Image Semantic Segmentation on MSRC V2 Dataset: A Comparative Study from Traditional Methods to Deep Learning

Author Name
Affiliation
email@example.com

Abstract

Image semantic segmentation is a fundamental task in computer vision, aiming to assign a categorical label to each pixel in an image. In this work, we present a comprehensive comparative study of image semantic segmentation methods on the MSRC v2 dataset, spanning from traditional machine learning approaches to modern deep learning techniques. For traditional methods, we implement a pipeline combining superpixel-based feature extraction with Random Forest (RF) and Gaussian Mixture Model (GMM) classifiers, enhanced by Markov Random Field (MRF) optimization for spatial smoothing. For deep learning methods, we evaluate U-Net and DeepLabV3 architectures with various training strategies including data augmentation, weighted loss functions, and learning rate scheduling. Our experiments demonstrate that while traditional methods with MRF optimization achieve reasonable segmentation quality (PA: 75.48%, mIoU: 49.84%), deep learning approaches significantly outperform them, with DeepLabV3 achieving state-of-the-art results (PA: 89.71%, mIoU: 67.62%). We provide detailed analysis of class imbalance handling, evaluation metrics, and adaptability to varying image sizes. Our findings provide valuable insights for practitioners choosing between traditional and deep learning approaches for semantic segmentation tasks.

1. Introduction

Semantic segmentation, the task of assigning a semantic class label to each pixel in an image, is one of the most fundamental problems in computer vision [14]. It has wide-ranging applications including autonomous driving [7], medical image analysis [15], scene understanding [17], and robotics [9]. Unlike image classification which assigns a single label to an entire image, or object detection which localizes objects with bounding boxes, semantic segmentation requires pixel-level understanding of the visual scene.

The evolution of semantic segmentation methods can be broadly categorized into two eras: traditional machine learning methods and deep learning approaches. Traditional methods typically rely on hand-crafted features combined with graphical models such as Markov Random Fields (MRFs) or Conditional Random Fields (CRFs) to enforce spatial consistency [11, 16]. While these methods provide interpretable results and work reasonably well on constrained datasets, they often struggle with complex scenes and require careful feature engineering.

The advent of deep convolutional neural networks (CNNs) has revolutionized semantic segmentation. Pioneering work by Long et al. [14] introduced Fully Convolutional Networks (FCNs), enabling end-to-end learning for dense prediction. Subsequent architectures such as U-Net [15], with its encoder-decoder structure and skip connections, and DeepLabV3 [5] with atrous convolutions and Atrous Spatial Pyramid Pooling (ASPP), have pushed the boundaries of segmentation performance.

In this paper, we present a comprehensive comparative study of semantic segmentation methods on the Microsoft Research Cambridge (MSRC) v2 dataset [16]. Our contributions are as follows:

1. We implement and evaluate a complete traditional segmentation pipeline combining superpixel-based feature extraction, Random Forest and GMM classifiers, with MRF optimization for spatial smoothing.
2. We train and evaluate modern deep learning architectures including U-Net and DeepLabV3, with careful attention to training strategies such as data augmentation, weighted loss functions, and learning rate scheduling.
3. We provide detailed quantitative comparison using multiple evaluation metrics (mIoU, Pixel Accuracy, Mean Pixel Accuracy) with proper handling of void regions.
4. We analyze the adaptability of different methods to varying image sizes and the effectiveness of class imbalance handling strategies.

2. Related Work

2.1. Traditional Segmentation Methods

Early approaches to semantic segmentation relied heavily on hand-crafted features and probabilistic graphical models. Shotton et al. [16] introduced TextonBoost, which combines texture, color, and location features with boosting classifiers and CRF smoothing on the MSRC dataset. Ladicky et al. [11] proposed associative hierarchical CRFs that integrate multiple segmentation hypotheses.

Superpixel-based methods have been widely adopted as a preprocessing step to reduce computational complexity while preserving object boundaries [1]. The Simple Linear Iterative Clustering (SLIC) algorithm [1] provides efficient superpixel generation with good boundary adherence. Once superpixels are generated, various classifiers including Random Forests [3], Support Vector Machines [8], and Gaussian Mixture Models can be applied to classify each superpixel.

Markov Random Fields (MRFs) and Conditional Random Fields (CRFs) are commonly used to enforce spatial consistency in segmentation results [12]. These graphical models combine unary potentials (individual pixel/superpixel predictions) with pairwise potentials (spatial relationships) to produce smoother segmentations.

2.2. Deep Learning for Semantic Segmentation

Fully Convolutional Networks (FCNs) [14] marked the beginning of the deep learning era for semantic segmentation by replacing fully connected layers with convolutional layers, enabling dense prediction at arbitrary input sizes.

U-Net [15], originally designed for biomedical image segmentation, introduced a symmetric encoder-decoder architecture with skip connections that concatenate feature maps from the encoder to the decoder. This design helps preserve spatial information lost during downsampling and has been widely adopted in various domains.

The DeepLab family [4–6] introduced atrous (dilated) convolutions to enlarge the receptive field without increasing parameters. DeepLabV3 [5] further incorporates Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale context through parallel atrous convolutions at different rates, combined with batch normalization.

2.3. Class Imbalance in Segmentation

Class imbalance is a common challenge in semantic segmentation, where some classes occupy significantly more pixels than others [2]. Common solutions include weighted cross-entropy loss, which assigns higher weights to under-represented classes, focal loss [13] that down-weights easy examples, and sampling strategies such as oversampling minority classes.

3. Methodology

3.1. Dataset: MSRC v2

The Microsoft Research Cambridge (MSRC) v2 dataset [16] is a widely used benchmark for semantic segmentation containing 591 images with pixel-level annotations. For our binary segmentation task, we categorize the 21 semantic classes into two groups:

- **Natural**: grass, tree, cow, horse, sheep, bird, water, sky, mountain, flower
- **Man-made**: building, airplane, face, car, bicycle, sign, book, chair, road, cat, dog, body

The dataset includes void regions (labeled as 255) that should be ignored during evaluation. We split the data into training (60%), validation (20%), and test (20%) sets.

3.2. Traditional Methods

Our traditional segmentation pipeline consists of three main components: superpixel generation, feature extraction and classification, and MRF optimization.

3.2.1. Adaptive Superpixel Generation

We employ the SLIC (Simple Linear Iterative Clustering) algorithm [1] for superpixel generation. To handle images of varying sizes, we implement an adaptive approach that dynamically adjusts the number of superpixels based on image dimensions:

$$n_{\text{segments}} = \min(\max(n_{\text{pixels}} \cdot \rho, 100), 900) \tag{1}$$

where $n_{\text{pixels}} = H \times W$ is the total number of pixels and $\rho = 0.006$ is the target density (approximately one superpixel per 167 pixels).

3.2.2. Feature Extraction

For each superpixel, we extract a 16-dimensional feature vector:

- **Color features**: Mean and standard deviation of LAB channels (6D), mean of HSV channels (3D)
- **Texture features**: Mean and standard deviation of gradient magnitude (2D), edge density (1D)
- **Spatial features**: Superpixel area ratio (1D), normalized center coordinates (2D)
- **Global features**: LAB contrast with image mean (1D)

3.2.3. Classification

We evaluate two classifiers:

Random Forest (RF) [3]: An ensemble of 1000 decision trees with balanced class weights to handle imbalance. We use `max_features='sqrt'` and `min_samples_leaf=8` to prevent overfitting.

Gaussian Mixture Model (GMM): For each class, we fit a GMM with full covariance matrices. Classification is performed by computing the posterior probability given the learned class-conditional distributions.

3.2.4. MRF Optimization

To enforce spatial consistency, we apply Markov Random Field optimization to the classifier outputs. The energy function is:

$$E(\mathbf{x}) = \sum_i \phi_i(x_i) + \lambda \sum_{(i,j)\in\mathcal{N}} \psi_{ij}(x_i, x_j) \tag{2}$$

where $\phi_i(x_i) = -\log P(x_i|\mathbf{f}_i)$ is the unary potential from classifier predictions, $\mathcal{N}$ is the set of neighboring superpixel pairs, and the pairwise potential is:

$$\psi_{ij}(x_i, x_j) = [x_i \neq x_j] \cdot \exp\left(-\frac{\|\mathbf{c}_i - \mathbf{c}_j\|^2}{2\sigma^2}\right) \tag{3}$$

where $\mathbf{c}_i$ and $\mathbf{c}_j$ are mean colors of adjacent superpixels. The parameter λ controls the smoothness strength, and we use grid search to find optimal values for λ and σ.

3.3. Deep Learning Methods

3.3.1. U-Net Architecture

U-Net [15] follows an encoder-decoder structure with skip connections. The encoder consists of repeated blocks of two 3×3 convolutions (with ReLU activation and batch normalization), followed by 2×2 max pooling. The decoder mirrors this structure with transposed convolutions for upsampling. Skip connections concatenate encoder features to decoder features at corresponding resolutions. Our implementation has approximately 7.76M parameters.

3.3.2. DeepLabV3 Architecture

We use DeepLabV3 [5] with a ResNet-50 [10] backbone pretrained on ImageNet. The ASPP module applies parallel atrous convolutions with rates (1, 6, 12, 18) followed by global average pooling. Features are concatenated and processed by 1×1 convolutions before bilinear upsampling to the original resolution. Our model has approximately 39.62M parameters.

3.3.3. Data Augmentation

To increase training data diversity and prevent overfitting, we apply the following augmentations:

- Random horizontal flip (p=0.5)
- Random rotation (±10 degrees)
- Random resized crop (scale 0.8-1.0)
- Color jittering (brightness, contrast, saturation, hue)

3.3.4. Training Strategy

Loss Function: We use weighted cross-entropy loss to handle class imbalance:

$$\mathcal{L} = -\sum_{c=1}^{C} w_c \cdot y_c \cdot \log(\hat{y}_c) \tag{4}$$

where weights w_c are computed using the inverse class frequency.

Optimizer: Adam optimizer with initial learning rate 1×10^{-4}.

Learning Rate Schedule: Cosine annealing with warm restarts.

Early Stopping: Training stops if validation mIoU does not improve for 10 consecutive epochs.

4. Experiments

4.1. Evaluation Metrics

We employ four standard metrics for semantic segmentation evaluation, with proper handling of void regions (label=255):

Pixel Accuracy (PA): Ratio of correctly classified pixels to total valid pixels:

$$\text{PA} = \frac{\sum_{i=1}^{C} p_{ii}}{\sum_{i=1}^{C}\sum_{j=1}^{C} p_{ij}} \tag{5}$$

Mean Pixel Accuracy (MPA): Average of per-class accuracies:

$$\text{MPA} = \frac{1}{C}\sum_{i=1}^{C} \frac{p_{ii}}{\sum_{j=1}^{C} p_{ij}} \tag{6}$$

Intersection over Union (IoU): For each class i:

$$\text{IoU}_i = \frac{p_{ii}}{\sum_{j=1}^{C} p_{ij} + \sum_{j=1}^{C} p_{ji} - p_{ii}} \tag{7}$$

Mean IoU (mIoU): Average IoU across all classes:

$$\text{mIoU} = \frac{1}{C}\sum_{i=1}^{C} \text{IoU}_i \tag{8}$$

where p_{ij} is the number of pixels of class i predicted as class j, and C is the number of classes.

4.2. Implementation Details

All experiments are conducted using Python 3.8 with PyTorch 1.12. Traditional methods use scikit-learn and scikit-image libraries. Deep learning models are trained on NVIDIA GPUs with batch size 8. Images are resized to 256×256 for deep learning models. We set random seeds for reproducibility.

4.3. Traditional Method Results

Table 1 presents the results of traditional methods. Key observations:

- Random Forest outperforms GMM across all metrics, benefiting from its ability to model complex decision boundaries and handle class imbalance through balanced weighting.
- MRF optimization provides consistent improvements: +2.27% PA, +3.07% MPA, and +5.52% mIoU for RF; similar gains for GMM.
- The best traditional method (RF + MRF) achieves 75.48% PA and 49.84% mIoU.

Table 1: Traditional method results on test set. MRF optimization consistently improves segmentation quality.

Method	PA (%)	MPA (%)	mIoU (%)
RF (No MRF)	73.21	65.87	44.32
RF + MRF	75.48	68.94	49.84
GMM (No MRF)	68.54	61.23	38.76
GMM + MRF	71.89	64.56	43.21

Table 2: Deep learning model results on test set.

Model	PA (%)	MPA (%)	mIoU (%)
U-Net	86.53	79.82	61.47
DeepLabV3	**89.71**	**83.45**	**67.62**

4.4. Deep Learning Results

Table 2 shows the deep learning results:

- Both deep learning models significantly outperform traditional methods.
- DeepLabV3 achieves the best performance with 89.71% PA, 83.45% MPA, and 67.62% mIoU.
- The pretrained backbone and ASPP module in DeepLabV3 provide advantages in capturing multi-scale features compared to the from-scratch trained U-Net.

4.5. Training Dynamics

Figure 1 shows the training dynamics. U-Net requires more epochs to converge compared to DeepLabV3, which benefits from ImageNet pretraining. Both models show minimal overfitting due to data augmentation and weighted loss.

4.6. Comprehensive Comparison

Table 3 provides a comprehensive comparison. DeepLabV3 achieves the highest performance at the cost of more parameters. The choice between methods depends on computational resources and accuracy requirements.

4.7. Class Imbalance Analysis

Table 4 shows per-class IoU values. The man-made class, being less frequent in the dataset, has consistently lower IoU across all methods. Weighted loss functions help mitigate this issue but do not fully eliminate it.

4.8. Size Adaptability Analysis

We evaluate model performance across different image sizes present in the MSRC v2 dataset. Traditional methods with adaptive superpixel generation maintain consistent performance across sizes. Deep learning models, trained on fixed 256×256 inputs, show slight performance variations when applied to images of different aspect ratios due to resizing artifacts.

Figure 1: Training and validation loss/mIoU curves for U-Net and DeepLabV3. Both models converge within 30 epochs, with DeepLabV3 showing faster initial convergence due to pretrained weights.

Table 3: Comprehensive comparison of all methods.

Method	PA	MPA	mIoU	Params
RF + MRF	75.48	68.94	49.84	–
GMM + MRF	71.89	64.56	43.21	–
U-Net	86.53	79.82	61.47	7.76M
DeepLabV3	**89.71**	**83.45**	**67.62**	39.62M

5. Qualitative Results

Figure 2 presents qualitative comparisons between methods. Key observations:

- Traditional methods without MRF produce noisy, fragmented predictions following superpixel boundaries.
- MRF optimization smooths predictions but may oversmooth fine details and thin structures.
- Deep learning methods, especially DeepLabV3, produce cleaner segmentations with better boundary adherence and less noise.

Table 4: Per-class IoU analysis showing class imbalance effects.

Class	RF+MRF	U-Net	DeepLabV3
Natural	56.23	68.94	73.51
Man-made	43.45	53.99	61.73

Figure 2: Qualitative comparison. From left to right: input image, ground truth, RF+MRF, U-Net, DeepLabV3. Deep learning methods produce cleaner segmentations.

- DeepLabV3's ASPP module helps capture objects at multiple scales, resulting in more consistent predictions.

6. Discussion

Traditional vs. Deep Learning: Our experiments clearly show that deep learning methods outperform traditional approaches by a significant margin (14-18% improvement in PA, 12-19% in mIoU). However, traditional methods remain valuable when:

- Computational resources are limited
- Interpretability is important
- Training data is scarce

MRF Effectiveness: MRF optimization provides consistent 2-5% improvements for traditional methods by enforcing spatial coherence. This suggests that incorporating spatial priors remains valuable even in the deep learning era, as evidenced by CRF post-processing in some deep learning pipelines [4].

Class Imbalance: While weighted loss functions help address class imbalance, the performance gap between majority and minority classes persists. Future work could explore more advanced techniques such as focal loss [13] or class-balanced sampling.

Limitations: Our study focuses on binary segmentation (natural vs. man-made), which simplifies the original 21-class problem. Extending to full multi-class segmentation would provide a more complete evaluation. Additionally, the MSRC v2 dataset is relatively small; evaluation on larger benchmarks like Cityscapes [7] or ADE20K [17] would further validate our findings.

7. Conclusion

We presented a comprehensive comparative study of image semantic segmentation methods on the MSRC v2 dataset, covering both traditional machine learning and deep learning approaches. Our traditional pipeline combining superpixel-based features, Random Forest/GMM classifiers, and MRF optimization achieves reasonable performance (PA: 75.48%, mIoU: 49.84%), while deep learning methods significantly surpass these results, with DeepLabV3 achieving state-of-the-art performance (PA: 89.71%, mIoU: 67.62%).

Key findings include: (1) MRF optimization consistently improves traditional method performance by 2-5%; (2) Deep learning methods benefit from pretrained backbones and multi-scale feature extraction; (3) Class imbalance remains challenging for all methods; (4) Adaptive superpixel generation enables consistent traditional method performance across varying image sizes.

Future work will extend this study to multi-class segmentation, explore additional deep learning architectures (e.g., Transformer-based models), and evaluate on larger benchmarks. The insights gained from this comparative study provide practical guidance for practitioners selecting appropriate segmentation approaches based on their specific requirements and constraints.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[2] M. Berman, A. R. Triki, and M. B. Blaschko. The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 2018.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015.

[5] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[8] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[9] S. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *Applied Soft Computing*, 70:41–65, 2018.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[11] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Associative hierarchical CRFs for object class image segmentation. In *ICCV*, 2009.

[12] J. D. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *ICCV*, 2017.

[14] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[15] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[16] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[17] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017.