Deep Learning Approaches in Lane Detection

Lane detection algorithms involving conventional algorithms like Hough transform algorithm, RANSAC fitting algorithm solve the problem to a limited extent and perform well in case of fewer vehicles and better road conditions.
Deep learning techniques have performed much better. Following is a review of different deep learning approaches that have been developed in lane detection.

Datasets for lane detection:
Caltech Lanes Dataset contains 1,225 images taken from four different places.
Road Marking Dataset contains 1,443 images manually labeled into 11 classes of road markings
COCO Dataset
The Lane Marker Dataset
Brook Roberts, Sebastian Kaltwang , A dataset for lane instance segmentation in urban environments, 2018
CULane dataset: It is collected by cameras mounted on six different vehicles driven by different drivers in Beijing. More than 55 hours of videos were collected and 133,235 frames were extracted. We have divided the dataset into 88880 for training set, 9675 for validation set, and 34680 for test set. The test set is divided into normal and 8 challenging categories-Nomal, crowded, night, no line, shadow, arrow, dazzle light, curve, crossroad
TUSimple dataset: The dataset contains images in:

1. Good and medium weather conditions
2. Different daytime
3. 2-lane/3-lane/4-lane/ or more highway roads
4. Different traffic conditions

BDD100K

Models

- Lane detection based on **combined Convolutional Neural Networks (CNN)** and **Random Sample Consensus (RANSAC) algorithm**
  The model proposed calculate edges in an image using a hat shape kernel and then detects lanes using CNN combined with RANSAC. This helps in dealing with complex road scenes that include road side trees, fence, intersections etc.
  In the pre-processing step, noise is reduced by blurring and edge detection. Blurred image is obtained by 5x5 gaussian smoothing function which helps in reducing environment noise. A hat shaped kernel is used to calculate edges of the blurred image. When the RANSAC fails to find a road lane from this image, a trained CNN provides a candidate of road lane. RANSAC is then repeatedly applied to candidates of road lanes obtained from CNN. The CNN used has 2 subsampling layers, 3 convolutional layers and a multilayer perceptron having 3 fully connected layers.

Input image->Convolutional layer1->Subsampling layer1->Convolutional layer2->Subsampling layer2->Convolutional layer3->Multilayer perceptron->Output image

Performance evaluation: The model was tested on three complex conditions:
Case1: detecting more than three lines
Case2: changing of lane position is too big
Case3: distance of vanishing points of left and right lanes is too big

Cases	Corrected detection(more than half region of detected lane overlaps with target lane)	MIssed detection(less than half region of detected lane overlaps with target lane)	False detection(detected lane never overlaps with target lane)
Case1	94.7%	5.1%	0%
Case2	93.9%	4.9%	1.2%
Case3	93.2%	4.5%	2.3%

- **DVCNN**
DVCNN model involves combining the front-view image and the top-view one for lane detection. The front-view image helps in excluding moving vehicles, barriers while the top-view one extracts club-shaped structures similar to the lane lines. In addition, a global optimization function is designed which takes the lane line probabilities, lengths, widths, orientations and the amount into account. From the top view image, different lengths are normalized so that lane lines can be recognized in global context. From the front-view image, it is convenient to utilize information such as the context and area which means that the lane lines are recognized in the local space.
The data layer accepts front view and top view patches as inputs whose sizes are normalised. The 2 sub-networks are composed of several layers of convolution, ReLu, pooling and fully connected layer. The 2 patches are then concatenated.
The dataset used is composed of 47 batch images where there are 20,000 images in each batch. Expressways, streets, country roads, avenues are included in order to ensure completeness.

Performance Evaluation: On an average recall ratio (detected ground truths /ground truths) obtained is 92.8% and average precision (detected ground truths / detected all) ratio is 95.49%. Combined CNN and RANSAC model produced an average recall ratio of 59.89% and average precision ratio of 93.67%.

- **VPGNet**

  This model aims to tackle lane detection in adverse weather conditions like rainy, low illumination conditions which cause low visibility of lanes. The dataset used contains images at daytime with different levels of precipitation (no rain, rain, heavy rain) and images at night time(contains images with low illumination). In such situations, humans can predict the locations of the lanes from global information such as nearby structures of roads or the flow of traffic. The approach used in this model involves predicting a **vanishing point** which could provide a global geometric context of a scene, which is important to infer the location of lanes and road markings. A "Vanishing Point (VP)" has been defined by the author as the nearest point on the horizon where lanes converge and disappear predictively around the farthest point of the visible lane. In simple words, it is a point where parallel lines in a three-dimensional space converge to a two-dimensional plane by a graphical perspective.

  VPGNet detects and recognizes lane and road markings as well as localizes vanishing point. The network thus performs 4 tasks: grid box regression, object detection, multi-label classification, and vanishing point prediction.

  VPGNet uses a data layer to induce grid-level annotation that enables training of both lane and road markings simultaneously. The corner points of lane and road markings are marked to form a closed polygon. When the closed region is filled, it is same as pixel-wise annotation. Here grid wise annotation is used where each grid is 8x8 pixels because lane and road markings can be easily bounded given a polygon and this technique is much faster than pixel-wise annotation. Points on the grid are regressed to the closest grid cell and combined by a multi-label classification task to represent an object.

  The network architecture involves the shared layers where each of the 4 tasks are occuring through the same layers and the branched layers where each of the tasks have different layers.

  Earlier methods involve vectorizing the spatial output of the network to predict the exact location of a Vanishing Point by using a softmax classifier. However, selecting exactly one point over the whole network's output size results in imprecise localization. Therefore, a quadrant mask has been used that divides the image into 4 sections which cover the structures of a global context, the intersection of which gives a Vanishing Point. To implement this, 5 channels have been defined for the output of vanishing point prediction task- 1 absence channel and 4 quadrant channels. Every pixel belongs to one of the 5 channels based on which the Vanishing Point is predicted.

In the first phase of training, only the Vanishing Point is predicted which helps in training the kernels to learn a global context of image. In the second phase, the tasks are further trained using the initialized kernels from the first phase. It is taken care that the learning rates of each task is balanced to avoid dependence on other tasks. Post processing involves point sampling, clustering, and lane regression for lane class and grid sampling, box clustering for road marking class.
Using multi task learning led to more neurons responding especially around boundary of roadways.

Performance evaluation: Lane detection F1 score on Caltech Dataset(Cordova1):88.4%
Lane detection F1 score on Caltech Dataset(Washington):86.9%

Lane detection F1 score on Benchmark Dataset:
Scenario1(Daytime-No rain): 87%
Scenario2(Daytime-Rain): 78.8%
Scenario3(Daytime-Heavy Rain): 76.8%
Scenario4(Night time): 74.3%

- **RFCN**
Region based fully convolutional networks is an object detection algorithm that has been used for lane detection. It shows an advantage in object localization. It involves multiple feature maps called position-sensitive score maps for detecting sub-regions of an object. RFCN takes in feature maps and applies convolution to create position-sensitive score maps for detecting sub-regions of an object. For each ROI, position-sensitive ROI pool is applied to generate a vote array. Array is averaged and softmax is used to classify the object. The input in RFCN is bounding box annotation, and its output is also bounding box information. Thus, RFCN provides the location information of the lane in the picture. The lane is fit in the bounding box and Hough transform is used to process the image in the bounding box range. The output is in pixel mode and it is compared with the grid-level ground truth to get the accuracy.
The dataset contains five types of data training. They are dashed white line, solid white line, dashed yellow line, double yellow solid line, solid yellow line.

Performance evaluation: Lane detection score

Model	Single white	Dashed white	Single yellow	Double yellow	Dashed yellow	Mean
RFCN	0.78	0.88	0.77	0.61	0.85	0.77

VPGNet	0.72	0.84	0.77	0.55	0.9	0.75

VPGNet shows a better accuracy.

ENet-SAD
Self attention distillation(SAD) is a method which allows a model to learn from itself and gains improvement without any additional supervision or labels. SAD allows a network to exploit attention maps derived from its own layers as the distillation targets for its lower layers. It helps in refining the lower layers based on rich contextual layers that deeper layers capture. This is done by the preceding layer mimicing the attention map of the deeper layer. ENet is an encoder-decoder structure comprised of E1~E4 (encoder of ENet) ,D1 and D2 (decoder of ENet). A small network P1 is added to predict the existence of lanes. The encoder module is shared to save memory space. Dilated convolution is added to replace the original convolution layers in the lane existence prediction branch to increase the receptive field of the network without increasing the number of parameters. In ENet-SAD model, SAD is applied to attention maps of ENet.

Performance Evaluation: The performance of ENet-SAD is evaluated on TUSimple, CULane and BDD100k datasets.

Dataset/Algorithm	TUSimple	CULane	BDD100K
	Accuracy	F1 Score	Accuracy
ENet-SAD	96.64%	70.8	36.56%

Benchmark
VPGNet gives the best performance on Caltech lanes dataset (Washington) and shows a F1 score of 86.9%. It also gives the best performance in Caltech lanes dataset (Cordova) and shows a F1 score of 88.4%. ENet-SAD has the best performance on BDD100K and TUSimple dataset. It shows a F1 score of 70.8 which comes next to the highest score of 71.6 by SCNN.

On CULane dataset

Approach	F1 Score
SCNN	71.6
ENet-SAD	70.8

ENet-label	68.8

On TUSimple dataset

Method	Accuracy
ENet-SAD	96.64%
Spatial CNN	96.53%
Pairwise pixel supervision + FCN	96.5%
LaneNet	96.4%
Discriminative Loss function	96.4%
ENet-Label	96.29%

On Caltech lanes Washington dataset

Method	F1 Score
VPGNet	0.869
Overfeat CNN Detector+ DBSCAN	0.861

On Caltech lanes Cordova dataset

Method	F1 Score
VPGNet	0.884
Overfeat CNN Detector + DBSCAN	0.866

Model we should use
ENet-SAD performs well in complex conditions like challenging weather conditions, illumination, different types of lanes, traffic conditions. Therefore, it would be the best model we could use for Indian roads.

References
1. Jing Feng, Xiaoyu Wu, Yu Zhang Lane detection based on deep learning IEEE, 2019
2. Seokju Lee, Junsik Kim, Jae Shin Yoon Vanishing Point Guide Network for lane and road marking detection and recognition, 2017
3. Bei He, Rui Ai, Xianpeng Lang Accurate and Robust lane detection based on dual view convolutional neural network
4. Jihun KIm, Minho Lee Robust Lane Detection based on convolutional neural network and random sample consensus, 2014
5. Yuenan Hou, Zheng Ma, Chunxiao Liu, Learning Lightweight Lane Detection CNNs by Self Attention Distillation, 2019