How to Extract Roads from Satellite Images ?

Saksham Jindal
Department of Electrical and Computer Engineering
University of California, San Diego
sjindal@ucsd.edu

1 Introduction

Road Extraction and land cover mapping, in general, from satellite imagery, is an important tool for monitoring and efficient map generation for an intelligent transportation system, automobile navigation and emergency support in times of natural disasters. With the recent advancement of technology in satellite imagery, remote sensing systems can provide data with very high spatial resolution which allows us to have high precision ground information and permit large-scale monitoring of roads. Automating road extraction plays an important role in dynamic spatial development and plays an important role in large scale mapping and urban planning. The following section will aim to give a brief walkthrough the training techniques adopted for experimentation as a part of converging on an approach to segment the satellite images. We use deep learning based semantic segmentation network architectures to train satellite images labelled with masks at a pixel level. The following paper experiments with mainly four training strategies while training the models: Data Preparation, Data Augmentation, Loss Functions, Schedulers, Optimizers, Network Architectures.

2 Method

2.1 Data Preparation

Massachusetts Roads Dataset [6] consists of 1171 aerial images of the state of Massushussets, each 1500 x 1500 pixels in size. We had only 804 labelled images in the training set which is split into a set of 643 images for training and 161 for validation. Further, the algorithms discussed are evaluated on a test set of 14 images. The pixel values in the images must be scaled prior to providing the images as input to a deep learning neural network model during the training or evaluation of the model. This ensures each input parameter (pixel, in the case) has a similar data distribution which makes convergence faster. Also, networks process inputs using small weight values and inputs with large integer values can slow or disrupts the learning process. The pixel intensity values in the images, present in 8-bit format, were normalised to bring the pixel values in the range 0-1 by dividing by 255. Further, the images processed by subtracting per-channel mean from pixel values calculated on the training set. Finally, the values obtained by subtracting per-channel mean were divided by per-channel standard deviation calculate on the training set.

This ensures that distribution of pixel values follows a normal distribution with mean 0 and standard deviation 1. In the current setup, per channel mean and per channel standard are pre-calculated for the training set and we obtain mean values and standard deviation values of (0.428, 0.431, 0.395) and (0.292, 0.285, 0.297) for RGB channels respectively.

2.2 Data Augmentation

Data Augmentation is a strategy that enables practitioners to significantly increase the diversity of data available, without actually collecting new data. It aims to address 2 important requirements while training supervised models - diversity of training data and amount of data. There are 2 ways which the data augmentation can be achieved in our pipeline - offline augmentation (perform all necessary

augmentations beforehand) and online augmentations (performing augmentation on a mini-batch just before feeding data to the model)

In this paper, we have performed online augmentations using albumentations library [1]. Following augmentations were applied on each mini-batch of training data before loading to feed into the model Random Crop (Images are randomly cropped at 512 x 512 pixel size), Random Horizontal Flip (with probability 0.5), Random Vertical Flip (with probability 0.5), Random Rotate (with probability 0.5), Transpose (with probability 0.5), Random Shift-Scale-Rotate, Random Brightness and Contrast added/removed, Random Gamma transformations and Random Blur (with probability of 1%)

An illustration of augmentation performed on an input image of size 1500 x 1500 pixels to get augmentation on output image size of 512 x 512 pixels can be seen in the following set of images.

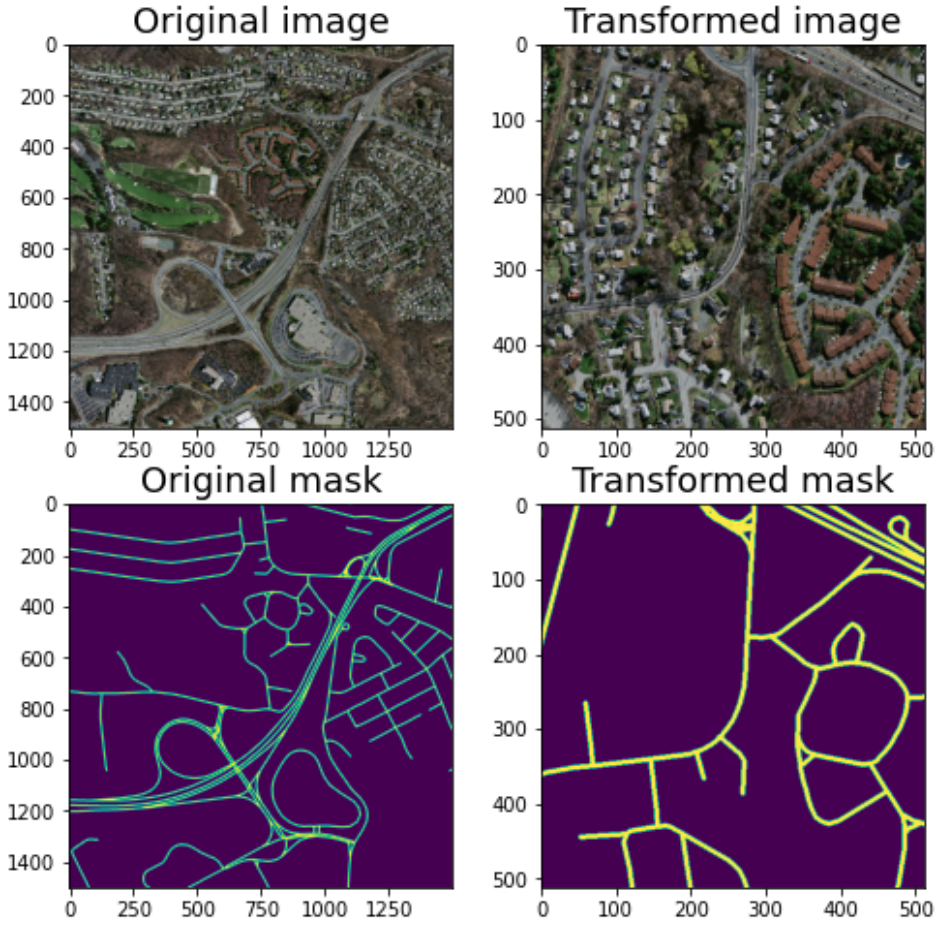


Figure 1: A comparison of original images and augmented images

2.3 Loss Functions

Semantic segmentation, a classification task performed on a pixel level, requires careful consideration of the choice of loss function or objective function in non-convex optimization. The selected loss function significantly influences the training process and the quality of the segmentation output.

One commonly used loss function for image segmentation is the Categorical Cross Entropy Loss. This loss function calculates the cross entropy between the predicted class probabilities and the ground truth labels on a pixel-wise basis. In the case of semantic segmentation, where background and road classes are assigned labels 0 and 1, respectively, the aim is to minimize the cross entropy loss to improve the accuracy of segmenting road pixels from the background.

However, when dealing with unbalanced datasets where certain classes are underrepresented, using a simple cross entropy loss can lead to issues. To address this problem, a Weighted Categorical Cross Entropy Loss can be utilized. This loss function applies weights to the different classes to account for the class imbalance. In the case of road segmentation, where the background class heavily outweighs the road class, weighting schemes like the inverse number of samples or the inverse of the square root of the number of samples can be employed to give more importance to the road class during training.

Another alternative is the Focal Loss, which introduces a modulating term to the cross entropy loss. This modulating term focuses the learning process on hard negative examples, allowing the model to pay more attention to challenging pixels. The scaling factor of the modulating term decreases as the model's confidence in the correct class prediction increases.

Additionally, the Dice Loss, also known as Jaccard Loss, can be used for image segmentation. The Dice Loss is based on the Dice coefficient, which measures the overlap between two samples. It ranges from 0 to 1, with a value of 1 indicating perfect and complete overlap. Since the goal is to maximize the dice coefficient or Intersection over Union (IOU), the Dice Loss is used as a proxy to approximate this objective. In the case of n-class segmentation, such as the road and background classes, the final Dice Loss is calculated separately for each class and then averaged to obtain a final score.

In summary, choosing an appropriate loss function is crucial for semantic segmentation tasks. Each loss function mentioned here has its advantages and considerations, such as addressing class imbalance, focusing on hard examples, or directly optimizing evaluation metrics like the Dice coefficient or IOU. Experimentation and evaluation with different loss functions can help determine the most suitable approach for achieving accurate and robust segmentation results for the given problem.

2.4 Optimizers and Schedulers

We use four different combinations of optimizers and schedulers used in machine learning or deep learning models.

First, we have Stochastic Gradient Descent (SGD) with Polynomial Learning Rate (LR) Scheduler. This combination involves using the SGD optimizer along with a polynomial LR scheduler. SGD is a popular optimizer that performs gradient descent by updating the model's parameters based on a small batch of randomly selected samples. The polynomial LR scheduler gradually decreases the learning rate over time using a polynomial function, which helps in fine-tuning the model's performance.

The second combination is Stochastic Gradient Descent (SGD) with Warm Restarts Scheduler [5]. Again, SGD is the chosen optimizer, but this time it is paired with the Warm Restarts scheduler. The Warm Restarts scheduler periodically resets the learning rate to a higher value during training. This resetting helps the model to escape local minima and explore different areas of the loss landscape.

The third combination is Stochastic Gradient Descent (SGD) with One Cycle LR Scheduler [8]. It involves using the SGD optimizer along with the One Cycle LR scheduler. The One Cycle LR scheduler varies the learning rate within a single cycle of training. It starts with a low learning rate, gradually increases it to a maximum value, and then decreases it again. This technique aims to achieve faster convergence and better generalization.

The fourth combination is Adam Optimizer [4] with Polynomial Learning Rate Decay. Here, the Adam optimizer is employed, which is an adaptive learning rate optimization algorithm commonly used in deep learning. The learning rate of Adam decreases over time, following a polynomial decay. This decay allows the model to make larger updates initially and gradually reduce the step size to refine the parameters.

These combinations of optimizers and schedulers are used to improve the training process and enhance the performance of machine learning or deep learning models. They manipulate the learning rate to control the convergence, exploration, and generalization capabilities of the models, leading to better accuracy and efficiency. The choice of optimizer and scheduler depends on the specific task, dataset, and model architecture, and experimenting with different combinations can help find the most suitable setup for the given problem.

2.5 Network Architectures

In the field of semantic segmentation, various network architectures have been developed to improve the accuracy and efficiency of pixel-level classification. Let's discuss three popular architectures: UNet-Resnet50, PSPNet (Resnet-50 backbone), and Deeplab V3+.

UNet-Resnet50 architecture combines the strengths of two groundbreaking models: UNet [7] and ResNet. UNet introduced the concept of long skip connections between the contracting and expanding paths, allowing for better information flow and feature extraction at different levels. On the other hand, ResNet introduced residual blocks with skip connections, enabling the construction of deeper networks. By combining these two approaches, UNet-Resnet50 achieves state-of-the-art performance in segmentation tasks. The architecture leverages the benefits of both UNet and ResNet, with skip connections for feature fusion and the ability to learn complex hierarchical representations.

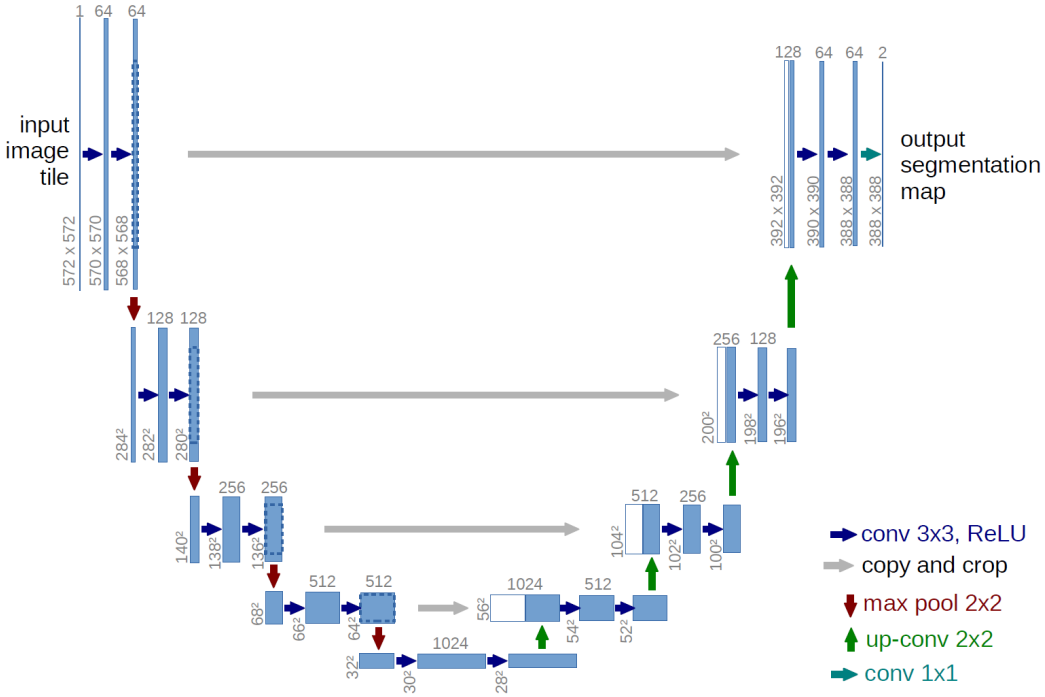


Figure 2: A specimen of UNet (with VGG backbone) from original paper [7]

PSPNet (Pyramid Scene Parsing Network) [9] is another prominent semantic segmentation architecture. It improves upon the Fully Convolutional Network (FCN) by incorporating global context information into the predictions. The encoder of PSPNet consists of dilated convolutions, which increase the receptive field and capture more contextual information. The key component of PSPNet is the Pyramid Pooling Module, which captures global context by pooling and aggregating information at multiple scales. This global context helps the model classify pixels based on the overall image context, leading to more accurate segmentations.

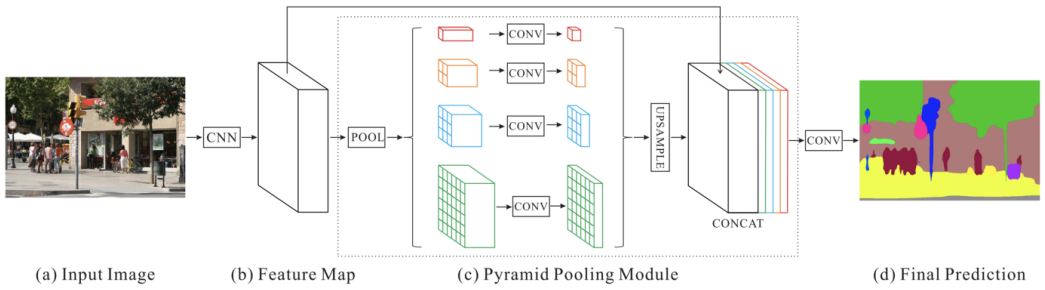
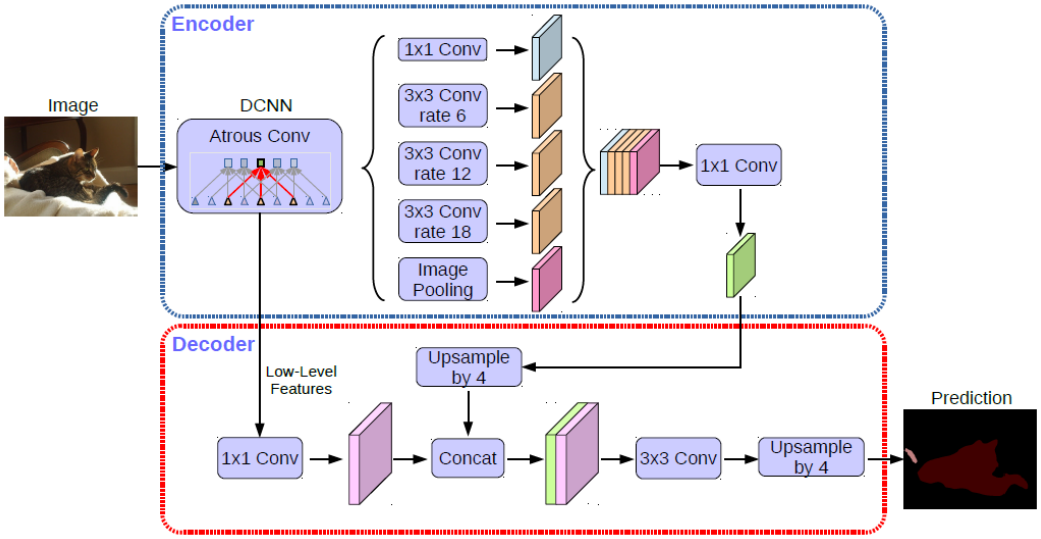


Figure 3: PSPNet Architecture

Deeplab V3+ [2] is a state-of-the-art architecture in the Deeplab series, which includes earlier versions like Deeplab V1, V2, and V3. Deeplab V3+ uses several innovative techniques to achieve superior performance. It incorporates Atrous Spatial Pyramid Pooling (ASPP), which encodes multi-scale contextual information by applying atrous convolutions at different rates and fusing the results. This allows the model to capture fine details and larger context simultaneously. Another important technique is depth-wise separable convolution, which separates the depth of the feature map and applies depth-wise convolution followed by 1x1 convolution for computational efficiency. In terms of the decoder, Deeplab V3+ introduces a different approach. Instead of upsampling the feature maps by a factor of 16 (as in previous versions), the upsampled features are first upsampled by a factor of 4 and then concatenated with low-level features from the encoder. This allows the model to preserve fine-grained details while upsampling, leading to sharper boundaries in the segmentation output.



2.6 Metrics

Intersection over Union (IoU)

The Intersection over Union (IoU), also known as the Jaccard index, is a metric used to measure the overlap between the predicted segmentation mask and the ground truth mask. It quantifies the percentage of overlap between the two masks, making it a suitable metric for segmentation tasks. The IoU is calculated by dividing the area of intersection by the area of union between the predicted and ground truth masks. In semantic segmentation, the IoU is often computed for each class separately and then averaged across all classes to obtain the mean IoU score. In the case of evaluating road segmentation, the IoU specifically for the road class is considered due to the high class imbalance in the dataset.

$$IoU = \frac{Area of Intersection}{Area of Union} \tag{1}$$

Pixel Accuracy

Pixel accuracy is a metric that measures the percentage of correctly classified pixels in the image. It provides a measure of how accurately the model predicts the class labels for individual pixels. Pixel accuracy is commonly reported for each class separately as well as globally across all classes. However, it can be misleading in the presence of class imbalance, as the metric may be biased towards the majority class or background class if it dominates the dataset.

Dice Score (or F1 score)

The Dice score, also known as the F1 score, is a metric commonly used in segmentation tasks to evaluate the performance of models. It is calculated as twice the area of overlap between the predicted and ground truth masks divided by the total number of pixels in both masks. The Dice score is similar to the IoU, as both metrics measure the similarity or overlap between two masks. It is particularly suitable for handling class imbalance in datasets and is defined as the harmonic mean of precision and recall.

$$Dice\ Score = \frac{2 \times Area of Intersection}{Total number of pixels in both masks} \tag{2}$$

Precision-Recall

Precision and recall are metrics used to evaluate the performance of binary classifiers in segmentation tasks. Precision measures the proportion of correctly predicted positive instances out of the total instances predicted as positive, while recall measures the proportion of correctly predicted positive instances out of the total actual positive instances. Precision and recall are important metrics in cases where the class of interest is imbalanced or where the focus is on correctly identifying positive instances.

$$Precision = \frac{True Positives}{True Positives + False Positives} \tag{3}$$

$$Recall = \frac{True Positives}{True Positives + False Negatives} \tag{4}$$

3 Experiments and Results

3.1 Choice of Loss Function

Over Experiment 1, Experiment 2 (weighted cross entropy loss), Experiment 3 (dice loss) and Experiment 4 (focal loss), the loss functions were varied with a Unet with Resnet-50 backbone (Figure 4).

During an initial analysis, it was observed that the ratio of pixels belonging to the road and background is 1:20 (**class imbalance**). The weighing of the cross entropy loss was experimented with the ratios 1:20 and $1/\sqrt{20}$ and it was observed that using inverse square root of pixel distributions exceeded results on the Class IOU (Road IOU) being monitored. Further, it was found to be performing much better than Dice Loss and Focal Loss. It can be argued that weighed cross entropy loss performs better than dice loss and focal loss because the former optimised for the overall segmentation IOU and focal loss has been reported to perform better when the class imbalance is of the order of 1:1000

3.2 Choice of Optimizer and Scheduler

The performance of Stochastic Gradient Descent with Cosine Annealing Warm Restarts (cyclic LR) , SGD with One CycleLR (superconvergence) and Adam were compared using Resnet-50 backbone of Unet and cross entropy loss weighted in 1:5 proportions. (Figure 5 and 6)

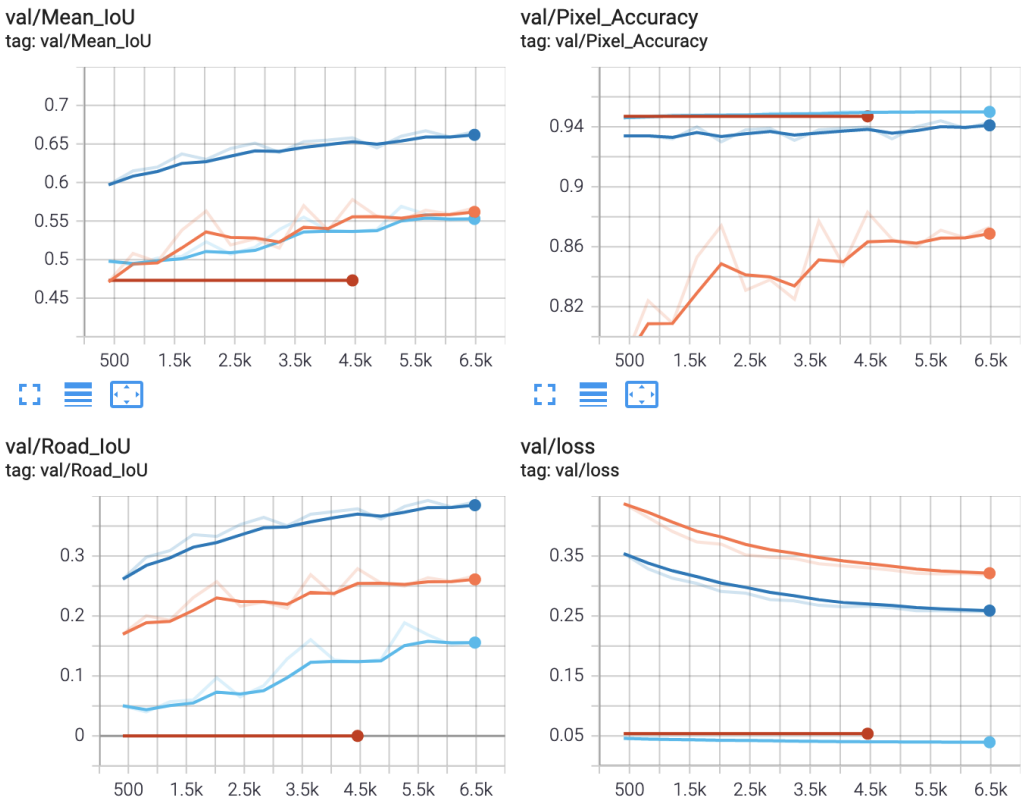


Figure 4: Exp. 1 (Weighted CE - 1:20) - Orange, Exp. 2 (Weighed CE - 1:5) - Blue, Exp. 3 Dice Loss, Exp. 4 - Focal Loss

For Experiment 5, using SGD with Cosine Annealing rates, maximum and minimum LR are important hyperparameters used in the experiments. They are calculated by using the "LR range test" as discussed in Leslie Smith's paper on Cyclic LR [8]. Also, the learning rate is "warm restarted" at 15%, 45% and 90% of the total iterations (found by trial-and-error)

For Experiment 6, SGD with One Cycle LR anneals the learning rate from an initial learning rate (base lr same as min lr in above) to some maximum learning rate (same max lr as above) and then from that maximum learning rate to some minimum learning rate (base lr divided by a factor of 25). The learning rate was increased from base lr to max lr during phase 1 (40% of iterations) and decreased to base LR/25

For Experiment 7, instead of "SGD-scheduler" combination, adaptive learning rate technique - Adam is used instead of SGD.

There are few interesting observations that could be made from the above training and validation plots (Figure 5 and 6). First, Warm Restarts Scheduler (Exp 5; Pink) converges much faster than One Cycle LR policy. Second, with careful selection of hyperparameters, both the schedulers seem to converge to, approximately, similar values. Finally, contrary to expectations, the performance of Adam pales in comparison to the above experiment (probably due to high max LR).

3.3 Choice of Architecture

Using SGD with Warm Restarts Scheduler and Weighed CE Loss (1:5), performance of UNet, PSPNet and Deeplabv3+ (all Resnet-50 backbones) were compared during the Experiments 5, 8 and 9 (Figure 7 and 8). We observe that performance of Deeplabv3+ far exceeds UNet and PSPNet on class-wise IOU calculated.

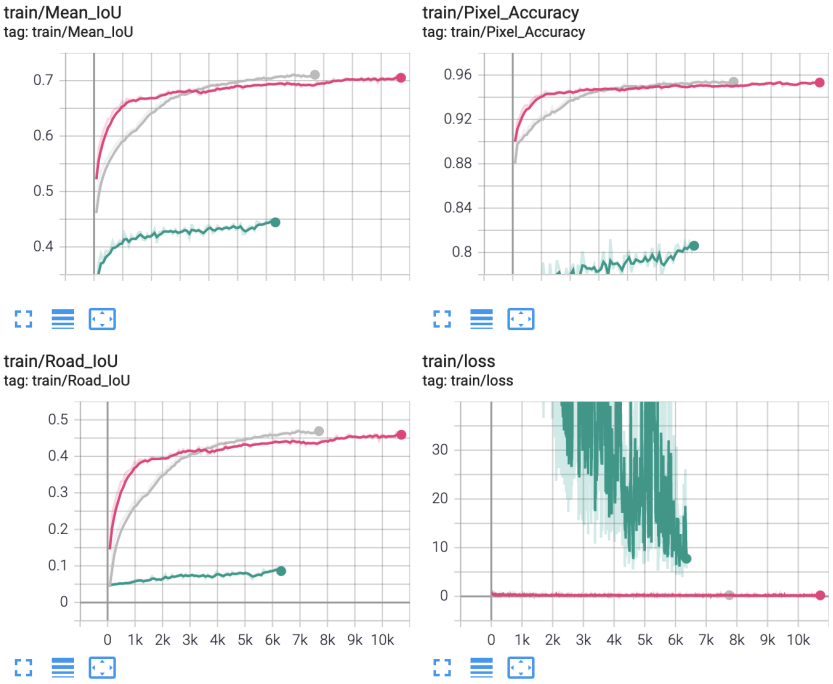


Figure 5: Results on training set with different optimiser and schedulers: Exp 5: Pink - SGD with Warm Restarts, Exp 6: Grey - SGD with One Cycle LR and Exp 7 : Grey : Adam optimizer - Training Plots

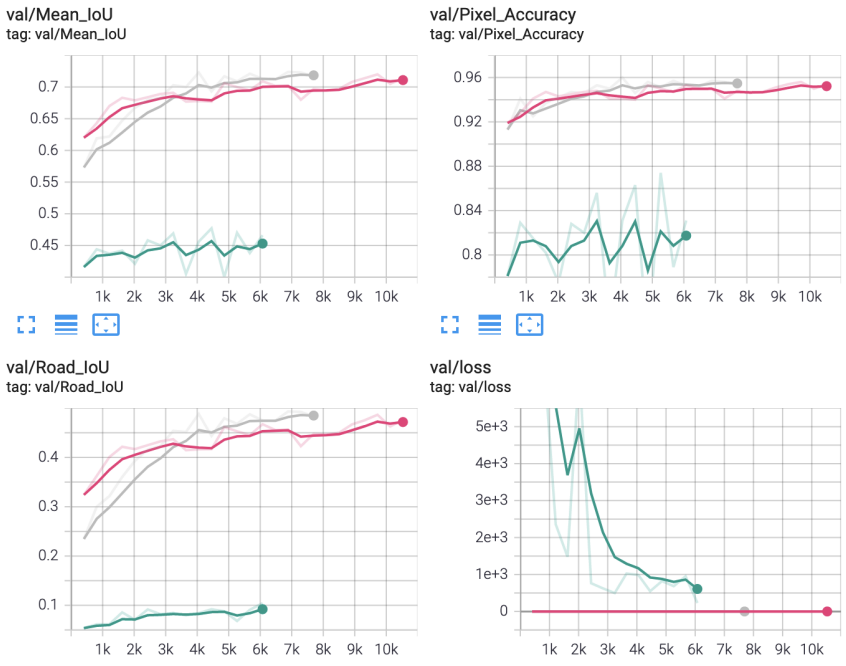


Figure 6: Results on training set with different optimiser and schedulers: Exp 5 : Pink - SGD with Warm Restarts, Exp 6: Grey - SGD with One Cycle LR and Exp 7 : Grey : Adam optimizer - Validation Plots

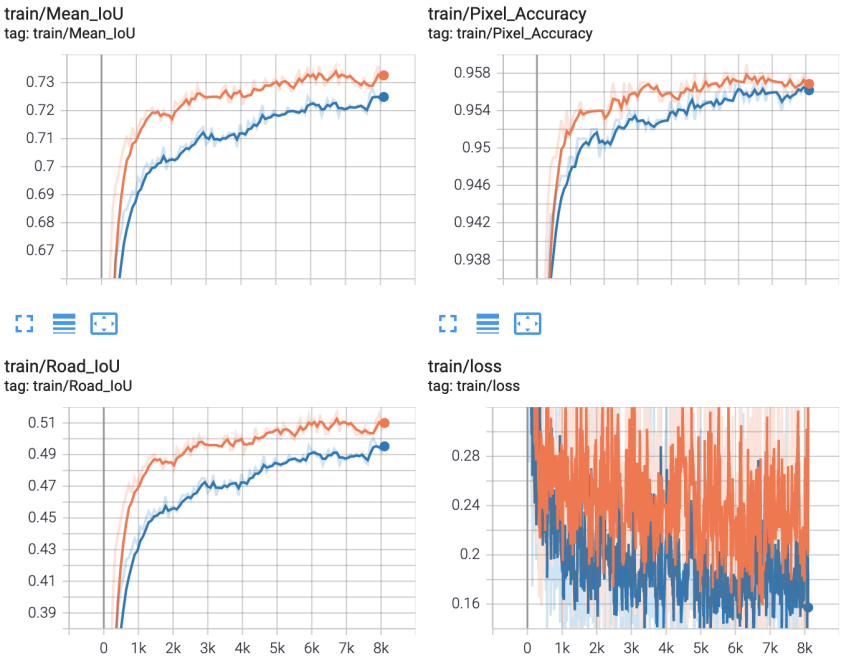


Figure 7: Results on training set for Exp 8 (Orange - PSPNet) and Exp 9 (Blue - Deeplabv3+)

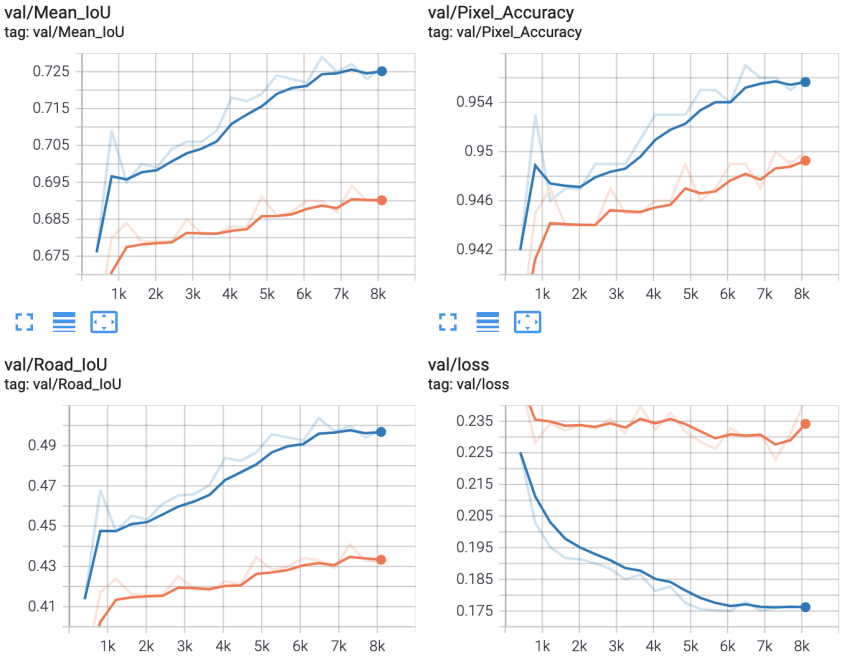


Figure 8: Results on training set for Exp 8 (Orange - PSPNet) and Exp 9 (Blue - Deeplabv3+)

3.4 Experiment summary

An overall summary of the experiments has been documented in the Table 1 and 2.

Table 1: Experiment Summary - Table 1

Experiment	Architecture	Loss	Optimizer Scheduler
Exp 1	Unet-Resnet50	Weighted CE (1:20)	SGD with Poly LR scheduler
Exp 2	Unet-Resnet50	Weighted CE (1:5)	SGD with Poly LR scheduler
Exp 3	Unet-Resnet50	Dice Loss	SGD with Poly LR scheduler
Exp 4	Unet-Resnet50	Focal Loss	SGD with Poly LR scheduler
Exp 5	Unet-Resnet50	Weighted CE (1:5)	SGD with Cosine Annealing Warm Restarts
Exp 6	Unet-Resnet50	Weighted CE (1:5)	SGD with OneCycle LR
Exp 7	Unet-Resnet50	Weighted CE (1:5)	Adam
Exp 8	PSPNet	Weighted CE (1:5)	SGD with Cosine Annealing Warm Restarts
Exp 9	Deeplabv3+	Weighted CE (1:5)	SGD with Cosine Annealing Warm Restarts

Table 2: Experiment Results - Table 2

Experiment	Road IOU	Pixel Accuracy	Dice Score	Precision	Recall
Exp 1	0.283	0.847	0.440	0.296	0.859
Exp 2	0.396	0.927	0.565	0.488	0.678
Exp 3	8.4e-13	0.929	8.4e-13	1.00	8.45
Exp 4	0.2098	0.934	0.336	0.600	0.248
Exp 5	0.502	0.945	0.667	0.583	0.783
Exp 6	0.497	0.932	0.657	0.579	0.781
Exp 7	0.104	0.839	0.187	0.144	0.2761
Exp 8	0.4528	0.9380	0.6222	0.5423	0.737
Exp 9	0.5101	0.9450	0.6742	0.574	0.8201

Figure 9: Visualization of Predictions and Comparison from Experiment 9. Image (left), Ground Truth (center) and Prediction (right)

3.5 Visualization of Results

We present the visualization of results in Figure 9 using configurations of Experiment 9 (refer Table 1)

3.6 Threshold Optimization

Threshold optimization is an important exercise in classification based predictive modelling, especially in cases of imbalanced data. Generally, using a default threshold (0.5) results in a sub-optimal performance for imbalanced datasets. A simple approach of doing so is to tune the threshold used to map probabilities to class labels.

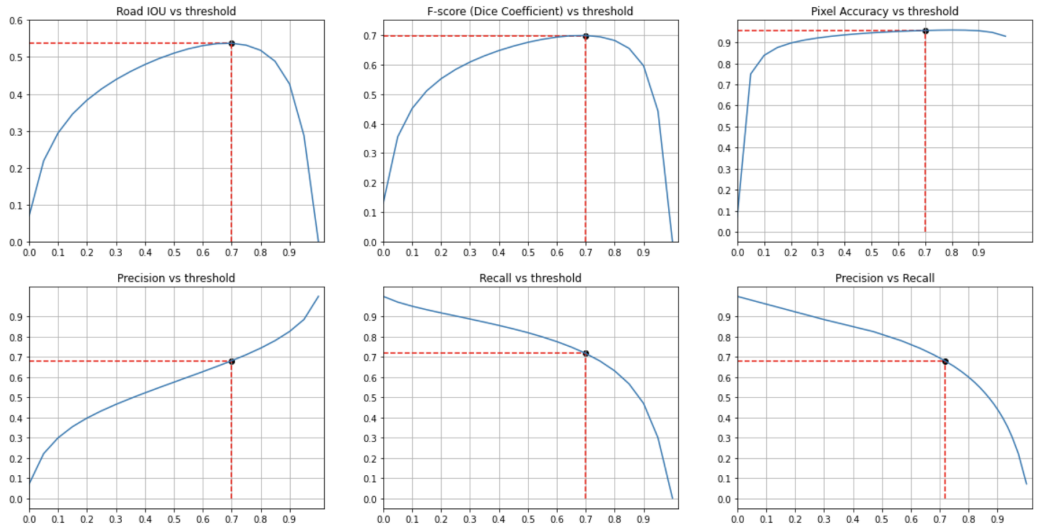


Figure 10: Variation of different metrics across threshold values

3.7 Notes on Post-processing

We can observe in the visualisation of predictions, there are certain “open holes"and breaks in a continuous stretch of road as documented in the above predictions. To obtain a road network, it could be an important exercise to fill in the blind spots in the mask extracted and thus obtain the masks.

As noted in the [3], the authors use morphological transformations (erosion and dilation) in image processing for post-processing. The process of erosion involves removal of foreground object and process of dilation involves adding foreground object around the boundary.While, erosion is used for diminish the features, dilation is used for accentuate features.

Morphological Opening - The opening operation erodes an image and then dilates the eroded image. This could be useful for noise removal

Morphological Closing -The closing operation dilates an image and then erodes the dilates image.. It could be useful for closing small holes in the pixels of the road

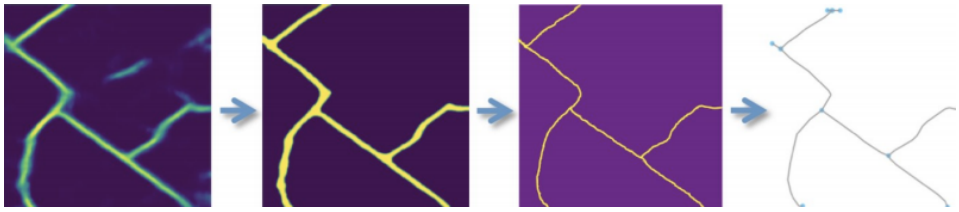


Figure 11: Approach as discussed in. The above figure is from the same paper for post-processing

Further, a **practical** extension of the above work is the extraction road network (graph of roads as vertices and junctions as nodes). Authors in the same paper, use skeletonization as a process of reducing road pixels or *thinning* to skeletal shape that preserves the extent and connectivity of the original region. Authors further process the above skeletons to obtain graphs structure.

4 Conclusion

Satellite imagery analysis is crucial for intelligent remote monitoring and building system to generate efficient routes. The use cases for the technology spans across humanitarian to military.

In the paper, methods to road segmentation maps from satellite images has been discussed. It was observed that using carefully tuned weights in cross entropy loss, hyperparameters in the optimiser and scheduler (SGD with warm restarts) with Deeplabv3+ yields a class wise IOU of 0.51 on the testing dataset. Further, the paper discusses post-processing techniques - thresholding, closing, opening and skeletonization to obtain road network graphs.

References

[1] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2):125, feb 2020.

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.

[3] Adam Van Etten. City-scale road extraction from satellite imagery v2: Road speeds and travel times. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, mar 2020.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[5] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2016.

[6] Volodymyr Mnih. Machine learning for aerial image labeling. 2013.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[8] Leslie N. Smith. Cyclical learning rates for training neural networks, 2015.

[9] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2016.