INTERNSHIP REPORT

Project Title

Semantic Segmentation of Indian Road Scenes through Unsupervised Domain Adaptation

Author Name: Saket Pradhan

Author Affiliation: Department of Information Technology,

Thakur College of Engineering and Technology, Mumbai

Contact Details:

Phone: +91 9869474272

Email: s1032190351@thakureducation.org

Course of Project: 15th October, 2022 – 15st March, 2023

Total Duration: 5 Months

Lab: Statistical Signal Processing (SSP) Lab

Department: Department of Electrical Communication Engineering (ECE)

Institution: Indian Institute of Science, Bangalore 560012

REPORT - NOV. 30

Tasks Completed in till now:

- Organised and arranged the raw images of the WIRIN dataset and their respective json annotations.
- Explored the existing benchmark datasets for autonomous navigation.
- Implemented the SOTA models for GTAV-to-Cityscapes on the Bengaluru roads images for static image semantic segmentation and Unsupervised Domain Adaptation.

Benchmark Datasets:

The GTAV dataset contains 24966 synthetic images with pixel level semantic annotation. The images have been rendered using the open-world video game Grand Theft Auto V and are all from the car perspective in the streets of American-style virtual cities. There are 19 semantic classes which are compatible with the ones of Cityscapes dataset.

Cityscapes is a large-scale database which focuses on semantic understanding of urban street scenes. It provides semantic, instance-wise, and dense pixel annotations for 30 classes grouped into 8 categories (flat surfaces, humans, vehicles, constructions, objects, nature, sky, and void). The dataset consists of around 5000 fine annotated images and 20000 coarse annotated ones. Data was captured in 50 cities during several months, daytimes, and good weather conditions. It was originally recorded as video so the frames were manually selected to have the following features: large number of dynamic objects, varying scene layout, and varying background.

Unsupervised Domain Adaptation:

Unsupervised Domain Adaptation (UDA) is training a statistical model on labelled data from a source domain to achieve better performance on data from a target domain, with access to only unlabelled data in the target domain. It is a learning framework to transfer knowledge learned from source domains with a large number of annotated training examples to target domains with unlabelled data only. It aims to adapt a model trained on synthetic data to real-world data without requiring expensive annotations of real-world images.

In principle, UDA mainly focuses on the global distribution alignment between domains while not including the local distribution properties. Its objective is to leverage features from a labelled source domain and use them on an unlabelled target domain, with a similar but different data distribution. Most deep learning approaches to domain adaptation consist of two steps:

(i) learn features that preserve a low risk on labelled samples (source domain) and
(ii) make the features from both domains to be as indistinguishable as possible, so that a classifier trained on the source can also be applied on the target domain.

As UDA methods for semantic segmentation are usually GPU memory intensive, most methods operate only on downscaled images.

Papers Studied:

HRDA: Context-Aware High-Resolution Domain-Adaptive Semantic Segmentation. (https://www.ecva.net/papers/eccv_2022/papers_ECCV/papers/136900370.pdf)

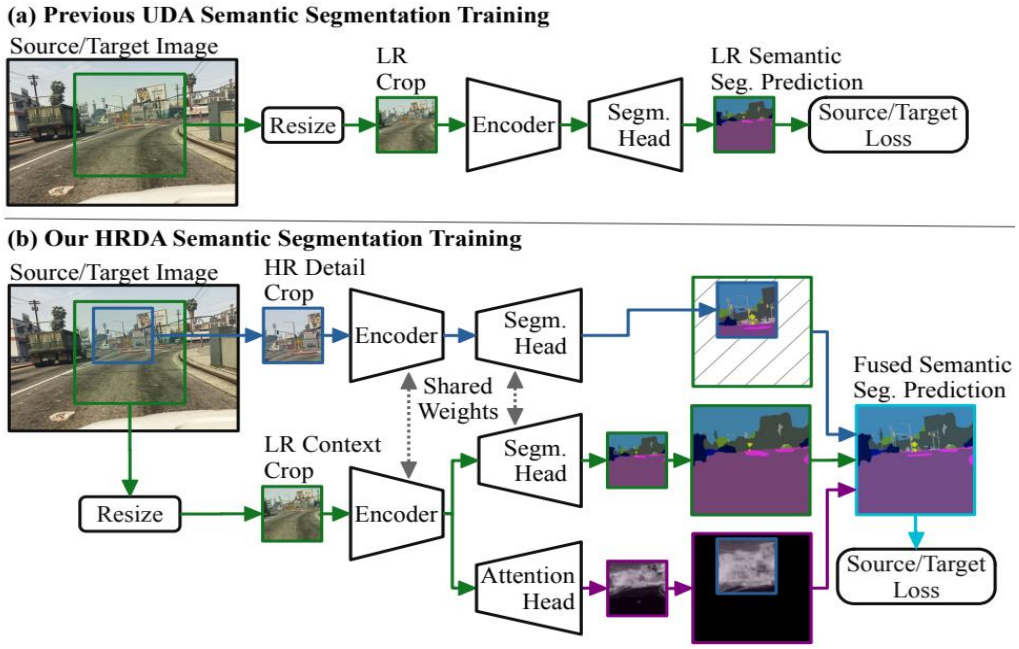


Fig. Proposed Training with Multi-Resolution Fusion.

DAFormer: Improving Network Architectures and Training Strategies for Domain-Adaptive Semantic Segmentation (https://arxiv.org/pdf/2111.14887v2.pdf).

The HRDA model is a continuation of the previous SOTA model DAFormer on the GTAV-to-Cityscapes dataset. The inference results for some images of the WIRIN dataset on the DAFormer model are given below.

Observations:

1. The DAFormer model is very susceptible to shadows, especially if they cover the area of the road immediately in front of the car.
2. It is unable to segment the roads properly if the sidewalks or the footpaths are not clearly defined on the road.
3. It is unable to create proper segmentation maps on entities in the presence of excessive exposure. This is predominantly observed over roads which under excess sunlight are classified incorrectly.
4. The model does not capture the road signs or segments them incorrectly.

Results

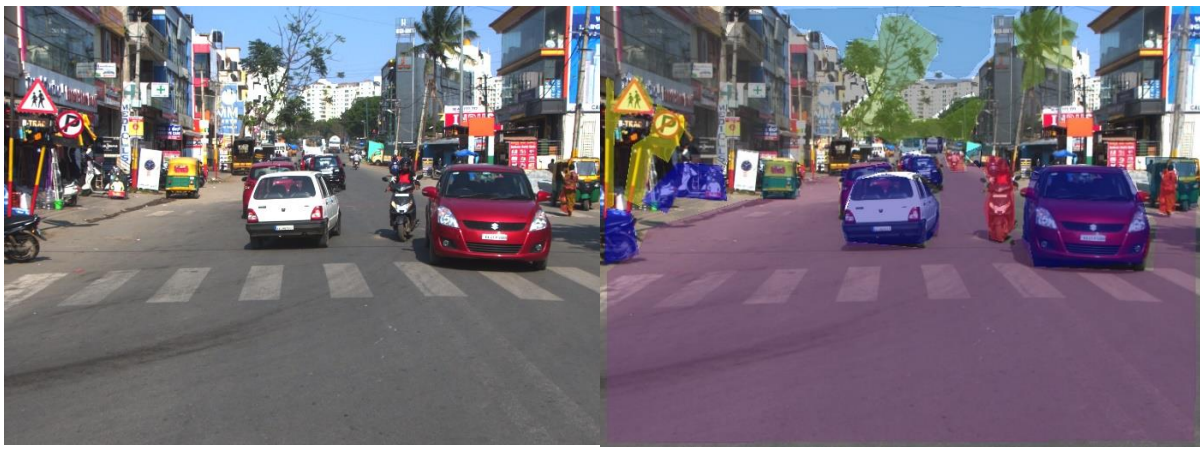
Input Image | Segmented Image | Overlay

REPORT - DEC. 6

Tasks Completed in the week:

- Implemented a script to generate masks over the WIRIN images using the json annotations provided. Implemented a bash script to mask the images in bulk.
- Studied the SegFormer framework, which is a precursor to the DAFormer model tested earlier.

Masked Images:



Original Image Masked Image

Observations:

- The area of the road directly in front of the car is not captured entirely in the annotations of several images. (See example 2 of masked images*)
- Some images have very coarse annotations. This is predominantly observed with trees, road signs and static vehicles. (See example 3 of masked images*)

Paper Studied:

SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. (https://arxiv.org/pdf/2105.15203.pdf)

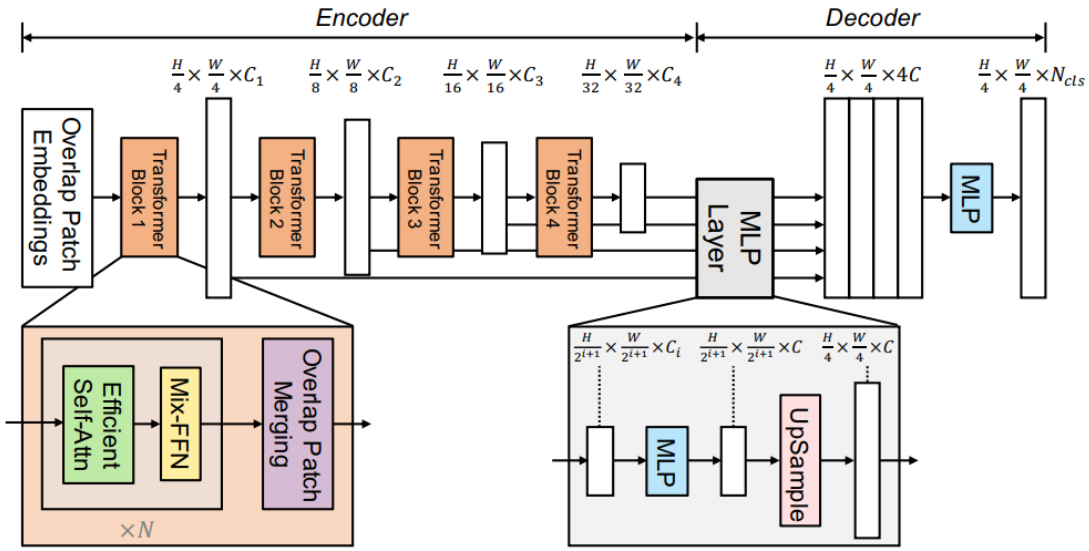


Fig.: The SegFormer framework has 2 modules: A hierarchical transformer encoder to extract the coarse and fine features, and a decoder to directly fuse these multi-level features and predict the segmentation mask.

There are two core ideas to SegFormer: the encoder in the model outputs multi-scale features, and the MLP-based decoder aggregates this information from different layers to output a segmentation map.

Working:

Encoder -

- An input image is first divided into 4 x 4 patches (similar to vision transformers), but in vision transformers, it generally uses a patch size of 16 x 16. A smaller patch size used to make better dense prediction tasks.
- The transformer block is composed of three sub-modules:

1. An efficient self-attention,
2. A mixed feed-forward network, and
3. An overlapping patch merging module.

- The first efficient self-attention module works like the original multi-head self-attention transformer, but it uses a sequence reduction technique in order to lower the computational costs.
- The next block, which is a mixed feed-forward network, is used to solve the fixed resolution problem.
- Instead of using fixed sized positional encoding, layers of convolution and multi-layer perceptron (MLP) are used to implement data-driven positional encoding.
- The last module, which is the overlapping patch merging block, is used to reduce the size of the feature map.
- The size of the feature is reduced as it goes to the higher part of the network.

Decoder -

- The decoder is quite simple compared to the modules in the encoder, because there are different features of different sizes in each layer of the encoder.
- The full MLP layer in the decoder takes the features from the encoder and fuses them together.
- This all MLP decoder is composed of four main steps:

1. Firstly, multi-level features from the encoder are fed into the multi-layer perceptron layer to unify in the channel dimension.
2. Next, the features are up-sampled to ¼ of its size and are concatenated together.
3. Thirdly, a MLP layer is adapted to fuse the concatenated features.
4. Finally, another MLP layer takes these fused features to predict the segmentation mask.

REPORT - DEC. 22

Tasks Completed in the week:

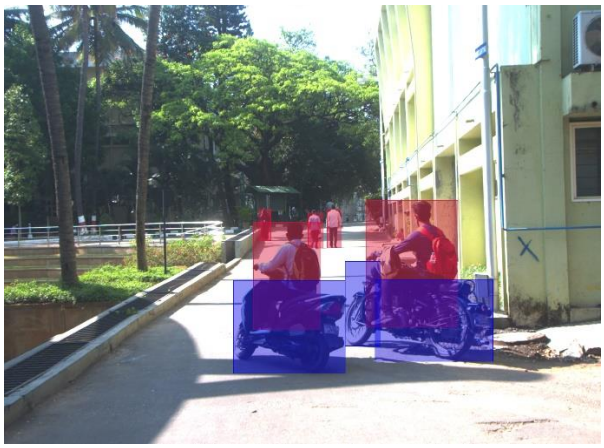
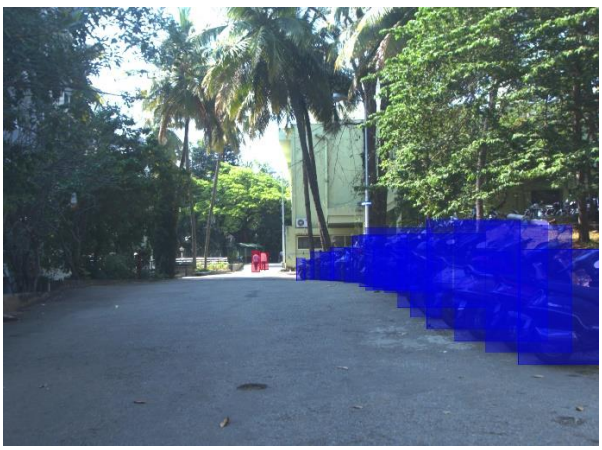
- Implemented a script to generate bounding boxes of different class labels over the WIRIN images using the json annotations provided. Implemented a bash script to process the images in bulk.
- Compiled a GitHub repository of all code implemented and models tested (kept private for the time being).
- Compiled the annotations provided into a single json file for ease of parsing through the dataset; the file can be found here.

Images with Bounding Boxes:

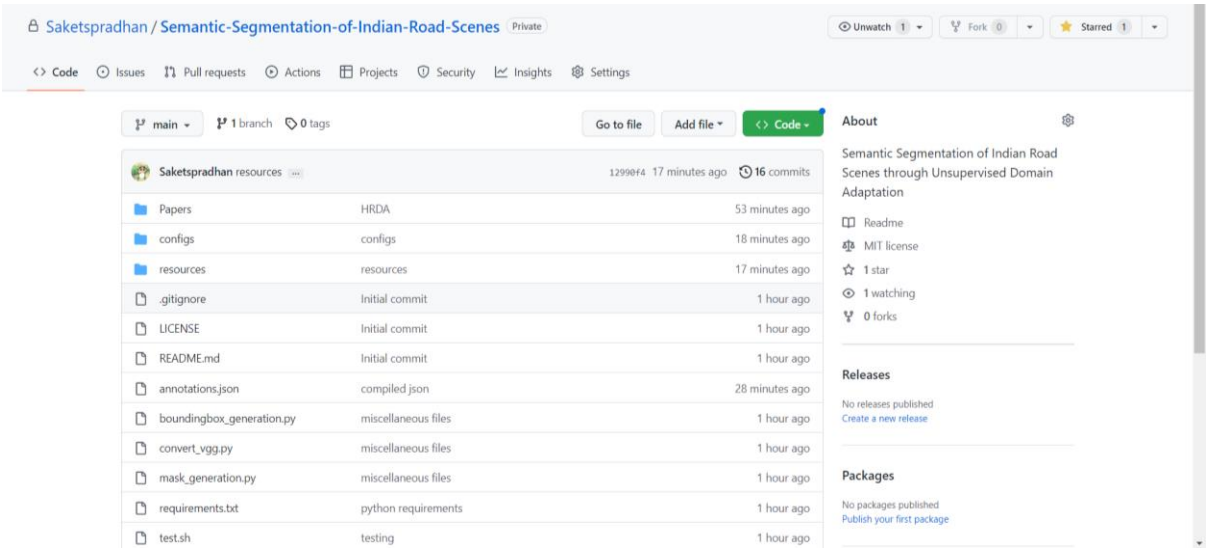




vivo | விவோ

GitHub Repository



Observations:

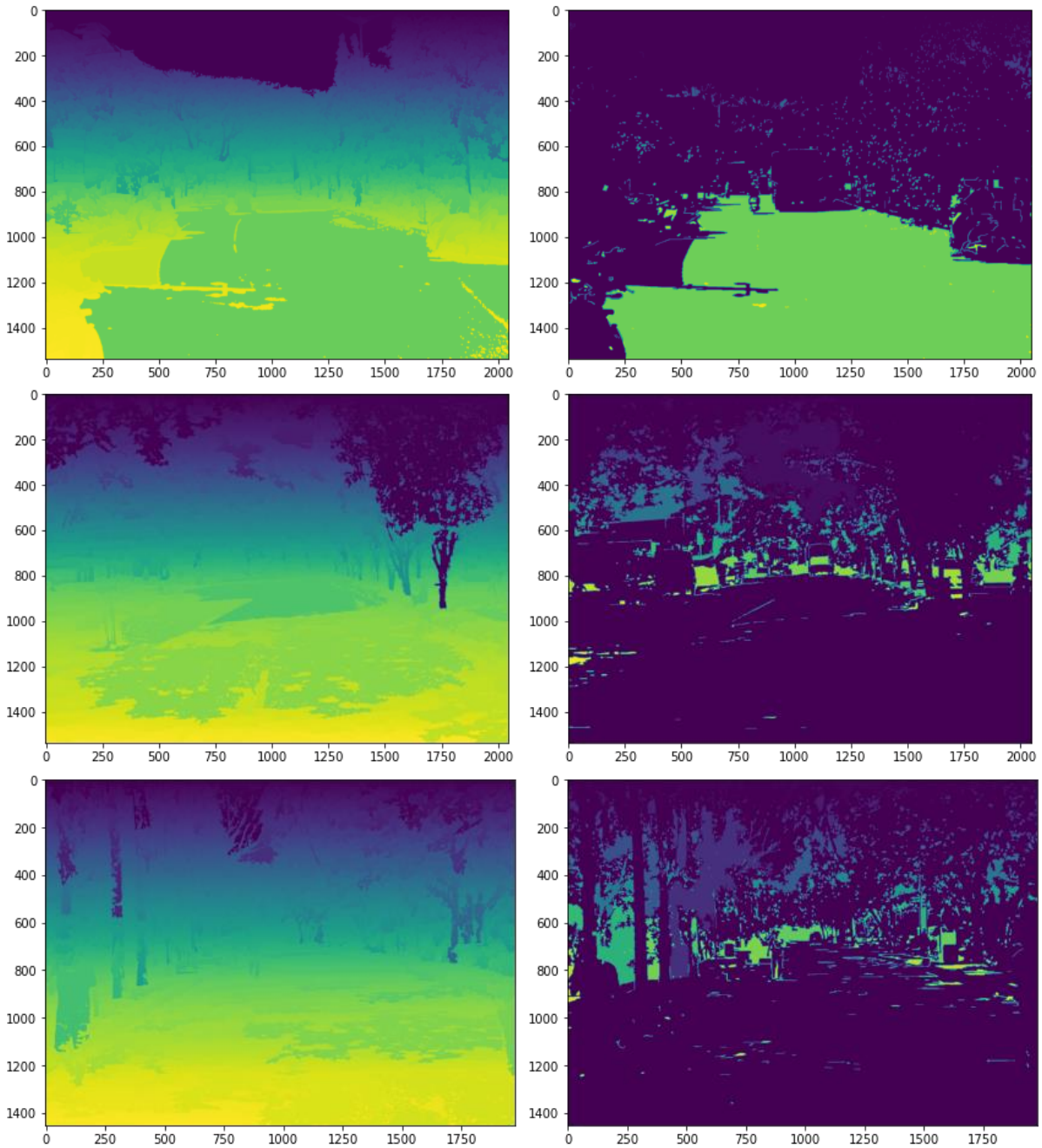
- In the case of motorcycles or bicycles, the passenger and the bike are annotated separately, leading to two different bounding boxes being drawn when only a single bounding box was expected.
- In some images, if any class completely overlaps another class, it tends to offset the bounding box from the image centre. This is most prominently observed in vehicles which are far away from the camera in the frame.
- When vehicles (bikes in particular) are parked adjacently, the individual vehicle annotations result in improper dimensions of the bounding boxes (see the last image*).
- If any car or a pedestrian is partially outside the image frame, the bounding boxes are improperly created around them.
- Multiple classes are intersecting with each other several times, therefore making the calculation of the mIoU score, or other accuracy metric (dice coefficient or F1 Score) very difficult.

REPORT - DEC. 29

Tasks Completed in the week:

- Implemented a script to calculate the mIoU score for multiple-class based semantic segmentation.
- Tested graph-based image segmentation using Felzenszwalb's algorithm.

Results for image segmentation with Felzenszwalb's algorithm



Felzenszwalb's Algorithm

The Felzenszwalb algorithm is a graph-based image segmentation method that can be used to partition an image into multiple regions or segments. It is typically used to identify and differentiate different objects or regions of interest in an image.

It is possible to use the Felzenszwalb algorithm to identify shadow regions in an image, but it would depend on the specific characteristics of the shadows and how they are represented in the image. Shadows can often have low contrast and be difficult to distinguish from other regions in the image, so it may be challenging to use the Felzenszwalb algorithm to accurately identify them.

Approach for calculating mIoU for multiple classes

For the calculation of mIoU for multiple classes, we need the labelled matrix of both predicted result and expected one (ground truth). Next, by going through a series of steps, we end up reaching the mIoU value. Here are two matrices, one representing the actual segmented output and the other predicted by the model.

Example:

0	0	0	0
0	1	1	4
5	5	2	4
5	3	3	4

actual

0	0	0	0
0	0	1	1
5	5	2	4
5	3	3	4

predicted

The elements of these matrices are the labels representing different classes to which pixels, at that particular location on the image belong. Here, there are altogether 6 classes with labels '0', '1', '2', '3', '4' and '5', and the matrices are of 2D Numpy type with size (4 x 4) each. To calculate the mIoU score for the image pair:

1. Find out the frequency count of each class for both the matrix. This can be done using the "bincount" function available in the numpy package.
2. Convert the matrix to 1D format. This step is done for easy computation, which can be done by reshaping the Numpy array.
3. Find out the category matrix.

Since, we assume 6 classes, there can be 6 x 6 = 36 possibilities. Tthe 6th pixel actually belongs to class '1' but is predicted to be present in class '0' and thus belonging to category '1–0'. Each

such possibility corresponds to a category. The possible categories could be ‘0–0’, ‘0–1’, ‘0–2’, ….., ‘4–5’, ‘5–5’. They are numbered as per their index; category ‘0–0’ has got number 0, category ‘0–1’ got number 1, and so on. The category matrix is one that will have the elements as the category numbers to which the pixels at that particular location belong.

Category = (number of classes x actual_1D) + pred_1D

4. Constructing the confusion matrix.

A confusion matrix is a (no. of classes x no. of classes) size matrix which stores the information about the number of pixels belonging to a particular category. The frequency count of the ‘category’ array gives a linear array which on reshaping to (6x6) gives us the confusion matrix. The confusion matrix also stores some useful information which help in the calculation of IoU. The diagonal of the confusion matrix represents the common region. So, these elements are the intersection values of the predicted output and ground truth. The upper triangular part of confusion matrix represents those areas where actual matrix is true but the predicted one is false and the lower triangular part represents the opposite.

5. Calculating IoU for individual classes.
6. Calculating MIoU for the actual-predicted pair.

It is found out using the ‘nanmean’ function available in numpy package. ‘Nanmean’ is preferred than ordinary mean to ignore the cases where individual IoU value may turn out to be ‘nan’ because of the absence of any particular class in an image.

I have uploaded the code for calculating the mIoU score in the project GitHub repository.

REPORT - JAN. 5

Tasks Completed in the week:

- Studied the paper OneFormer: One Transformer to Rule Universal Image Segmentation (https://arxiv.org/pdf/2211.06220v2.pdf)

About OneFormer

OneFormer is a new segmentation model that beats former SOTA solutions—MaskFormer and Mask2Former, and it is now ranked number one in the instance, semantic and panoptic segmentation. OneFormer is based on transformers and built using Detectron2.

OneFormer is the first multi-task image segmentation framework. This means the model only needs to be trained once with universal architecture and a single dataset. Previously, even if the model scored high in all three segmentation tasks, it needed to be trained individually on the semantic, instance, or panoptic datasets.

OneFormer introduces a task-conditional joint training strategy. The model uniformly samples training examples from different ground truth domains. As a result, the model architecture is task-guided for training and task-dynamic for inference, all with a single model.

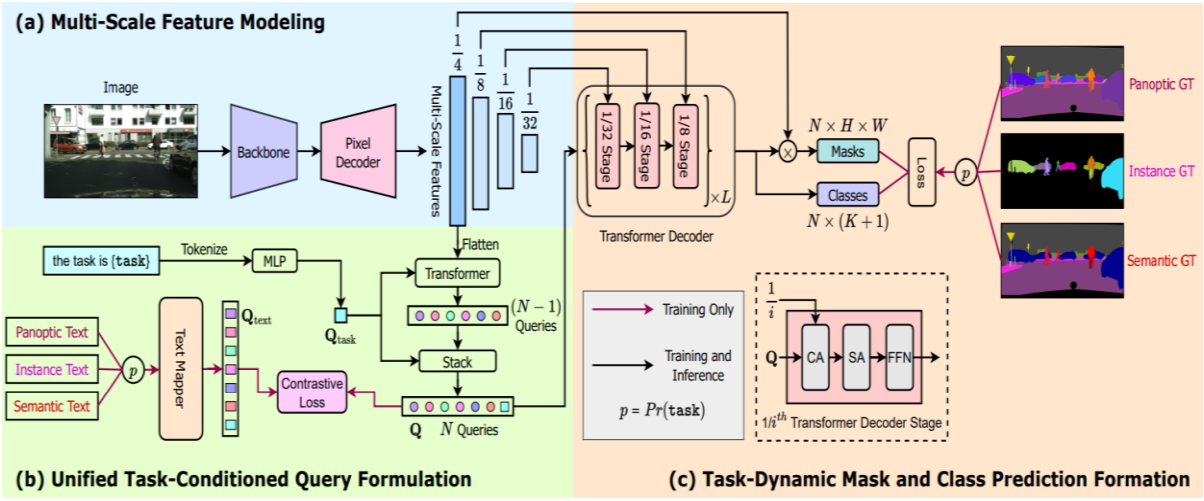
Working

The paper suggests a multi-task universal image segmentation framework (OneFormer) to completely unify image segmentation, which outperforms the current state-of-the-art on all three image segmentation tasks by just training once on a single dataset.

The framework design is a transformer-based method, which query tokens may direct, in response to the recent success of transformer-based frameworks. They initialize their queries as repetitions of the task token (obtained from the task input) to add task-specific context to their model. Then they compute a query-text contrastive loss using the text derived from the corresponding ground-truth label for the sampled task.

According to their hypothesis, a contrastive loss on the queries aids in guiding the model to become more task-sensitive. Additionally, it lessens incorrect category predictions. OneFormer is tested on three significant segmentation datasets, each with all three segmentation tasks: ADE20K, Cityscapes, and COCO.

Model Architecture



Model Performance on the Cityscapes dataset

All backbones are pretrained on ImageNet-22k.

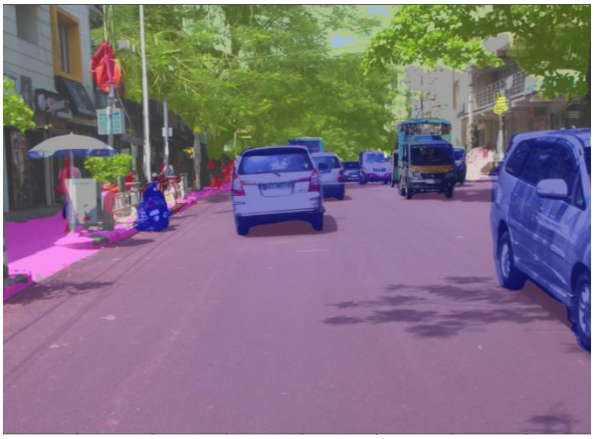
Model + Backbone	mIoU (s.s)	mIoU (ms+flip)	Parameters
OneFormer + **Swin-L**	83.0	84.4	219M
OneFormer + **ConvNeXt-L**	83.0	84.0	220M
OneFormer + **DiNAT-L**	83.1	84.0	223M
OneFormer + **ConvNeXt-XL**	83.6	84.6	372M

REPORT - JAN. 12

Tasks Completed in the week:

- Implemented the OneFormer model for semantic segmentation on the dataset provided.
- Organised and appended all code to the project GitHub repository.

Results










Original Image

Output Image

Observations:

- The OneFormer model often misidentifies and mislabels the classes when the objects are considerably far away in the frame. This is most frequently observed with vehicles and humans (See the mislabelled truck in the last example*).
- OneFormer cannot identify the surface of the road or the sky if the image is not coarse enough (See last second last example, where the ground is incompletely segmented*).
- OneFormer does not deal well with shadows, or bright patches of light, if present in the image. In such cases, the area is often left unclassified (See example 5*).
- At times, a single vehicle is categorized into two or more classes in the presence of sharp boundaries or margins in the image.

REPORT - JAN. 21

Tasks Completed in the week:

- Started compiling the contents for the paper; wrote the sections – semantic segmentation and unsupervised domain adaptation under the heading 'Related Works'.
- Calculated the mIoU scores for the GTA5 to Cityscapes dataset across different models.
  - Number of classes in the dataset: **30**
  - Number of classes considered for comparison: **19** (given below)

Results

Table: Performance for various models on the GTA5 to Cityscapes dataset

Models	**Classes and Score**																			
	Road	S.walk	Build.	Wall	Fence	Pole	Tr. Light	Tr. Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bicycle	mIoU
CBST	91.8	53.5	80.5	32.7	21	34	28.9	20.4	83.9	34.2	80.9	53.1	24	82.7	30.3	35.9	16	25.9	42.8	91.8
DACS	89.9	39.7	87.9	30.7	39.5	38.5	46.4	52.8	88	44	88.8	67.2	35.8	84.5	45.7	50.2	0	27.3	34	89.9
CorDA	94.7	63.1	87.6	30.7	40.6	40.2	47.8	51.6	87.6	47	89.7	66.7	35.9	90.2	48.9	57.5	0	39.8	56	94.7
BAPA	94.4	61	88	26.8	39.9	38.3	46.1	55.3	87.8	46.1	89.4	68.8	40	90.2	60.4	59	0	45.1	54.2	94.4
ProDA	87.8	56	79.7	46.3	44.8	45.6	53.5	53.5	88.6	45.2	82.1	70.7	39.2	88.8	45.5	59.4	1	48.9	56.4	87.8
DAFormer	95.7	70.2	89.4	53.5	48.1	49.6	55.8	59.4	89.9	47.9	92.5	72.2	44.7	92.3	74.5	78.2	65.1	55.9	61.8	95.7
HRDA	96.4	74.4	91	61.6	51.5	57.1	63.9	69.3	91.3	48.4	94.2	79	52.9	93.9	84.1	85.7	75.9	63.9	67.5	96.4

REPORT - JAN. 31

Tasks Completed in the week:

- Wrote content for the 'Introduction' and 'Methodology' sub-sections of the paper.
- Calculated the mIoU scores for the SYNTHIA to Cityscapes dataset across different models.
  - Number of classes considered for comparison: **16** (**13** in SYNTHIA common with Cityscapes)

Results

Table: Performance for various models on the SYNTHIA to Cityscapes dataset

Models	**Classes and Score**																			
	Road	S.walk	Build.	Wall	Fence	Pole	Tr. Light	Tr. Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bicycle	mIoU
CBST	68	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	-	78.3	60.6	28.3	81.6	-	23.5	-	18.8	39.8	68
DACS	80.6	25.1	81.9	21.5	2.9	37.2	22.7	24	83.7	-	90.8	67.6	38.3	82.9	-	38.9	-	28.5	47.6	80.6
BAPA	91.7	53.8	83.9	22.4	0.8	34.9	30.5	42.8	86.6	-	88.2	66	34.1	86.6	-	51.3	-	29.4	50.5	91.7
CorDA	93.3	61.6	85.3	19.6	5.1	37.8	36.6	42.8	84.9	-	90.4	69.7	41.8	85.6	-	38.4	-	32.6	53.9	93.3
ProDA	87.8	45.7	84.6	37.1	0.6	44	54.6	37	88.1	-	84.4	74.2	24.3	88.2	-	51.1	-	40.5	45.6	87.8
DAFormer	84.5	40.7	88.4	41.5	6.5	50	55	54.6	86	-	89.8	73.2	48.2	87.2	-	53.2	-	53.9	61.7	84.5
HRDA	85.2	47.7	88.8	49.5	4.8	57.2	65.7	60.9	85.3	-	92.9	79.4	52.8	89	-	64.7	-	63.9	64.9	85.2

REPORT - FEB. 17

Tasks Completed in the week:

- Calculated the mIoU scores for the National Dataset for Indian Roads (WIRIN dataset).
  - Number of classes in the given dataset: **29**
  - Number of classes considered for comparison: **19** (marked in black)
  - Number of classes rejected: **10** (marked in red)

1. road
2. sidewalk
3. building
4. wall
5. fence
6. pole
7. traffic_light
8. traffic_sign
9. vegetation
10. terrain
11. sky
12. person
13. car
14. truck
15. bus
16. train
17. motorcycle
18. bicycle
19. rider
20. unknown
21. tunnel
22. autorickshaw
23. animal
24. rail_track
25. guard_rail
26. miscellaneous_vehicles
27. pillar
28. bridge
29. divider

Results (yet to be completed for some sections) *

Table: Performance for various models on the National Dataset for Indian Roads (WIRIN dataset)

Models	**Classes and Score**																			
	Road	S.walk	Build.	Wall	Fence	Pole	Tr. Light	Tr. Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bicycle	mIoU
CBST	79.3	45.4	69.2	27.9	17.7	28.1	26.6	17.4	73.5	30.0	67.5	42.1	20.4	72.9	27.1	31.5	13.4	22.8	38.3	79.2
DACS*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CorDA*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BAPA	84.6	54.2	79.2	23.8	36.1	34.0	42.9	49.7	77.4	41.2	78.6	62.3	35.6	80.8	51.9	52.1	0	41.0	48.5	84.6
ProDA	82.6	52.0	74.2	43.9	43.4	42.8	47.5	49.5	82.3	42.0	76.3	65.7	36.5	82.2	42.3	55.4	0.9	45.7	52.4	82.6
DAFormer	90.9	66.9	84.3	51.3	45.7	47.6	53.2	57.1	85.9	45.3	87.6	69.1	42.8	87.8	70.7	74.4	61.8	53.4	59.0	90.8

REPORT – FEB. 25

Tasks Completed in the week:

- Evaluated and tabulated the class-wise IoU scores for the classes: Road, Sidewalk, Building, Wall, Fence, Pole, Traffic Light, Sign, and Vegetation.

The class-wise IoU score tables are given below as follows:

Class/Model	Road		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	91.8	68	**79.3**
BAPA	94.4	93.3	**84.6**
ProDA	87.8	87.8	**82.6**
DAFormer	95.7	84.5	**90.9**

Class/Model	Sidewalk		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	53.5	29.9	**45.4**
BAPA	61	61.6	**54.2**
ProDA	56	45.7	**52**
DAFormer	70.2	40.7	**66.9**

Class/Model	Building		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	80.5	76.3	**69.2**
BAPA	88	85.3	**79.2**
ProDA	79.7	84.6	**74.2**
DAFormer	89.4	88.4	**84.3**

Class/Model	Wall		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	32.7	10.8	**27.9**
BAPA	26.8	19.6	**23.8**
ProDA	46.3	37.1	**43.9**
DAFormer	53.5	41.5	**51.3**

Class/Model	Fence		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	21	1.4	**17.7**
BAPA	39.9	5.1	**36.1**
ProDA	44.8	0.6	**43.4**
DAFormer	48.1	6.5	**45.7**

Class/Model	Pole		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	34	33.9	**28.1**
BAPA	38.3	37.8	**34**
ProDA	45.6	44	**42.8**
DAFormer	49.6	50	**47.6**

Class/Model	Traffic Lights		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	28.9	22.8	**26.6**
BAPA	46.1	36.6	**42.9**
ProDA	53.5	54.6	**47.5**
DAFormer	55.8	55	**53.2**

Class/Model	Sign		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	20.4	29.5	**17.4**
BAPA	55.3	42.8	**49.7**
ProDA	53.5	37	**49.5**
DAFormer	59.4	54.6	**57.1**

Class/Model	Vegetation		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	83.9	77.6	**73.5**
BAPA	87.8	84.9	**77.4**
ProDA	88.6	88.1	**82.3**
DAFormer	89.9	86	**85.9**

REPORT – MAR. 5

Tasks Completed in the week:

- Evaluated and tabulated the class-wise IoU scores for the classes: Terrain, Sky, Person, Car, Truck, Bus, Train, Motorcycle, Bicycle, and Rider, and the mIoU score.

The class-wise IoU score tables are given below as follows:

Class/Model	Terrain		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	34.2	NA	**30**
BAPA	46.1	NA	**41.2**
ProDA	45.2	NA	**42**
DAFormer	47.9	NA	**45.3**

Class/Model	Sky		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	80.9	78.3	**67.5**
BAPA	89.4	90.4	**78.6**
ProDA	82.1	84.4	**76.3**
DAFormer	92.5	89.8	**87.6**

Class/Model	Person		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	53.1	60.6	**42.1**
BAPA	68.8	69.7	**62.3**
ProDA	70.7	74.2	**65.7**
DAFormer	72.2	73.2	**69.1**

Class/Model	Rider		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	24	28.3	**20.4**
BAPA	40	41.8	**35.6**
ProDA	39.2	24.3	**36.5**
DAFormer	44.7	48.2	**42.8**

Class/Model	Car		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	82.7	81.6	**72.9**
BAPA	90.2	85.6	**80.8**
ProDA	88.8	88.2	**82.2**
DAFormer	92.3	87.2	**87.8**

Class/Model	Truck		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	30.3	NA	**27.1**
BAPA	60.4	NA	**51.9**
ProDA	45.5	NA	**42.3**
DAFormer	74.5	NA	**70.7**

Class/Model	Bus		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	35.9	23.5	**31.5**
BAPA	59	38.4	**52.1**
ProDA	59.4	51.1	**55.4**
DAFormer	78.2	53.2	**74.4**

Class/Model	Train		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	16	NA	**13.4**
BAPA	0	NA	**0**
ProDA	1	NA	**0.9**
DAFormer	65.1	NA	**61.8**

Class/Model	Motorbike		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	25.9	18.8	**22.8**
BAPA	45.1	32.6	**41**
ProDA	48.9	40.5	**45.7**
DAFormer	55.9	53.9	**53.4**

Class/Model	Bike		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	42.8	39.8	**38.3**
BAPA	54.2	53.9	**48.5**
ProDA	56.4	45.6	**52.4**
DAFormer	61.8	61.7	**59**

Class/Model	mIoU		
	GTA5 to Cityscapes	SYNTHIA to Cityscapes	**WIRIN**
CBST	91.8	68	**78**
BAPA	94.4	93.3	**84.6**
ProDA	87.8	87.8	**81.6**
DAFormer	95.7	84.5	**90.7**

FINAL REPORT – MAR. 15

Tasks Completed in the week:

- Paper titled ‘Semantic Segmentation of Indian Road Scenes using Unsupervised Domain Adaptation’ completed and attached to the end of the report.

***

Semantic Segmentation of Indian Road Scenes using Unsupervised Domain Adaptation

Saket Pradhan
Department of Information Technology
Thakur College of Engineering & Technology
Mumbai, India
saketspradhan@gmail.com

Abstract—Semantic segmentation classifies each pixel of an image into one of several predefined classes. In the context of road scenes, it is used for self-driving cars, traffic monitoring, and map generation. Recent advances have been driven by the development of large, publicly available road scene datasets, such as Cityscapes and SYNTHIA. The current SOTA is dominated by FCNs and encoder-decoder architectures that directly produce a dense, per-pixel prediction. Encoder decoder architectures, such as the U-Net, use a combination of convolutional and transposed convolutional layers to upsample a lower-resolution feature map to output a dense, per-pixel prediction. One area of particular interest is the development of models that handle variations in lighting and weather conditions. We use DAFormer for semantic segmentation, which is trained end-to-end with cross-entropy loss and the SGD optimizer. The use of data augmentation techniques such as random flipping, scaling, and cropping is also performed during training to improve the robustness of the model to different image variations. The performance is evaluated using several standard metrics, with mean IoU scores of over 95.7% on the Cityscapes and 84.5% on the SYNTHIA datasets.

Keywords—*semantic, segmentation, DAFormer, SegFormer, transformers, self-driving, cityscapes, GTA5*

I. Introduction

Semantic segmentation is a task in computer vision that involves classifying each pixel in an image into one of several predefined categories. This technology can be used in a wide range of applications, including self-driving cars, traffic monitoring and surveillance, and urban planning. In the context of Indian road scenes, semantic segmentation can be used to identify different types of vehicles, pedestrians, buildings, and road markings, which can be used to improve traffic safety and efficiency [1].

However, training semantic segmentation models on Indian road scenes can be challenging due to the wide variability in lighting, weather, and other factors that can affect the appearance of the scene. Additionally, the availability of labeled data for Indian road scenes is often limited, which can make it difficult to train high-performing models.

Unsupervised domain adaptation is a method that can be used to improve the performance of semantic segmentation models on Indian road scenes. This approach involves training a model on one set of data, and then fine-tuning it to perform well on a different, but related, set of data. This can be done by using a technique called adversarial training, which involves training two models simultaneously: one that generates images that are similar to the target domain, and another that tries to distinguish between the generated images and the real images.

II. Related Works

A. Semantic Segmentation

Semantic segmentation is a computer vision task that involves classifying each pixel in an image into one of several predefined categories. The goal of semantic segmentation is to assign a semantic label, such as "car" or "person," to each pixel in an image. This technique is used to identify and locate various objects and features in an image, such as vehicles, pedestrians, traffic signs, and lane markings.

Semantic segmentation is typically performed using deep learning techniques, such as convolutional neural networks (CNNs) and fully convolutional networks (FCNs). These models are trained on large amounts of labelled data and are able to learn to recognize patterns and features in images that are relevant for semantic segmentation. One of the key advantages of semantic segmentation is its ability to provide a detailed understanding of the scene.

By assigning a label to each pixel, semantic segmentation can be used to identify and locate specific objects and features in an image, such as vehicles, pedestrians, traffic signs, and lane markings. This information can be used for a variety of applications, including self-driving cars, robotics, and surveillance systems.

Semantic segmentation is also used in the field of computer graphics, where it can be used to create high-quality images and animations. The technique can also be used in medical imaging and satellite imaging, where it can be used to identify and locate specific structures or regions in images.

B. Unsupervised Domain Adaptation

Unsupervised Domain Adaptation (UDA) is training a statistical model on labelled data to a target domain's data from a source domain more efficiently, having access to only unlabeled data in the target domain. It is a framework for learning that allows information to be transferred from source domains with lots of annotated training examples to target domains with just unlabeled data [2]. It aims to adapt a model trained on synthetic data to real-world data without requiring expensive annotations of real-world images.

In principle, UDA mainly focuses on the global distribution alignment between domains while not including the local distribution properties. Its objective is to leverage features from a labelled source domain and use them on an unlabeled target domain, with a similar but different data distribution. The majority of deep learning methods for domain adaptation involve two phases:

i. learn features that preserve a low risk on labelled samples (source domain) and,
ii. make the features from both domains to be as indistinguishable as possible, so that a classifier trained on the source can also be applied on the target domain.

Most approaches only work with downscaled pictures since UDA methods for semantic segmentation are often GPU memory heavy.

C. Indian Road Scenes

Semantic segmentation of Indian road scenes is the process of classifying each pixel in an image of an Indian road scene into one of several predefined categories. This technique is used to identify and locate various objects and features such as vehicles, pedestrians, traffic signs, and lane markings. One of the key challenges in the semantic segmentation of Indian road scenes is the variability in the appearance of objects and features. Indian road scenes often contain a large number of pedestrians and other non-vehicular objects, which can further complicate the task of semantic segmentation.

To address these challenges, researchers have developed several approaches that leverage deep learning techniques such as convolutional neural networks (CNNs) and fully convolutional networks (FCNs) to improve the performance of semantic segmentation models. The application of semantic segmentation of Indian road scenes is mainly used in the development of advanced driver assistance systems (ADAS) for self-driving cars, road safety, traffic management, surveillance, and robotics [3].

ADAS systems use cameras and other sensors to perceive the environment and make decisions about how to control the vehicle. Semantic segmentation can be used to identify and locate objects and features such as vehicles, pedestrians, and traffic signs, which can be used to improve the performance of self-driving cars. Semantic segmentation can also be used to improve road safety by identifying and locating potential hazards such as pedestrians, bicycles, and motorcycles on the road. This information can be used to alert drivers or to control the speed and trajectory of self-driving cars [4].

III. DATASETS COMPARATIVE STUDY

A. Cityscapes

Cityscapes is a comprehensive database that concentrates on comprehending urban street scenes semantically. It offers detailed annotations for 30 different classes, which are divided into 8 categories (such as flat surfaces, humans, vehicles, constructions, objects, nature, sky, and void), and includes both instance-wise and dense pixel annotations. The dataset encompasses around 5000 images with fine annotations and 20000 with coarse annotations. The data was captured in 50 cities over a period of several months and only during daylight hours and good weather conditions.

The footage was initially recorded as video, and specific frames were selected for inclusion based on the presence of a high number of dynamic objects, diverse scene layouts, and changing backgrounds. The dataset is comprised of a diverse set of stereo video sequences from 50 cities and boasts high-quality pixel-level annotations. This makes it significantly larger in scale compared to similar previous efforts. It is designed for evaluating the performance of vision algorithms in crucial tasks of semantic urban scene understanding, such as pixel-level, instance-level, and panoptic semantic labelling, and to support research that utilizes large amounts of (weakly) annotated data, like for training deep neural networks.

B. GTA5

The GTA5 dataset comprises 24966 synthetic images with pixel-level semantic labelling. The images were created through the open-world video game Grand Theft Auto 5 and feature a car's view of American-style virtual cities. There are 19 semantic categories, similar to those in the Cityscapes dataset. The dataset encompasses an extensive range of road scenes, from urban to rural, and provides comprehensive annotations for objects such as buildings, roads, vehicles, and pedestrians. Its diversity and large number of images make it a valuable resource for training models to handle diverse input and generalize effectively to new images.

C. SYNTHIA

The SYNTHIA dataset is a synthetic collection of images and annotations designed to support research in semantic segmentation and other scene-understanding tasks related to driving scenarios. It features a large volume of data with precise pixel-level semantic annotations, including over 200,000 high-definition images from video streams and 20,000 images from separate snapshots.

The dataset offers a diverse range of scenes, including a European-style town, modern city, highway, and green areas, as well as a variety of dynamic objects such as cars, pedestrians, and cyclists. Additionally, the dataset includes different seasons, lighting conditions, weather patterns, and a simulation of multiple sensors, including 8 RGB cameras and 8 depth sensors.

D. National Dataset for Indian Roads

The National Dataset for Indian Roads is a dataset created for research and development of autonomous

systems—specifically for Indian road scene analysis, in partnership by IISc and Wipro under the WIRIN initiative.

It was created to address the lack of publicly available datasets for Indian road scenes, having unique characteristics of such as crowded roads, diverse traffic, and a wide range of vehicle types. It includes 1000 images captured from various locations in India with a resolution of 1920*1080 pixels. Each picture is annotated with 29 classes, including roads, buildings, vehicles, pedestrians, and traffic signs.

The dataset is composed of images and their annotations which are useful for the semantic segmentation of Indian road scenes as well as traffic analysis. The dataset was created using real-world data captured in Bengaluru urban and suburban, including a wide variety of urban to rural scenes, which makes it unique and valuable for researchers working on autonomous systems in India.

One of its key features is that it includes detailed annotations and labels for different types of roads (such as highways, residential streets, and dirt roads), as well as for different types of vehicles (such as cars, trucks, and buses) and different types of buildings (such as houses and factories). These annotations can be used to train models to accurately identify and segment different objects in the images, which is an important step in developing models for autonomous vehicles and other applications that rely on understanding the environment. This dataset also includes annotations for unique and specific Indian road conditions such as traffic lights, traffic signs, autorickshaws, and guard rails. This allows for the development of models that are specifically tailored to Indian road conditions and regulations.

IV. Methodology

In this paper, we implement two models for semantic segmentation, namely SegFormer [5] and DAFormer [11], and discuss their results and performance.

A. SegFormer

SegFormer [5] includes a new hierarchically structured transformer encoder structure that outputs multi-scale features. SegFormer does not require positional encoding, thereby preventing the interpolation of positional codes leading to a lower performance while the resolution during testing differs from that while training. It also has the tendency to disprove complex decoders. The MLP decoder combines local and global attention by merging information from various layers, resulting in effective representations.

The SegFormer framework has 2 modules: A hierarchical transformer encoder to extract the coarse and fine features, and a decoder to directly fuse these multi-level features and predict the segmentation mask. There are two core ideas to SegFormer: the encoder in the model outputs multi-scale features, and the MLP-based decoder aggregates this information from different layers to output a segmentation map.

1. Encoder

An input image is first divided into 4*4 patches (similar to vision transformers), but in vision transformers, it generally uses a patch size of 16*16. A smaller patch size used to make better dense prediction tasks. The transformer block is composed of three sub-modules:

1. An efficient self-attention,
2. A mixed feed-forward network, and
3. An overlapping patch merging module.

The first efficient self-attention module works like the original multi-head self-attention transformer, but it uses a sequence reduction technique in order to lower the computational costs. The next block, which is a mixed feed-forward network, is used to solve the fixed resolution problem. Instead of using fixed sized positional encoding, layers of convolution and multi-layer perceptron (MLP) are used to implement data-driven positional encoding. The last module, which is the overlapping patch merging block, is used to reduce the size of the feature map. The size of the feature is reduced as it goes to the higher part of the network.

2. Decoder

The decoder is quite simple compared to the modules in the encoder, because there are different features of different sizes in each layer of the encoder. • The full MLP layer in the decoder takes the features from the encoder and fuses them together. This all MLP decoder is composed of four main steps:

1. Firstly, multi-level features from the encoder are fed into the multi-layer perceptron layer to unify in the channel dimension.
2. Next, the features are up-sampled to ¼ of its size and are concatenated together.
3. Thirdly, a MLP layer is adapted to fuse the concatenated features.
4. Finally, another MLP layer takes these fused features to predict the segmentation mask.

B. DAFormer

Training a neural network for semantic segmentation usually requires expensive pixel-wise annotations of real-world images. Therefore, it is more desirable to exploit other domains that are easier to annotate, such as synthetic data. However, a model trained on the source domain typically experiences a performance drop when applied to the target domain. The goal of Unsupervised Domain Adaptation (UDA) is to increase the performance over the target domain by using unlabeled target images.

Many state-of-the-art UDA methods are based on self-training. The network is trained using ground truth labels for source images and pseudo-labels for target images. The pseudo-labels are generated by taking confident predictions of a teacher. The teacher network is an exponential moving average of the student for temporally stable predictions. In that way, the networks are iteratively adapted to the target domain. Previous UDA methods mostly use a DeepLabV2 network architecture. However, DeepLabV2 is significantly outperformed by more recent networks in supervised semantic segmentation.

DAFormer [11] studies the influence of the network architecture on UDA and compiles a more sophisticated architecture, representing a major advance in UDA.

Fig. 1. Qualitative results for semantic segmentation on the National Dataset for Indian Roads. Example predictions showing a better recognition of most classes by DAFormer as opposed to SegFormer. Compared to SegFormer, the DAFormer model predicts the Ground Truth masks with substantially finer details near object boundaries. It also reduces long-range errors as highlighted in white in images 1, 4, and 6.

TABLE I. GTA5 TO CITYSCAPES

Models	Classes and Score																			
	Road	S.walk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
CBST [6]	91.8	53.5	80.5	32.7	21	34	28.9	20.4	83.9	34.2	80.9	53.1	24	82.7	30.3	35.9	16	25.9	42.8	91.8
DACS [7]	89.9	39.7	87.9	30.7	39.5	38.5	46.4	52.8	88	44	88.8	67.2	35.8	84.5	45.7	50.2	0	27.3	34	89.9
CorDA [8]	94.7	63.1	87.6	30.7	40.6	40.2	47.8	51.6	87.6	47	89.7	66.7	35.9	90.2	48.9	57.5	0	39.8	56	94.7
BAPA [9]	94.4	61	88	26.8	39.9	38.3	46.1	55.3	87.8	46.1	89.4	68.8	40	90.2	60.4	59	0	45.1	54.2	94.4
ProDA [10]	87.8	56	79.7	46.3	44.8	45.6	53.5	53.5	88.6	45.2	82.1	70.7	39.2	88.8	45.5	59.4	1	48.9	56.4	87.8
DAFormer [11]	95.7	70.2	89.4	53.5	48.1	49.6	55.8	59.4	89.9	47.9	92.5	72.2	44.7	92.3	74.5	78.2	65.1	55.9	61.8	95.7
HRDA [12]	96.4	74.4	91	61.6	51.5	57.1	63.9	69.3	91.3	48.4	94.2	79	52.9	93.9	84.1	85.7	75.9	63.9	67.5	96.4

TABLE II. SYNTHIA TO CITYSCAPES

Models	Classes and Score																			
	Road	S.walk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
CBST [6]	68	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	-	78.3	60.6	28.3	81.6	-	23.5	-	18.8	39.8	68
DACS [7]	80.6	25.1	81.9	21.5	2.9	37.2	22.7	24	83.7	-	90.8	67.6	38.3	82.9	-	38.9	-	28.5	47.6	80.6
BAPA [8]	91.7	53.8	83.9	22.4	0.8	34.9	30.5	42.8	86.6	-	88.2	66	34.1	86.6	-	51.3	-	29.4	50.5	91.7
CorDA [9]	93.3	61.6	85.3	19.6	5.1	37.8	36.6	42.8	84.9	-	90.4	69.7	41.8	85.6	-	38.4	-	32.6	53.9	93.3
ProDA [10]	87.8	45.7	84.6	37.1	0.6	44	54.6	37	88.1	-	84.4	74.2	24.3	88.2	-	51.1	-	40.5	45.6	87.8
DAFormer [11]	84.5	40.7	88.4	41.5	6.5	50	55	54.6	86	-	89.8	73.2	48.2	87.2	-	53.2	-	53.9	61.7	84.5
HRDA [12]	85.2	47.7	88.8	49.5	4.8	57.2	65.7	60.9	85.3	-	92.9	79.4	52.8	89	-	64.7	-	63.9	64.9	85.2

TABLE III. NATIONAL DATASET FOR INDIAN ROADS

Models	Classes and Score																			
	Road	S.walk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
CBST [6]	79.3	45.4	69.2	27.9	17.7	28.1	26.6	17.4	73.5	30.0	67.5	42.1	20.4	72.9	27.1	31.5	13.4	22.8	38.3	79.2
BAPA [8]	84.6	54.2	79.2	23.8	36.1	34.0	42.9	49.7	77.4	41.2	78.6	62.3	35.6	80.8	51.9	52.1	0	41.0	48.5	84.6
ProDA [10]	82.6	52.0	74.2	43.9	43.4	42.8	47.5	49.5	82.3	42.0	76.3	65.7	36.5	82.2	42.3	55.4	0.9	45.7	52.4	82.6
DAFormer [11]	90.9	66.9	84.3	51.3	45.7	47.6	53.2	57.1	85.9	45.3	87.6	69.1	42.8	87.8	70.7	74.4	61.8	53.4	59.0	90.8

It improves the state-of-the-art performance by a significant margin of 10.8 mIoU on GTA→Cityscapes. A hierarchical transformer is utilized for the encoder, which is revealed to be more domain-robust than the predominant CNNs.

For the decoder, a context-aware feature fusion is used, which utilizes domain-robust context clues from different encoder levels. Compared to the DeepLabV2 architecture, DAFormer significantly reduces the performance gap between UDA and the supervised oracle. In many cases, the source dataset is imbalanced as some rare classes appear only in a few images. Therefore, the performance of these rare classes heavily depends on the random seed of the data sampling. By frequently sampling images with rare classes, the network can learn them more stably, which improves the quality of pseudo-labels and reduces confirmation bias.

During UDA, the network overfits the source domain, and difficult classes of the target domain are not distinguished clearly. Therefore, a Thing-Class ImageNet Feature Distance is introduced to regularize the source training. DAFormer shows significant successive improvements of the proposed components over a strong UDA baseline. In particular, it learns even difficult classes that previous methods struggled with.

V. RESULTS

The observations made while applying the DAFormer model are as follows:

The DAFormer struggles to deal with shadows, particularly if they cover the part of the road right in front of the vehicle. It is not effective at dividing the road into proper segments if the sidewalks or walkways are not well-defined. The model is vulnerable to creating inaccurate segmentation maps in situations where there is too much light exposure, particularly on roads that are overexposed to sunlight. The DAFormer also fails to recognize or segment road signs incorrectly. The observations made while using the SegFormer model are:

It frequently makes incorrect identifications and classifications for objects that are far away in the frame, particularly for vehicles and humans. The model struggles to recognize the road surface or sky if the image is not clear enough, as seen in an example where the ground is partially segmented. Shadows or bright patches of light present in the image can cause the model to leave the affected area unclassified. In some cases, a single vehicle may be divided into multiple classes due to sharp boundaries or margins in the image.

The results of the benchmark of previous models on various classes and shown in Table 1, Table 2, and Table 3. Fig. 1 shows the inference results for some images from the National Dataset for Indian Roads.

VI. CONCLUSION

The task of semantic segmentation of Indian road scenes is difficult due to the complexity and variety of these scenes, as well as the lack of sufficient labeled data. Unsupervised domain adaptation has proven to be a promising solution by fine-tuning pre-trained models on the target domain using limited labeled data. This approach has shown positive results for different types of Indian road scenes, with the ability toaccurately segment road surfaces and objects like vehicles, pedestrians, and buildings. The unsupervised domain adaptation also increases the generalization of the models to new scenes, which is crucial for practical applications. However, there are still some challenges to be overcome, such as the models' incapacity to fully capture variations like weather conditions, etc.

Furthermore, further research is necessary to assess the robustness and scalability of unsupervised domain adaptation for semantic segmentation of Indian road scenes. In conclusion, unsupervised domain adaptation is a hopeful method for semantic segmentation of Indian road scenes. It effectively uses pre-trained models and limited labeled data to enhance the performance and generalization of the models. Despite the existing limitations, this field continues to evolve with advancements in computer vision, deep learning, and the availability of large labeled datasets, which could result in even better and more reliable models for applications like self-driving cars, traffic monitoring, and smart city planning.

VII. References

[1] Dewangan, D. K., & Sahu, S. P. (2021). Road detection using semantic segmentation-based convolutional neural network for intelligent vehicle system. In *Data engineering and communication technology* (pp. 629-637). Springer, Singapore.

[2] Baheti, B., Gajre, S., & Talbar, S. (2019, October). Semantic scene understanding in unstructured environment with deep convolutional neural network. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)* (pp. 790-795). IEEE.

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

[4] Chen, Y., Li, W., & Van Gool, L. (2018). Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7892-7901).

[5] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, *34*, 12077-12090.

[6] Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: ECCV. pp. 289–305 (2018).

[7] Tranheden, W., Olsson, V., Pinto, J., Svensson, L.: DACS: Domain Adaptation via Cross-domain Mixed Sampling. In: WACV. pp. 1379–1389 (2021).

[8] Wang, Q., Dai, D., Hoyer, L., Fink, O., Van Gool, L.: Domain adaptive semantic segmentation with self-supervised depth estimation. In: ICCV. pp. 8515–8525 (2021).

[9] Liu, Y., Deng, J., Gao, X., Li, W., Duan, L.: Bapa-net: Boundary adaptation and prototype alignment for cross-domain semantic segmentation. In: ICCV. pp.8801–8811 (2021).

[10] Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., Wen, F.: Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In: CVPR. pp. 12414–12424 (2021).

[11] Hoyer, L., Dai, D., Van Gool, L.: DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In: CVPR (2022).

[12] Hoyer, L., Dai, D., & Van Gool, L. (2022). HRDA: Context-Aware High-Resolution Domain-Adaptive Semantic Segmentation. arXiv preprint arXiv:2204.13132.