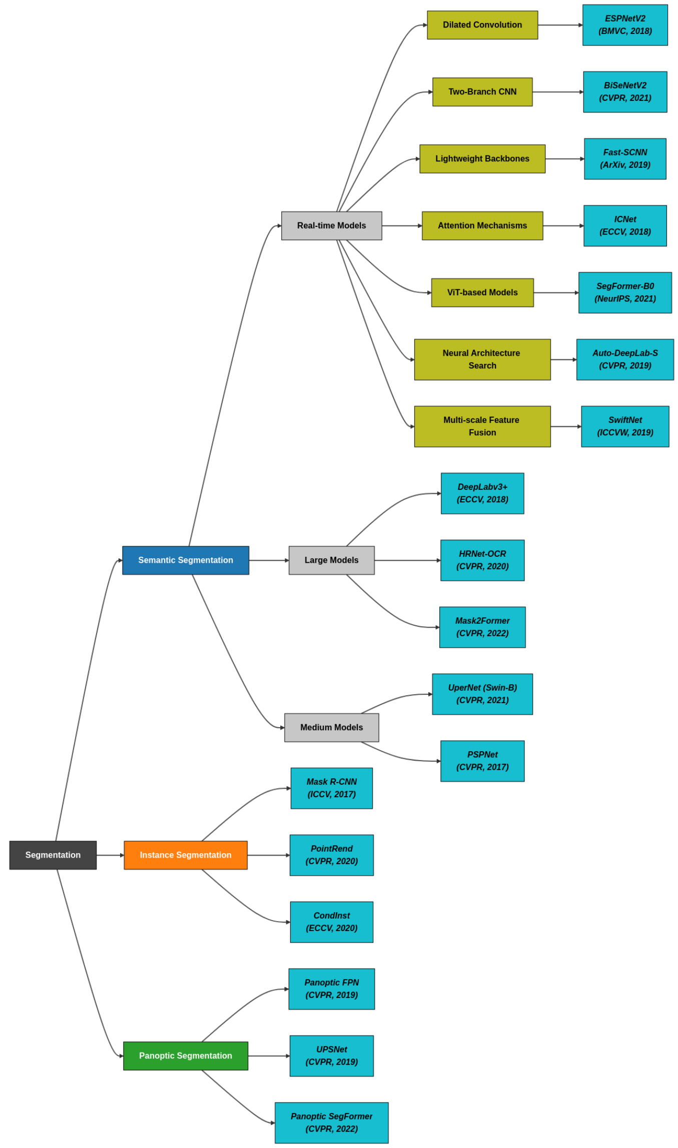

Fig. 1. Hierarchical taxonomy of state-of-the-art segmentation models. This diagram presents a detailed breakdown of the segmentation landscape, starting from the root category of image segmentation and expanding into semantic, instance, and panoptic subtypes. Semantic segmentation is further divided into large, medium, and real-time model classes, with real-time models organised by architectural strategies such as dilated convolutions, attention mechanisms, ViTs, and NAS. For each approach, the top-performing model on the Cityscapes benchmark is highlighted, with the corresponding publication venue and year indicated. This structured overview helps new researchers navigate the segmentation model space efficiently. The figures was created by us using Mermaid library and a high resolution version can be made accessible.

attention modules [22], [23] and 5) employing AutoML technologies [24], [25], [26]. Although these advancements have significantly boosted semantic segmentation performance, they introduced numerous empirical modules, resulting in a more computationally intensive and complex framework. More recently the focus has been on designing architectures which can give high accuracy and perform real-time inference as shown in Figure 2.

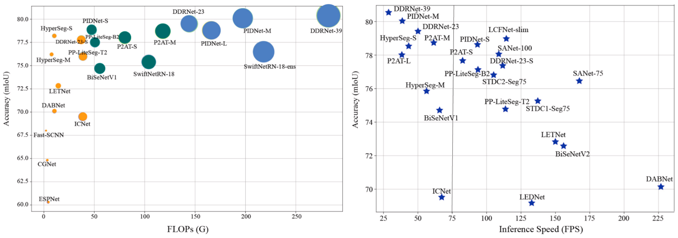

Fig. 2. There exists a trade-off between a model's inference speed and accuracy with preference for models which can consistently deliver higher performance with minimum reduction in inference speed. Some of the most notable semantic segmentation models were reviewed by Elhassan et al. [27] and presented in this figure.

2.2 Popular approaches

2.2.1 *Dilated convolution*

An increase in convolution network's kernel size increases the number of parameters which was a constraint faced by many early semantic segmentation works such as FCN [12], SegNet [28] and U-Net [29]. These designs for semantic segmentation were based on an encoder-decoder architecture where the encoder expands its receptive field through strided convolutions or pooling operations, while the decoder restored detailed information from high-level semantics using deconvolutions or upsampling. However, this downsampling process often led to the loss of spatial details in encoder-decoder networks which was addressed by dilated convolution operation as shown in equation 1. DeepLab [30] significantly improved previous works by incorporating this operation with varying dilation rates r within the network. However, as the operation requires non-contagious memory accesses, it wasn't ideal for hardware implementation despite its theoretical strengths. PSPNet [13] introduced a pyramid pooling module which significantly overcame these bottlenecks associated with dilated convolution while being able to parse multi-scale context information.

$$y[i] = \sum_{k=0}^{K} x[i + r.k].W[k] \tag{1}$$

where, i represents a specific pixel in a 1D signal, y represents the 1D output, x represents the 1D input, K represents the kernel size, r represents the convolution dilation rate.

2.2.2 *Light weight encoder and decoder*

Following the success of PSPNet, further approaches mostly concentrated on designing light weight encoder-decoder block such as SwiftNet [31] which used two inputs as shown in Figure 3. The first is a low-resolution input which provides high-level semantics to the encoder and another is a high-resolution input which offers sufficient details to its lightweight decoder. Other approaches include designs on light-weight backbones for existing architectures, reduced input size for faster inference without losing accuracy [32]. Since these networks use an encoder-decoder architecture, there is a high latency caused by deep information flow. Additionally, traditional convolution is faster than depth-wise separable convolution on GPUs, prompting the search for more efficient models without convolution factorisation and encoder-decoder structures.

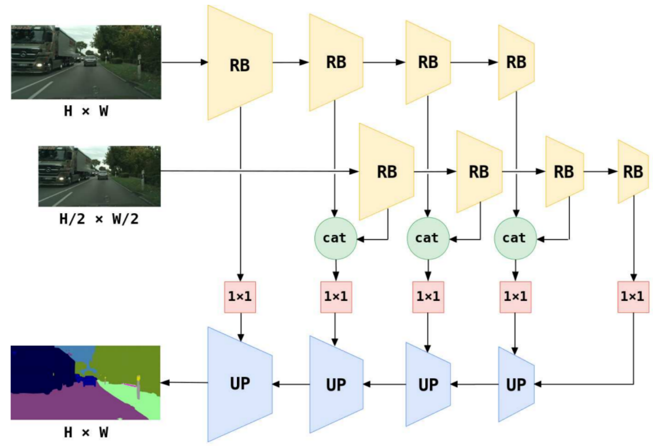

Fig. 3. Structural diagram of SwiftNet which introduces the concept of pyramidal fusion [31]. Encoder parameters (yellow) are shared across all pyramid levels and pre-trained on ImageNet. Features of identical resolutions are concatenated (green circles), passed through a 1×1 bottleneck convolution (red squares), and integrated within the decoder (blue). Image sourced from SwiftNet paper.

2.2.3 *Two branch network*

Another mechanism besides pyramid fusion which gained popularity is two branch network (TBN). Here, the idea involves capturing contextual dependency by a large receptive field, as spatial details are crucial for boundary delineation and small-scale object recognition. BiSeNet [33] proposed the TBN architecture, involving two branches with different depths for context embedding and detail parsing. May works inspired from it attempted enhancing at reducing its complexity such as DDRNet which introduced bilateral connections for improving information exchange between the context and detailed branches. A further optimisation to this was made by Xu et al. [34] who highlight the problems associated with fusing detailed semantics with context information as shown in Figure 6 (blurring of boundaries by surrounding pixels).

2.2.4 *Vision Transformer*

Transformer architecture proposed in 2017 by Vaswani et al. for machine translation achieved superior performance and demonstrated removal of sequential components such as RNNs and CNNs improves scalability as model architecture gets simplified and solely relies on attention mechanism [35]. Following the massive success of transformer architecture in natural language processing tasks due to the development of large language models, attempts were made to employ the architecture in computer vision as well. Since the attention mechanism requires sequence modelling, Dosovitskiy et al. in 2020 proposed tokenizing an image by splitting it into 16x16 blocks and then passing it to the transformer architecture, an approach now popularly known as Vision Transformer (ViT) [11]. ViT was the first approach demonstrating that a pure Transformer could achieve SOTA performance in image classification by treating each image as a sequence of tokens and feeding them through multiple Transformer layers. Following this, DeiT [36] explored a data-efficient training strategy and a distillation approach for ViT. More recent methods, such as Token to Token ViT [37], CrossViT [38], and LocalViT [39], introduced specific modifications to further enhance image classification performance.

Beyond classification, Pyramid ViT [40] was the first to incorporate a pyramid structure in Transformers, showcasing the potential of a pure Transformer backbone over CNN counterparts in dense prediction tasks. Subsequent methods like Swin [41] and Twins [42] improved the local continuity of features and eliminated fixed-size position embeddings to enhance Transformer performance in dense prediction tasks. For semantic segmentation, SETR [43] utilised ViT as a backbone to extract features, achieving impressive results. However, these Transformer-based methods tend to have low efficiency, making them challenging to deploy in real-time applications. More recent methods such as SegFormer by NVIDIA [44] and Mask2Former by Meta [45] have proved the effectiveness of Transformer-based architectures for semantic segmentation and used as foundation models for many derivative works improving performance such as VLTSeg which currently ranks highest amongst the open source models on City Scape benchmark [46]. However, since most benchmark leaders on CityScape[4] are computationally demanding, we would instead focus on SOTA models for real-time inference and evaluate their performance[5].

2.3 Models evaluated

2.3.1 BiSeNet

Bilateral Segmentation Network (BiSeNet) demonstrated that real-time inference can be achieved for semantic segmentation task without compromising on spatial resolution which was popular until then [33]. The design includes a spatial path with a small stride which preserves the spatial information and generates high-resolution features. Meanwhile, a context path with a fast down sampling strategy is employed to obtain sufficient receptive field. On top of these two paths, there exists a Feature Fusion Module for combining features efficiently. The use of two branch network design in real time semantic segmentation tasks was popularised after BiSeNet demonstrated its results.

2.3.2 STDC

A significant limitation of BiseNet [33] is its use of additional path specifically for encoding spatial information. Although this extra path is beneficial for capturing spatial details, it introduces significant computational overhead, making the network slower and less efficient for real-time applications. Moreover, BiSeNet often relies on backbones that are pre-trained on tasks such as image classification. These backbones, are not always optimized for the specific requirements of image segmentation which can lead to inefficiencies and suboptimal performance because the design of these backbones does not cater to the unique challenges of segmentation tasks, such as the need for precise boundary delineation and the handling of small-scale objects. To address these issues, Fan et al. [47] proposed a novel and efficient network structure called the Short-Term Dense Concatenate (STDC) network. The STDC network aims to eliminate structural redundancies found in traditional networks. One of the key innovations in the STDC network is its gradual reduction of feature map dimensions. By progressively decreasing the size of the feature maps and aggregating them, STDC creates a compact and efficient representation of the image (Figure 4). This aggregation process forms the basic module of the STDC network, ensuring that important spatial and contextual information is retained without unnecessary complexity. In the decoder part of the STDC network, a Detail Aggregation module is implemented which integrates the learning of spatial information directly into the low-level layers of the network. Finally, the STDC network combines low-level features, which capture fine details, with deep features, which encapsulate high-level semantic information. This fusion of low-level and deep features allows the network to predict the final segmentation results accurately.

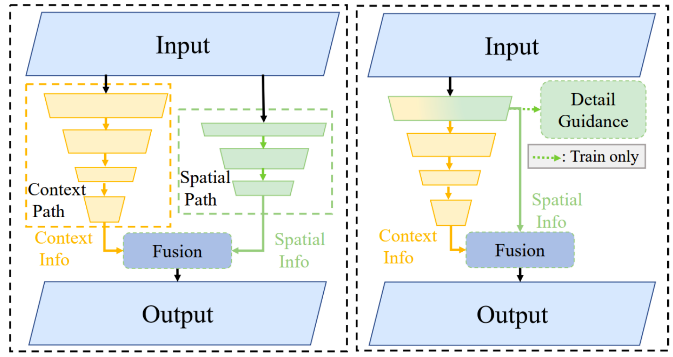

Fig. 4. The architectures of BiSeNet [33] and STDC [47] are displayed from left to right. BiSeNet employs an additional spatial path to encode spatial information, whereas the STDC model uses a detail guidance module to encode spatial information in low-level features, avoiding an extra time-consuming path. Image sourced from STDC paper.

2.3.3 DDRNet

Deep dual-resolution networks (DDRNet) by Hong et al. [48] was the first architecture which outperformed dilated convolution based models while permitting real-time inference. The proposed design consisted of two deep branches between which multiple bilateral fusions were performed using a new contextual information extractor named Deep Aggregation Pyramid Pooling Module (DAPPM). This extractor helped enlarging the receptive field and fusing multi-scale context based on low-resolution feature maps. As shown in Figure 5, DDRNet start from one trunk and then divide into two parallel deep branches with different resolutions. One deep branch generates relatively high-resolution feature maps and the other extracts rich semantic information through multiple downsampling operations. Multiple bilateral connections are bridged between two branches to achieve efficient information fusion. Then, a DAPPM is added to the output of the low-resolution branch, which extracts rich contextual information from the high-level feature maps of 1/64 image, using large pooling

4. https://paperswithcode.com/sota/semantic-segmentation-on-cityscapes
5. https://paperswithcode.com/sota/real-time-semantic-segmentation-on-cityscapes

TABLE 1
A list of all the models which will be evaluated in this study. Each model has multiple variants say small, base, large. We have considered multiple variants for each model which offer real-time inference. The mIoU and FLOPS are taken from the respective paper for the smallest variant which might not have the highest performance but has the fastest inference on CityScapes validation set. Later during our benchmarking we will compute and present the metrics for each variant of these models.

Model name	Venue	Variants considered	Backbone	mIoU	FLOPS
BiSeNetv2 [33]	ECCV 2018	1	XCeption, ResNet	72.0	2.9G
STDC [47]	CVPR 2021	3	CNN (custom)	77.0	-
DDRNet [48]	arXiv preprint 2021	3	Dual resolution network	77.8	36.3G
PIDNet [34]	CVPR 2023	3	ResNet	78.8	46.3G
SegFormer [44]	NeurIPS 2021	3	Hierarchical Transformer Encoder	76.2	125.5G
Mask2Former [45]	CVPR 2022	3	Swin transformer, ResNet	79.4	-

kernels with exponential strides and generates feature map of 1/128, 1/256 and 1/512 image resolutions. These multi-scale contextual information is then upsampled, followed by 3×3 convolutions to fuse contextual information of different scales in a hierarchial-residual way.

2.3.4 PIDNet

In 2023, Xu et al. [34] demonstrated that the two branch networks architecture which is used in most real-time state-of-the-art semantic segmentation models is equivalent to a proportional integral (PI) controller used in classical control systems as shown in Equation 2. Similar to PI, these architectures suffer from overshoot issues as the high resolution details are often overwhelmed by low frequency surrounding context. They thus propose PIDNet, which brings in a derivative control to alleviate the overshooting issue as shown in Figure 6 and is the currently highest ranked SOTA model when it comes to real-time inference on Cityscapes semantic segmentation. A P controller focuses on the current signal, an I controller accumulates past signals causing potential overshoot, and a D controller dampens overshoot, while TBNs use multiple convolutional layers to parse context and details.

$$c_{out}[n] = k_p e[n] + k_i \sum_{i=0}^{n} e[i] \tag{2}$$

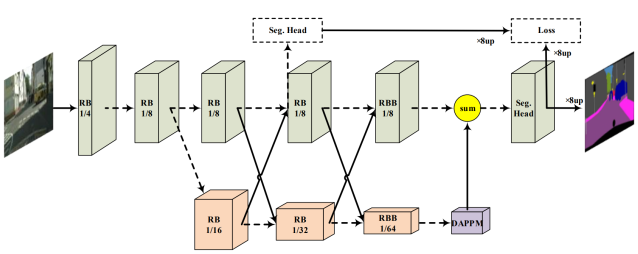

Fig. 5. Fig. 4. Overview of DDRNets for semantic segmentation: RB are sequential residual basic blocks, RBB is a single residual bottleneck block, DAPPM is the Deep Aggregation Pyramid Pooling Module, and Seg. Head is the segmentation head. Solid lines indicate data processing paths, dashed lines indicate paths without processing, and dashed boxes are components discarded during inference. Image sourced from DDRNet paper.

2.3.5 SegFormer

SegFormer [44] is a ViT based architecture for semantic segmentation, combining Transformers with lightweight MLP decoders. It features a novel hierarchical Transformer encoder that outputs multiscale features without needing positional encoding, avoiding performance drops when testing resolution differs from training. Additionally, it uses a simple All-MLP decoder, avoiding complex and computationally heavy modules. Given an image of size $H \times W \times 3$, it is divided into patches of size 4×4, which are input to the hierarchical Transformer encoder to obtain multi-level features at $\{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$ of the original image resolution. These features are then passed to the All-MLP decoder to predict the segmentation mask at a $H/4 \times W/4 \times N_{cls}$ resolution, where N_{cls} is the number of categories (Figure 7). The main computational bottleneck of the encoders is the self-attention layer. In the original multi-head self-attention process, each head Q, K, V has dimensions $N \times C$, where $N = H \times W$ is the sequence length. The self-attention is calculated as shown in equation 3. The computational complexity is $O(N^2)$, which is prohibitive for large resolutions. Instead, a sequence reduction process is used, reducing the sequence length by a ratio R as shown in equations 4 and 5. This reduces the self-attention complexity from $O(N^2)$ to $O(N^2/R)$. In experiments, R is set to $[64, 16, 4, 1]$ from stage-1 to stage-4 as shown in Figure 8.

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_{\text{head}}}}\right) V \tag{3}$$

$$\hat{K} = \text{Reshape}\left(\frac{N}{R}, C \cdot R\right)(K) \tag{4}$$

$$K = \text{Linear}(C \cdot R, C)(\hat{K}) \tag{5}$$

2.3.6 Mask2Former

Mask2Former [45] stands for Masked Attention Mask Transformer and was designed to address any image segmentation task, including panoptic, instance, and semantic segmentation. Most models aim at specialising in one particular task but this model provides a unified approach to all three segmentation tasks. Mask2Former's key components include masked attention, which extracts localised features by constraining cross-attention within predicted mask regions (Figure 9). This approach reduces research effort and outperforms specialised architectures on multiple datasets. The architecture consists of three main components: Backbone for extracting low-resolution features from an image, Pixel Decoder for upsampling low-resolution features to generate high-resolution per-pixel embeddings and Transformer Decoder which processes object queries on image features to predict segmentation masks. Given an image, the pixel decoder generates multi-scale feature maps, which are fed into the Transformer decoder. The masked attention operator in the Transformer decoder extracts localised features by focusing on the foreground region of the predicted mask for each query. It follows similar strategy of Segformer for reducing computational complexity (equations 3 to 5). Mask2Former adopts a multi-scale deformable attention Transformer (MSDeformAttn) as the default pixel decoder, using 6 MSDeformAttn layers applied to feature maps with resolutions $\frac{1}{8}, \frac{1}{16}, \frac{1}{32}$, and a simple upsampling layer to generate the feature map of resolution $\frac{1}{4}$. The final binary mask predictions are decoded from per-pixel embeddings with object queries. The loss function combines binary cross-entropy loss and dice loss as shown in equation 6 and the final loss is a combination of mask loss and classification loss (equation 7). Mask2Former demonstrates strong performance across different segmentation tasks, providing a versatile and efficient solution for image segmentation.

$$L_{\text{mask}} = \lambda_{\text{ce}} L_{\text{ce}} + \lambda_{\text{dice}} L_{\text{dice}} \tag{6}$$

$$L_{\text{final}} = L_{\text{mask}} + \lambda_{\text{cls}} L_{\text{cls}} \tag{7}$$

2.4 Datasets

2.4.1 *Cityscapes*

The Cityscapes dataset [49] is designed for semantic understanding of urban street scenes. It includes 5,000 finely annotated images and 20,000 coarsely annotated images. These can be downloaded from their website[6] after registration. The photos were captured in 50 different cities in Germany across various months (spring, summer, fall) in daytime. The dataset was created in 2015 with each image having high resolution of 2048x1024 pixels. The dataset is divided into training, validation, and testing sets, with 2,975 finely annotated images for training, 500 for validation, and 1,525 for testing. Since the semantic annotation for testing set is available only with the cityscapes team, we have performed our tests on these 500 finely annotated validation set. There are 30 classes in the dataset, but only 19 of these classes are used for evaluation purposes. The dataset supports various tasks such as pixel-level, instance-level, and panoptic semantic labelling.

6. https://www.cityscapes-dataset.com

2.4.2 *Dark Zurich*

A notable limitation of cityscapes is the lack of data in adverse weather conditions or lowlight conditions. The Dark Zurich dataset [50] was designed for tasks such as semantic segmentation in varying lighting conditions (nighttime, twilight, and daytime). It contains a total of 151 images captured at night time, each with a resolution 1920x1080 pixels. Each image is annotated with GPS coordinates, allowing for cross-time-of-day correspondences, meaning each nighttime or twilight image can be matched to its daytime counterpart. The images were captured in Zurich, Switzerland, and the dataset was introduced around 2019. The dataset supports various tasks, including domain adaptation, image translation, and robust image matching. It uses the same 19 classes as are used in Cityscapes evaluation.

2.4.3 *Sydneyscapes*

The SydneyScapes dataset [51] offers street level semantic segmentation in Australian landscape. It consists of 756 high-quality pixel-level annotated images. The images were captured in Sydney and surrounding cities in New South Wales. This dataset aims to support the development and testing of autonomous vehicle perception systems and other computer vision applications. Sydneyscapes also uses the same 19 classes that were used in Cityscapes evaluation and tries to provide a good mix of scenes in daytime and night time conditions.

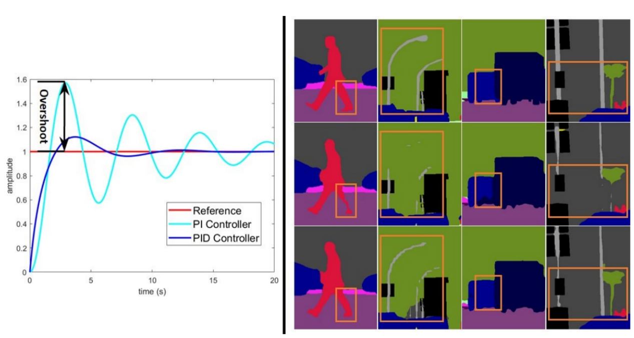

Fig. 6. Step responses of PI and PID controllers for a second-order system are shown on the left, highlighting overshoot issue for dynamic system. On the right, the images (top to bottom) represent ground truth, outputs of DDRNet-23 (a two branch network) [48] and ADB-Bag-DDRNet-23 (a reimplementation by Xu et al. [34] with PIDNet architecture), respectively. Image sourced from the original paper.

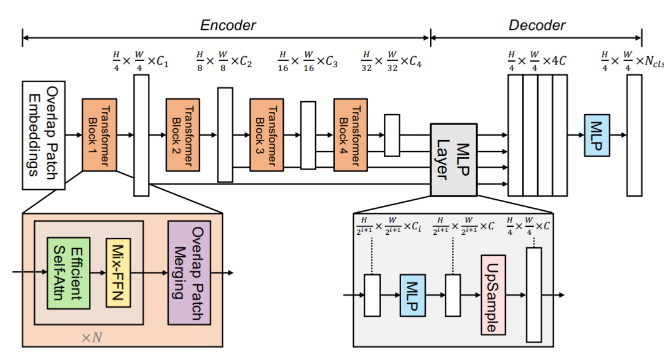

Fig. 7. SegFormer consists of two main modules: a hierarchical Transformer encoder to extract coarse and fine features; and a lightweight All-MLP decoder to directly fuse these multi-level features and predict the semantic segmentation mask. Image derived from SegFormer paper [44].

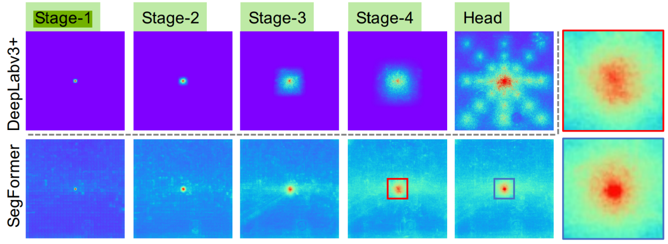

Fig. 8. Effective Receptive Field of DeepLabv3+ (top) compared against SegFormer (bottom) on Cityscapes dataset. Image derived from SegFormer paper [44].

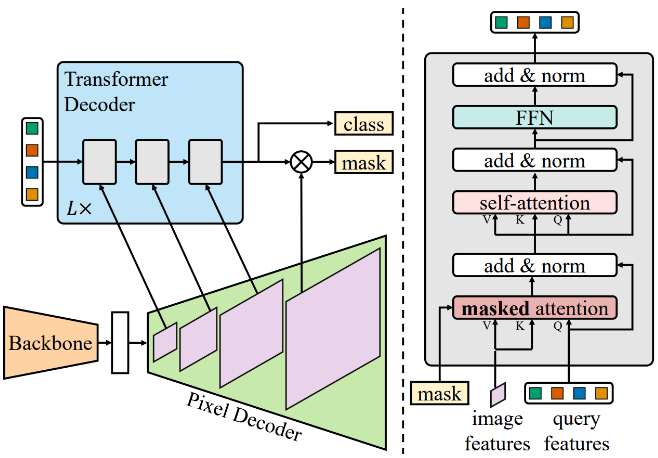

Fig. 9. Mask2Former propose a new Transformer decoder with masked attention instead of the standard cross-attention for dealing with small objects. It also switches the order of self and cross-attention for making query features learnable, and removes dropout to make computation more effective. Image derived from Mask2Former paper [45].

3 Benchmarking Results

3.1 Metrics

We evaluate the performance of various semantic segmentation methods guided by two performance metrics and three metrics representing computational efficiency. The following metrics are used to provide a comprehensive assessment.

3.1.1 Mean Intersection over Union (mIoU)

Mean Intersection over Union (mIoU) is a common evaluation metric for semantic segmentation tasks. It measures the average overlap between the predicted segmentation and the ground truth across all classes. The mIoU is calculated as follows:

$$\text{mIoU} = \frac{1}{N}\sum_{i=1}^{N}\frac{|P_i \cap G_i|}{|P_i \cup G_i|} \tag{1}$$

where N is the number of classes, P_i is the predicted set for class i, and G_i is the ground truth set for class i.

3.1.2 Mean Pixel Accuracy (mAcc)

Mean Pixel Accuracy (mAcc) evaluates the proportion of correctly classified pixels over the total number of pixels. It is defined as:

$$\text{mAcc} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i + FN_i} \tag{2}$$

where TP_i, FP_i, and FN_i represent the true positives, false positives, and false negatives for class i, respectively.

3.1.3 Frames Per Second (FPS)

Frames Per Second (FPS) measures the computational efficiency of the segmentation model. It indicates how many frames the model can process per second, reflecting its real-time performance capability. Higher FPS values are desirable for applications requiring real-time processing.

3.1.4 Floating Point Operations Per Second (FLOPS)

Floating Point Operations Per Second (FLOPS) is a measure of the computational complexity of the model. It quantifies the number of floating-point operations the model performs per second. Lower FLOPS values indicate more efficient models, which are preferable for deployment on resource-constrained devices.

3.1.5 Number of parameters

Number of parameters represents the complexity of the model with larger model requiring more amount of memory which can be a constraint when deploying on devices with limited computational resources.

3.2 Ablation Study

3.2.1 Results on CityScapes

The class wise IoU and pixel accuracy for the six models under consideration are presented in Tables 2 and 3. We split it into two tables due to space constraints. We can observe exceptionally high accuracy ($> 90\%$) when segmenting classes of road, building, sky, vegetation which form the background during road travel but alarmingly poor accuracy across classes rider, truck, motorcycle which does lead to the question of practical feasibility of deploying such systems. On a good note, other classes such as car, bicycle, person are segmented with sufficiently good accuracy. The mean IoU, mean pixel accuracy, number of parameters, FLOPS, FPS is presented in Table 4. Since, Mask2Former variants which were chosen had the highest number of parameters compared to other models, it isn't surprising that it performs better across the board.

3.2.2 Results on Dark Zurich

The results for Dark Zurich are presented in Table 5 and showcase a worrying reduction in model's performance. While mIoU reduces by half, mean pixel accuracy reduces by almost 75% which shows that models do not handle low light conditions effectively and need significant training for improving adaptability to such scenarios. The qualitative analysis shown in Figure 10 shows decent accuracy for few models such as SegFormer (shown) and Mask2Former. We note that Mask2Former and SegFormer provide good performance at 74.2% and 58.5% overall but poor performance on classes involving humans (rider, person).

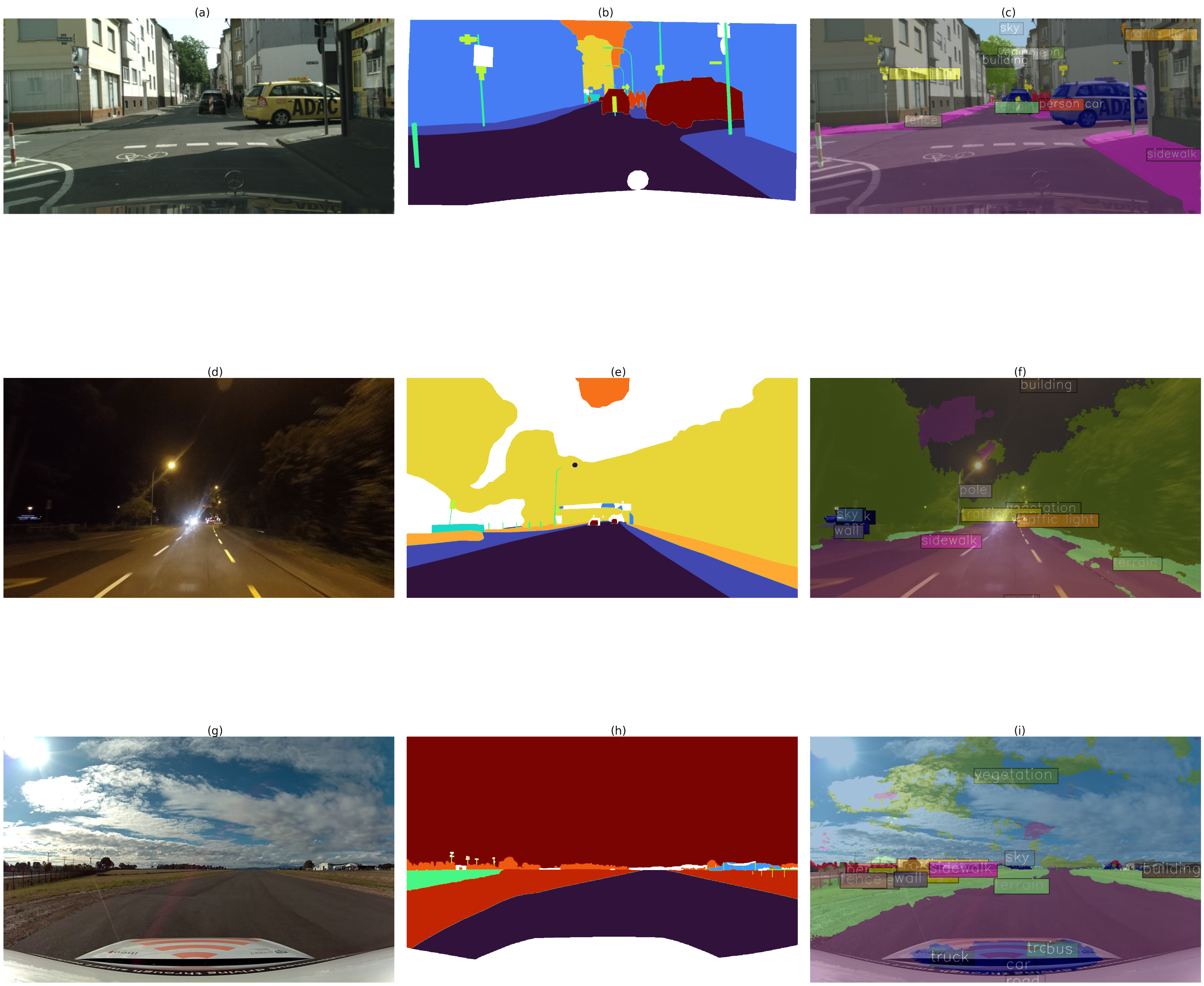

Fig. 10. Qualitative results across the three datasets for SegFormer MiT-b0 model are showcased. We observe near accurate predictions for cityscape and slightly inaccurate results for Dark Zurich and Sydneyscape. From Top to Bottom, each row represents one image from CityScapes (a, b, c), Dark Zurich (c, d, e) and Sydney Scapes(d, e, f). From left to right, the columns are: Original image, segmented image as in the dataset, segmented predictions by the model overlayed on the original image for quick comparison.

TABLE 2
Performance of different SOTA models on the classes 1 to 10 for the CityScape validation dataset.

Model	Metric	Classes									
		road	pole	bus	traffic sign	fence	wall	rider	building	car	sky
BiseNetV2 FCN	IoU	97.96	60.79	74.31	75.12	56.71	50.53	54.46	91.54	94.14	94.45
	Acc	98.91	72.21	90.03	83.8	66.81	57.8	66.73	96.22	97.11	97.98
STDC1	IoU	97.63	55.04	81.19	72.31	52.06	48.11	54.18	90.78	93.1	93.47
	Acc	98.62	66.53	86.56	81.1	62.62	52.26	68.68	95.8	97.29	97.79
STDC1-pre	IoU	97.97	58.19	85.8	76.74	56.19	42.36	57.99	91.42	94.33	94.2
	Acc	98.98	70.51	90.46	85.21	63.24	46.22	71.13	96.33	97.71	97.7
STDC2-pre	IoU	98.26	61	86.62	77.93	61.04	51.3	62.51	92.23	94.69	94.42
	Acc	99.09	72.4	92.09	86.57	69.57	55.6	75.8	96.42	97.72	97.88
DDRNet	IoU	98.27	66.83	90.19	80.87	64.25	55.25	66.98	93.07	95.48	95.24
	Acc	98.99	78.62	93.87	88.12	76.01	60.86	81.07	96.81	97.93	98.24
PIDNet S	IoU	98.31	65.14	88.03	79.06	60.59	54.39	64.98	92.71	95.32	94.64
	Acc	99.1	77.51	91.78	87.24	72.4	60.85	79.62	96.49	97.74	97.83
PIDNet B	IoU	98.24	66.33	90.6	80.49	64.71	57.93	67.53	93.1	95.58	94.8
	Acc	99.05	78.67	94.43	88.97	73.95	64.65	80.4	96.8	97.96	97.81
PIDNet L	IoU	98.34	67.26	92.42	82.11	65.64	64.41	68.3	93.4	95.62	94.92
	Acc	99.11	79.49	95.62	89.43	74.64	70.49	81.56	96.86	97.83	98.05
SegFormer MiT-b0	IoU	98.03	62.6	83.88	77.72	56.82	58.76	57.39	92.23	94.33	94.98
	Acc	98.95	72.51	89.76	84.88	66.19	65.28	69.63	96.73	97.59	98.22
MiT-b1	IoU	98.2	65.94	86.71	80.77	57.97	62.56	60.46	92.82	95.24	95.19
	Acc	98.99	76	93.25	87.5	65.01	69.22	72.11	96.9	97.84	98.26
MiT-b2	IoU	98.38	68.63	91.33	81.78	62.61	66.17	65.61	93.36	95.63	95.36
	Acc	99.12	78.95	95.45	89.01	70.24	72.39	78.56	96.93	98.01	98.25
Mask2Former r101	IoU	98.17	69.53	90.9	81.05	64.55	58.36	67.15	93.15	95.73	95.29
	Acc	98.96	79.85	95.36	87.81	75	65.37	77.51	97.08	98.04	98.29
swin b	IoU	98.53	72.2	92.08	**85.01**	**74.46**	66	69.81	**94.17**	**96.3**	**95.74**
	Acc	99.1	82.45	96.24	**90.89**	**85.28**	77.27	81.68	97.17	**98.14**	98.64
swin l	IoU	**98.61**	**72.44**	**92.71**	84.43	68.91	**68.96**	**72.29**	94.02	96.23	95.69
	Acc	**99.25**	**83.16**	**96.84**	90.58	77.08	**77.41**	**83.59**	**97.28**	98.13	**98.69**

3.2.3 Results on SydneyScapes

The results for SydneyScapes are presented in Table 6. While the performance of the models decreases we believe this is a result of poor quality of images in SydneyScapes compared to CityScapes. So some finetuning should improve model performance significantly. The qualitative analysis shown in Figure 10 shows decent accuracy for few models such as SegFormer (shown). Again, Mask2Former and SegFormer provide good performance overall but poor performance on classes involving humans (rider, person). Surprisingly, their performance though is lower than that on Dark Zurich. We have visualised the mean pixel accuracy which varies widely across the three datasets in a figure 11 and analyse it in combination with model effeciency (FLOPS and number of parameters).

4 Methodology

4.1 Experimental setup

The experiments were performed using `mmsegmentation` framework [52] which is a popular toolbox providing high level APIs to the most popular segmentation models and wrappers for most of the available datasets. The use of `mmsegmentation` significantly saved time but its setup is time consuming as active development has mostly paused for the framework. We used a `mmsegmentation` docker image[7] and created our own container locally. Notebooks associated for reproducing our results are available in Appendix.

4.2 Selection of models

Since our primary objective involves the use of semantic segmentation models for improving truck driver safety, our models should be capable of real-time inference. While the current benchmark leaders in CityScapes all showcase impressive accuracy they are very large models and not a good fit for this problem. A truck will have limited on-board computational capabilities offered by Jetson Nano, Jetson Xavier, and other devices which motivated us to look for models which give impressive performance and are also capable of real-time inference. This led to the selection of BiSeNet [33] which was the first model to achieve real-time inference while maintaining spatial resolution. Subsequent models such as STDC [47] and DDRNet [48] also leveraged two branch network popularised by BiSeNet to further improve performance. Later, PIDNet [34] resolved the overshooting

7. Docker hub image link. Thanks Sonny for sharing this.

TABLE 3
Performance of different SOTA models on the classes 11 to 19 for the CityScape validation dataset.

Model	Metric	Classes								
		traffic light	train	sidewalk	terrain	person	bicycle	vegetation	truck	motorcycle
BiseNetV2 FCN	IoU	67.22	61.5	83.39	61.79	78.89	73.83	91.99	66.02	53.72
	Acc	79.1	64.92	91.54	70.32	89.55	86.06	96.57	72.01	65.22
STDC1	IoU	61.95	74.01	81.43	59.85	77.03	73.24	91.1	55.47	52.56
	Acc	72.93	77.47	91.29	69.36	89.11	84.7	96.37	60.22	59.96
STDC1-pre	IoU	67.97	75.56	83.76	62.4	79.18	75.51	91.71	70.97	61.54
	Acc	78.92	79.87	91.47	72.04	90.49	86.66	96.47	73.97	73.12
STDC2-pre	IoU	69.7	77.99	85.18	59.28	80.57	76.86	91.87	70.84	64.54
	Acc	81.63	84.29	92.44	70.11	91.24	87.64	96.62	75.08	74.23
DDRNet	IoU	73.51	83.78	86.07	66.48	83.4	78.28	92.72	81.57	67.63
	Acc	85.62	88.69	93.15	76.42	91.86	90.08	96.63	86.75	79.16
PIDNet S	IoU	71.48	83.29	85.89	66.56	81.75	77.25	92.35	82.06	62.33
	Acc	83.57	89.78	92.51	76.35	90.99	89.78	96.56	90.35	71.88
PIDNet B	IoU	74.17	84.51	85.63	65.35	83.32	78.93	92.68	84.44	65.78
	Acc	86	88.44	92.58	74.51	92.06	90.02	96.64	88.88	76.02
PIDNet L	IoU	74.32	84.34	86.21	64.88	84.19	79.25	92.76	81.81	66.71
	Acc	86.41	88	93.08	74.24	92.28	90.27	96.63	90.92	78.43
SegFormer MiT-b0	IoU	69.83	75.66	84.03	64.13	80.72	76.27	92.55	69.36	64.93
	Acc	81	81.34	91.92	73.71	90.19	87.73	96.64	73.78	77.38
MiT-b1	IoU	72.73	70.92	85.36	65.12	82.36	78.05	92.89	82.16	67.18
	Acc	83.97	74.82	93.07	73.97	91.68	89.01	96.86	86.51	78.15
MiT-b2	IoU	74.85	82.32	86.45	63.47	84.16	79.44	93.06	85.97	71.86
	Acc	85.91	86.36	93.28	71.91	92.25	90.29	97.05	89.19	82.32
Mask2Former r101	IoU	75.12	82.61	85.02	63.32	84.38	80.07	92.99	87.87	69.96
	Acc	85.38	86.53	93.17	74.11	91.64	89.74	96.42	92.02	80.21
swin b	IoU	77	**85.22**	87.9	67.49	85.9	81.88	**93.4**	**90.05**	73.81
	Acc	86.51	**88.33**	94.59	**78.19**	92.72	**90.86**	96.63	**94.28**	83.69
swin l	IoU	**77.37**	83.86	**88.74**	**67.74**	**86.58**	**82.09**	93.36	89.94	**75.4**
	Acc	**87.79**	88.05	**94.64**	76.62	**93.15**	90.77	**96.64**	92.97	**85.95**

TABLE 4
Performance and computational efficiency of the models evaluated on CityScapes. FPS was derived from respective paper as it was difficult to reproduce similar results as the one presented in the paper.

Model	mIoU↑	mAcc↑	FLOPS↓	Params↓	FPS↑
bisev2 fcn	73.07	81.2	98.9G	3.4M	47.3
stdc1	71.82	79.4	68G	8.3M	74.8
stdc1-pre	74.94	82.13	68G	8.3M	74.8
stdc2-pre	76.67	84.02	94G	12.3M	58.2
ddrnet	79.99	87.31	144G	20.3M	108.1
pidnet-s	78.74	86.44	48G	7.7M	93.2
pidnet-m	80.22	87.25	178G	28.8M	39.8
pidnet-l	80.89	88.07	276G	37.3M	31.1
seg mit-b0	76.54	83.86	122G	3.7M	15.2
seg mit-b1	78.56	85.43	240G	13.7M	11.2
seg mit-b2	81.08	87.66	421G	24.7M	7.4
m2f r101	80.8	87.5	-	63M	-
m2f swin-b	83.52	90.14	-	107M	-
m2f swin-l	83.65	89.93	-	216M	-

TABLE 5
Performance on Dark Zurich validation set. Models are trained on CityScapes data. We note the reduction in performance between the two datasets in parenthesis.

Model	Overall		Class	
	mIoU ↑	mAcc ↑	rider ↑	person ↑
bisev2 fcn	46.0 (-30.6)	9.9 (-74.1)	11.75	21.08
stdc1	44.7 (-36.1)	12.9 (-74.6)	15.0	12.33
stdc1-pre	34.3 (-49.25)	10.7 (-79.5)	17.88	16.54
stdc2-pre	48.9 (-34.79)	14.13 (-75.8)	59.53	23.82
ddrnet	33.6 (-47.3)	9.2 (-78.9)	5.77	14.11
pidnet-s	40.0 (-33.1)	13.3 (-67.9)	4.65	17.52
pidnet-m	41.2 (-30.6)	12.8 (-66.6)	5.07	25.69
pidnet-l	46.8 (-28.1)	14.4 (-67.7)	11.13	17.2
seg mit-b0	53.1 (-26.9)	15.5 (-71.8)	0.32	16.74
seg mit-b1	53.8 (-25)	16.5 (-69.9)	0.32	19.13
seg mit-b2	58.8 (-21.4)	23.6 (-63.6)	4.47	5.76
m2f r101	50.2 (-26.3)	19.0 (-64.8)	9.58	7.29
m2f swin-b	62.0 (-16.54)	29.9 (-55.5)	11.18	7.64
m2f swin-l	74.2 (-6.86)	35.3 (-52.4)	11.51	11.20

TABLE 6
Performance on SydneyScapes validation set. Models are trained on CityScapes data. We note the reduction in performance between the two datasets in parenthesis.

Model	Overall		Class	
	mIoU ↑	mAcc ↑	rider ↑	person ↑
bisev2 fcn	37.67 (-39)	51.46 (-32.56)	11.75	21.08
ddrnet	43.75 (-37.14)	55.87 (-32.2)	5.77	14.11
stdc1	37.32 (-43.48)	54.37 (-33.13)	15.0	12.33
stdc1-pre	37.59 (-45.93)	52.4 (-37.74)	17.88	16.54
stdc2-pre	44.1 (-39.55)	57.08 (-32.85)	59.53	23.82
pidnet-s	45.96 (-27.11)	60.09 (-21.11)	4.65	17.52
pidnet-m	49.52 (-22.3)	62.3 (-17.1)	5.07	25.69
pidnet-l	47.62 (-27.32)	63.12 (-19.01)	11.13	17.2
seg mit-b0	49.94 (-30.05)	60.73 (-26.58)	0.32	16.74
seg mit-b1	52.66 (-26.08)	65.13 (-21.31)	0.32	19.13
seg mit-b2	54.64 (-25.58)	66.18 (-21.07)	4.47	5.76
m2f r101	47.08 (-29.46)	65.91 (-17.95)	9.58	7.29
m2f swin-b	65.07 (-13.49)	79.84 (-5.59)	11.18	7.64
m2f swin-l	66.51 (-14.57)	80.01 (-7.65)	11.51	11.20

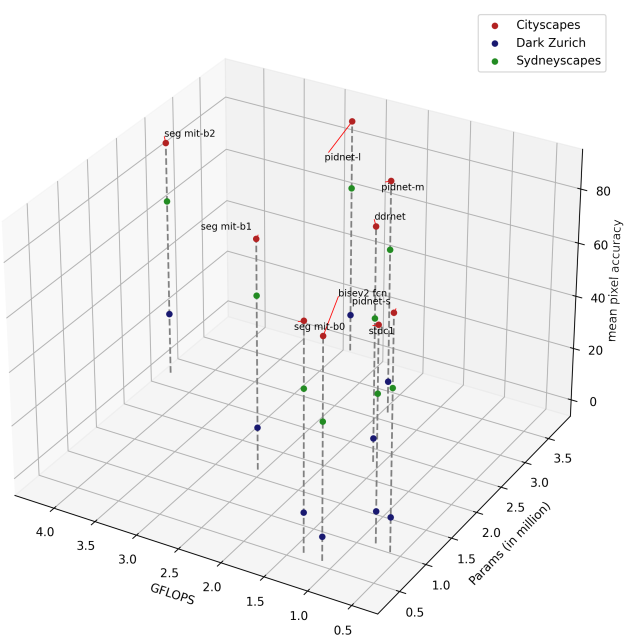

Fig. 11. The mean pixel accuracy measured for the models against its computational efficiency (FLOPS, number of parameters) is presented as a 3d scatter plot across the three datasets in this study. No points are present for Mask2Former whose computational efficiency couldn't be measured due to lack of support by mmsegmentation library. The scatter points for stdc1-pre and stdc2-pre were excluded as they lied too close to other points and made the overall plot less interpretable.

issue faced by two branch network and significantly enhanced overall performance. We have also covered smaller variants of SegFormer [44] and Mask2Former [45] which are capable of near real-time performance.

4.3 Data preprocessing

All three benchmarking datasets were sourced from their respective websites and preprocessed based on chosen model's configuration. The default configuration steps were slightly varied to support the differing image resolutions, dataset name and location. The preprocessing step takes care of masking the 30 classes in CityScapes to 19 classes of interest. This is done internally by `mmsegmentation` using the official `cityscapesScripts` library provided by the CityScapes team. All other transformations (resizing, normalization, cropping, augmentation) for each model are carried as it is without any variations.

4.4 Deployment considerations

Using a library such as `mmsegmentation` provides us with several benefits especially once the model is ready for deployment. First, we would need to convert the trained model into a format suitable for deployment. This is typically done using the MMDeploy Model Converter module, which can transform a PyTorch model into an intermediate representation (IR) such as `ONNX` or `TorchScript`. This IR model can then be converted into a backend-specific model compatible with various inference engines like `TensorRT`, `ONNX Runtime`, or `OpenVINO`. Next, we need to prepare the deployment environment. This involves ensuring that our production device meets the hardware requirements, such as having a compatible GPU or sufficient memory. This step is crucial for ensuring that the model runs efficiently and effectively on the target device. Finally, we need to integrate the model into our application. This involves writing the code to load the model, preprocess input data, run inference, and post-process the results. The MMDeploy SDK simplifies this process by providing pre-built functions for these tasks. Additionally, we may need to optimize the model for real-time performance, such as by reducing the model size or using quantization techniques. By following these steps, we can effeciently deploy a high-performance AI model to a low resource device, leveraging the powerful tools provided by `OpenMMLab`.

5 AI Ethics

We have considered each of the eight guiding principles for ethical AI development in Australia[8].

Human, societal and environmental wellbeing: We believe these semantic segmentation models can complement existing safety mechanisms and help reduce the over 1,128 fatalities that were reported on Australian roads by the freight report for 2021. As highlighted in our results this models should be used an additional source for developing safe mechanisms as they have lower performance for motorcycle and rider classes. The use of these models would

8. AI Ethics principles

increase overall energy requirement which can deteriorate the environment and needs to be suitably offset.

Human-centred values: These models should work well on diverse group of individuals that are commonly encountered in an urban environment. They are also not good enough to develop fully or semi-autonomous solutions and thus not be solely relied upon for decision making processes. The fallback for any major decision should fall back to a human being.

Fairness: AI systems should be inclusive and accessible, and should not involve or result in unfair discrimination against individuals, communities or groups. These models are mostly trained to operate in urban environment and might face challenges in regional environments or with individuals who are less represented in the training dataset such as individual with disabilities.

Privacy protection and security: AI systems should respect and uphold privacy rights and data protection, and ensure the security of data. The overall video footage recorded will be deleted periodically after a set period (a week) to respect individual privacy. These records will not be deleted in real-time to ensure that sufficient data is available for analysis post an accident, allowing us in discovering its causes and assess steps for mitigation.

Reliability and safety: Since the models were trained on CityScapes dataset they work well in European environment but the performance of most models was observed to drop one third as it was provided with Australian scene images from SydneyScapes. We also observe, extremely poor performance in images captured in low lighting conditions such as Dark Zurich with performance dropping almost three quarters. The model might also be unreliable in adverse climatic conditions where the chances of accidents are the highest. These limitations require further finetuning of the model on a local case specific dataset, creation of which would be costly. Drivers would also be incentivised to actively report any anomalies observed during system usage which would aid development efforts.

Transparency and explainability: There should be transparency and responsible disclosure so people can understand when they are being significantly impacted by AI, and can find out when an AI system is engaging with them. The drivers who would be primary users of this system would be adequately trained and made aware of the strengths and weaknesses of the model. In the deep learning era, the explainability of models has become difficult. A recently proposed idea by Mossina et al. [53] does enable quantifying the pixel level uncertainty in semantic segmentation using a technique called conformal prediction which could be leveraged here. Measurement of uncertainty can help measure reliability of prediction with highly uncertain pixels set to null, reducing false positives.

Contestability: The use of our pipeline would serve as an enhancement to drivers and is not meant to supersede or impact them in the decision make process at any point.

Accountability: With the pipeline being used in production, it would be utmost importance that users are able to report strange behaviours and anomalies that they would like fixed. A suitable platform would thus be setup which would be publicly accessible to report such incidents.

6 Discussion

6.1 UN Sustatinable Development Goals

In 2015, United Nations proposed 17 Sustainable Development Goals designed to achieve a better and more sustainable future for all by 2030. These goals address a wide range of global challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice. Four of these goals listed below are suitably addressed by this work:

- Good Health and Well-being: This work promotes healthy lives well-being for all humans who use the road transportation by enhancing overall safety.
- Decent Work and Economic Growth: The increased safety promotes sustained, inclusive, and sustainable economic growth for the Australian freight industry, ensuring full, productive employment for all.
- Industry, Innovation, and Infrastructure: This work achieves the objective of building resilient infrastructure, promotes inclusive and sustainable industrialisation, and fosters innovation.
- Sustainable Cities and Communities: Cities and human settlements would be more inclusive, safe, resilient, and sustainable as a result of this work.

6.2 Carbon Footprint

Experiments were conducted using a private infrastructure, which has a carbon efficiency of 0.432 $kgCO_2eq/kWh$. A cumulative of 10 hours of computation was performed on hardware of type RTX 4090 16 GB (TDP of 300W). Total emissions are estimated to be 1.3 $kgCO_2eq$ of which 0 percents were directly offset. Estimations were conducted using the MachineLearning Impact calculator presented in [54].

6.3 Data security, privacy

In our proposed solution, we prioritise data security and privacy through several key measures. We use advanced encryption to protect data both in transit and at rest. Access control mechanisms, including multi-factor authentication and role-based access control, ensure that only authorised personnel can access sensitive information. We also employ data anonymisation techniques to safeguard personal information. Regular security audits and continuous monitoring help us detect and respond to potential threats promptly. Our solution complies with relevant data protection regulations, such as GDPR and CCPA. Additionally, we follow data minimisation principles, collecting only necessary data and retaining it only as long as required.

6.4 Practical considerations

We note that while designs such as PIDNet, DDRNet achieve good performance in real-time, they are not highly scalable and do not adapt well to unseen data. SegFormer and Mask2Former are much more scalable and have strong generalisation capabilities. Deployment of all six models is possible on extremely limited hardware such as NVIDIA Jetson Nano and does not require more powerful GPUs. A singificant future work could involve evaluating model's performance on interstate highway images. All three benchmarking datasets that were discussed so far do not represent scenes outside urban areas so Google Street View can be used as a medium for obtaining images from interstate highways, remote areas which would be representative of the performance of our models outside city boundaries. For doing this, Google Street View Static API[9] is available. Other considerations besides the semantic segmentation model, include reducing the noise in data passed by the camera to the model by taking average of multiple images. Improving model performance in low light by preprocessing it with suitable filter before passing it to the model and training on more data having climatically adverse conditions. The algorithms for driving truck also need to be adapted as truck is considerably heavier than a car and requires more time to slow down.

7 Conclusion

In this study, we present standardised performance of state of the art real-time inference semantic segmentation models on three different datasets, representing diversity spatially and temporally. Most models' performance on CityScapes aligns with what has been reported in their respective paper but shows significant deterioration on Sydneyscapes and drops even further in low light environment conditions. The models thus are not good enough to work in diverse scenarios and require further fine tuning. We further highlight that most models have poor performance on classes representing human beings which signifies the need for a human to continuously be present in the decision making process and this models to serve and complementary aid at best. More representative images of local conditions would prove very crucial in model fine-tuning and can significantly enhance overall performance of the model. The quantification of semantic segmentation uncertainty would serve as a very important avenue for future improvement as it would prevent raising false alarms and improve user experience and trust. We have covered a relevant recent study which achieves the same in Transparency and explainability subheading. The best model in terms of accuracy is xx% which achieves an impressive mIoU of but offers poor inference speed at xx FPS. The fastest inference model is xx which can infer at xx FPS but offers poor accuracy (xx%). We do not identify any impact on existing processes with the deployment of this pipeline in production and exhaustive testing, followed by sufficient education of drivers and users would help in improving model adoption. We hope our approach can aid in the rapid development of Australian freight industry while improving safety.

Appendix 1 - Video Presentation

https://echo360.net.au/media/
08f88af4-937b-44f3-97d0-6828784a4ed9/public

Appendix 2 - Others

All code for reproduction has been provided in well documented notebooks accessible in my GitHub repository: https://github.com/nepython/COMP6011.

9. https://developers.google.com/maps/documentation/street-view/overview

Acknowledgments

If you need to acknowledge anyone please include here.

References

[1] M. D. Keall, W. J. Frith, and T. L. Patterson, "The contribution of alcohol to night time crash risk and other risks of night driving," *Accident Analysis & Prevention*, vol. 37, no. 5, pp. 816–824, 2005.

[2] L. N. Sharwood, J. Elkington, M. Stevenson, and K. K. Wong, "Investigating the role of fatigue, sleep and sleep disorders in commercial vehicle crashes: a systematic review," *Journal of the Australasian College of Road Safety*, vol. 22, no. 3, pp. 24–30, 2011.

[3] K. Gharehbaghi, I. Clarkson, N. Hurst, and F. Rahmani, "Transportation development for regional infrastructure: Implications for australian rural areas," *Transportation Research Procedia*, vol. 48, pp. 4003–4011, 2020.

[4] A. M. Zanni and T. J. Ryley, "The impact of extreme weather conditions on long distance travel behaviour," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 305–319, 2015.

[5] F. Heimes and H.-H. Nagel, "Towards active machine-vision-based driver assistance for urban areas," *International Journal of Computer Vision*, vol. 50, pp. 5–34, 2002.

[6] M. M. Trivedi, T. Gandhi, and J. McCall, "Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 108–120, 2007.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[13] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[14] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3684–3692.

[15] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4353–4361.

[16] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7151–7160.

[17] Y. Zhou, X. Sun, Z.-J. Zha, and W. Zeng, "Context-reinforced semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4046–4055.

[18] C. Yu, J. Wang, C. Gao, G. Yu, C. Shen, and N. Sang, "Context prior for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 416–12 425.

[19] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6819–6829.

[20] Y. Yuan, J. Xie, X. Chen, and J. Wang, "Segfix: Model-agnostic boundary refinement for segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*. Springer, 2020, pp. 489–506.

[21] M. Zhen, J. Wang, L. Zhou, S. Li, T. Shen, J. Shang, T. Fang, and L. Quan, "Joint semantic segmentation and boundary detection using iterative pyramid contexts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 666–13 675.

[22] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[23] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3146–3154.

[24] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei, "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 82–92.

[25] A. Shaw, D. Hunter, F. Landola, and S. Sidhu, "Squeezenas: Fast neural architecture search for faster semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision workshops*, 2019, pp. 0–0.

[26] Y. Li, L. Song, Y. Chen, Z. Li, X. Zhang, X. Wang, and J. Sun, "Learning dynamic routing for semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8553–8562.

[27] M. A. Elhassan, C. Zhou, A. Khan, A. Benabid, A. B. Adam, A. Mehmood, and N. Wambugu, "Real-time semantic segmentation for autonomous driving: A review of cnns, transformers, and beyond," *Journal of King Saud University-Computer and Information Sciences*, p. 102226, 2024.

[28] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[30] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[31] M. Orsic, I. Kreso, P. Bevandic, and S. Segvic, "In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 607–12 616.

[32] H. Li, P. Xiong, H. Fan, and J. Sun, "Dfanet: Deep feature aggregation for real-time semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9522–9531.

[33] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 325–341.

[34] J. Xu, Z. Xiong, and S. P. Bhattacharyya, "Pidnet: A real-time semantic segmentation network inspired by pid controllers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 19 529–19 539.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[36] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[37] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 558–567.

[38] C.-F. R. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 357–366.

[39] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, "Localvit: Bringing locality to vision transformers," *arXiv preprint arXiv:2104.05707*, 2021.

[40] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[41] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[42] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen, "Twins: Revisiting the design of spatial attention in vision transformers," *Advances in neural information processing systems*, vol. 34, pp. 9355–9366, 2021.

[43] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[44] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in neural information processing systems*, vol. 34, pp. 12 077–12 090, 2021.

[45] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1290–1299.

[46] C. Hümmer, M. Schwonberg, L. Zhou, H. Cao, A. Knoll, and H. Gottschalk, "Strong but simple: A baseline for domain generalized dense perception by clip-based transfer learning," in *Computer Vision – ACCV 2024*, M. Cho, I. Laptev, D. Tran, A. Yao, and H. Zha, Eds. Singapore: Springer Nature Singapore, 2025, pp. 463–484.

[47] M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo, and X. Wei, "Rethinking bisenet for real-time semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9716–9725.

[48] Y. Hong, H. Pan, W. Sun, and Y. Jia, "Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes," *arXiv preprint arXiv:2101.06085*, 2021.

[49] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset," in *CVPR Workshop on the Future of Datasets in Vision*, vol. 2, 2015, p. 1.

[50] C. Sakaridis, D. Dai, and L. V. Gool, "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7374–7383.

[51] J. S. Berrio Perez, M. Shan, S. Worrall, and H. Lyu, "Sydneyscapes: Image segmentation for australian environments," 2024. [Online]. Available: https://ses.library.usyd.edu.au/handle/2123/33051

[52] M. Contributors, "Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark," 2020.

[53] L. Mossina, J. Dalmau, and L. Andéol, "Conformal semantic image segmentation: Post-hoc quantification of predictive uncertainty," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3574–3584.

[54] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, "Quantifying the carbon emissions of machine learning," *arXiv preprint arXiv:1910.09700*, 2019.