An Enhanced Deep Convolutional Encoder-Decoder Network for Road Segmentation on Aerial Imagery

Teerapong Panboonyuen

Paper ID: IC2IT2017-26, July 7, 2017

Contact: teerapong.panboonyuen@gmail.com

Overview

- Introduction
- Related Work
- Methodology
- Experiment
- Results
- Discussion
- Conclusion

Overview

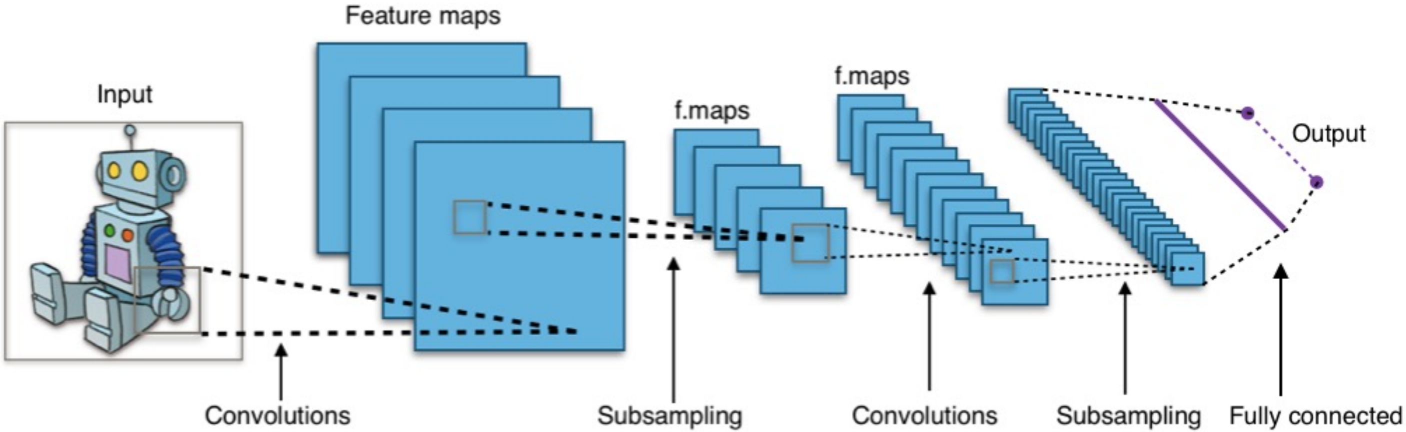
- Introduction
- Related Work
- Methodology
- Experiment
- Results
- Discussion
- Conclusion

Introduction

- Object classification from images is among the many practical examples where deep learning algorithms have successfully been applied.
- In our work, we present an improved deep convolutional encoder-decoder network (DCED) with Landscape metrics for segmenting road objects from aerial images.
- The most recent DCED approach for object segmentation, namely SegNet, is used as one of the benchmarks in evaluating our method.
- The experiments were conducted on a well-known aerial imagery, Massachusetts roads dataset.

Introduction

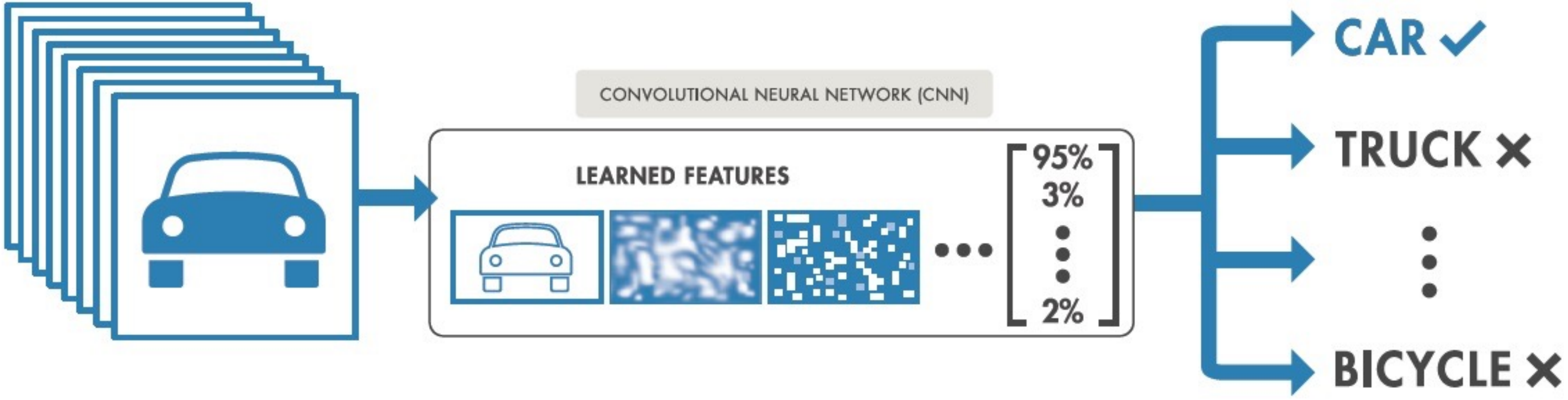
- Object classification from images is among the many practical examples where deep learning algorithms have successfully been applied.



https://upload.wikimedia.org/wikipedia/commons/6/63/Typical_cnn.png

Introduction

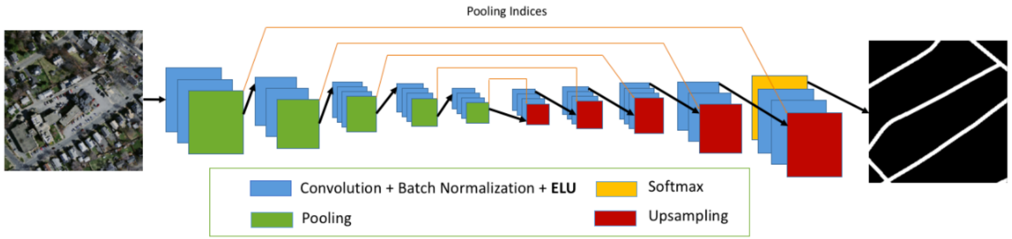
- Object classification from images is among the many practical examples where deep learning algorithms have successfully been applied.



https://es.mathworks.com/content/mathworks/es/es/discovery/convolutional-neural-network/_jcr_content/mainParsys/image.img.jpg/1492406019315.jpg

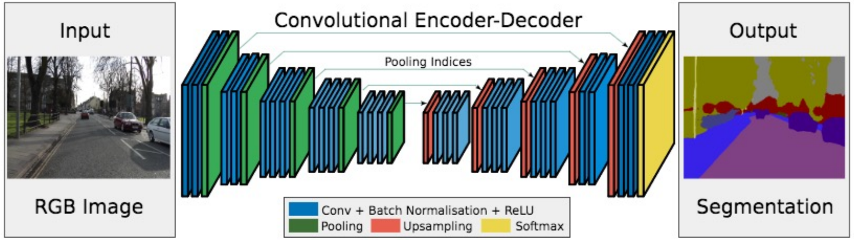
Introduction

- Object classification from images is among the many practical examples where deep learning algorithms have successfully been applied.
- In our work, we present an improved deep convolutional encoder-decoder network (DCED) with landscape metrics for segmenting road objects from aerial images.
- The most recent DCED approach for object segmentation, namely SegNet, is used as one of the benchmarks in evaluating our method.
- The experiments were conducted on a well-known aerial imagery, Massachusetts roads dataset.



Introduction

- Object classification from images is among the many practical examples where deep learning algorithms have successfully been applied.
- In our work, we present an improved deep convolutional encoder-decoder network (DCED) with Landscape metrics for segmenting road objects from aerial images.
- **The most recent DCED approach for object segmentation, namely SegNet, is used as one of the benchmarks in evaluating our method.**
- The experiments were conducted on a well-known aerial imagery, Massachusetts roads dataset.



Vijay Badrinarayanan, Alex Kendall and Roberto Cipolla "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." PAMI, 2017.

Introduction

- Object classification from images is among the many practical examples where deep learning algorithms have successfully been applied.
- In our work, we present an improved deep convolutional encoder-decoder network (DCED) with Landscape metrics for segmenting road objects from aerial images.
- The most recent DCED approach for object segmentation, namely SegNet, is used as one of the benchmarks in evaluating our method.
- The experiments were conducted on a well-known aerial imagery, Massachusetts roads dataset.

Overview

- Introduction
- **Related Work**
- Methodology
- Experiment
- Results
- Discussion
- Conclusion

Related Work

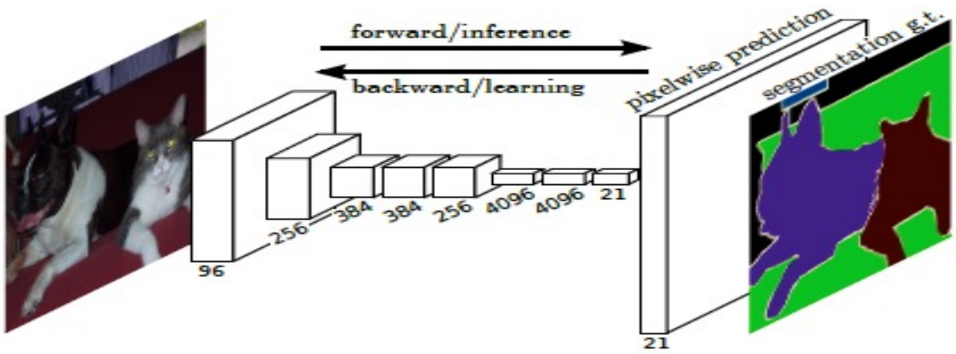
- (1) Deep Learning for Semantic Segmentation
- (2) Deep Learning for Road Segmentation
- (3) Activation Functions in Deep Learning

Related Work

- (1) Deep Learning for Semantic Segmentation
- (2) Deep Learning for Road Segmentation
- (3) Activation Functions in Deep Learning

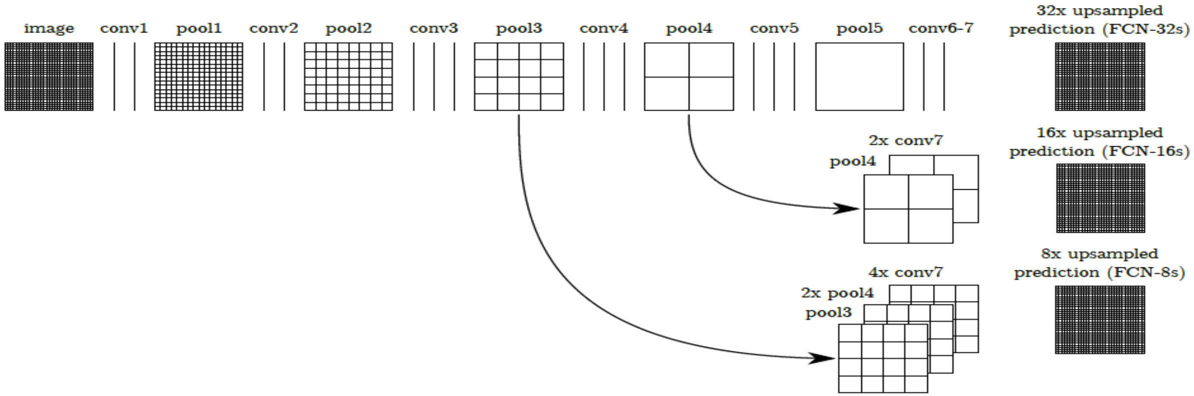
(1) Deep Learning for Semantic Segmentation

- Fully convolutional networks (FCN) was presented by J. Long, E. Shelhamer and T. Darrell. (2015)
  - Some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s).
- Deconvolutional neural network (DeCNN) was proposed by H. Noh et al. (2015)
- Deep convolutional encoder-decoder (DCED) was proposed by V. Badrinarayanan (2015)



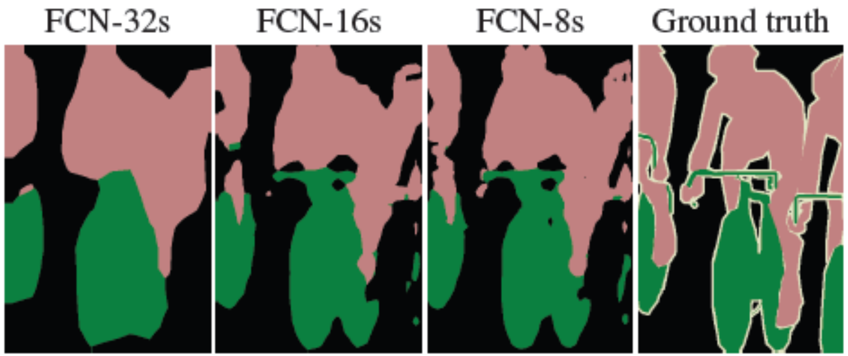
(1) Deep Learning for Semantic Segmentation

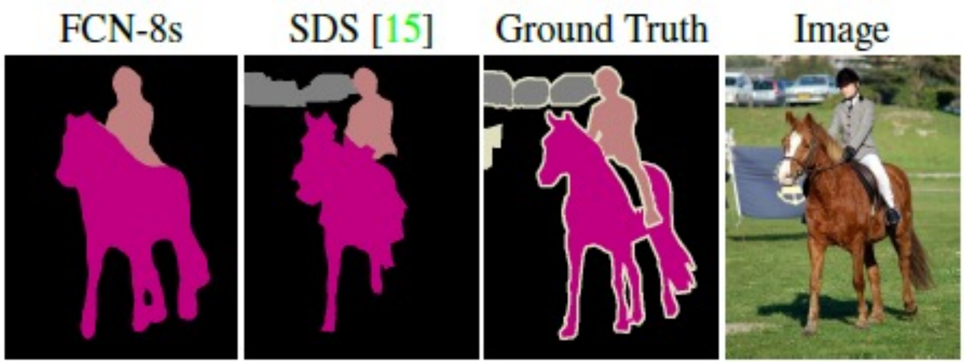
- Fully convolutional networks (FCN) was presented by J. Long, E. Shelhamer and T. Darrell. (2015)
  - Some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s).
- Deconvolutional neural network (DeCNN) was proposed by H. Noh et al. (2015)
- Deep convolutional encoder-decoder (DCED) was proposed by V. Badrinarayanan (2015)



(1) Deep Learning for Semantic Segmentation

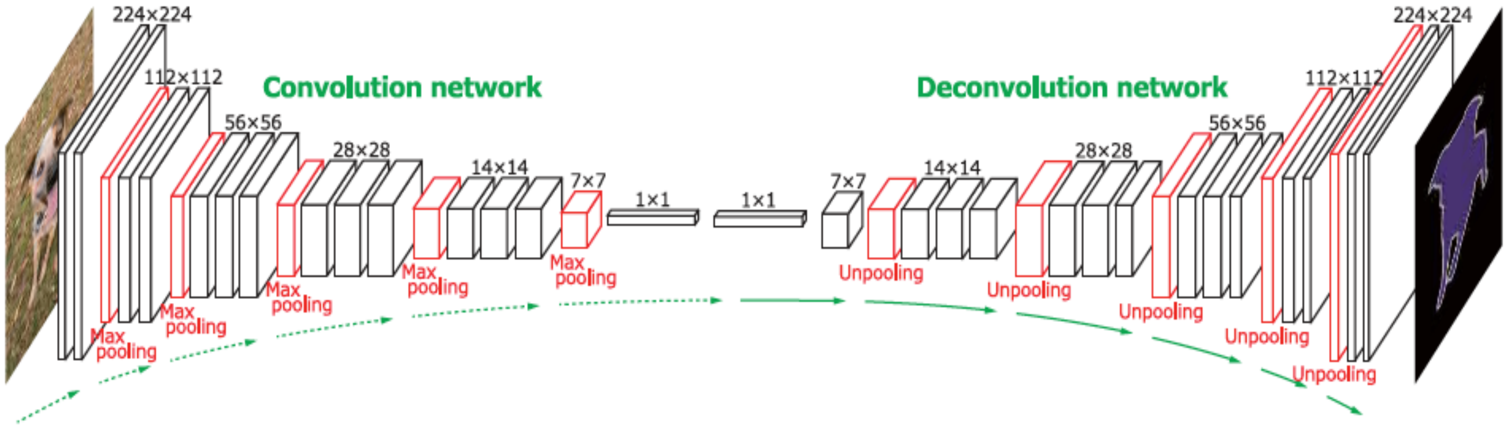
- Fully convolutional networks (FCN) was presented by J. Long, E. Shelhamer and T. Darrell. (2015)
  - Some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s).
- Deconvolutional neural network (DeCNN) was proposed by H. Noh et al. (2015)
- Deep convolutional encoder-decoder (DCED) was proposed by V. Badrinarayanan (2015)





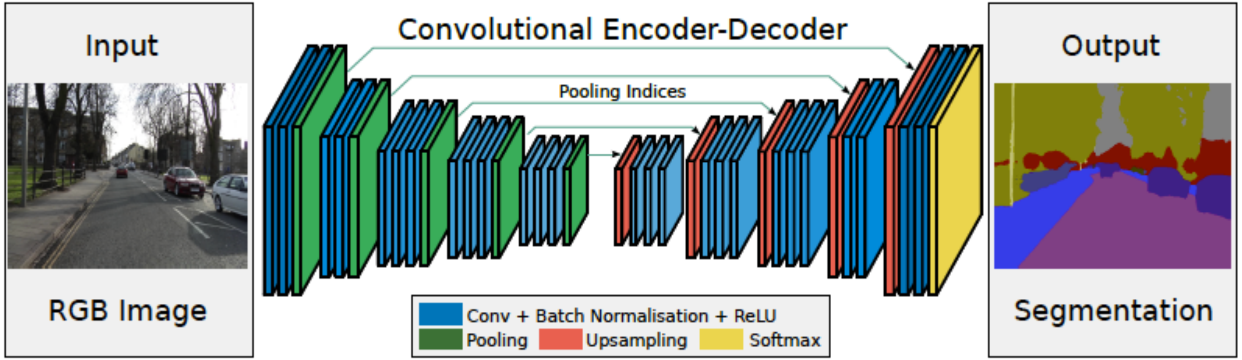
(1) Deep Learning for Semantic Segmentation

- Fully convolutional networks (FCN) was presented by J. Long, E. Shelhamer and T. Darrell. (2015)
  - Some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s).
- Deconvolutional neural network (DeCNN) was proposed by H. Noh et al. (2015)
- Deep convolutional encoder-decoder (DCED) was proposed by V. Badrinarayanan (2015)



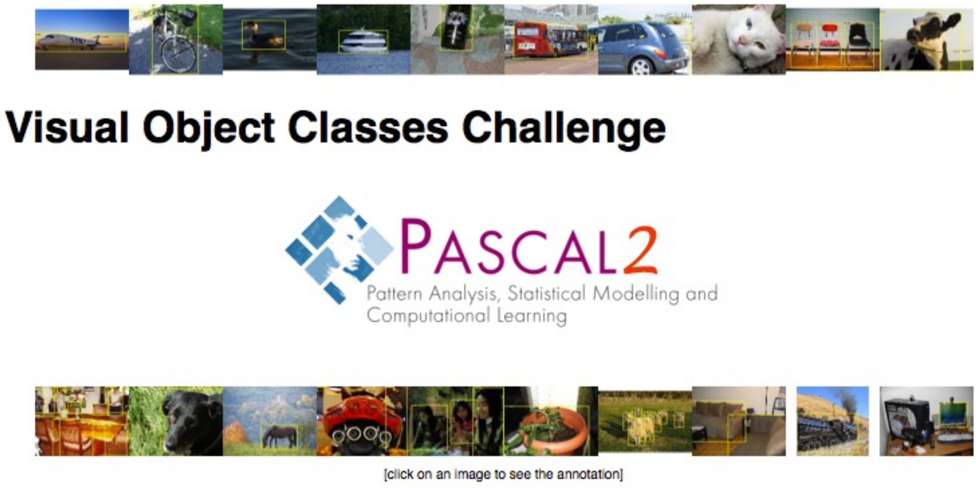
(1) Deep Learning for Semantic Segmentation

- Fully convolutional networks (FCN) was presented by J. Long, E. Shelhamer and T. Darrell. (2015)
  - Some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s).
- Deconvolutional neural network (DeCNN) was proposed by H. Noh et al. (2015)
- Deep convolutional encoder-decoder (DCED) was proposed by V. Badrinarayanan (2015)



PASCAL VOC Data Sets

- PASCAL VOC is a standard recognition dataset and benchmark with detection and semantic segmentation challenges.
  - The semantic segmentation challenge annotates 20 object classes and background.



(1) Deep Learning for Semantic Segmentation (Results)

- Fully convolutional networks (FCN) was presented by J. Long, E. Shelhamer and T. Darrell. (2015)
- Performance: Accuracy = 59.4 percent (FCN)
  - Some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s).
  - Performance: Accuracy = 62.7 percent (FCN-8s) , 62.4 percent (FCN-16s)
- Deconvolutional neural network (DeCNN) was proposed by H. Noh et al. (2015)
- Performance: Accuracy = 72.5 percent (DeCNN)
- Deep convolutional encoder-decoder (DCED) was proposed by V. Badrinarayanan (2015)
- Performance: Accuracy = 80.1 percent (DCED)

(1) Deep Learning for Semantic Segmentation (Results)

- Fully convolutional networks (FCN) was presented by J. Long, E. Shelhamer and T. Darrell. (2015)
- Performance: Accuracy = 59.4 percent (FCN)
  - Some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s).
  - Performance: Accuracy = 62.7 percent (FCN-8s) , 62.4 percent (FCN-16s)
- Deconvolutional neural network (DeCNN) was proposed by H. Noh et al. (2015)
- Performance: Accuracy = 72.5 percent (DeCNN)
- Deep convolutional encoder-decoder (DCED) was proposed by V. Badrinarayanan (2015)
- Performance: Accuracy = 80.1 percent (DCED)

(1) Deep Learning for Semantic Segmentation (Results)

- Fully convolutional networks (FCN) was presented by J. Long, E. Shelhamer and T. Darrell. (2015)
- Performance: Accuracy = 59.4 percent (FCN)
  - Some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s).
  - Performance: Accuracy = 62.7 percent (FCN-8s) , 62.4 percent (FCN-16s)
- Deconvolutional neural network (DeCNN) was proposed by H. Noh et al. (2015)
- Performance: Accuracy = 72.5 percent (DeCNN)
- Deep convolutional encoder-decoder (DCED) was proposed by V. Badrinarayanan (2015)
- Performance: Accuracy = 80.1 percent (DCED)

(1) Deep Learning for Semantic Segmentation (Results)

PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning

- Fully convolutional networks (FCN) was presented by J. Long, E. Shelhamer and T. Darrell. (2015)
- Performance: Accuracy = 59.4 percent (FCN)
    - Some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s).
    - Performance: Accuracy = 62.7 percent (FCN-8s) , 62.4 percent (FCN-16s)
- Deconvolutional neural network (DeCNN) was proposed by H. Noh et al. (2015)
- Performance: Accuracy = 72.5 percent (DeCNN)
- Deep convolutional encoder-decoder (DCED) was proposed by V. Badrinarayanan (2015)
- Performance: Accuracy = 80.1 percent (DCED)

(1) Deep Learning for Semantic Segmentation (Results)

PASCAL2 Pattern Analysis, Statistical Modelling and Computational Learning

- Fully convolutional networks (FCN) was presented by J. Long, E. Shelhamer and T. Darrell. (2015)
- Performance: Accuracy = 59.4 percent (FCN)
    - Some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s).
    - Performance: Accuracy = 62.7 percent (FCN-8s) , 62.4 percent (FCN-16s)
- Deconvolutional neural network (DeCNN) was proposed by H. Noh et al. (2015)
- Performance: Accuracy = 72.5 percent (DeCNN)
- Deep convolutional encoder-decoder (DCED) was proposed by V. Badrinarayanan (2015)
- Performance: Accuracy = 80.1 percent (DCED)

DCED was given the best performance on PASCAL VOC data set

(1) Deep Learning for Semantic Segmentation (Results)

PASCAL2 Pattern Analysis, Statistical Modelling and Computational Learning

- Fully convolutional networks (FCN) was presented by J. Long, E. Shelhamer and T. Darrell. (2015)
- Performance: Accuracy = 59.4 percent (FCN)
    - Some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s).
    - Performance: Accuracy = 62.7 percent (FCN-8s) , 62.4 percent (FCN-16s)
- Deconvolutional neural network (DeCNN) was proposed by H. Noh et al. (2015)
- Performance: Accuracy = 72.5 percent (DeCNN)
- Deep convolutional encoder-decoder (DCED) was proposed by V. Badrinarayanan (2015)
- Performance: Accuracy = 80.1 percent (DCED)

DCED was given the best performance on PASCAL VOC data set

So, DCED was chosen be our standard architecture and baseline

Related Work

- (1) Deep Learning for Semantic Segmentation
- (2) Deep Learning for Road Segmentation
- (3) Activation Functions in Deep Learning

Related Work (Standard data set)

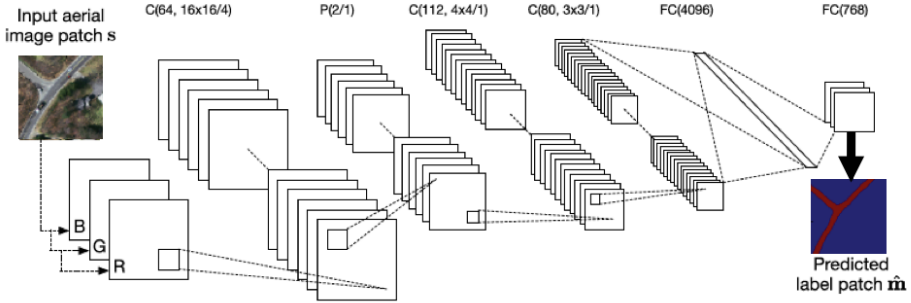
- Massachusetts roads dataset (Mass. Roads). (Very high resolution imagery)
  - 1500x1500 pixels (Resolution = 1 $meter^2$/pixel)
  - Training set = 1108, Validation set = 14, Test set = 49
  - made publicly available on website: https://www.cs.toronto.edu/~vmnih/data/

(2) Deep Learning for Road Segmentation

- S. Saito et al. (2016) use FCN architecture for segmenting road objects on Mass. road data set.
  - Performance: F1 = 0.742
- S. Muruganandham (2016) use DeCNN and FCN-8s for segmenting road objects on Mass. Road data set.
  - Performance: F1 (DeCNN) = 0.657
  - Performance: F1 (FCN-8s) = 0.762

(2) Deep Learning for Road Segmentation

- S. Saito et al. (2016) use FCN architecture for segmenting road objects on Mass. road data set.
  - Performance: F1 = 0.742
- S. Muruganandham (2016) use DeCNN and FCN-8s for segmenting road objects on Mass. Road data set.
  - Performance: F1 (DeCNN) = 0.657
  - Performance: F1 (FCN-8s) = 0.762



(2) Deep Learning for Road Segmentation

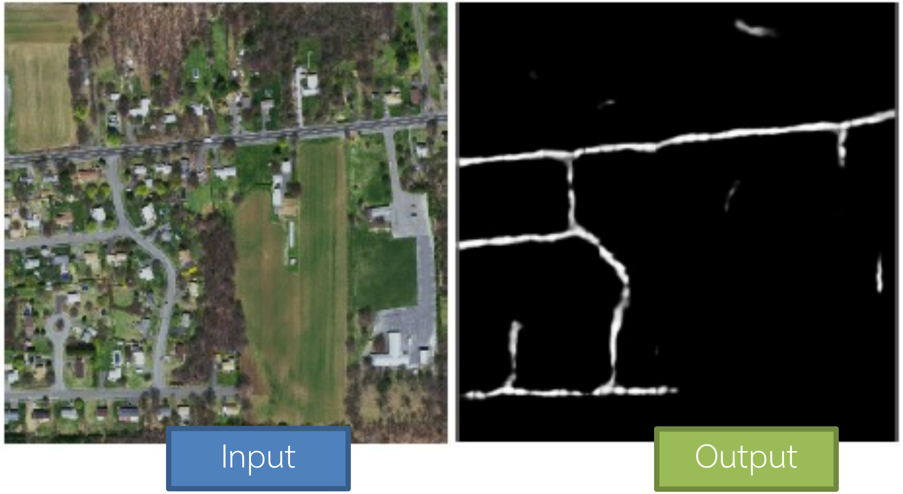
PASCAL2 Pattern Analysis, Statistical Modelling and Computational Learning

- S. Saito et al. (2016) use FCN architecture for segmenting road objects on Mass. road data set.
  - Performance: F1 = 0.742
- S. Muruganandham (2016) use DeCNN and FCN-8s for segmenting road objects on Mass. Road data set.
  - Performance: F1 (DeCNN) = 0.657
  - Performance: F1 (FCN-8s) = 0.762

(2) Deep Learning for Road Segmentation

PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning

- S. Saito et al. (2016) use FCN architecture for segmenting road objects on Mass. road data set.
    - Performance: F1 = 0.742
- S. Muruganandham (2016) use DeCNN and FCN-8s for segmenting road objects on Mass. Road data set.
    - Performance: F1 (DeCNN) = 0.657
    - Performance: F1 (FCN-8s) = 0.762



(2) Deep Learning for Road Segmentation

PASCAL2 Pattern Analysis, Statistical Modelling and Computational Learning

- S. Saito et al. (2016) use FCN architecture for segmenting road objects on Mass. road data set.
  - Performance: F1 = 0.742
- S. Muruganandham (2016) use DeCNN and FCN-8s for segmenting road objects on Mass. Road data set.
  - Performance: F1 (DeCNN) = 0.657
  - Performance: F1 (FCN-8s) = 0.762

FCN FCN-8s and DeCNN were poposed in this task

(2) Deep Learning for Road Segmentation

PASCAL2 Pattern Analysis, Statistical Modelling and Computational Learning

- S. Saito et al. (2016) use FCN architecture for segmenting road objects on Mass. road data set.
  - Performance: F1 = 0.742
- S. Muruganandham (2016) use DeCNN and FCN-8s for segmenting road objects on Mass. Road data set.
  - Performance: F1 (DeCNN) = 0.657
  - Performance: F1 (FCN-8s) = 0.762

So, These architecture will be baseline for comparison in our work.

FCN FCN-8s and DeCNN were poposed in this task

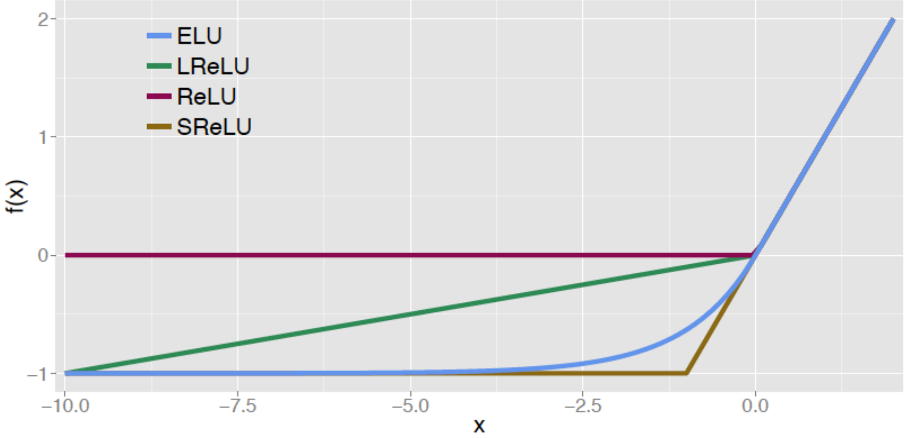
Related Work

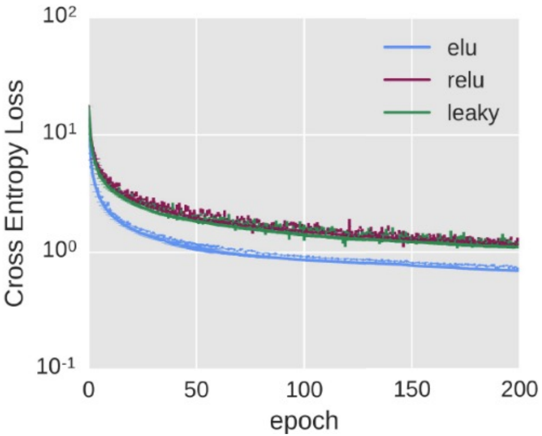
- (1) Deep Learning for Semantic Segmentation
- (2) Deep Learning for Road Segmentation
- (3) Activation Functions in Deep Learning

(3) Activation Functions in Deep Learning

PASCAL2 Pattern Analysis, Statistical Modelling and Computational Learning

- Clevert et al. (2016) proposed "exponential linear unit" (ELU) which speeds up learning in deep neural networks and leads to higher classification accuracies.
  - Training and testing on CIFAR-10 data set, CIFAR-100 sets
  - Performance: Error = 6.55 and 24.28
  - AlexNet has an error: 18.04 and 45.80, CNN has an error 7.25 and 33.71





(3) Activation Functions in Deep Learning

- Clevert et al. (2016) proposed "exponential linear unit" (ELU) which speeds up learning in deep neural networks and leads to higher classification accuracies.
  - Training and testing on CIFAR-10 data set, CIFAR-100 sets
  - Performance: Error = 6.55 and 24.28
  - AlexNet has an error: 18.04 and 45.80, CNN has an error 7.25 and 33.71

Network	CIFAR-10 (test error %)	CIFAR-100 (test error %)	augmented
AlexNet	18.04	45.80	
DSN	7.97	34.57	√
NiN	8.81	35.68	√
Maxout	9.38	38.57	√
All-CNN	7.25	33.71	√
Highway Network	7.60	32.24	√
Fract. Max-Pooling	**4.50**	27.62	√
ELU-Network	6.55	**24.28**	

(3) Activation Functions in Deep Learning

PASCAL2 Pattern Analysis, Statistical Modelling and Computational Learning

- Clevert et al. (2016) proposed "exponential linear unit" (ELU) which speeds up learning in deep neural networks and leads to higher classification accuracies.
  - Training and testing on CIFAR-10 data set, CIFAR-100 sets
  - Performance: Error = 6.55 and 24.28
  - AlexNet has an error: 18.04 and 45.80, CNN has an error 7.25 and 33.71

So, ELU-Network is better than ReLU-Network

Network	CIFAR-10 (test error %)	CIFAR-100 (test error %)	augmented
AlexNet	18.04	45.80	
DSN	7.97	34.57	√
NiN	8.81	35.68	√
Maxout	9.38	38.57	√
All-CNN	7.25	33.71	√
Highway Network	7.60	32.24	√
Fract. Max-Pooling	**4.50**	27.62	√
ELU-Network	6.55	**24.28**	

Overview

- Introduction
- Related Work
- **Methodology**
- Experiment
- Results
- Discussion
- Conclusion

Methodology

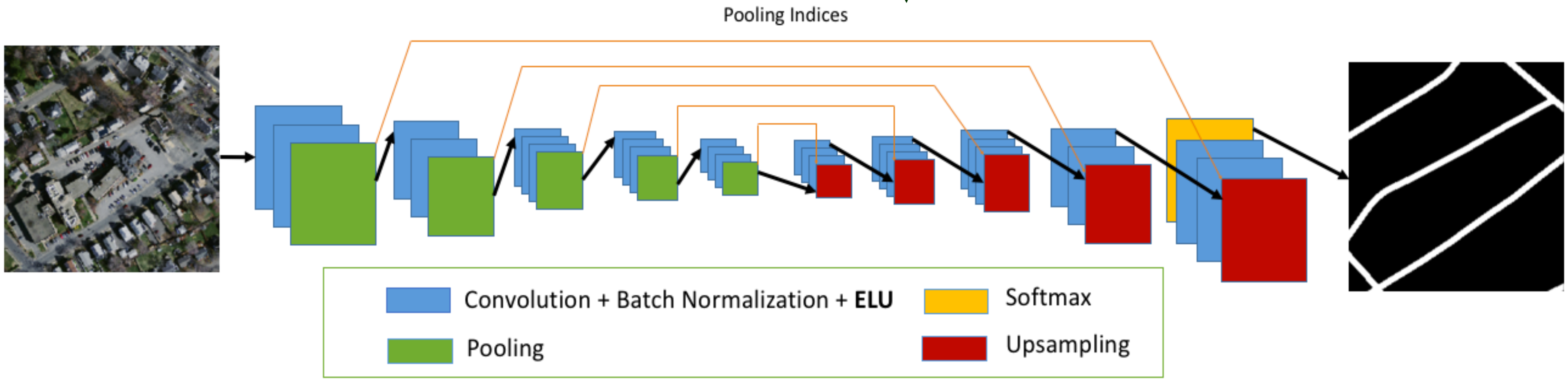
- Three aspects of the proposed method are enhanced:
- (1) Modification of DCED architecture
- (2) Data amplification
- (3) Adoption of landscape metrics..

Methodology (Proposed Method)

- Three aspects of the proposed method are enhanced:
- (1) Modification of DCED architecture. → ELU-DCED
- (2) Data amplification
- (3) Adoption of landscape metrics.

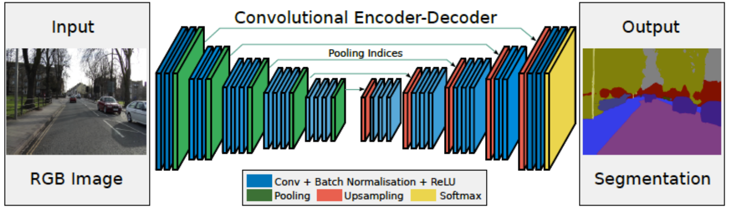
Methodology (Proposed Method)

- Three aspects of the proposed method are enhanced:
- (1) Modification of DCED architecture. → ELU-DCED
- (2) Data amplification
- (3) Adoption of landscape metrics.

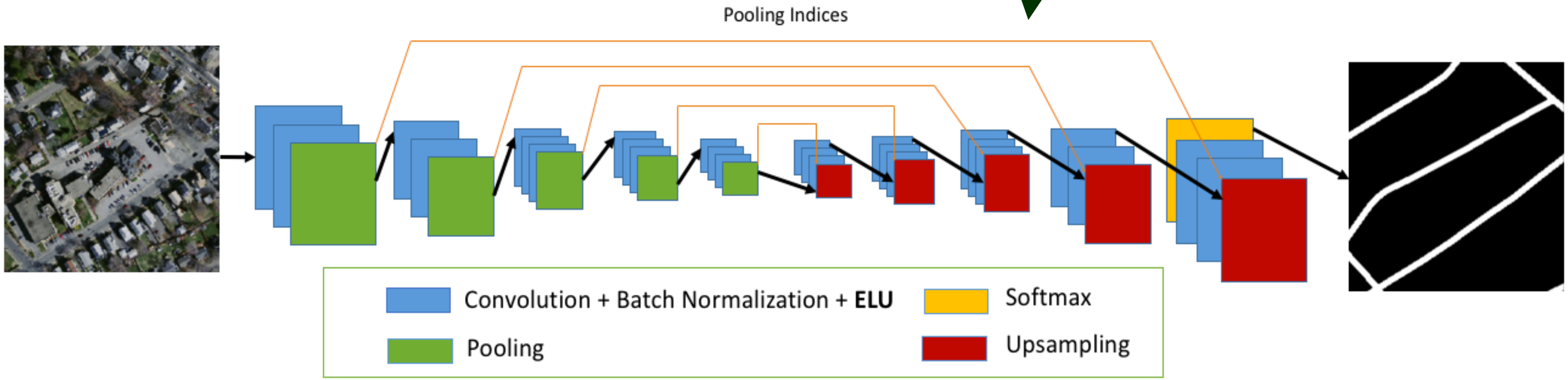


Methodology (Proposed Method)

- Three aspects of the proposed method are enhanced:
- (1) Modification of DCED architecture. → ELU-DCED
- (2) Data amplification
- (3) Adoption of landscape metrics.



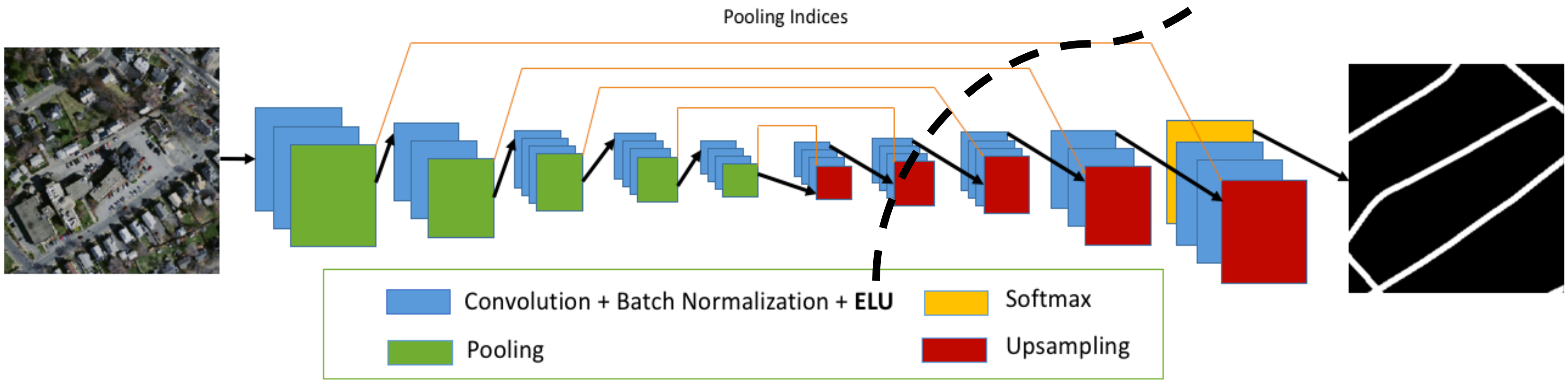
DCED Architecture



Methodology (Proposed Method)

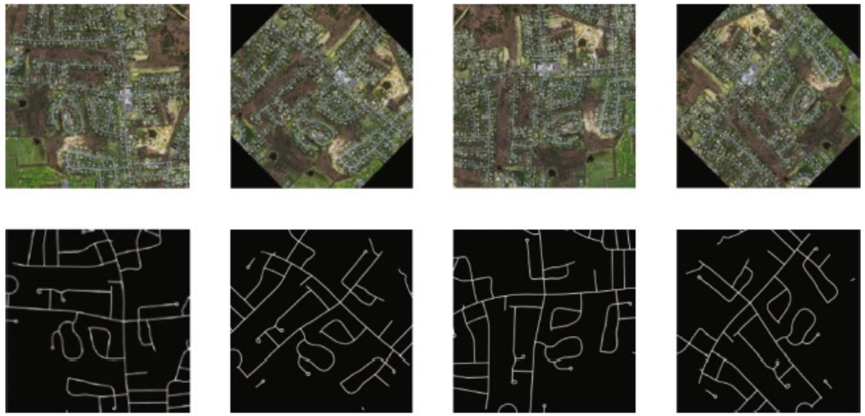
- Three aspects of the proposed method are enhanced:
- (1) Modification of DCED architecture. → ELU-DCED
- (2) Data amplification
- (3) Adoption of landscape metrics.

ReLU → ELU



Methodology (Proposed Method)

- Three aspects of the proposed method are enhanced:
- (1) Modification of DCED architecture. → ELU-DCED
- (2) Data amplification → ELU-DCED-R (**R means Rotate images)
- (3) Adoption of landscape metrics.

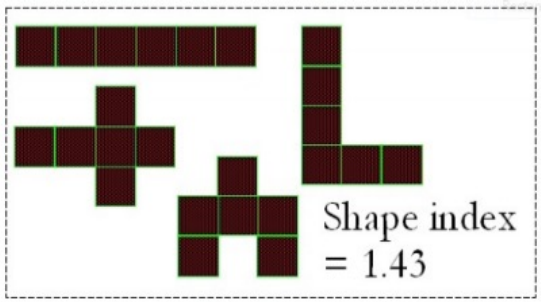


0 degree 45 degree 90 degree 135 degree

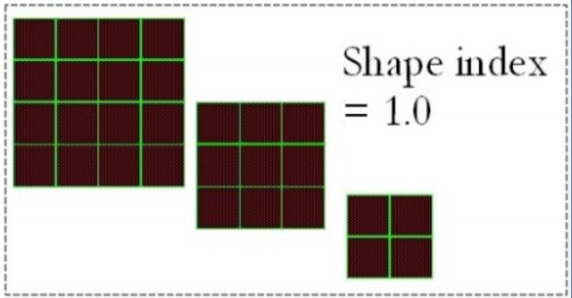
Methodology (Proposed Method)

- Three aspects of the proposed method are enhanced:
- (1) Modification of DCED architecture. → ELU-DCED
- (2) Data amplification → ELU-DCED-R (**R means Rotate images)
- (3) Adoption of landscape metrics. → ELU-DCED-RL (**L means Landscape Metrics)

$$\text{Shape Index} = \frac{\text{perimeter}}{4\text{x}\sqrt{\text{area}}}$$



Perimeter =14, area = 6; shape idx = 1.43



Perimeter 16, area = 16; shape idx = 1.00

Overview

- Introduction
- Related Work
- Methodology
- **Experiment**
- Results
- Discussion
- Conclusion

Experiment

- The experiments were conducted on a standard benchmark, "Massachusetts roads dataset" (Mass. Roads)
- Compared our proposed method (ELU-DCED, ELU-DCED-R, ELU-DCED-RL) to four baselines: basic-model (CNN), FCN-no-skip, FCN-8s, and SegNet.
- In terms of precision, recall, and F1 (Performance Evaluations)




Experiment

- The experiments were conducted on a standard benchmark, “Massachusetts roads dataset” (Mass. Roads)
- Compared our proposed method (ELU-DCED, ELU-DCED-R, ELU-DCED-RL) to four baselines: basic-model (CNN), FCN-no-skip, FCN-8s, and SegNet.
- In terms of precision, recall, and F1 (Performance Evaluations)

Experiment

- The experiments were conducted on a standard benchmark, “Massachusetts roads dataset” (Mass. Roads)
- Compared our proposed method (ELU-DCED, ELU-DCED-R, ELU-DCED-RL) to four baselines: basic-model (CNN), FCN-no-skip, FCN-8s, and SegNet.
- In terms of precision, recall, and F1 (Performance Evaluations)

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1) \qquad \text{Recall} = \frac{TP}{TP + FN} \quad (2) \qquad F1 = \frac{2 \text{ x Precision x Recall}}{\text{Precision + Recall}} \quad (3)$$

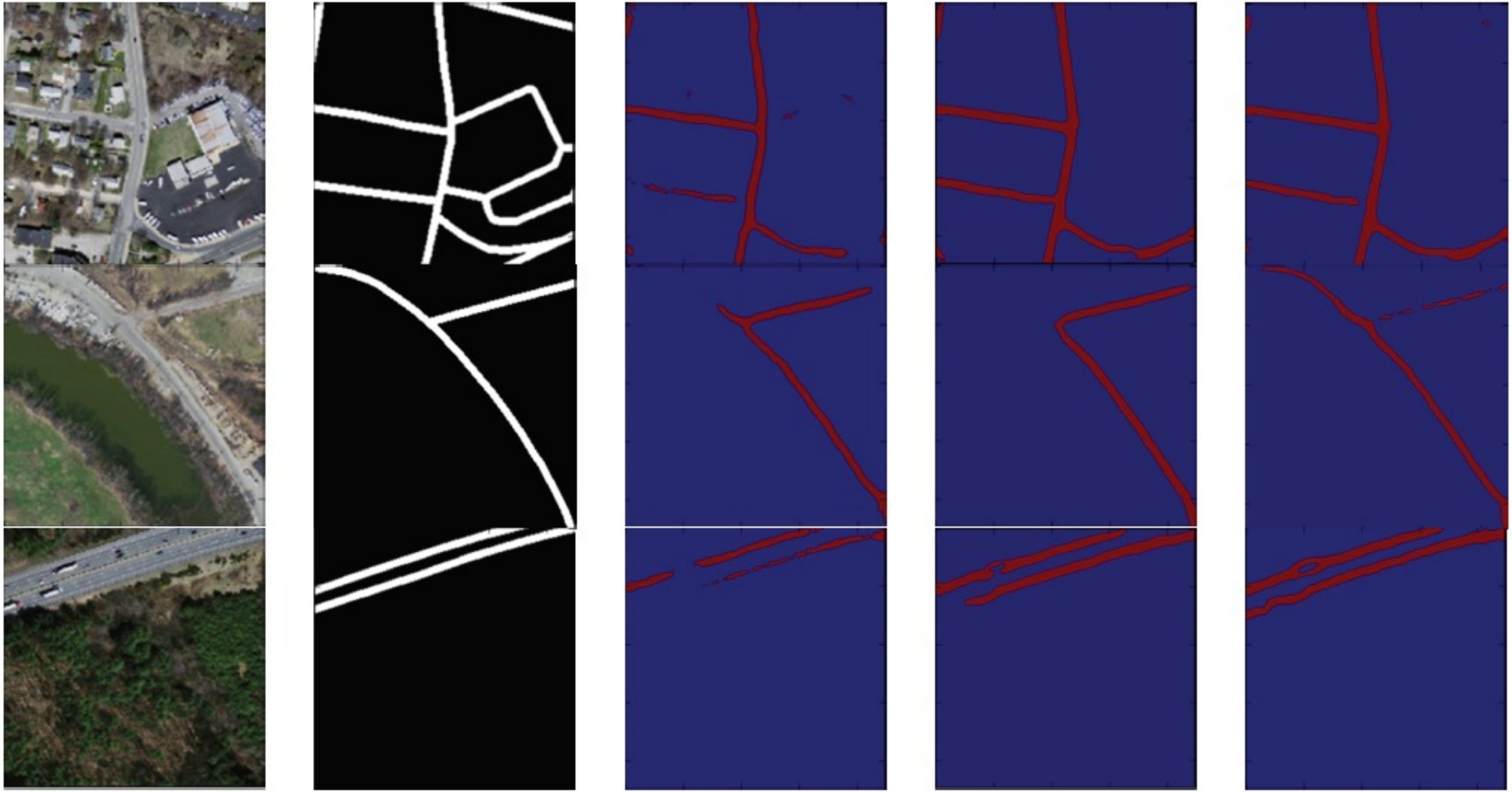
TP denote the number of true positives (the number of correctly classified road pixels), TN denote the number of true negatives (the number of correctly classified non-road pixels), FP denote the number of false positives (the number of mistakenly classified road pixels), and FN denote the number of false negatives (the number of mistakenly classified non-road pixels).

Overview

- Introduction
- Related Work
- Methodology
- Experiment
- **Results**
- Discussion
- Conclusion

Results

	Model	Precision	Recall	F1
Baseline	Basic-model [2]	0.657	0.657	0.657
	FCN-no-skip [2]	0.742	0.742	0.742
	FCN-8s [2]	0.762	0.762	0.762
	SegNet [6]	0.773	0.765	0.768
Proposed Method	ELU-DCED	0.852	0.733	0.788
	ELU-DCED-R	0.78	0.847	0.812
	ELU-DCED-RL	**0.854**	**0.861**	**0.857**

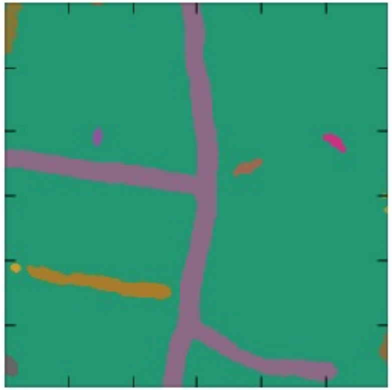


Satellite Image | Target | DCED (Baseline) | ELU-DCED | ELU-DCED-Rotate

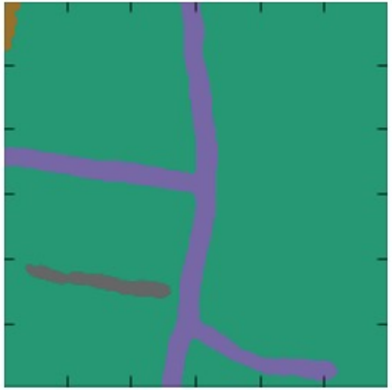
Results

Satellite Image

Target



ELU-DCED-R

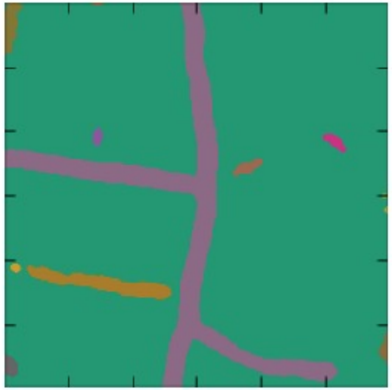


ELU-DCED-RL

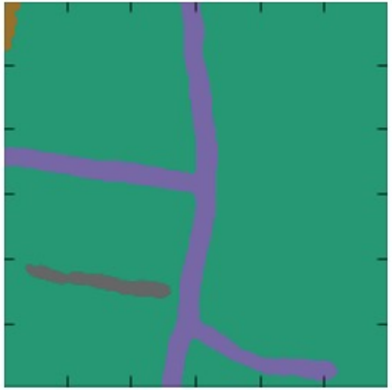
Results

Satellite Image

Target



ELU-DCED-R



ELU-DCED-RL

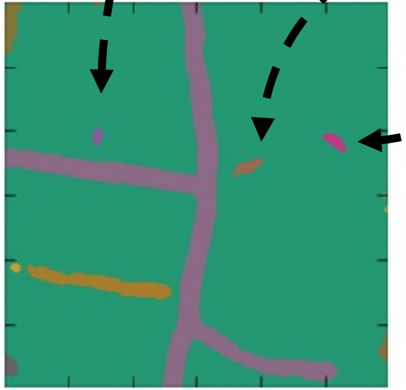
Using Landscape Metrics (Consider the shape index)

Results

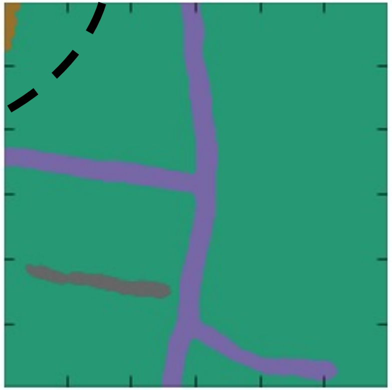
Extracted roads with removing noises.

Satellite Image

Target



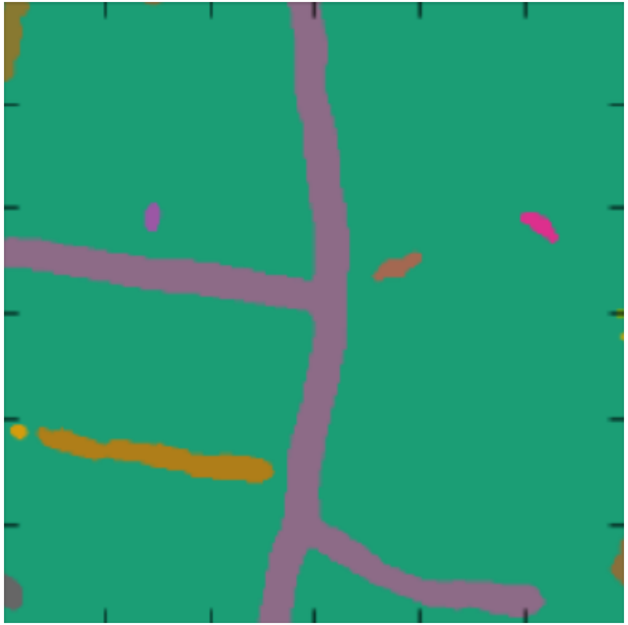
ELU-DCED-R

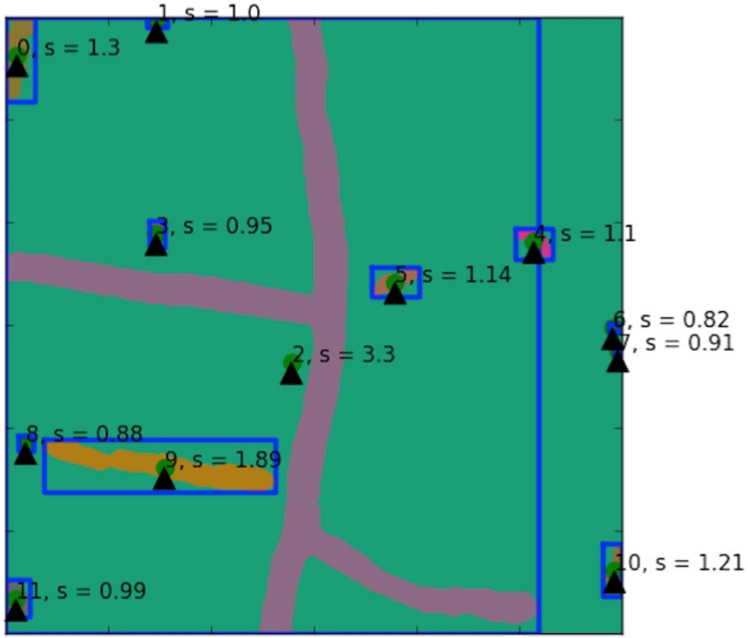


ELU-DCED-RL

Using Landscape Metrics (Consider the shape index)

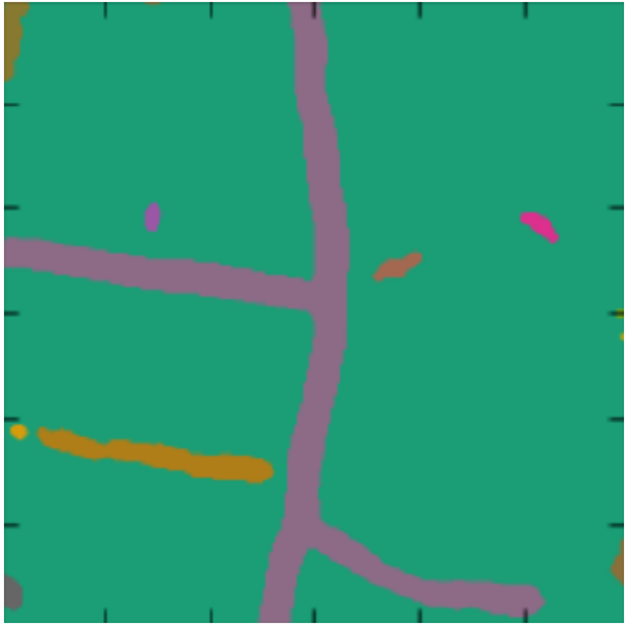
Results

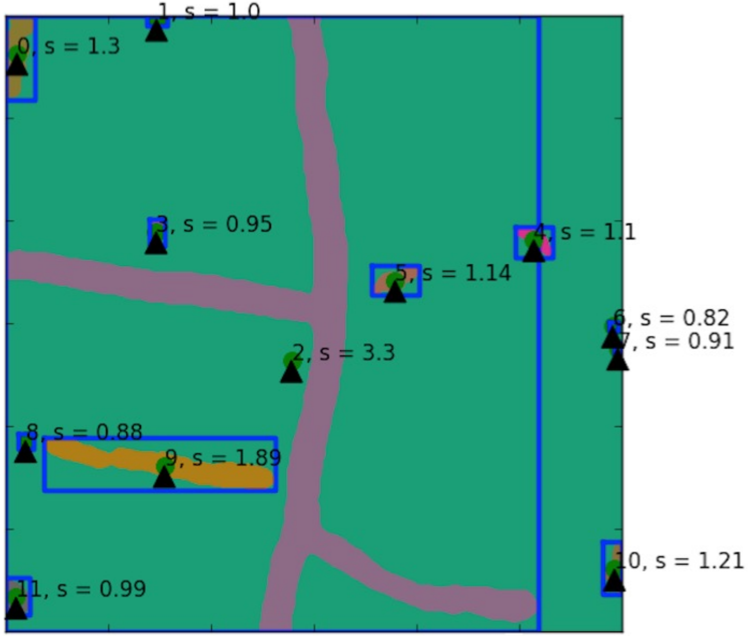



0, s = 1.3
1, s = 1.0
3, s = 0.95
4, s = 1.1
5, s = 1.14
6, s = 0.82
7, s = 0.91
2, s = 3.3
8, s = 0.88
9, s = 1.89
10, s = 1.21
11, s = 0.99

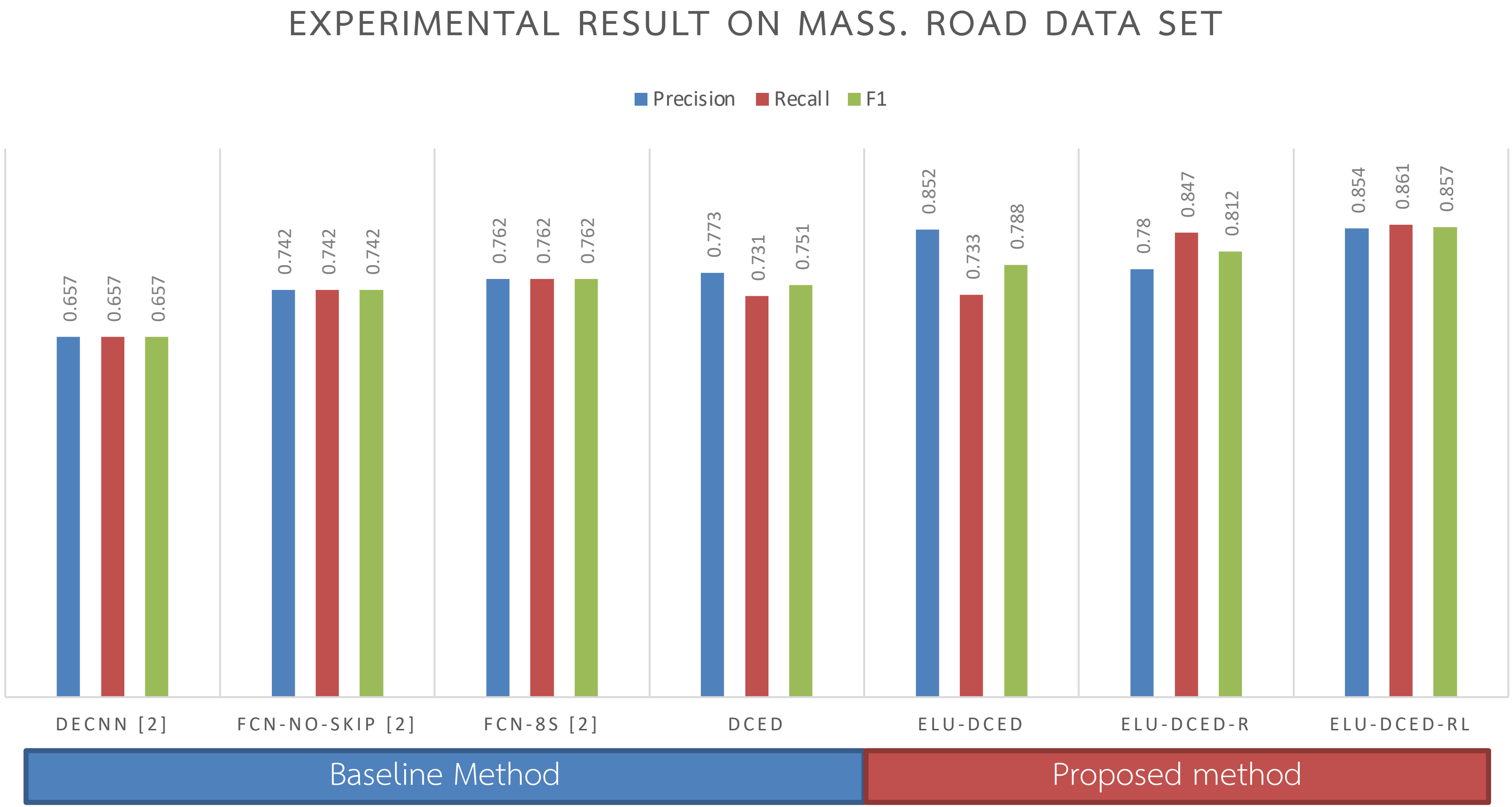
Results

Shape Index (Score)





Result


EXPERIMENTAL RESULT ON MASS. ROAD DATA SET
Precision
Recall
F1
0.657
0.657
0.657
0.742
0.742
0.742
0.762
0.762
0.762
0.773
0.731
0.751
0.852
0.733
0.788
0.78
0.847
0.812
0.854
0.861
0.857
DECNN [2]
FCN-NO-SKIP [2]
FCN-8S [2]
DCED
ELU-DCED
ELU-DCED-R
ELU-DCED-RL
Baseline Method
Proposed method

Overview

- Introduction
- Related Work
- Methodology
- Experiment
- Results
- Discussion
- Conclusion

Discussion

- Discussion on Enhanced Deep Learning Framework
  - For the effect of ELU, Result shows that the precision of ELU-SegNet is higher than that of the original SegNet by 7.9%—without losing recall. This can imply that ELU is more robust than ReLU to detect road pixels.
  - The result also shows that the recall of ELU-SegNet-R is higher than that of ELU-SegNet by 11.4%, meaning that it can detect more patterns of the roads.
- Discussion on Landscape Metrics
  - The landscape metrics are applied to our framework in order to remove all inaccurately extracted roads (false positives: FP), considered as a negative effect of the rotated image strategy as discussed in the previous section.
  - Result shows that the precision of the network is increased by 7.4% by applying the landscape metrics filtering. This shows that the FP issue has been resolved.

Discussion

- Discussion on Enhanced Deep Learning Framework
  - For the effect of ELU, Result shows that the precision of ELU-SegNet is higher than that of the original SegNet by 7.9%—without losing recall. This can imply that ELU is more robust than ReLU to detect road pixels.
  - The result also shows that the recall of ELU-SegNet-R is higher than that of ELU-SegNet by 11.4%, meaning that it can detect more patterns of the roads.
- Discussion on Landscape Metrics
  - The landscape metrics are applied to our framework in order to remove all inaccurately extracted roads (false positives: FP), considered as a negative effect of the rotated image strategy as discussed in the previous section.
  - Result shows that the precision of the network is increased by 7.4% by applying the landscape metrics filtering. This shows that the FP issue has been resolved.

Discussion

- Discussion on Enhanced Deep Learning Framework
  - For the effect of ELU, Result shows that the precision of ELU-SegNet is higher than that of the original SegNet by 7.9%—without losing recall. This can imply that ELU is more robust than ReLU to detect road pixels.
  - The result also shows that the recall of ELU-SegNet-R is higher than that of ELU-SegNet by 11.4%, meaning that it can detect more patterns of the roads.
- Discussion on Landscape Metrics
  - The landscape metrics are applied to our framework in order to remove all inaccurately extracted roads (false positives: FP), considered as a negative effect of the rotated image strategy.
  - Result shows that the precision of the network is increased by 7.4% by applying the landscape metrics filtering. This shows that the FP issue has been resolved.

Discussion

- Discussion on Enhanced Deep Learning Framework
  - For the effect of ELU, Result shows that the precision of ELU-SegNet is higher than that of the original SegNet by 7.9%—without losing recall. This can imply that ELU is more robust than ReLU to detect road pixels.
  - The result also shows that the recall of ELU-SegNet-R is higher than that of ELU-SegNet by 11.4%, meaning that it can detect more patterns of the roads.
- Discussion on Landscape Metrics
  - The landscape metrics are applied to our framework in order to remove all inaccurately extracted roads (false positives: FP), considered as a negative effect of the rotated image strategy as discussed in the previous section.
  - Result shows that the precision of the network is increased by 7.4% by applying the landscape metrics filtering. This shows that the FP issue has been resolved.

Overview

- Introduction
- Related Work
- Methodology
- Experiment
- Results
- Discussion
- **Conclusion**

Conclusion

- In this work, we present a novel deep learning network framework to extract road objects from aerial images.
  - The network is based on Deep Convolutional Encoder-Decoder Network (DCED), called "SegNet."
- To improve the network's precision, we incorporate the recent activation function, called Exponential Linear Unit (ELU), into our proposed method.
- The proposed model is also improved to detect more road patterns by adding eight different rotated images.
- Excessive detected roads are further be eliminated by applying landscape metrics thresholding.
- The experiments were conducted on Massachusetts roads dataset and compared to the existing road extraction techniques.
- The results show that the enhanced SegNet outperforms the original one—10.6% for F1—as well as all other baselines.

Conclusion

- In this work, we present a novel deep learning network framework to extract road objects from aerial images.
    - The network is based on Deep Convolutional Encoder-Decoder Network (DCED), called "SegNet."
- To improve the network's precision, we incorporate the recent activation function, called Exponential Linear Unit (ELU), into our proposed method.
- The proposed model is also improved to detect more road patterns by adding eight different rotated images.
- Excessive detected roads are further be eliminated by applying landscape metrics thresholding.
- The experiments were conducted on Massachusetts roads dataset and compared to the existing road extraction techniques.
- The results show that the enhanced SegNet outperforms the original one—10.6% for F1—as well as all other baselines.

Conclusion

- In this work, we present a novel deep learning network framework to extract road objects from aerial images.
  - The network is based on Deep Convolutional Encoder-Decoder Network (DCED), called "SegNet."
- **To improve the network's precision, we incorporate the recent activation function, called Exponential Linear Unit (ELU), into our proposed method.**
- The proposed model is also improved to detect more road patterns by adding eight different rotated images.
- Excessive detected roads are further be eliminated by applying landscape metrics thresholding.
- The experiments were conducted on Massachusetts roads dataset and compared to the existing road extraction techniques.
- The results show that the enhanced SegNet outperforms the original one—10.6% for F1—as well as all other baselines.

ELU-DCED Architecture

Conclusion

- In this work, we present a novel deep learning network framework to extract road objects from aerial images.
  - The network is based on Deep Convolutional Encoder-Decoder Network (DCED), called "SegNet."
- To improve the network's precision, we incorporate the recent activation function, called Exponential Linear Unit (ELU), into our proposed method.
- **The proposed model is also improved to detect more road patterns by adding eight different rotated images.**
- Excessive detected roads are further be eliminated by applying landscape metrics thresholding.
- The experiments were conducted on Massachusetts roads dataset and compared to the existing road extraction techniques.
- The results show that the enhanced SegNet outperforms the original one—10.6% for F1—as well as all other baselines.

Conclusion

- In this work, we present a novel deep learning network framework to extract road objects from aerial images.
  - The network is based on Deep Convolutional Encoder-Decoder Network (DCED), called "SegNet."
- To improve the network's precision, we incorporate the recent activation function, called Exponential Linear Unit (ELU), into our proposed method.
- **The proposed model is also improved to detect more road patterns by adding eight different rotated images.**
- Excessive detected roads are further be eliminated by applying landscape metrics thresholding.
- The experiments were conducted on Massachusetts roads dataset and compared to the existing road extraction techniques.
- The results show that the enhanced SegNet outperforms the original one—10.6% for F1—as well as all other baselines.

ELU-DCED-R Architecture (*R = Rotate)

Conclusion

- In this work, we present a novel deep learning network framework to extract road objects from aerial images.
  - The network is based on Deep Convolutional Encoder-Decoder Network (DCED), called "SegNet."
- To improve the network's precision, we incorporate the recent activation function, called Exponential Linear Unit (ELU), into our proposed method.
- The proposed model is also improved to detect more road patterns by adding eight different rotated images.
- Excessive detected roads are further be eliminated by applying landscape metrics thresholding.
- The experiments were conducted on Massachusetts roads dataset and compared to the existing road extraction techniques.
- The results show that the enhanced SegNet outperforms the original one—10.6% for F1—as well as all other baselines.

Conclusion

- In this work, we present a novel deep learning network framework to extract road objects from aerial images.
    - The network is based on Deep Convolutional Encoder-Decoder Network (DCED), called "SegNet."
- To improve the network's precision, we incorporate the recent activation function, called Exponential Linear Unit (ELU), into our proposed method.
- The proposed model is also improved to detect more road patterns by adding eight different rotated images.
- Excessive detected roads are further be eliminated by applying landscape metrics thresholding.
- The experiments were conducted on Massachusetts roads dataset and compared to the existing road extraction techniques.
- The results show that the enhanced SegNet outperforms the original one—10.6% for F1—as well as all other baselines.

ELU-DCED-RL Architecture (*L = Landscape Metrics)

Conclusion

- In this work, we present a novel deep learning network framework to extract road objects from aerial images.
  - The network is based on Deep Convolutional Encoder-Decoder Network (DCED), called "SegNet."
- To improve the network's precision, we incorporate the recent activation function, called Exponential Linear Unit (ELU), into our proposed method.
- The proposed model is also improved to detect more road patterns by adding eight different rotated images.
- Excessive detected roads are further be eliminated by applying landscape metrics thresholding.
- **The experiments were conducted on Massachusetts roads dataset and compared to the existing road extraction techniques.**
- The results show that the enhanced SegNet outperforms the original one—10.6% for F1—as well as all other baselines.

Conclusion

- In this work, we present a novel deep learning network framework to extract road objects from aerial images.
  - The network is based on Deep Convolutional Encoder-Decoder Network (DCED), called "SegNet."
- To improve the network's precision, we incorporate the recent activation function, called Exponential Linear Unit (ELU), into our proposed method.
- The proposed model is also improved to detect more road patterns by adding eight different rotated images.
- Excessive detected roads are further be eliminated by applying landscape metrics thresholding.
- The experiments were conducted on Massachusetts roads dataset and compared to the existing road extraction techniques.
- The results show that the enhanced SegNet outperforms the original one—10.6% for F1—as well as all other baselines.

Future work

- In future work, more choices of image segmentation techniques, optimization techniques and/or other activation functions will be investigated and compared to obtain the best DCED-based framework for semantic road segmentation.

Q&A

Thank you

Teerapong panboonyuen

Reference

1. J. Wang, J. Song, M. Chen, and Z. Yang, “Road network extraction: a neural-dynamic framework based on deep learning and a finite state machine,” International Journal of Remote Sensing, vol. 36, 2015.
2. S. Muruganandham, “Semantic Segmentation of Satellite Images using Deep Learning,” Master’s Thesis, 2016.3.
3. S. Saito, T. Yamashita and Y. Aoki, “Multiple Object Extraction from Aerial Imagery with Convolutional Neural Networks,” Journal of Imaging Science and Technology, vol. 60, no. 1, pp. 1–9, 2016.4.
4. J. Long, E. Shelhamer, and T. Darrell. “Fully Convolutional Networks for Semantic Segmentation,” In CVPR, 2015.5.
5. H. Noh, S. Hong, and B. Han, “Learning Deconvolution Network for Semantic Segmentation,” In ICCV, 2015.6.
6. V. Badrinarayanan, A. Handa, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling,” arxiv 2015.7.
7. Volodymyr Mnih. “Machine Learning for Aerial Image Labeling,” PhD thesis. University of Toronto, 2013.8.
8. R. Maurya, P. R. Gupta, and A. S. Shukla, “Road Extraction Using K-Means Clustering and Morphological Operations,” In ICIIP, 2011.9.
9. V. Badrinarayanan, A. Kendall, R. Cipolla, and S. Member, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” arxiv 2016.10.
10. Clevert et al, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),” In ICLR, 2016.

Reference

11. G. Xu, D. Zhang and X. Liu, "Road Extraction in High Resolution Images from Google Earth," 2009, In ICICS, 2009. 12.
12. F. Visin, M. Ciccone, and A. Romero, “ReSeg: A Recurrent Neural Network-based Model for Semantic Segmentation,” arxiv 2015.
13. M. Volpi and V. Ferrari, "Semantic Segmentation of Urban Scenes by Learning Local Class Interactions," In CVPRW, 2015.
14. J. Liu, B. Liu, and H. Lu, “Detection guided deconvolutional network for hierarchical feature learning,” Pattern Recognition., vol. 48, no. 8, pp. 2645–2655, 2015.
15. S. Hong, H. Noh, and B. Han, “Decoupled deep neural network for semi-supervised semantic segmentation,” in NIPS, pp. 1495–1503, 2015.
16. O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical.” arxiv 2015.
17. K. Mcgarigal, “Landscape metrics for categorical map patterns,” vol. 2001, no. Chapter 5, pp. 1–77, 2001.
18. C. Poullis, “Tensor-Cuts: A simultaneous multi-type feature extractor and classifier and its application to road extraction from satellite images,” ISPRS J. Photogramm. Remote Sens., vol. 95, pp. 93–108, 2014.
19. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by ReducingInternal Covariate Shif," International Conference on Machine Learning, 2015.
20. Simonyan, K. and Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition", In ICLR, 2015.
21. M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” International Journal of Computer Vision, vol. 111, no. 1, pp. 98–136, 2015.
22. X. Huang and L. Zhang, “Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines,” Int. J. Remote Sens., vol. 30, no. 8, pp. 1977–1987, 2009.
23. T. Panboonyuen, P. Vateekul and K. Jitkajornwanich, “Image Vectorization of Road Satellite Data Sets,” Journal of Remote Sensing and GIS Association of Thailand, Vol.17 (ISSN 1513-4261), Special Issue (Page 310-321), 2016.
24. C. Sujatha and D. Selvathi, “Connected component-based technique for automatic extraction of road centerline in high resolution satellite images,” EURASIP J. Image Video Process., vol. 2015, no. 1, p. 8, 2015.
25. D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, and M. Datcu, “Classification with an edge: improving semantic image segmentation with boundary detection.” In CVPR, 2014.

Appendix

Experiment (Implementation)

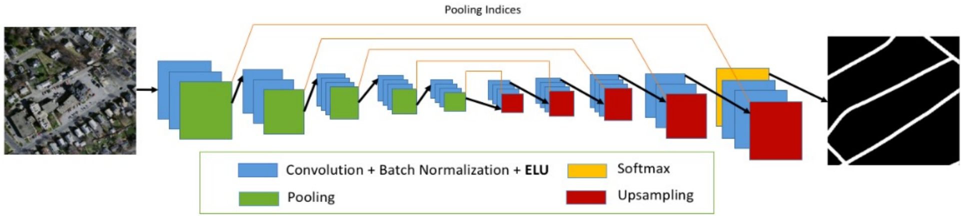
- Lasagne (Based Theano): Deep Learning Library (Python)
- Ubuntu 14.04.5 LTS (Trusty Tahr)
- CUDA 7.5
- CuDNN 5.1
- Nvidia GeForce GTX 960 (4 GB)

Experiment (Implementation)

- The training procedure took approximately 32 hours for the original training datasets.
- 48 hours for the amplified training datasets, and finished after 200 epochs.
- In each epoch, 576 seconds were used for the original training datasets
- 864 seconds were used for the amplified training datasets.

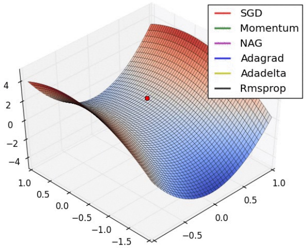
Experiment (Implementation)

- The encoder network consists of convolution layer and pooling layer. A technique, called batch normalization is used to speed up the learning process of the CNN by reducing internal covariate shift. In the encoder network, the number of layers is reduced to 13 layers (VGG16) by removing the last three layers (fully connected layers).
- To do optimization for training networks, stochastic gradient descent (SGD) with a fixed learning rate of 0.1 and momentum of 0.9 are used. In each training round (epoch), a mini-batch (a set of 12 images) is chosen such that each image is used once. The model with the best performance on the validation dataset in each epoch will be selected.



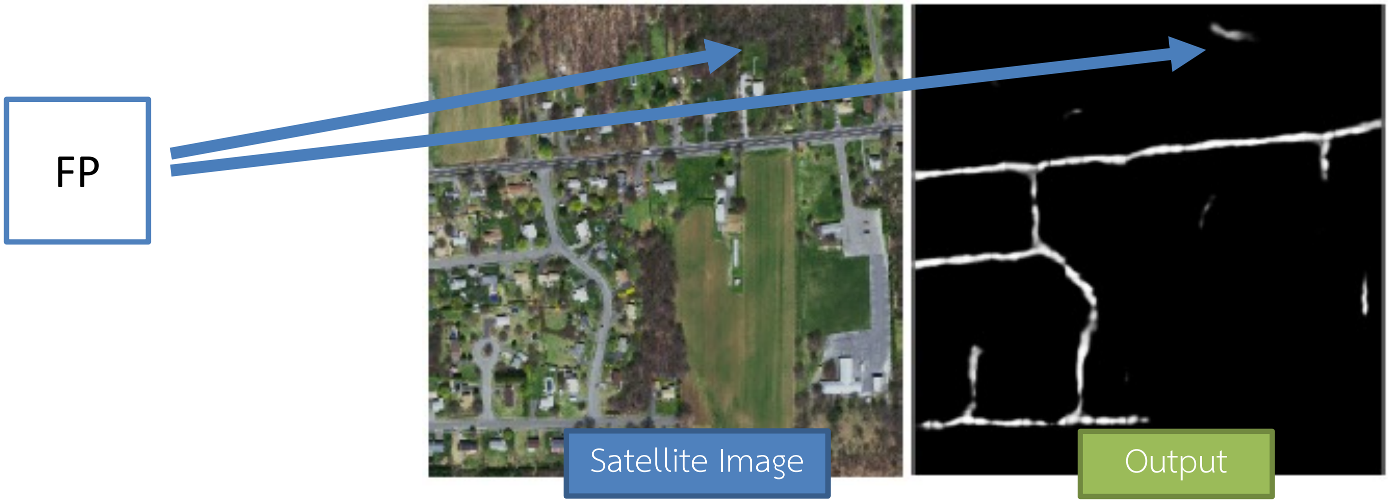
Experiment (Implementation)

- The encoder network consists of convolution layer and pooling layer. A technique, called batch normalization is used to speed up the learning process of the CNN by reducing internal covariate shift. In the encoder network, the number of layers is reduced to 13 layers (VGG16) by removing the last three layers (fully connected layers).
- To do optimization for training networks, stochastic gradient descent (SGD) with a fixed learning rate of 0.1 and momentum of 0.9 are used. In each training round (epoch), a mini-batch (a set of 12 images) is chosen such that each image is used once. The model with the best performance on the validation dataset in each epoch will be selected.



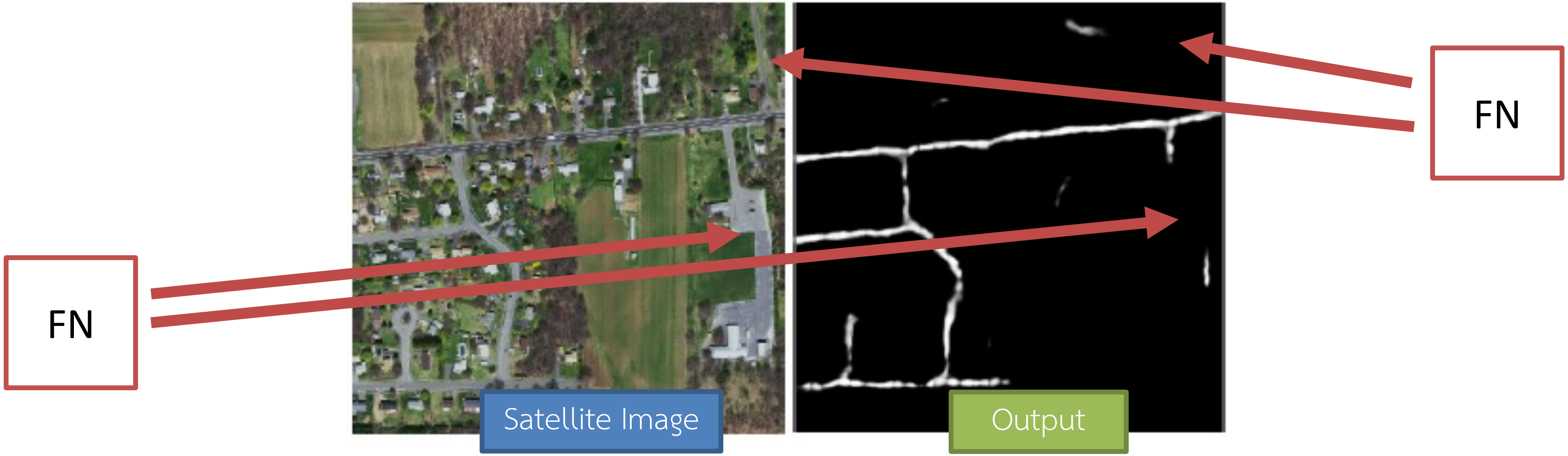
Statement of the problems

- Deep Learning Problem
  - False Positive Problem
  - False Negative Problem



Statement of the problems

- Deep Learning Problem
  - False Positive Problem
  - False Negative Problem

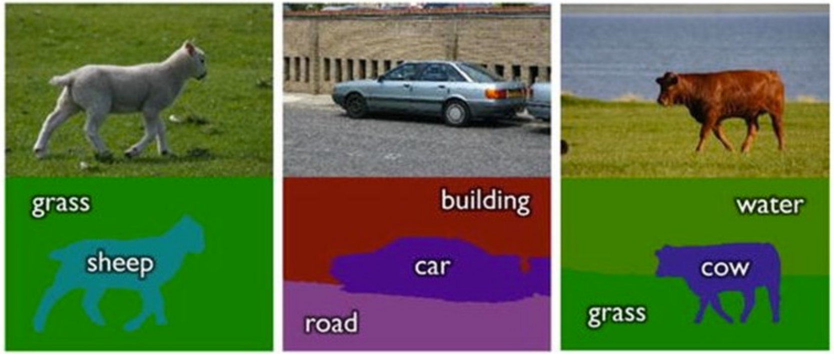


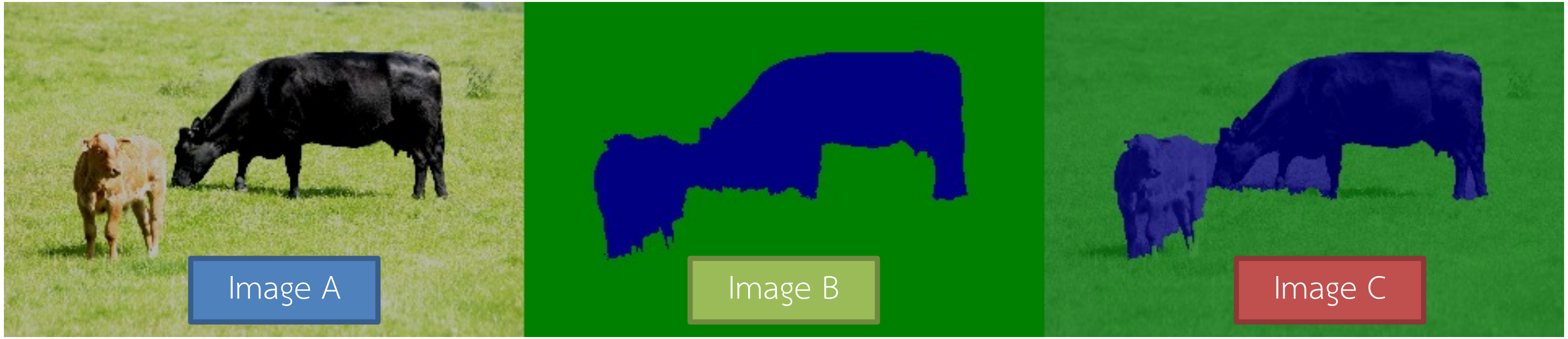
Theory

- Semantic Segmentation
- Neural Network
- Convolution Neural Network
- Connected Component
- Gaussian Smoothing
- Performance Evaluation

Theory

- **Semantic Segmentation**
- Neural Network
- Convolution Neural Network
- Connected Component
- Gaussian Smoothing
- Performance Evaluation

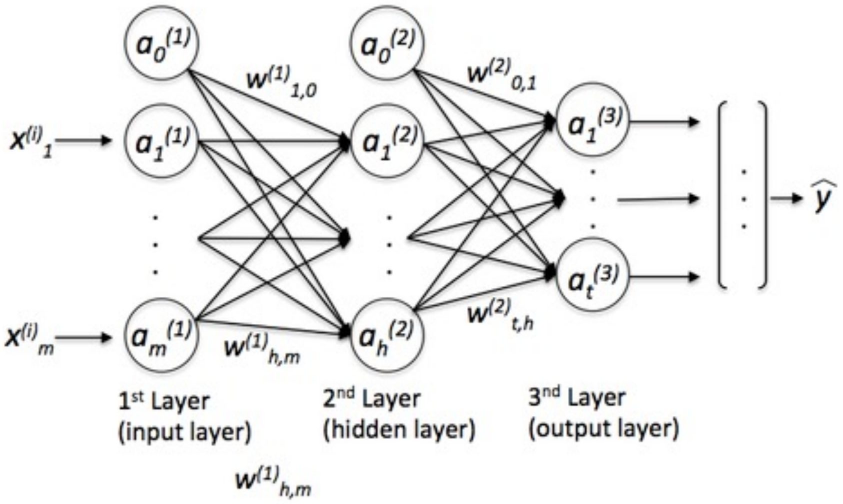


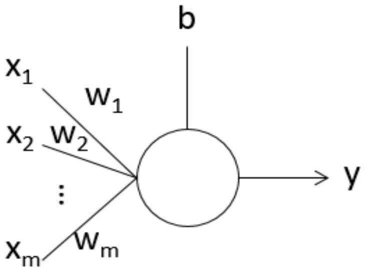


Reference : http://stackoverflow.com/questions/33947823/what-is-semantic-segmentation-compared-to-segmentation-and-scene-labeling

Theory

- Semantic Segmentation
- **Neural Network**
- Convolution Neural Network
- Connected Component
- Gaussian Smoothing
- Performance Evaluation



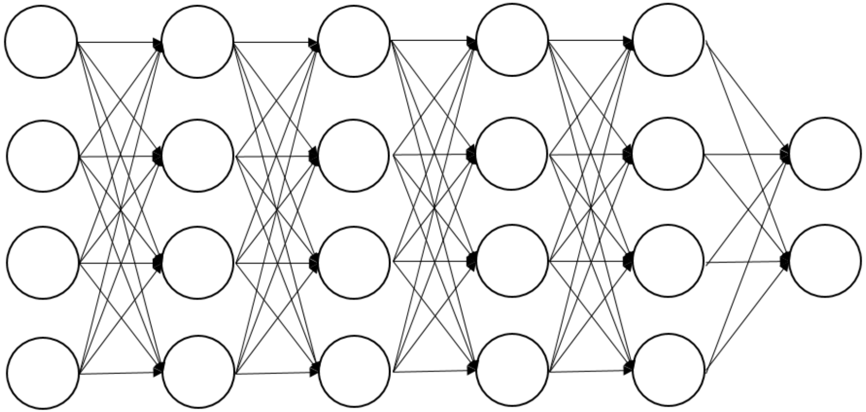


$$a_j^l = g(\sum_{k=1}^{n} w_{jk}^l x_k^{l-1} + b_j^l)$$

Reference : Adit Deshpande, CS Undergrad at UCLA ('19)

Theory

- Semantic Segmentation
- Neural Network
- Convolution Neural Network
- Connected Component
- Gaussian Smoothing
- Performance Evaluation

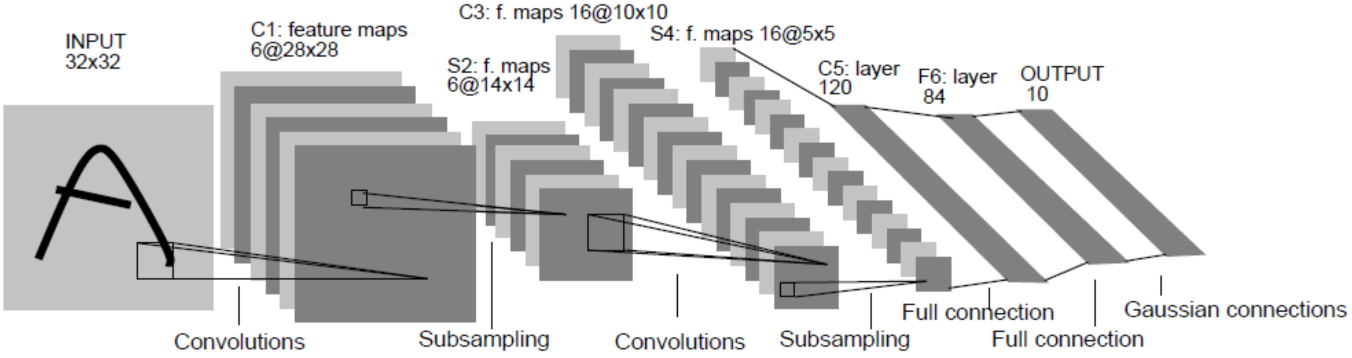


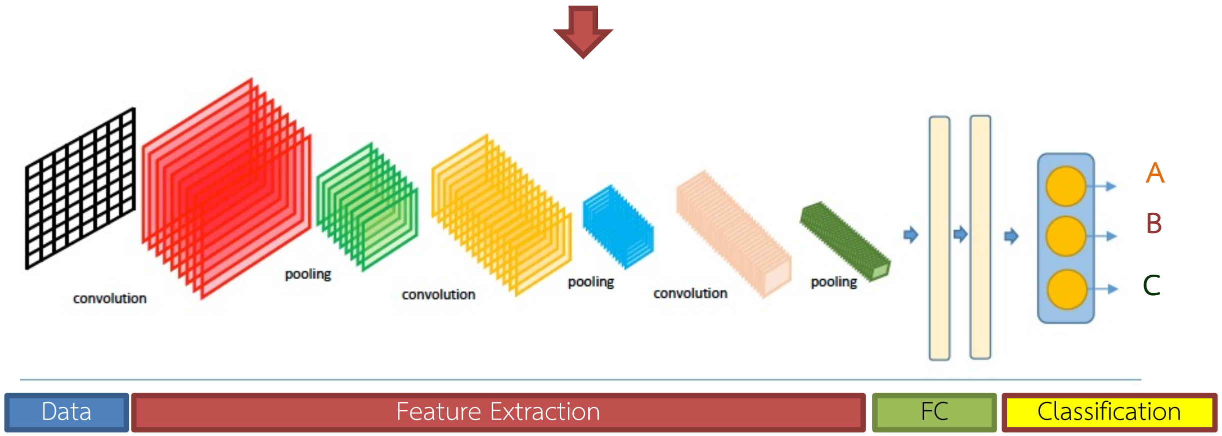
Deep Neural Network

Convolution Neural Network

In CNN architecture shared weight together

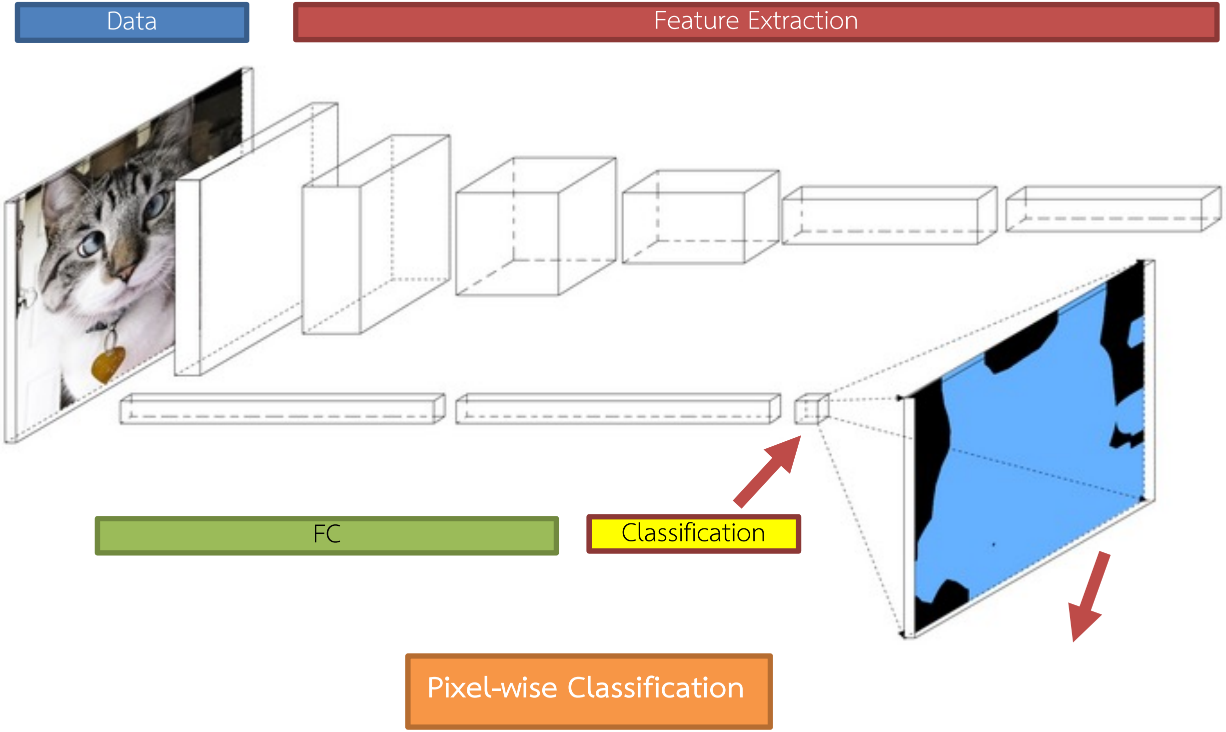
- **Overview (1)**
- Overview (2)
- Convolution Layer
- Deconvolution Layer
- Pooling Layer
- Unpooling Layer
- Fully Connected Layer
- Classification Layer





Convolution Neural Network

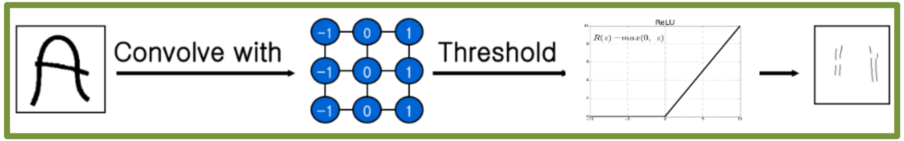
- Overview (1)
- **Overview (2)**
- Convolution Layer
- Deconvolution Layer
- Pooling Layer
- Unpooling Layer
- Fully Connected Layer
- Classification Layer



Convolution Neural Network

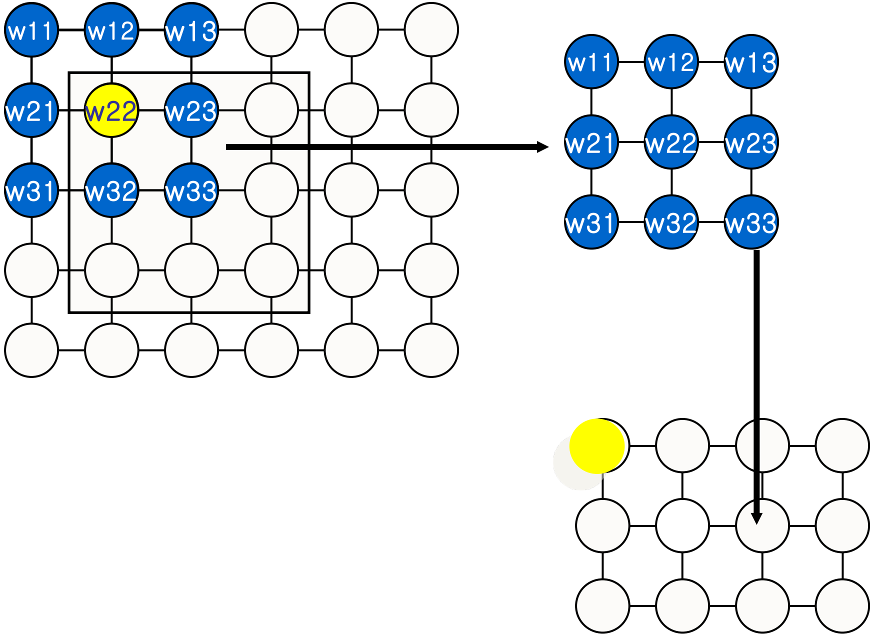
Activation Function

ReLU is the most interesting



- Overview (1)
- Overview (2)
- **Convolution Layer**
- Deconvolution Layer
- Pooling Layer
- Unpooling Layer
- Fully Connected Layer
- Classification Layer

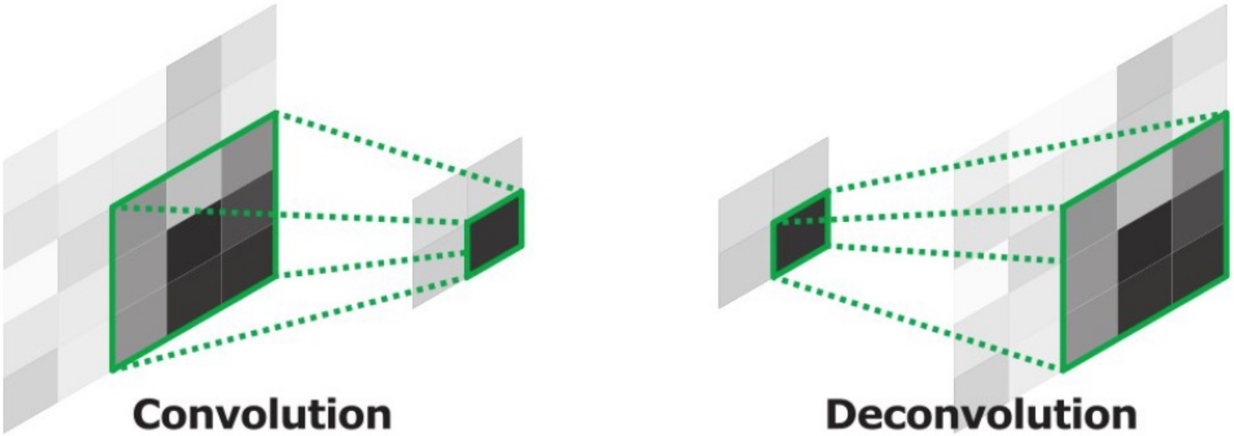
Input



Output

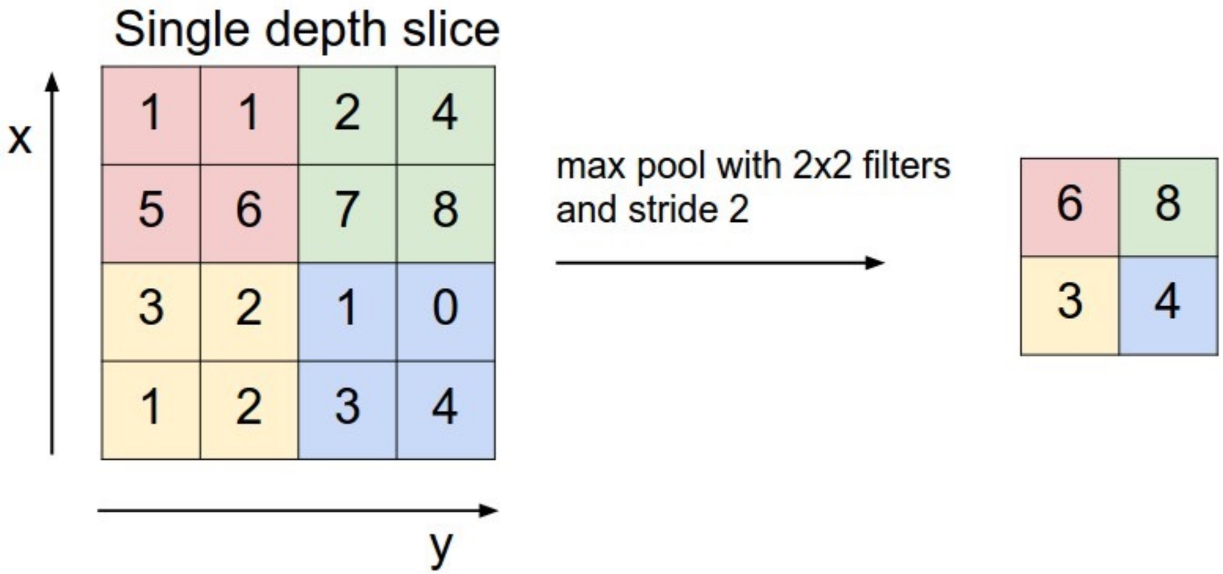
Convolution Neural Network

- Overview (1)
- Overview (2)
- Convolution Layer
- **Deconvolution Layer**
- Pooling Layer
- Unpooling Layer
- Fully Connected Layer
- Classification Layer



Convolution Neural Network

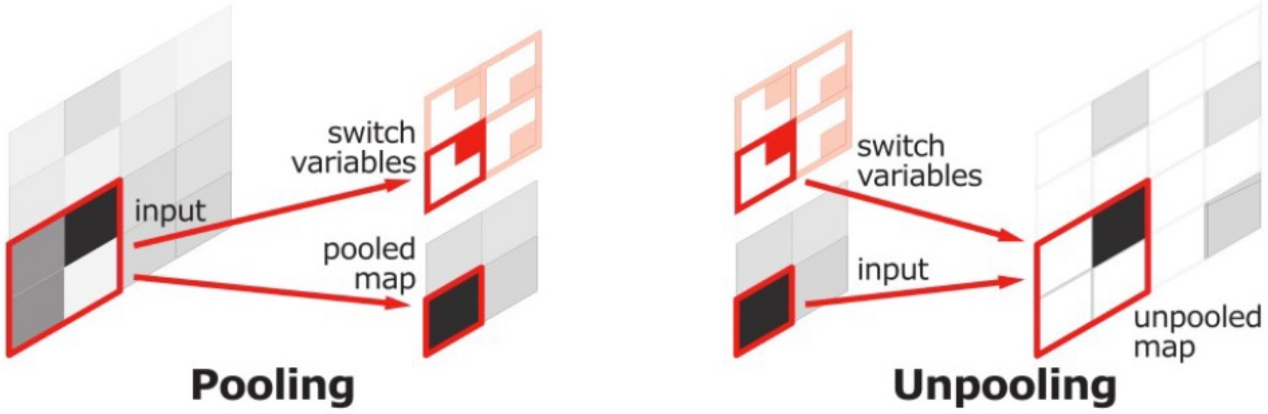
- Overview (1)
- Overview (2)
- Convolution Layer
- Deconvolution Layer
- **Pooling Layer**
- Unpooling Layer
- Fully Connected Layer
- Classification Layer



Reference : Adit Deshpande, CS Undergrad at UCLA ('19)

Convolution Neural Network

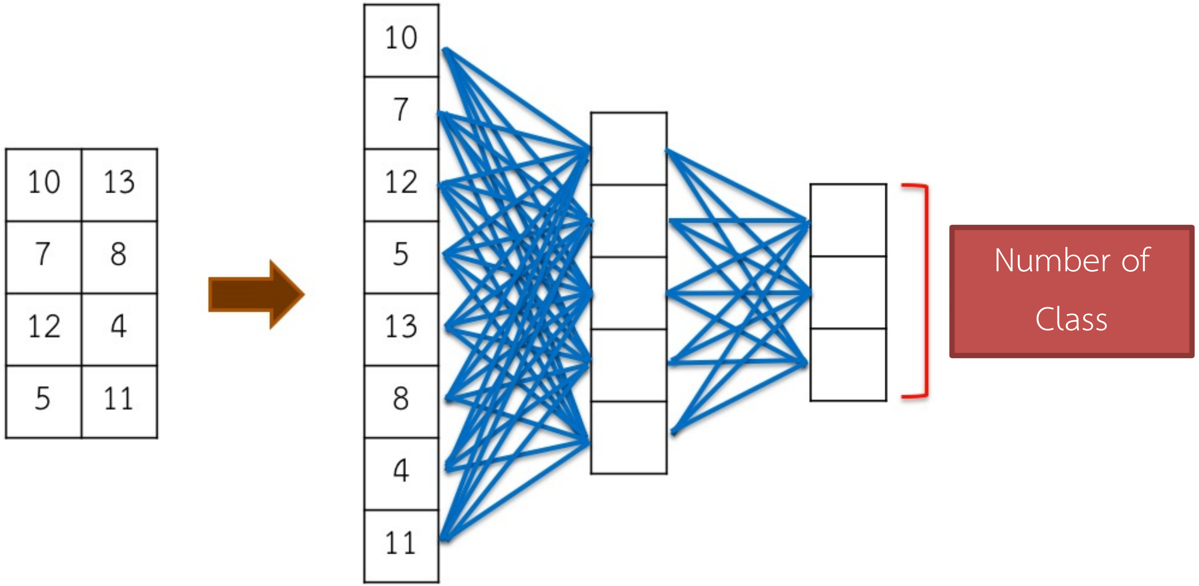
- Overview (1)
- Overview (2)
- Convolution Layer
- Deconvolution Layer
- Pooling Layer
- **Unpooling Layer**
- Fully Connected Lay
- Classification Layer



Reference : Adit Deshpande, CS Undergrad at UCLA ('19)

Convolution Neural Network

- Overview (1)
- Overview (2)
- Convolution Layer
- Deconvolution Layer
- Pooling Layer
- Unpooling Layer
- **Fully Connected Layer**
- Classification Layer

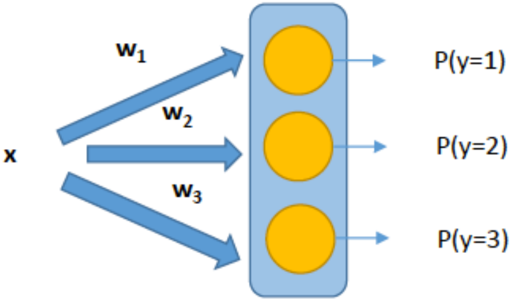


Reference : Adit Deshpande, CS Undergrad at UCLA ('19)

Convolution Neural Network

- Overview (1)
- Overview (2)
- Convolution Layer
- Deconvolution Layer
- Pooling Layer
- Unpooling Layer
- Fully Connected Layer
- **Classification Layer**

Softmax Function



P(y=1)+P(y=2)+P(y=3) = 1

$$P(y=1) = \frac{\exp(\boldsymbol{w_1} \cdot \boldsymbol{x})}{\sum_{k=1}^{3} \exp(\boldsymbol{w_k} \cdot \boldsymbol{x})}$$

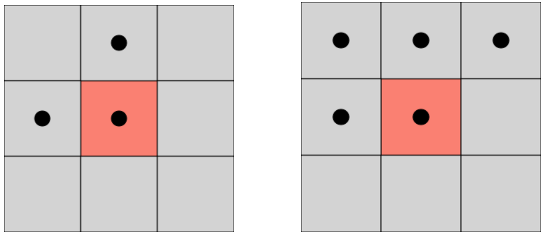
$$P(y=2) = \frac{\exp(\boldsymbol{w_2} \cdot \boldsymbol{x})}{\sum_{k=1}^{3} \exp(\boldsymbol{w_k} \cdot \boldsymbol{x})}$$

$$P(y=3) = \frac{\exp(\boldsymbol{w_3} \cdot \boldsymbol{x})}{\sum_{k=1}^{3} \exp(\boldsymbol{w_k} \cdot \boldsymbol{x})}$$

Reference : Adit Deshpande, CS Undergrad at UCLA ('19)

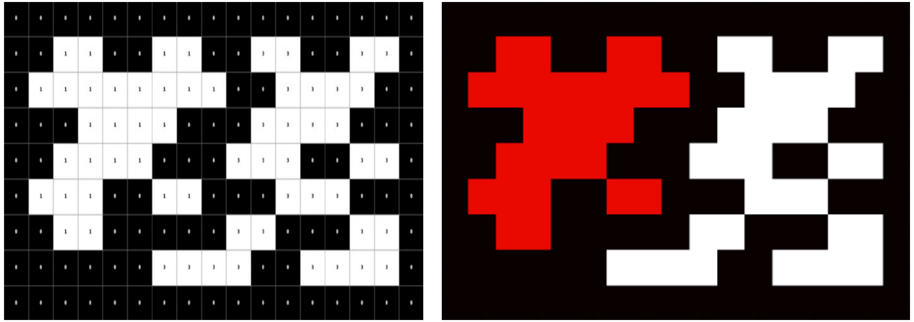
Theory

- Semantic Segmentation
- Neural Network
- Convolution Neural Network
- **Connected Component**
- Gaussian Smoothing
- Performance Evaluation



4 Point

8 Point



Example: **Connected-Component Labeling**

https://en.wikipedia.org/wiki/Connected-component_labeling

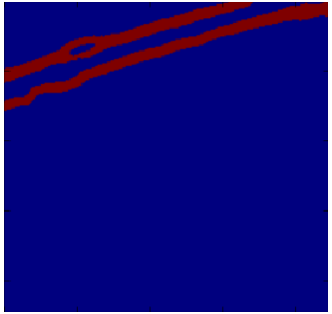
Theory

- Semantic Segmentation
- Neural Network
- Convolution Neural Network
- Connected Component
- **Gaussian Smoothing**
- Performance Evaluation

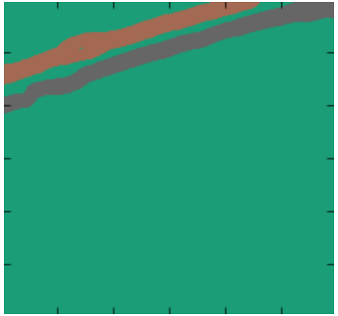
$$S_{i,j} = I(i,j) * G(i,j,\sigma)$$

$S_{i,j}$ is output
$I(i,j)$ is a finding edge in image
$G(i,j,\sigma)$ is Gaussian function

ภาพ A



ภาพ B

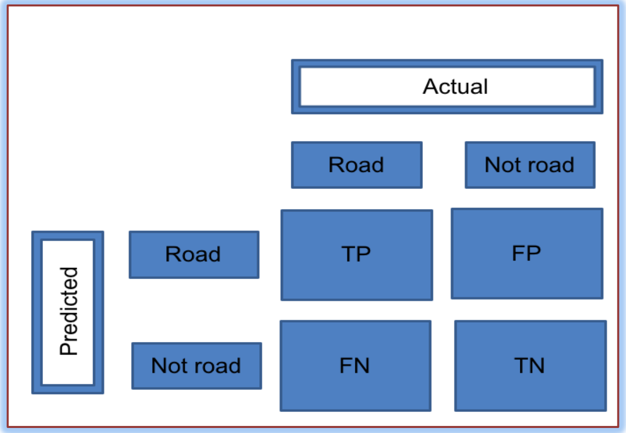


ภาพ C

http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

Theory

- Semantic Segmentation
- Neural Network
- Convolution Neural Network
- Connected Component
- Gaussian Smoothing
- **Performance Evaluation**



$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Thank you

Teerapong panboonyuen

Deep Learning Research

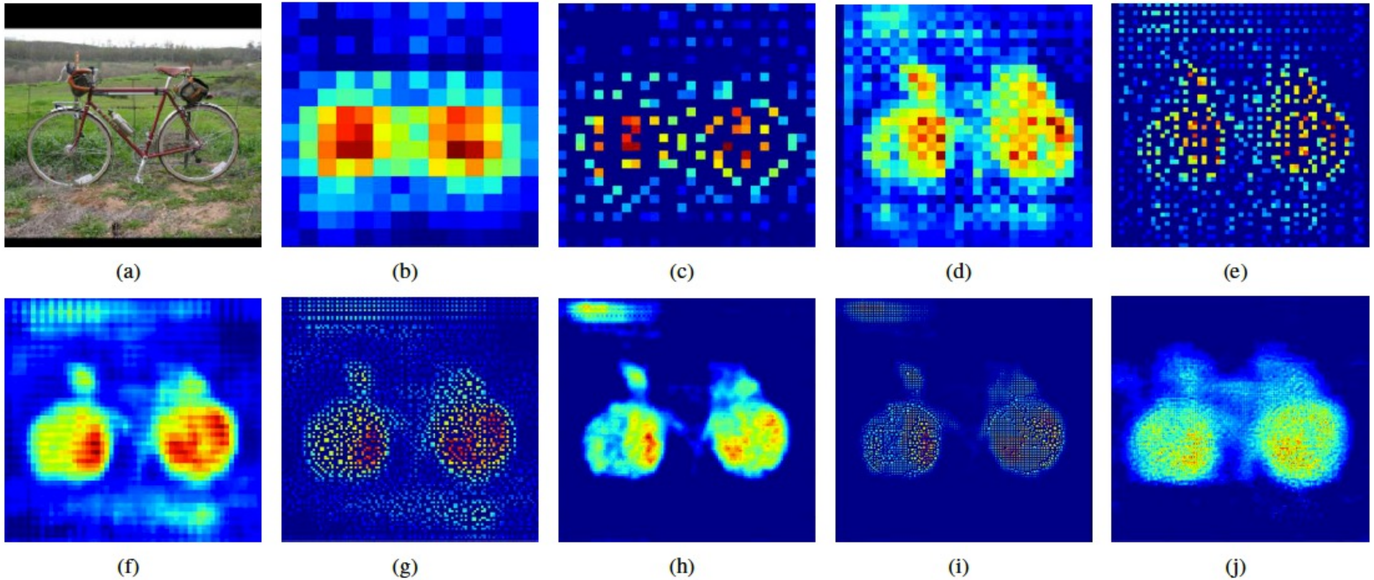


Figure 4. Visualization of activations in our deconvolution network. The activation maps from (b) to (j) correspond to the output maps from lower to higher layers in the deconvolution network. We select the most representative activation in each layer for effective visualization. The image in (a) is an input, and the rest are the outputs from (b) the last 14×14 deconvolutional layer, (c) the 28×28 unpooling layer, (d) the last 28×28 deconvolutional layer, (e) the 56×56 unpooling layer, (f) the last 56×56 deconvolutional layer, (g) the 112×112 unpooling layer, (h) the last 112×112 deconvolutional layer, (i) the 224×224 unpooling layer and (j) the last 224×224 deconvolutional layer. The finer details of the object are revealed, as the features are forward-propagated through the layers in the deconvolution network. Note that noisy activations from background are suppressed through propagation while the activations closely related to the target classes are amplified. It shows that the learned filters in higher deconvolutional layers tend to capture class-specific shape information.

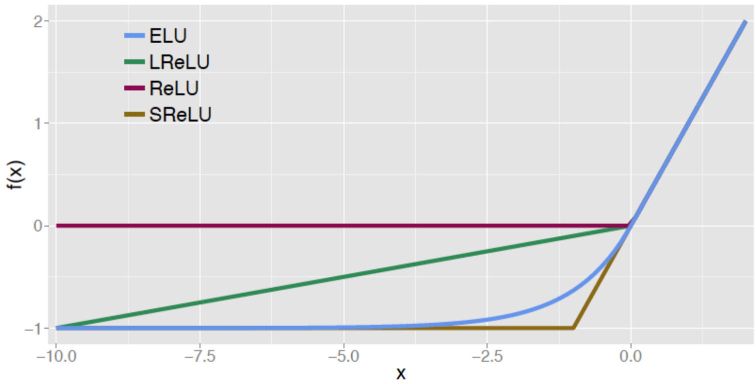
Reference : Adit Deshpande, CS Undergrad at UCLA ('19)

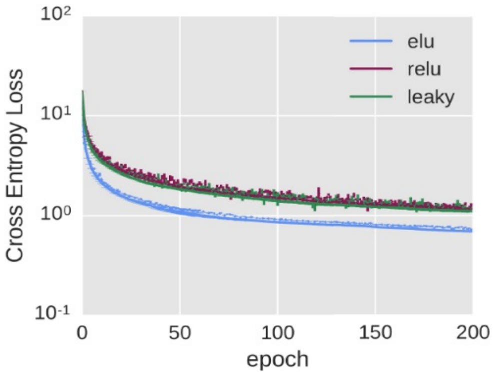
Deep Learning Research

- Clevert (2016)

The *exponential linear unit* (ELU) with $0 < \alpha$ is

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha\,(\exp(x) - 1) & \text{if } x \le 0 \end{cases}, \quad f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ f(x) + \alpha & \text{if } x \le 0 \end{cases}.$$



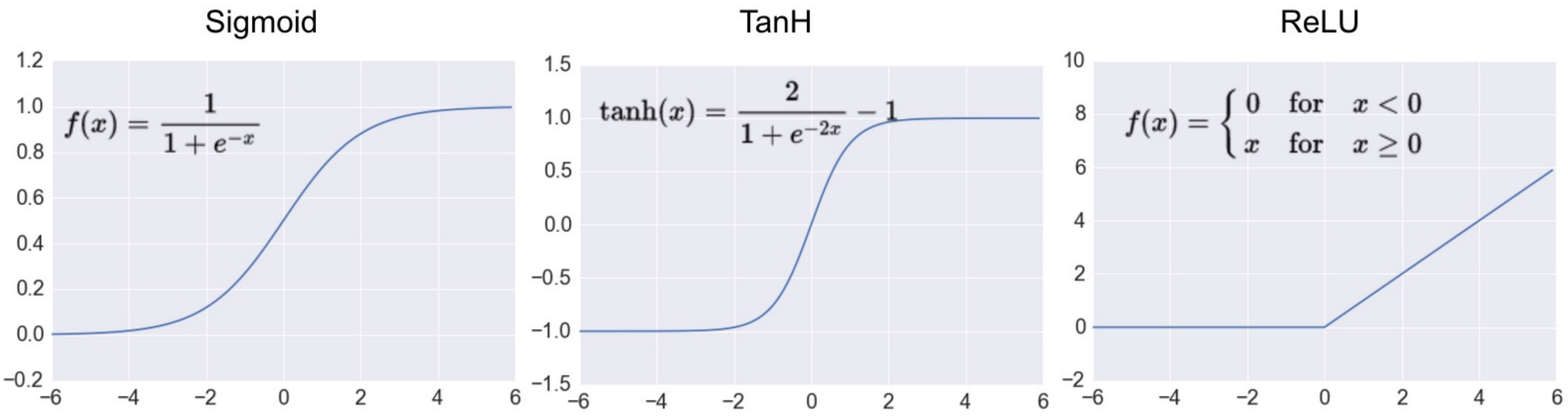


- The Rectified linear unit (ReLU), the leaky ReLU (LReLU, alpha = 0.1), the shifted ReLUs (SReLUs), and the exponential linear unit (ELU, alpha = 1.0).

Clevert et al, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),” In ICLR, 2016.

Deep Learning Research

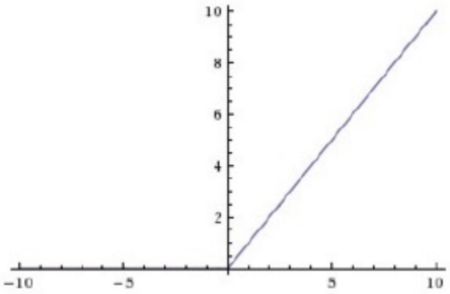
popularity activation function

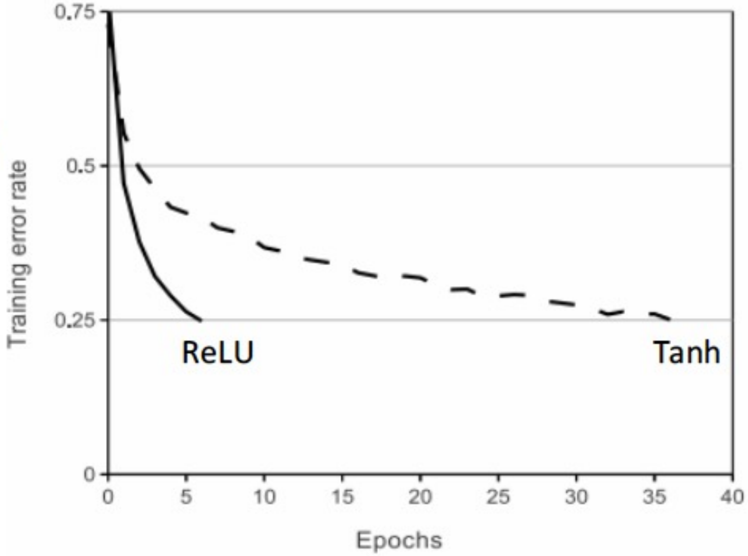


Clevert et al, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," In ICLR, 2016.

Deep Learning Research

- Sigmoid
- Tanh
- Rectified linear unit (ReLU): $f(x) = \max(0, x)$
  - Become very popular in the last few years.





[http://cs231n.github.io/convolutional-networks/]

Clevert et al, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," In ICLR, 2016.

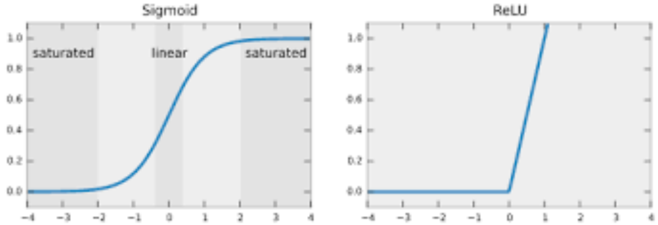
Deep Learning Research

How does ELU activation function help convergence, and what's its advantages over ReLU or sigmoid or tanh function?

Saiprasad Koturwar, studied at Indian Institute of Technology, Bombay
Written Sep 15

ELU(Exponential linear unit) function takes care of the Vanishing gradient problem. The other mentioned activation functions are prone to reaching a point from where the gradient of the functions does not change.

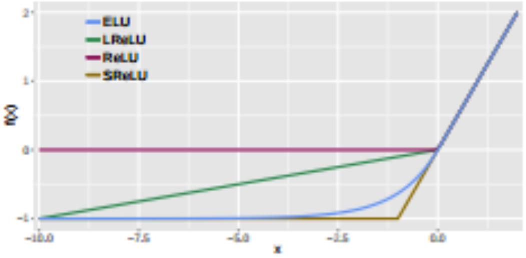
Lets start with the advantages that relu gives over sigmoid and tanh (tanh and sigmoid are simailar) so if you look at the following plot sigmoid/tanh gets saturated for large values of x, so as your activation value increases the corresponding gradient approaches zero and the corrosponding neurons effectively learn nothing. But with relu you do not have that problem as you will get finite gradient no matter what the value of x.



https://www.quora.com/How-does-ELU-activation-function-help-convergence-and-whats-its-advantages-over-ReLU-or-sigmoid-or-tanh-function

Deep Learning Research

But the problem with relu is that it's mean is not zero. If the mean value of activation is zero you get a faster learning.But if you use just a linear activation function (which would have mean activation zero) your overall network becomes linear and which will effectively be equal to a single layer network and with linear networks there is very little that you can learn from the data, that's why we use non linear activation functions. Now what ELU does is that it tries to make the mean activation close to zero and as it is an exponential function it does not saturate(I have not used this activation function yet), you can conclude this from ELU graph.



https://www.quora.com/How-does-ELU-activation-function-help-convergence-and-whats-its-advantages-over-ReLU-or-sigmoid-or-tanh-function

Deep Learning Research

How does softmax function work in AI field?

Steven Schmatz, studying machine learning

Written Jul 31, 2014 · Upvoted by Nikhil Dandekar, worked on machine learning at Microsoft, Foursquare and Quora and Jay Verkuilen

The softmax function is important in the field of machine learning because it can map a vector to a probability of a given output in binary classification.

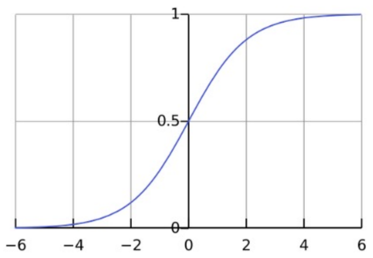
The softmax (logistic) function is defined as:

$$h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)},$$

where θ represents a vector of weights, and x is a vector of input values. This function is used to approximate a target function $y \in \{0, 1\}$ in binary classification. The softmax function produces a scalar output $h_\theta(x) \in \mathbb{R}, 0 < h_\theta(x) < 1$.

https://www.quora.com/How-does-ELU-activation-function-help-convergence-and-whats-its-advantages-over-ReLU-or-sigmoid-or-tanh-function

Deep Learning Research



This can be seen as the *confidence* that your test point has an output value of 1. When $-\theta^T x$ is very small, then the probability $y = 1$ is small. When $-\theta^T x$ is very large, $h_\theta(x)$ approaches 1 as the probability that $y = 1$ approaches 100%.

Note that this is also widely used in artificial neural network design, as the "activation function" of each neuron. Each neuron receives a vector of outputs from other neurons that fired, each axon with its own weighting. These are then linearly combined and used in the softmax function to determine if the next neuron fires or not.

https://www.quora.com/How-does-ELU-activation-function-help-convergence-and-whats-its-advantages-over-ReLU-or-sigmoid-or-tanh-function

Deep Learning Research

Why is it better to use Softmax function than sigmoid function?

Sri Krishna, Works on Deep learning for vision. BITSian.
Written Oct 21

I'm guessing you're asking **only wrt** the last layer for classification, in general Softmax is used (Softmax Classifier) when 'n' number of classes are there. Sigmoid or softmax both can be used for binary (n=2) classification.

Sigmoid:

$$S(t) = \frac{1}{1+e^{-t}}$$

Softmax:

$$h_\theta(x) = \frac{1}{1+\exp(-\theta^T x)},$$

Softmax is kind of Multi Class Sigmoid, but if you see the function of Softmax, the sum of all softmax units are supposed to be 1. In sigmoid it's not really necessary.

https://www.quora.com/How-does-ELU-activation-function-help-convergence-and-whats-its-advantages-over-ReLU-or-sigmoid-or-tanh-function

Deep Learning Research

Digging deep, you can **also** use **sigmoid** for **multi-class** classification. When you use a softmax, basically you get a probability of each class, (**join distribution and a multinomial likelihood**) whose sum is bound to be one. In case you use sigmoid for multi class classification, it'd be like a **marginal distribution and a Bernoulli likelihood**, p(y0/x) , p(y1/x) etc

As told earlier, in the case of softmax, **increasing** the output value of one class makes the the others go down (sigma=1). So, sigmoids can probably be preferred over softmax when your outputs are **independent** of one another. To put it more simple, if there are multiple classes and each input can belong to **exactly** one class, then it absolutely makes sense to use softmax, in the **other** cases, sigmoid seems better.

One more thing is, people mostly use **ReLu** activations these days (in the hidden layers) and using sigmoid blows up ReLu apparently, might be one of the reason why people prefer softmax.

PS - In case you're talking about activation functions in hidden layers, softmax isn't really used. And ReLu is better to use than sigmoid.

https://www.quora.com/How-does-ELU-activation-function-help-convergence-and-whats-its-advantages-over-ReLU-or-sigmoid-or-tanh-function

Deep Learning Research

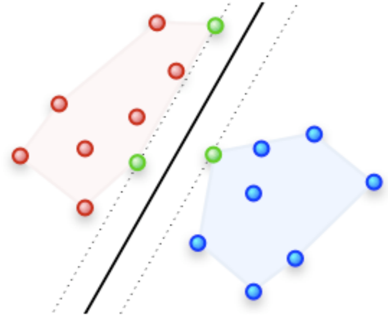
What is difference between SVM and Neural Networks?

Kaushik Kasi, (Data Science && Bitcoin) Enthusiast
Written Apr 12, 2015

SVM and NN are both supervised learning methods, but they work a bit differently. As Eren mentioned, in theory they may not be that different from each other.

Support Vector Machine

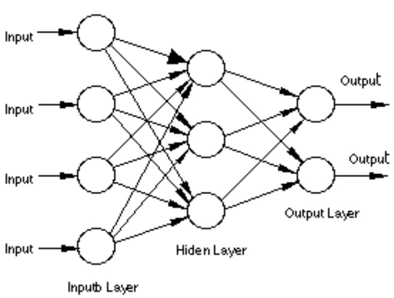
SVM fits a hyperplane/function between 2 different classes given a maximum margin parameter. This hyperplane attempts to separate the classes so that each falls on either side of the plane, and by a specified margin. There is a specific cost function for this kind of model which adjusts the plane until error is minimized.



Neural Network

A neural network has several input, hidden, and output nodes. Each node applies a function some data (could be softmax, linear, logistic), and returns an output. Every node in the proceeding layer takes a weighted average of the outputs of the previous layer, until an output is reached. The reasoning is that multiple nodes can collectively gain insight about solving a problem (like classification) that an individual node cannot. The cost function differs for this type of model -- the weights between nodes adjust to minimize error.

A Typical Neural Network



https://www.quora.com/How-does-ELU-activation-function-help-convergence-and-whats-its-advantages-over-ReLU-or-sigmoid-or-tanh-function

Deep Learning Research

Road network extraction: a neural-dynamic framework based on deep learning and a finite state machine

Jun Wang[a]*, Jingwei Song[a], Mingquan Chen[b], and Zhi Yang[a]

International Journal of Remote Sensing, 2015
Vol. 36, No. 12, 3144–3169, http://dx.doi.org/10.1080/01431161.2015.1054049
Taylor & Francis
Taylor & Francis Group

(2) Learning-based methods.

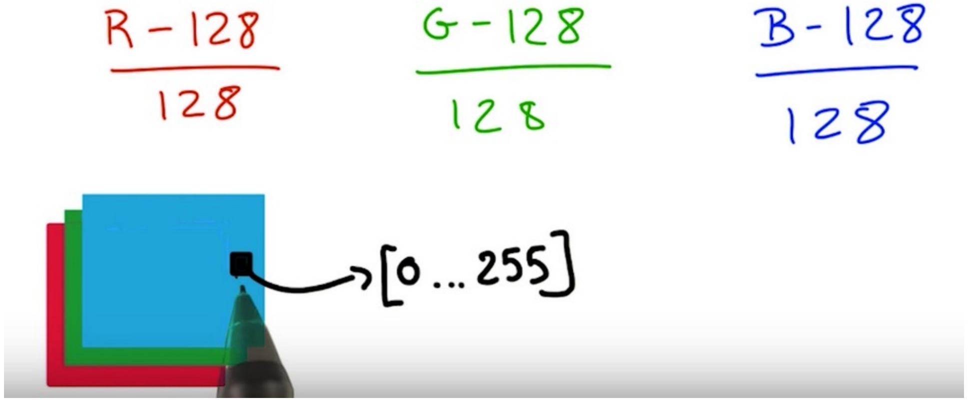
Road extraction from imagery is actually a classification problem (Mnih 2013; Singh and Garg 2014). Features can be extracted and trained by many machine learning algorithms, e.g. neural networks and support vector machine (SVM) (Huang and Zhang 2009). However, due to lack of labelled samples and the complexity of road features, it is hard to make the most of the power in modern classifiers.

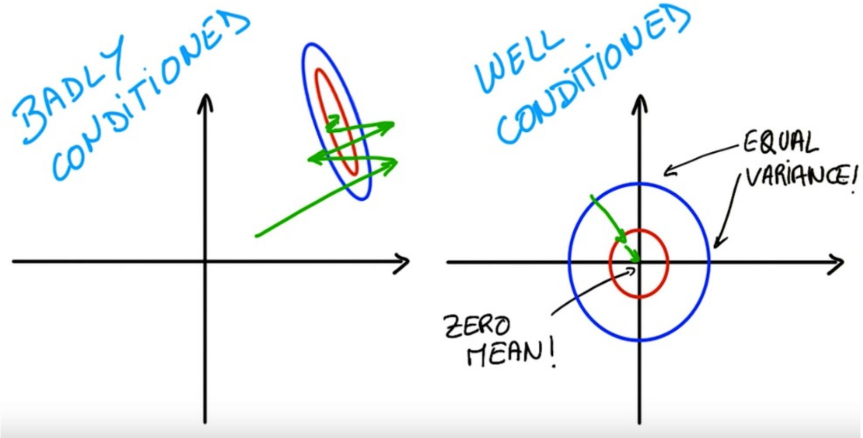
Barsi and Heipke (2003) proposed a road junction detector by training a neural network on a training data set of 542 junction and non-junction samples. Spectral and structural features at different scales are extracted and trained in several SVMs (Huang and Zhang 2009). The results of spectral and structural classification are integrated by applying a fusion method of majority voting. Principal component analysis (PCA) is applied to reduce the dimensionality of the input image patch, in this way large context can be used in training (Mnih and Hinton 2010; Mnih 2013). An unsupervised learning procedure was utilized for pretraining, before the restricted Boltzmann machines were trained with complicate large challenging urban data sets. These methods that learn to predict whether the objects are roads prove that it is very arduous because of the complexity of the surface.

Considering these deficiencies, the authors propose a semi-automatic neural-dynamic tracking framework based on deep convolutional neural networks (DNNs) and finite state machine (FSM). The framework consists of two processing steps: the training step and the tracking step.

Deep Learning Research

- Normalization om each channel of satellite image.
  - R G B [[0,255] [0,255] [0,255]] => [[-1, 1), [-1, 1), [-1, 1)]

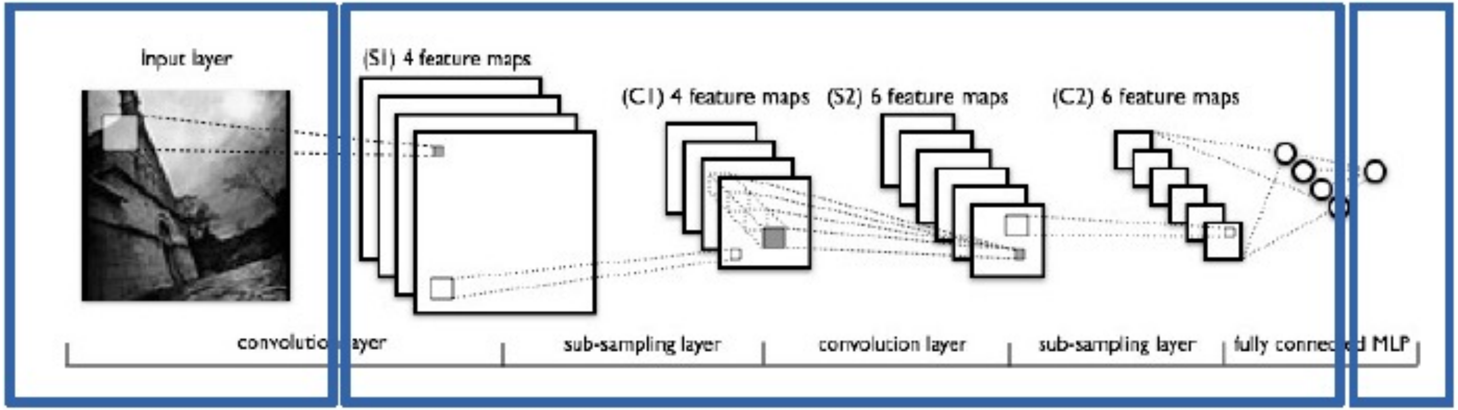




- Normalization on one channel of target image.
  - Gray [0,255] => {0,1}

Reference : https://classroom.udacity.com/courses/ud730/lessons/6370362152/concepts/71191606550923

Deep Learning Research

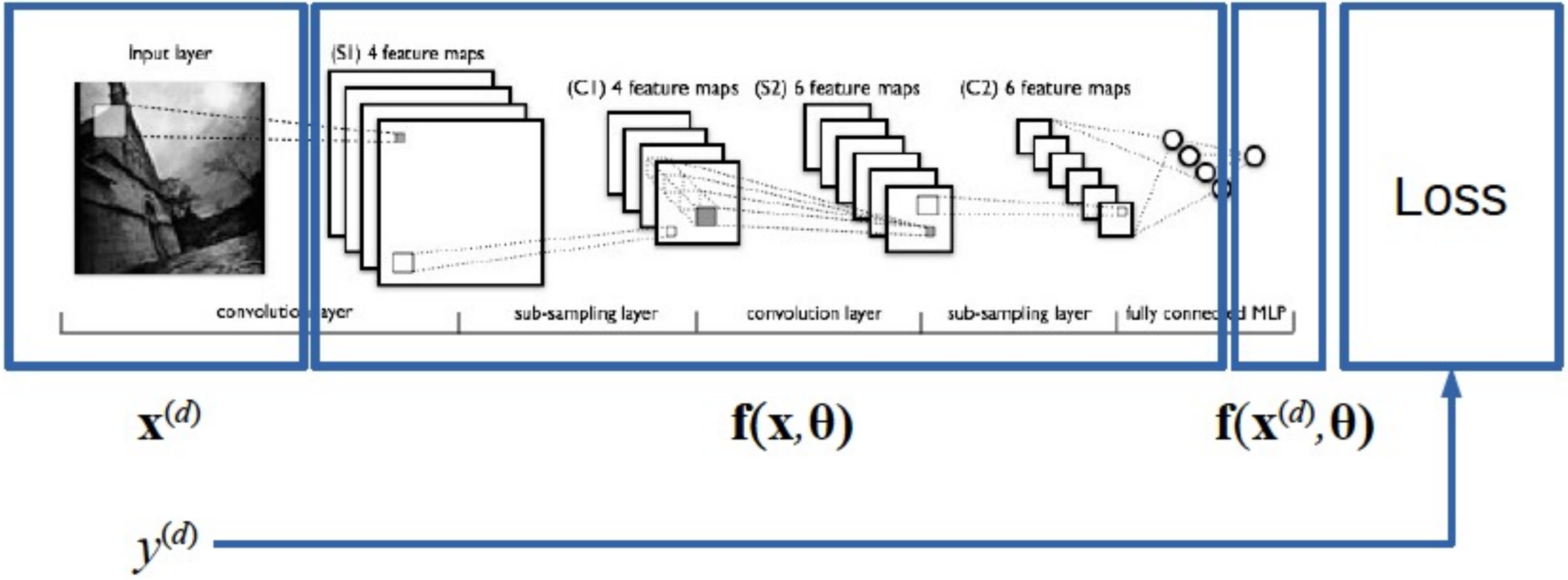


$\mathbf{x}^{(d)}$ $\mathbf{f}(\mathbf{x},\theta)$ $\mathbf{f}(\mathbf{x}^{(d)},\theta)$

$y^{(d)}$ $y^{(d)}$

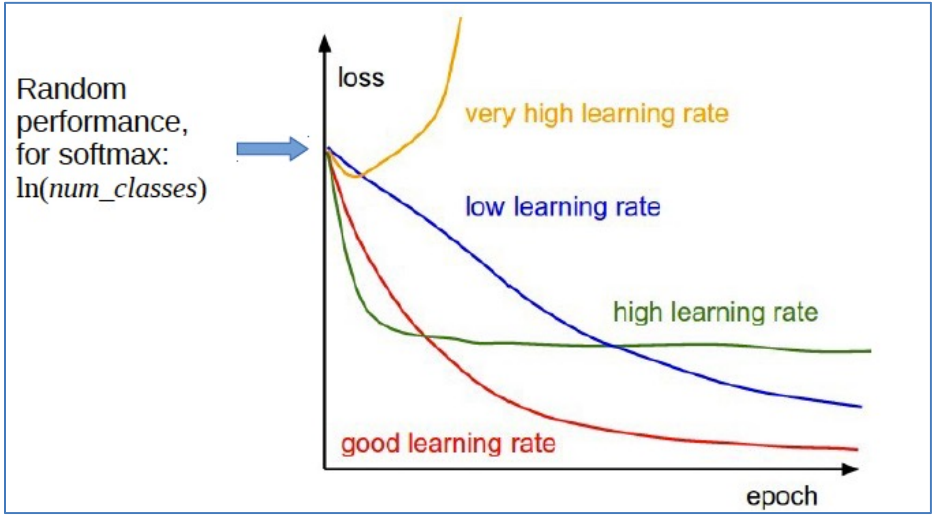
https://www.quora.com/How-does-ELU-activation-function-help-convergence-and-whats-its-advantages-over-ReLU-or-sigmoid-or-tanh-function

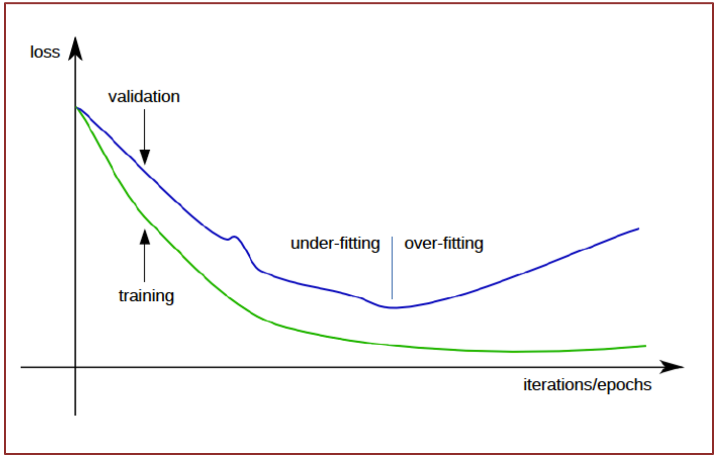
Deep Learning Research

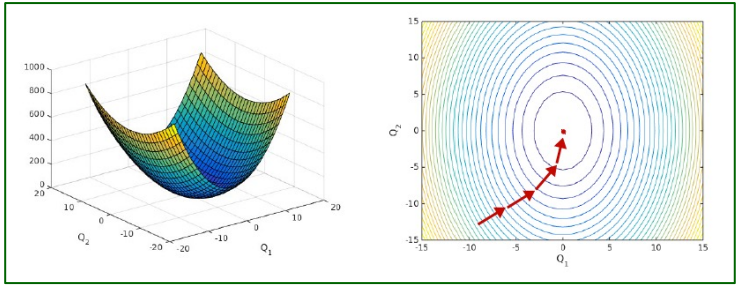


https://www.quora.com/How-does-ELU-activation-function-help-convergence-and-whats-its-advantages-over-ReLU-or-sigmoid-or-tanh-function

Deep Learning Research







https://www.quora.com/How-does-ELU-activation-function-help-convergence-and-whats-its-advantages-over-ReLU-or-sigmoid-or-tanh-function

Deep Learning Research

Softmax

- Generalization of the logistic function
  - Squashes the inputs to the [0 1] range

Logistic function: $$\sigma(z) = \frac{1}{1+e^{-z}}$$

Softmax function: $$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_i e^{z_i}}$$

In mathematics, the **softmax function**, or **normalized exponential function**,[1]:198 is a generalization of the logistic function that "squashes" a K-dimensional vector $\mathbf{z}$ of arbitrary real values to a K-dimensional vector $\sigma(\mathbf{z})$ of real values in the range (0, 1) that add up to 1. The function is given by

https://www.quora.com/How-does-ELU-activation-function-help-convergence-and-whats-its-advantages-over-ReLU-or-sigmoid-or-tanh-function

Thank you

Teerapong Panboonyuen

Contact: teerapong.panboonyuen@gmail.com